



UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II
FACOLTÀ DI INGEGNERIA

Corso di Laurea in Ingegneria Elettronica

TESI DI LAUREA

**Un toolbox Matlab per la “Computer Assisted Instruction”:
un’applicazione alla didattica di circuiti e campi**

RELATORI:

Prof. Massimiliano de Magistris

Prof. Stefano Russo

CANDIDATO:

Massimo Nicolazzo

Matr. 15/14440

ANNO ACCADEMICO 1999-2000

INDICE

INTRODUZIONE	4
CAPITOLO I: <i>DIDATTICA ASSISTITA DAL CALCOLATORE</i>	7
I.1: Il calcolatore come strumento di fruizione di materiale	7
I.2: Cenni alla storia delle tecnologie didattiche	9
I.3: Vantaggi dell'uso del calcolatore nella didattica	13
I.4: Educazione a distanza e sue problematiche	15
I.5: Alcuni esempi di ambienti per la didattica	
interattiva ed a distanza	18
I.5.1: CyberProf	18
I.5.2: WebCT (World Wide Web Course Tools)	19
I.5.3: Learning Environment	21
CAPITOLO II : <i>UN AMBIENTE PER L'ORGANIZZAZIONE DI MATERIALE</i>	
<i>DIDATTICO SUI CIRCUITI E SUI CAMPI</i>	24
II.1: Lo sviluppo di software didattico	24
II.2: Le scelte per la realizzazione del toolbox	28
II.2.1: MatLab come “shell”	29
II.2.2: Modalità di “delivery” e fruizione	30
II.2.3: Interazione con il Web	31
II.3: Il “Learning Environment” per i circuiti ed i campi	32
CAPITOLO III: <i>ASPETTI REALIZZATIVI</i>	35
III.1: Strutturazione generale	35
III.2: Aspetti realizzativi	37
III.3: Il nuovo “Toolbox” Matlab	40
III.3.1: EnvShell.m	40
III.3.2: RunMask.m	41

III.3.3: MultiIO.m	45
III.3.4: ReadDir.m.....	53
III.3.5: FileExist.m	56
III.3.6: Listato del programma WR.C	57
III.4: Un “dimostrativo”	60
 CONCLUSIONI	 67
 APPENDICE I : <i>MANUALE D’USO</i>	 69
A.I 1: Regole generali.....	69
A.I 2: Formati e Regole per l’ organizzazione del materiale didattico nell’ambiente.....	70
A.I 2.1: Istruzioni Aggiuntive ai File MatLab.....	71
A.I 2.2: Regole per i formati PSpice.....	73
A.I 2.3: Regole per i file in formato Html (index.html o index.htm)	75
A.I.2.4: Collegamenti di File in formato MS-Word e Adobe Acrobat.....	76
A.I 3: Programmi di utilità acclusi all'ambiente	76
 APPENDICE II : <i>FUNZIONI AGGIUNTE AL MATLAB</i>	 78
A.II 1: figureposition(handle,position):	79
A.II 2: uipause (secs):	80
A.II 3: Paramwindow(np,...).....	81
A.II.4: Listati delle functions	83
A.II.4.1: figureposition.m	83
A.II.4.2: uipause.m.....	83
A.II.4.3: ParamWindow.m	83
A.II.4.4: ParamOk.m	86
 BIBLIOGRAFIA	 87

INTRODUZIONE

“È veramente difficile insegnare a qualcuno ad eseguire un compito complesso. Per diventare veramente esperto in qualcosa di complesso, uno studente deve apprendere i principi astratti del lavoro, nonché imparare come questi principi si applicano nella pratica” (Kass, 1994).

Alcuni metodi di insegnamento sono abbastanza efficaci nel trasmettere principi astratti, non altrettanto nell’insegnare come applicare tali principi.

Nell’educazione scolastica formale, ad esempio, le conoscenze e le competenze solitamente sono astratte dalle loro utilizzazioni nel modo reale.

Le lezioni in aula spesso perdono di vista la pratica che dovrebbero trasmettere. Di conseguenza, la motivazione è spesso scarsa e molto di quanto si

impara viene presto dimenticato o rimane non integrato o inerte (Collins, Brown e Newman, 1989).

Alcuni metodi di insegnamento, invece, tralasciano completamente i principi astratti; per cui lo studente apprende mnemonicamente.

Il risultato di questo tipo di formazione è che gli studenti hanno memorizzato i passi standard da compiere ma potrebbero non essere in grado di confrontarsi con situazioni reali ed originali.

Molti ricercatori ed educatori concordano che gli ambienti di apprendimento delle competenze dovrebbero attribuire allo studente un ruolo attivo. Gli ambienti di “Learning by Doing” presentano molti vantaggi rispetto a quelli in cui lo studente assorbe passivamente informazioni impartite da un insegnante.

Per sfruttare appieno l'interazione complementare tra l'attività di soluzione dei problemi e l'istruzione, un insegnante deve poter impartire l'istruzione adeguata nel momento in cui lo studente ne ha bisogno. Questa condizione ideale viene quasi realizzata nei migliori apprendistati, in cui l'esperto controlla le attività dell'apprendista, eventualmente intervenendo con opportuni consigli e critiche. Tuttavia quando ci sono più studenti che insegnanti, l'insegnante non può essere continuamente disponibile per impartire l'istruzione adeguata. Così nei più comuni regimi moderni di formazione, l'istruzione e la soluzione realistica dei problemi sono separati l'uno dall'altro nel tempo e nello spazio.

Uno dei vantaggi principali dell'insegnamento su sistemi computerizzati è la possibilità di superare gli ostacoli pratici che spesso obbligano ad una separazione tra soluzione di problemi e l'apprendimento. Le simulazioni al computer possono consentire l'integrazione di apprendimento e soluzioni dei problemi dove altrimenti sarebbe stato piuttosto difficile ottenerla.

La didattica assistita dal calcolatore è già da tempo uno strumento diffuso ed indispensabile per molti settori della formazione. Ciò è sicuramente ancor più vero per la didattica delle materie ingegneristiche, dove l'uso di strumenti di CAD è sempre in più stretta relazione con i contenuti delle materie insegnate.

Questo lavoro di tesi vuol riferirsi alla didattica del settore dell'Ingegneria dell'Informazione, ed in particolare all'area disciplinare dell'Elettrotecnica, dove negli ultimi anni sono andati significativamente affermandosi due ambienti di simulazione, lo SPICE ed il Matlab, entrambi assai noti. In particolare, il primo è

specificamente rivolto all'analisi di circuiti, mentre il secondo si configura come un potente ambiente di programmazione matematica che dispone, oltre agli strumenti elementari attraverso cui anche l'analisi dei circuiti è assai agevole, potenti e flessibili strumenti per analisi di campo.

Molti docenti, in tali ambiti disciplinari, già da tempo integrano le esercitazioni dei rispettivi corsi ed il materiale didattico tradizionale con esempi ed esercizi realizzati e/o proposti agli studenti basandosi su tali ambienti. Tuttavia spesso il materiale così proposto non risulta di immediata fruizione per tutti gli studenti, ed inoltre si presenta legato dagli altri supporti didattici più tradizionali.

Nasce dunque l'esigenza di progettare un ambiente di apprendimento integrato, dove tutto il materiale didattico, con i suoi molteplici collegamenti, possa utilmente essere proposto. Inoltre è necessario ripensare criticamente alle modalità di trasmissione del materiale, di fruizione, nonché di realizzazione e di aggiornamento da parte del docente.

In questa tesi è stato studiato, nonché realizzato in versione sperimentale, un ambiente di apprendimento integrato che provi a dare soluzione organica ad alcune delle problematiche citate. Sono state studiate così varie possibili soluzioni, partendo da una rassegna sullo stato dell'arte, dall'analisi del materiale didattico già disponibile, dalla considerazione di possibili sviluppi futuri di un tale progetto.

È stato così realizzato un pacchetto di funzioni Matlab (un piccolo "Toolbox" per la didattica) che permette, in un'unica "shell" di esposizione-simulazione, la presentazione di materiale testo tradizionale, ipertesto, animazione, simulazione, editing, nonché visualizzazione di audiovisivi. Il pacchetto completo, contenente a sua volta il materiale didattico, software di dominio pubblico, utilities e quant'altro ritenuto utile ai fini della fruizione, viene masterizzato su di un CD autoinstallante da distribuire agli studenti.

CAPITOLO I:

DIFFICILE ASSISTITA DAL CALCOLATORE

I.1: Il calcolatore come strumento di fruizione di materiale

È ormai assodato che l'elaboratore elettronico, non è più solo uno strumento di lavoro, per fare calcolo massivo o elaborare testi, matrici e tabelle, ma è anche strumento di svago e fruizione di materiale di più disparato tipo. Fra questo, esiste ed è presente sul mercato, anche materiale di tipo didattico e software rivolti ad un ambito scolastico ed universitario.

Attualmente la grande enciclopedia di trenta voluminosi tomi, si riduce in una serie di pochi CD-ROM dove, non solo sono presenti le stesse informazioni, ma sono anche meglio organizzate, meglio collegate tra loro, più facili da consultare ed in più arricchite da quelle informazioni aggiuntive che, ovviamente un libro non può dare: filmati, musiche, commenti vocali.

“Il vantaggio che offre il supporto multimediale rispetto al libro nel processo di apprendimento è che attraverso il computer, l’ipertestualità, il CD ROM, si lavora meglio, quando per libro intendiamo enciclopedia, dizionari, manuali di apprendimento. Soltanto una ideologia e anche una nostalgia legittima, ma che comunque va vinta, può far credere che il libro funzioni meglio” (Abruzzese, 1997).

Non solo, ma attualmente l'enciclopedia non ha semplicemente arricchito la sua fruibilità dal punto di vista delle conoscenze in essa contenute, ma date le sue ridotte dimensioni (quella del Compact Disc), ha anche semplificato e migliorato la sua diffusione, essendo presente anche in edicola.

Ancora, le possibilità di poter arricchire le proprie conoscenze, sono certamente aumentate a dismisura con il capillare avvento di Internet, infatti, senza muoversi dalla propria scrivania, si ha la possibilità di accedere dal proprio computer ad informazioni di qualsiasi tipo.

Documentarsi su un argomento specifico o accrescere le proprie competenze usando Internet è sempre stato possibile in teoria, ma in pratica, l’utente inesperto finisce quasi sempre per perdersi in una marea d’informazioni non strutturate.

Grazie a Internet, però, a quasi vent’anni dai primi esperimenti di didattica assistita dal calcolatore, il mondo delle *tecnologie didattiche* sta vivendo un grande slancio.

Alla base di questo fenomeno ci sono in primo luogo motivi culturali: non c’è dubbio che la domanda di competenze che proviene dalle imprese e dalla società sia in continua, rapidissima crescita.

La formazione permanente è infatti oggi indispensabile per entrare o per restare nel mondo del lavoro, ma anche per coltivare i propri interessi extralavorativi e persino per divertirsi. D’altro canto le sorgenti “storiche” di formazione, ossia la scuola, l’università e il sistema delle imprese faticano a tenere il passo con il continuo rinnovamento dei contenuti, almeno usando gli strumenti didattici tradizionali.

La soluzione proposta dagli esperti è semplice: *usare la teledidattica su Internet* per trasformare le informazioni presenti sulla Rete in contenuti formativi che possano accrescere le competenze delle persone.

La *Rete* sta consentendo alle Università di tutto il mondo di "scambiare" interi corsi o singole lezioni, assicurando agli allievi didattica ad alto livello anche su argomenti per cui non dispongono, in sede, dei docenti adatti.

Dal punto di vista tecnico, il motore di questo cambiamento è il matrimonio tra la tecnologia del *Web*, affidabile e a basso costo, e le tecniche d'insegnamento assistito messe a punto negli scorsi anni dai formatori in ambito universitario e aziendale

I.2: Cenni alla storia delle tecnologie didattiche

Le prime macchine per insegnare di cui si ha notizia sono nate alla Ohio University negli anni 20 ad opera dello psicologo Sidney Pressey (Pressey, 1960). Tuttavia se dovessimo stabilire qual è la data di nascita delle tecnologie didattiche, potremmo ragionevolmente scegliere il 1954, data di pubblicazione del articolo di Skinner "The science of learning and the art of teaching" (Skinner, 1954). L'articolo, che proponeva un parallelismo tra gli studi di laboratorio relativi alle modifiche del comportamento degli animali e le pratiche che avrebbero potuto migliorare l'educazione, segnò l'inizio di una fioritura di studi che inizialmente si concentrarono sull'istruzione programmata e sull'impiego di macchine nei processi di apprendimento. Va detto che forse Skinner non pensava di fare da padre fondatore di un nuovo settore disciplinare visto che affermava nelle conclusioni dell'articolo: "il compito è semplice... le tecniche necessarie sono note". Tuttavia il nuovo settore disciplinare si sviluppò rapidamente, soprattutto nel mondo anglosassone, dove fu identificato col termine *educational technology* (talvolta con una diversa sfumatura di significato *instructional technology*). In Italia l'interesse per questo settore si sviluppò solo molto più tardi e solo all'inizio degli anni 70 fece la sua timida comparsa il termine tecnologie didattiche.

In cinquant'anni di vita questo settore è cresciuto e si è trasformato sia per dinamiche sue proprie che per l'influenza dell'innovazione tecnologica e dei mutamenti sociali.

Le tecnologie didattiche si svilupparono inizialmente secondo due linee distinte, parallele, ma non sempre comunicanti: scienza dei mezzi (all'epoca prevalentemente audiovisivi) utilizzabili nella didattica, ed applicazione alla didattica dei principi dell'istruzione programmata.

Le tecnologie didattiche come scienza dei mezzi si concentrarono in un primo momento sullo studio delle caratteristiche e delle possibilità dei vari mezzi audiovisivi. L'attenzione si focalizzò dapprima sull'audiovisivo inteso come strumento (hardware) e poi sul materiale didattico relativo (software) (Dale, 1969). Un successivo allargamento del campo di studio si ebbe con le prime formulazioni di modelli di comunicazione sorgente-ricevitore. L'attenzione, che era inizialmente centrata sulla macchina, le sue caratteristiche, il suo impatto nei processi formativi, il confronto fra l'istruzione tradizionale e quella tecnologica, si estese così al processo di comunicazione uomo-macchina (Berlo, 1968). In un terzo stadio di sviluppo ci fu un ulteriore allargamento del campo ed i modelli di comunicazione sviluppati iniziarono ad essere considerati come componenti di un vero e proprio sistema didattico (Hoban, 1974).

La seconda linea si sviluppò invece contemporaneamente su due fronti. Da una parte ci fu una fioritura di proposte metodologiche connesse alla produzione di corsi di istruzione programmata (obiettivi comportamentali, task analysis, criterion referenced testing, sistemi didattici adattivi ...). E dall'altra ci si impegnò sullo sviluppo di mezzi adatti a gestire non solo la presentazione della materia, ma anche il rinforzo. Qui le macchine per insegnare lasciarono ben presto il passo alla cosiddetta istruzione assistita dal calcolatore (CAI o Computer Assisted Instruction) ed ebbe inizio la lunga storia, tuttora in corso, della didattica interattiva.

Queste due linee iniziarono a convergere verso la fine degli anni sessanta quando apparirono i primi modelli di progettazione di interi corsi non orientati soltanto all'istruzione programmata (Gagné e Briggs, 1979), (Merril, 1971). Gradualmente si arrivò dunque a una visione sistemica (che implica cioè un tutto organizzato) e sistematica (che prevede cioè procedure logiche organizzate in fasi) dell'istruzione e delle tecnologie didattiche e si determinò il passaggio da istruzione programmata a programmazione dell'istruzione. Questo è emblematico di un graduale spostamento dell'interesse che si era allargato dai problemi microscopici legati a particolari aspetti del processo didattico, a quelli macroscopici legati alla progettazione e valutazione di interi corsi e curricula.

Verso la fine degli anni sessanta ci fu quindi un primo assestamento del dominio delle tecnologie didattiche: non più soltanto scienza dei mezzi ed istruzione programmata ma "applicazione sistematica di conoscenze scientifiche (mediate dal campo della psicologia, della teoria della comunicazione...) ai compiti pratici

dell'educazione" (Saettler , 1968). Nel 1970 il direttore della Commissione Instructional Technology istituita dal Governo Federale USA scriveva nel suo rapporto al Congresso: "... le tecnologie didattiche [sono una disciplina che studia] l' approccio sistematico al progetto, allo sviluppo, alla valutazione di processi di insegnamento/apprendimento in termini di obiettivi specifici basato sia su risorse umane che tecnologiche e finalizzato ad una istruzione più efficace". La definizione di tecnologie didattiche fornita all'inizio degli anni 80 dalla Association for Educational Communication and Technology (USA): "Le tecnologie didattiche ... hanno come oggetto processi complessi ed integrati che coinvolgono persone, procedure, idee, mezzi ed organizzazione per l'analisi di problemi relativi all'apprendimento e per l'elaborazione, l'implementazione, la valutazione e il controllo di soluzioni a quei problemi in situazioni in cui l'apprendimento è finalizzato e controllato".

Questo atteggiamento ha condotto in tempi abbastanza recenti alla nascita di una terminologia di sapore ingegneristico (courseware engineering) (Ferraris, Midoro e Olimpo, 1984). Questa terminologia vuole sottolineare l'esigenza di un approccio allo sviluppo e alla gestione di interventi formativi orientato all'integrazione e all'utilizzazione coordinata di conoscenze esistenti; e basato sul riconoscimento delle diverse fasi in cui si articola lo sviluppo di un intervento formativo riconosciuto nella sua totalità. Non si tratta di voler ricondurre la formazione a un puro fatto tecnologico, bensì di sottolineare l'importanza per l'autore-progettista di una struttura concettuale di riferimento all'interno della quale organizzare il proprio lavoro e la propria creatività.

Oggi, in ambito internazionale, le Tecnologie Didattiche, intese come approccio sistemistico e multidisciplinare ai problemi dell'istruzione, sono un settore disciplinare consolidato e ritenuto di importanza strategica per lo sviluppo. Già da molti anni sono attivi in numerosi paesi progetti specifici sulle applicazioni delle tecnologie informatiche all'educazione ed organismi nazionali che coordinano attività e ricerche nel settore (si pensi al Regno Unito dove è attivo dall'inizio degli anni settanta il Council for Educational Technology). Esistono nel mondo oltre 50 riviste internazionali di alto livello scientifico sulle Tecnologie Didattiche. La Comunità Europea ha attivato dal 1988 programmi di ricerca pluriennali nel settore (COMETT, DELTA). Moltissime università offrono corsi e post graduate curricula in Educational Technology. Dal 1987, alla Indiana University, si svolge ogni anno il

convegno dei professori di Instructional Design and Technology che si riuniscono con l'obiettivo dichiarato di promuovere l'eccellenza dei programmi e di indirizzare la ricerca nel settore (Olimpo, 1993).

Oltre all'evoluzione in campo didattico, anche l'evoluzione in campo informatico ha contribuito ad introdurre novità nel settore delle *Tecnologie Didattiche*. Le principali innovazioni consistono nell'introduzione dei "multimedia" e degli "ipertesti"; unendo le due cose ci si riferisce, con un unico termine, agli "ipermedia".

Per multimedia si intende un percorso lungo il quale si trovano fonti di diverso tipo come testi scritti, immagini, animazioni, suoni...

Il termine ipertesto invece si riferisce alla forma del percorso e non al suo contenuto. Un ipertesto è un percorso non lineare che può essere esplorato in modi diversi seguendo indicazioni e collegamenti posti lungo il percorso in funzione dei propri interessi.

Le prime esperienze di applicazioni multimediali furono militari. Gli scopi che ci si prefiggeva di raggiungere tramite l'uso di queste tecniche erano essenzialmente due:

- miglioramento della qualità dell'addestramento ai compiti da apprendere.
- uniformità di comportamenti (indipendentemente dalla locazione geografica).

Verso la fine degli anni '80, dato il drastico ridimensionamento delle spese militari, si pensò ad un doppio utilizzo delle tecnologie sfruttate fino a quel momento solo per scopi militari. Ci si chiese se i multimedia potessero ritenersi adatti per un uso didattico e si fecero varie considerazioni riportate qui di seguito.

Nell'acquisizione delle informazioni esiste una superiorità delle informazioni visive sia sul tatto che sull'udito. Comunque, se si utilizzano più sensi per acquisire una informazione si ottiene un effetto molto più forte di quello ottenibile impiegando fonti di un unico tipo.

I concetti vengono organizzati più facilmente con informazioni spaziali che con parole. Ecco perché, ad esempio è più facile comprendere nuove cose con diagrammi che con lunghi discorsi.

Con i multimedia si possono manovrare oggetti e non concetti, questa è la manipolazione che comunque avviene, a livello fondamentale. Quindi, con strumenti di questo tipo è intuitivo scoprire come interagire.

Queste considerazioni fanno capire perché l'apprendimento con multimedia possa essere molto più rapido, efficace ed efficiente di quello con mezzi tradizionali. Però esiste un grosso problema: l'informazione visiva ha accesso diretto in memoria senza, praticamente, essere elaborata. Quando, invece, l'accesso di una informazione è fonologico il tempo impiegato è maggiore ma risulta maggiore anche il tempo di mantenimento in memoria e questo facilita l'elaborazione dell'informazione e il suo accesso alla memoria a lungo termine.

I multimedia, da soli sono quindi adatti per sviluppare delle capacità che vengono messe in atto automaticamente. Però non è possibile utilizzare gli stessi strumenti in campo didattico con le medesime modalità utilizzate in campo militare perché, nel primo caso, è necessario che l'utente sia attivo. Questo risultato lo si può ottenere costruendo dei multimedia con capacità di esplorabilità al loro interno. Ancor meglio, con la possibilità di interagire con essi.

Questo si è rivelato essere il filone vincente. La costruzione di oggetti di questo tipo è ovviamente più faticosa della costruzione dei multimedia sopra descritti, ad uso militare, ma il risultato finale è migliore, ai fini didattici (Bagnara, 1993).

I.3: Vantaggi dell'uso del calcolatore nella didattica

L'uso del computer nella didattica presenta indiscutibilmente dei vantaggi.

La prima cosa da considerare è che il computer è uno strumento che generalmente incuriosisce ed attira. È molto probabile trovare studenti ben disposti ad utilizzare il calcolatore per attività didattiche.

Le simulazioni al computer possono consentire l'integrazione di apprendimento e soluzioni dei problemi dove altrimenti sarebbe stato piuttosto difficile ottenerla.

Il computer presenta molti vantaggi potenziali.

Prima di tutto, le simulazioni al computer possono permettere agli studenti di esercitarsi in compiti che altrimenti sarebbero troppo onerosi e/o pericolosi.

Le esperienze pratiche non sono sempre realizzabili per motivi di sicurezza, dimensioni, tempi di esecuzione o costi. Le simulazioni possono essere utili in tutte queste occasioni (Clarck, 1993). Il tempo di allestimento ed i costi dell'apparato sperimentale per qualunque attività si riduce a zero ed è possibile osservare in poco tempo fasi cruciali di un fenomeno che nella realtà potrebbe evolvere in un periodo lungo o brevissimo.

Con il calcolatore è possibile riprodurre il fenomeno sia così come avviene nella realtà, sia “sterilizzato”, visto cioè come si svilupperebbe in condizioni ideali, seguendo le leggi della fisica elementare e trascurando le deviazioni prodotte da effetti secondari. Entrambi i punti di vista possono risultare molto utili. Nella simulazione è possibile identificare i vari effetti, e dunque i vari fenomeni, che portano insieme al verificarsi di quanto si osserva. Questo è realizzabile, ad esempio, annullando uno ad uno vari parametri ed osservando i mutamenti ottenuti in seguito a queste variazioni. D'altra parte, in alcune circostanze, gli effetti secondari possono perturbare il fenomeno o ridurre l'intervallo di condizioni sperimentali disponibili. A volte i disturbi sono di tale entità da viziare pesantemente i dati raccolti. In occasioni come queste risulta molto difficile legare quanto osservato a leggi e regole che si stanno studiando, valide solo in condizioni “pulite” (Beaufils, Le Touzè e Blondl, 1994). Non solo, ma l'aspetto matematico e quello fisico-pratico possono essere disgiunti nel senso che si può riservare per intero l'incombenza dei calcoli al computer; questo permette di studiare solo la fisica del fenomeno e di fare osservazioni qualitative.

In aggiunta a questo, in un mondo simulato gli studenti sono al sicuro, possono correre rischi d'imparare dai propri errori.

Altro vantaggio rilevante della simulazione al computer è quello di poter sospendere, sezionare, ripetere una simulazione computerizzata per tutto il tempo necessario ad impartire l'istruzione ed in qualunque momento dell'attività.

Nel mondo reale non è sempre possibile rallentare o sospendere una esecuzione.

Ad esempio, non sarebbe opportuno per un medico in servizio al pronto soccorso sospendere la visita di un paziente potenzialmente colpito da un attacco cardiaco per consentire al tutor di insegnargli a distinguere un attacco cardiaco da un disturbo meno grave.

Infine, una simulazione al computer può contribuire a rafforzare l'interconnessione tra soluzione dei problemi e apprendimento fornendo un insegnamento individualizzato. Se l'insegnamento individualizzato da parte di una persona sarebbe l'ideale, in realtà un rapporto computer/studente di 1 a 1 è molto più fattibile economicamente di un rapporto insegnante/studente di 1 a 1 (Kass, 1994) .

I.4: Educazione a distanza e sue problematiche

Ci sono attualmente un certo numero di definizioni che definiscono il concetto di Educazione a Distanza o anche Distance Education (DE). Questa può essere definita come il sistema o processo che colleghi studenti o *learner* (colui che apprende, non necessariamente uno studente nel senso classico del termine) a risorse didattiche remote. Tale sistema può costituire il modo principale di apprendere o una forma di complemento a quello classico (AT&T).

Citiamo inoltre la definizione data dallo United States Distance Learning Association (USDLA), un'associazione nata nel 1983 con lo scopo di promuovere lo sviluppo e la diffusione dell'educazione a distanza. Secondo la USDLA l'educazione a distanza è "il processo di educazione o *training* attraverso un'istruzione mediata elettronicamente che comprenda satellite, video, audio, grafica, computer, tecnologie multimediali ed altre forme di educazione a distanza" . Lo USDLA sottolinea poi che l'educazione a distanza fa riferimento a situazioni di apprendimento ed insegnamento in cui i docenti e gli studenti sono geograficamente distanti e debbono necessariamente ricorrere a mezzi alternativi di comunicazione e di presentazione del materiale didattico. L'educazione a distanza comprende nel suo complesso due elementi che sono l'apprendimento a distanza e l'insegnamento a distanza. Il primo fa riferimento al ruolo dello studente ed il secondo a quello del docente. Le definizioni di cui prima possono essere sintetizzate elencando gli elementi caratteristici di uno scenario di educazione a distanza. Tali elementi sono:

- La separazione spaziale e temporale del docente dagli studenti per la maggior parte del processo educativo.
- L'uso di mezzi di comunicazione che uniscano il docente agli studenti e permettano la distribuzione e l'accesso a risorse didattiche remote.

- Controllo volitivo da parte dello studente piuttosto che del docente del processo di educazione.

Tale modello porta con se una serie di cambiamenti radicali che coinvolgono sia il personale accademico che la popolazione studentesca. Il personale docente vede sempre più profilarsi all'orizzonte nuovi ruoli nei quali presto o tardi, per forza o per ragione, esso si dovrà identificare; piuttosto che docenti in senso stretto essi dovranno assolvere il compito di mentori e guide nel processo educativo dello studente. All'aumentare delle risorse didattiche orientate allo studio autonomo quali corsi interattivi su CD-ROM il ruolo del docente diventerà sempre più marginale, di supporto più che d'insegnamento.

Tradizionalmente il docente ha ricoperto nell'ambito del proprio lavoro una serie di ruoli: preparare le lezioni del corso, materiale di supporto, esami ed esercitazioni ed ha poi avuto un contatto diretto ed autonomo con i propri studenti. L'affermarsi del modello d'insegnamento a distanza ha portato alla discretizzazione dei vari ruoli in cui le facoltà si stanno specializzando come *content expert* che lavorano fianco a fianco con tecnici, editori, graphic designer, esperti di comunicazione al fine di produrre materiale didattico di qualità. E' questo il caso in cui società di training lavorano in collaborazione con le università cui esse forniscono l'assistenza tecnica professionale per la produzione del materiale didattico finale. Molti corsi stanno subendo un processo di reingegnerizzazione al fine di poter offrire un accesso libero e remoto al materiale didattico e di fornire mezzi e modi alternativi d'apprendimento. A mano a mano che le risorse didattiche vengono rese in forma digitale ne viene sempre più favorito lo scambio tra le università che generalmente rendono tali risorse disponibili attraverso il World Wide Web.

Grazie a tale diffusione i piani di studio ed i corsi tendono ad essere sempre più personalizzati e modulari in quanto gli studenti avranno nella pratica la possibilità di accedere e scegliere tra un "pool" di risorse didattiche messe a disposizione da parte di diverse istituzioni. Molti studenti infatti non sono alla ricerca di una laurea e la modularizzazione permette di soddisfare esigenze specifiche. Nella pratica tale modularità e scambio d'informazioni é però resa complicata dalla mancanza di standard affermati che facilitino lo scambio e la condivisione di risorse tra le istituzioni educatrici. Al momento molte delle risorse create e pubblicate

digitalmente si perdono nell'oceano immenso del World Wide Web e solo parte di queste é realmente condivisa e riusata da altri.

Finora sono state analizzate le motivazione ed i vantaggi che giustificano largamente il bisogno di un cambio radicale nel mondo dell'educazione. Sono tuttavia da considerare gli aspetti negativi che rendono lo scenario meno roseo. Sembra non esserci alcuna università in cui sia gli studenti che i docenti non abbiano espresso una forma di malcontento rispetto alle nuove e flessibili forme di learning.

Se da un lato l'introduzione di un modello accademico flessibile senza limitazioni di spazio e tempo estende le opportunità di educazione a persone tradizionalmente tagliate fuori, il cambiamento porta degli innegabili svantaggi sia al personale docente che ai tradizionali studenti. Educazione a distanza significa innanzitutto nuove tecnologie, hardware e software, che siano di supporto al nuovo modello per l'insegnamento ed apprendimento. Tale tecnologia introduce dei costi d'installazione, manutenzione e di upgrading che possono anche richiedere personale specializzato o comunque che sia dedicato alla gestione di tali risorse. Per un'istituzione tali costi sono in realtà ripagati dall'aumentato numero di studenti che essa arriva a poter servire e per tale motivo le amministrazioni delle università sono in genere favorevoli al cambiamento (Twigg, 1995). D'altra parte, proprio all'aumentare del numero degli studenti virtuali ed al diminuire del rapporto diretto con il docente, gli studenti percepiscono che nel lungo periodo tale processo abbasserà la qualità dell'educazione. In più, i suddetti costi si ripercuotono ovviamente sugli studenti che, se costretti ad accedere a materiale didattico via Internet, incorreranno sicuramente in costi aggiuntivi dovuti all'acquisto di un modem e alle spese di connessione.

Un altro problema in scenari di educazione a distanza é la mancanza di un contatto diretto con i docenti del corso e con i propri colleghi. Non essendovi molta interazione tra studenti di un corso viene a mancare lo spirito competitivo e la capacità di autovalutazione nel confronto con gli altri. Non ultimo un senso di alienazione.

Aldilà dei problemi pedagogici che rimangono tuttora oggetto di discussione ci sono altri problemi con cui lo studente si confronta. Il principale ostacolo é la mancanza di esperienza o training nell'adoperare le nuove tecnologie. Se si pensa allo studente come una persona di una certa età con piccola o nessuna esperienza

d'uso del computer, non si fa fatica ad immaginare che anche l'uso di un semplice web browser non risulti affatto intuitivo ma altresì frustrante.

Anche per quanto riguarda il personale docente non mancano voci di discontento. I “nuovi” docenti avvertono il cambiamento di ruolo cui è soggetta la propria professione a favore di mansioni sempre più definite e discretizzate e questo vuol dire rinunciare ad anni di esperienza e di consolidate metodologie di insegnamento. Ad essi viene inoltre demandato il compito di sviluppare materiale didattico e risorse che tengano conto di una massa studentesca sempre più eterogenea e con esigenze diverse. Affinché un corso di distance learning sia efficiente ed efficace al pari di un corso tradizionale, il personale docente ha bisogno di una folta squadra di supporto che lo assista nello sviluppo e nella presentazione delle risorse. In particolare il docente dovrebbe essere formato all'uso delle nuove tecnologie laddove necessario; non tutti sanno ad esempio utilizzare un editor HTML, JAVA o tool di authoring più complessi. in assenza dell'autore.

I.5: Alcuni esempi di ambienti per la didattica interattiva ed a distanza

Attualmente, alcuni ambienti per la didattica interattiva e a distanza sono adottati in casi reali d'insegnamento. Tutti però, prevedono un certo lavoro e di applicazione da parte del creatore del pacchetto

Di seguito, ne citiamo i più diffusi:

I.5.1: CyberProf

Interfaccia per il training e l'insegnamento asincrono a distanza sviluppato al Department of Physics e al Center for Complex Systems Research presso il Beckman Institute della University of Illinois a Urbana-Champaign. Attualmente usato in numerose università, tra cui l'University of Illinois at Urbana-Champaign, University of Kentucky, University of Central Florida etc.

È essenzialmente un'interfaccia Web tra quattro entità: lo studente, l'insegnante, il World Wide Web, ed un software intelligente di classificazione-valutazione (“grading”) (Hubler A.W.,Assad A.M. ,1995). L'interazione con gli studenti e gli insegnati viene gestita tramite maschere “form” HTML processati da script in Perl e routine in C. Prevede un proprio editor HTML con il quale i docenti possono creare lezioni on line, assegnare compiti o anche problemi da risolvere sotto

forma di quiz. L'editor é arricchito con un numero di “aggiunte” (“*tags*”) proprietarie sviluppate per assistere il *course creator* nello sviluppo del corso

Necessita di una certa familiarità con il linguaggio HTML. Alternativamente un *course editor* può sviluppare il materiale didattico da presentare facendo uso di un'editor HTML piú avanzato ed utilizzare poi il CyberProf *file browser* per salvare e/o aggiornare le modifiche fatte. Le tag supportate dall'editor sono quelle base dell'HTML 2.0 tags, con in piú quelle definite in Netscape e Internet Explorer. Sono in particolare presenti delle tag che permettono d'incorporare codice LaTeX.

In piú, sono presenti estensioni per la creazione di quiz on-line. Anche questo, una volta realizzato con un testo HTML “ad hoc”, Cyberprof provvede ad implementarla con un FORM. E' poi disponibile un programma scritto in Java per il disegno di semplici figure che può essere usato sul Web sia dai docenti/course director che dagli studenti.

I.5.2: WebCT (World Wide Web Course Tools)

Sviluppato presso la University of British Columbia (UBC), è un tool che permette la creazione di sofisticati ambienti d'educazione basati sulla tecnologia web; può essere utilizzato per creare interi corsi on-line o semplicemente per pubblicare materiale di supporto alla didattica.

Sicuramente, il piú diffuso attualmente, fino ad oggi sono state registrate piú di 600 installazioni di WebCT in istituzioni accademiche di tutto il mondo. Il solo WebCT server della University of British Columbia ospita al momento oltre 150 corsi e questo numero è in continua crescita.

L'interfaccia d'accesso a WebCT, avviene attraverso un semplice web browser.

Per uno studente, un corso basato sul WebCT è una homepage che rappresenta la prima pagina che lo studente vede quando accede al corso on-line. Tra le tante cose, la homepage contiene messaggi testuali, immagini, link a parti del corso come esercizi ed appunti delle lezioni. Possono inoltre essere presenti link ad altre particolari homepage del corso dette *toolpage* cui non si può accedere se non attraverso altre *toolpage* o direttamente dalla homepage principale. Le *toolpage* sono in pratica delle homepage secondarie usate per accorpare parti diverse di un corso ed offrire un insieme di tool per gli studenti evitando così di affogare d'informazioni la

home page. Si può per esempio avere una toolpage per le lezioni del corso ed un'altra per le prove di laboratorio.

Per quanto riguarda il course designer, rispetto all'interfaccia studente, questa presenta in più anche una *designer frame* attraverso la quale è possibile accedere ad una serie di funzionalità di supporto alla creazione del corso e alla gestione degli studenti.

C'è da dire a questo punto che WebCT non offre deliberatamente alcun editor HTML per la creazione del materiale didattico vista l'enorme numero di questi attualmente disponibili sul mercato. Il course designer può semplicemente usare tool di conversione o scrivere le pagine HTML per intero. Ciò premesso, il designer dispone di una ricca suite di tool per la creazione e la gestione dei corsi e degli studenti. Un file manager, funzionalmente completo, è accessibile on-line attraverso cui il course designer può aggiornare il contenuto del corso sul WebCT server; egli dispone anche di una student view attraverso la quale può continuamente verificare come si presenterebbero agli studenti le pagine create. Sono disponibili tool di comunicazione quali bulletting board, sistemi di posta elettronica e chat per comunicazione real time tra gli studenti di un corso o per sessioni di esercitazione on-line. All'interno di una bulletin board poi, il course designer può creare dei forum di discussione alcuni dei quali privati, per esempio utili in lavori di gruppo.

Il course designer può anche associare al materiale di corso dei quiz di autovalutazione in cui gli studenti ricevono unitamente al voto dei commenti ed eventuali spiegazioni. Un glossario per il corso può essere creato e acceduto attraverso un tool di ricerca disponibile on-line, oppure attraverso semplici link HTML dalle diverse toolpage del corso. Ed inoltre il course designer può concedere agli studenti il diritto d'inserire delle annotazioni personali di pagina in punti specifici del corso on-line.

Infine, ma non per ultimo, è disponibile un set di tool amministrativi che permette di monitorare l'attendance del corso da parte degli studenti. Per ogni studente è possibile registrare, tra le tante informazioni, la data del primo accesso al corso, l'ultima data di accesso, il numero di accessi e soprattutto monitorare quali pagine lo studente visita. Il tutto è inserito in un database che può essere diversamente interrogato e ordinato restituendo per esempio informazioni sullo studente che ha seguito di meno o sullo studente che non legge effettivamente il materiale didattico on-line etc..

Uno dei principali vantaggi di WebCT è che, offrendo un'interfaccia d'accesso comune, non impone agli studenti di imparare interfacce diverse per ogni nuovo corso che viene portato on-line. Sia gli studenti che gl'insegnanti possono accedere a WebCT senza bisogno di installare alcun programma essendo richiesto un semplice Web browser. Questo introduce una notevole flessibilità visto che qualsiasi cambiamento al materiale didattico é prontamente disponibile per l'uso (Klessig R.W., Tesink K.).

I.5.3: Learning Environment

Nell'ambito del progetto RENAISSANCE è stato fondato il club Learning Environment al fine di fornire delle risposte al complesso problema dell'integrazione dell'Information Technology con i modelli d'insegnamento classici. Il club contava 11 istituzioni accademiche sotto la direzione della Fretwell & Downing, a capo del progetto RENAISSANCE, ed è stato attivo dal luglio del 1995 al maggio del 1997. Durante questi anni, sono state prodotte tre versioni del software Learning Environment (LE)

Il Learning Environment è stato implementato con un modello client-server supportandone l'uso su reti locali e wide area. Le informazioni sono mantenute da uno o più server Oracle mentre il client è previsto per il sistema operativo Windows95/NT 4.0.

Nel progettare quest'applicazione la Fretwell & Downing, con il feedback fornito dai membri dell' LE Club, ha focalizzato l'attenzione su alcuni punti (Miller M.A.):

- Realizzare un modello aperto che permettesse allo studente un'accesso libero alle risorse didattiche
- Possibilità di avere un piano di studio flessibile
- Possibilità di seguire i progressi dello studente (student tracking)
- Supporto on-line durante l'intero percorso educativo
- Possibilità di gestire agevolmente programmi di studio e gruppi di studenti da parte dei tutor

L'ambiente didattico implementato dal Learning Environment è completamente modularizzato. Il primo passo da compiere è quello di creare le risorse didattiche che s'intende usare per lo sviluppo del corso. Le risorse didattiche cui si ha accesso possono essere di diversi tipi. Esse possono essere ad esempio:

- Documenti in formato Word
- Documenti in formato Adobe Acrobat che permettono di allegare risorse multimediali e hyperlink
- Risorse didattiche multimediale che possono essere create usando diverse applicazioni di authoring
- Pagine Web

In realtà le risorse supportate possono essere di qualsiasi tipo e formato cui è però necessario associare l'applicazione opportuna. Se per esempio si vuole visionare una pagina web è possibile associare a tale risorsa un qualsiasi web browser.

Una volta create le risorse il tutor o course designer procede allo sviluppo del corso vero e proprio a partire dal *module*. Per ogni module il tutor ha a disposizione un template attraverso il quale specificare informazioni di carattere generale; si procede poi a definire le *activity* per il module attraverso cui il tutor definisce i compiti previsti affinché lo studente completi il module. Ad ogni activity il tutor può associare degli *attachment* che contengano risorse di supporto, e può associare spiegazioni aggiuntive ed annotazioni ognuna caratterizzata dalla propria icona.

Infine il tutor ha il compito di iscrivere gli studenti ai relativi module. Tali studenti sono stati precedentemente inseriti nel LE database dal Learning Environment administrator il quale si preoccupa anche di definire i tutor.

Per quanto riguarda lo student assessment il Learning Environment permette di associare ad ogni activity un esercizio o compito di valutazione (assessment). Da un punto di vista pratico l'assessment è in realtà un'altra activity cui sono associati attachment e note di spiegazione. Ciò che manca è però il supporto di una qualunque forma di autovalutazione e tool per la creazione di test on-line; è tuttavia presente una forma di student tracking.

L'interfaccia utente per lo studente ed il tutor si basa su tre diversi modi d'interazione: un modo fortemente strutturato di lavoro guidato, un altro di libera esplorazione delle risorse ed un ultimo di comunicazione on-line. Il primo modo

prevede uno “workspace” d’utente attraverso il quale vengono presentate le informazioni con le quali l’utente si troverà a lavorare.

Il modo non strutturato o modo ricerca, permette invece all’utente di effettuare ricerche personalizzate, esplorando liberamente tutte le risorse didattiche disponibili.

Il modo di comunicazione on-line è invece implementato attraverso applicazioni di posta elettronica e/o video conferenza che è possibile lanciare attraverso il Learning Environment.

Rispetto agli applicativi finora analizzati, il Learning Environment costituisce l’unico vero ambiente per la didattica a distanza. Non a caso, il Learning Environment costituisce il primo prodotto nel campo dell’educazione a distanza della Fretwell & Downing, specializzata nel passato proprio nello sviluppo di sistemi di gestione di studenti per college e istituti di *further education*.

In secondo luogo esso presenta un’elevata modularità che permette il riuso di activity, risorse didattiche, module attraverso diversi piani di studio.

Tuttavia esso non supporta in maniera adeguata lo student assessment ad anche lo stesso student tracking è piuttosto limitato, essendo stato progettato in un’epoca in cui la tecnologia web non era ancora completamente sviluppata che ne ha soprattutto limitata la portabilità. Le prove effettuate con il Learning Environment hanno dimostrato l’importanza di avere un’interfaccia d’accesso che sia la più semplice ed intuitiva possibile. Nella pratica il Learning Environment è stato caratterizzato da un’interfaccia molto poco intuitiva che gettava facilmente anche l’utente non novizio in uno stato di confusione.

CAPITOLO II :

UN AMBIENTE PER L'ORGANIZZAZIONE DI MATERIALE

DIDATTICO SUI CIRCUITI E SUI CAMPI

II.1: Lo sviluppo di software didattico

In generale, prima di realizzare un qualunque software che debba interfacciarsi con utenti più o meno esperti, essere “*user-friendly*” e che possa definirsi utile, occorre meditare su alcune problematiche e porsi una serie di domande, su come costruire e cosa è opportuno inserire in un tale software. Sarebbe, per esempio, utile che il programma potesse funzionare su tutte le macchine, a prescindere dal sistema operativo installato, o, comunque, sui sistemi usati più comunemente.

Nel particolare, nel caso di un software che debba fare da ambiente per un uso didattico, occorre prestare grande attenzione alla “interfaccia utente”, intesa come quella parte del software che permette all'utilizzatore di interagire con esso: la

facilità d'uso è di importanza fondamentale per software di qualsiasi genere. Un programma ben fatto con una cattiva interfaccia perde gran parte del suo valore.

Se un utente alle prime armi non può realizzare una operazione semplice senza essere costretto a consultare un manuale di istruzioni allora il pacchetto è probabilmente destinato ad essere messo da parte. Per ovviare a problemi di questo tipo è *necessario innanzitutto conoscere bene il tipo di persone che userà il prodotto* e quindi utilizzare alcuni accorgimenti per mettere l'utente a proprio agio, dotando il prodotto di una struttura autoevidente e facendo uso di metafore, che permettono di fare leva sulle conoscenze “a priori” dell'utilizzatore (Donnelly, 1993; Myers, 1993).

Per un software didattico, è di estrema importanza anche la visualizzazione: vedere rappresentazioni appropriate dei fenomeni e di concetti, ne aiuta la loro comprensione (Donnelly, 1990). Inoltre bisogna considerare che i meccanismi di apprendimento variano da individuo a individuo. Quello che per una persona è meglio assimilato e fissato con una discussione, per un'altra potrebbe esserlo grazie a immagini o animazioni o in altri modi. Di certo quindi, l'utilizzo di fonti di informazioni diverse, permette un percorso didattico più ricco e meno monotono.

Alcune, sono le questioni di carattere generale, valide per chi voglia realizzare un software, di qualsiasi tipo esso sia:

- L'utente trova indicazioni brevi e chiare su come interagire?
- Le cose funzionano come ci si aspetta?
- Le chiavi, i controlli, le interazioni con l'utente, hanno sempre un effetto rapido e visibile?
- Ci sono molte occasioni di trovarsi in una situazione di conflitto interno in cui il programma si blocca?
- Gli errori possono essere facilmente rettificati con una azione singola o quantomeno semplice?
- Ci sono pause lunghe? Se è così l'utente ne è avvisato?

Praticamente questo significa chiedersi se la struttura del pacchetto è sufficientemente chiara e se l'utente è in grado di muoversi agevolmente anche senza esperienza.

Domande più specifiche, riguardanti la realizzazione di pacchetti didattici veri e propri, e quindi le simulazioni, possono essere invece le seguenti:

- Le nozioni esposte sotto forma di testo o simulazioni o altro sono corrette?
- Sono incluse opzioni rilevanti che stimolino controlli di tipo “che cosa succede se”?
- Gli studenti sono stimolati a predire le conseguenze ed i risultati?
- Gli studenti sono in grado di svolgere esercizi quantitativi in cui si applica la legge o l'esperimento simulato?
- Il software può essere integrato in una o più lezioni?

Delle questioni poste, l'intento è stato di provvedere a compilare la parte riguardante lo sviluppo dell'interfaccia docente-discente, lasciando al docente il porsi e la risoluzione individuale delle domande prettamente didattiche.

Questo comporta un'altra serie di questioni riguardante quindi la interazione che dovrà avere il docente preparatore del pacchetto didattico con il programma, e che sono in sostanza:

- Il software permette di ottenere il risultato che ci si proponeva?
- Il manuale d'uso è sintetico ed esaustivo?
- Le regole compilative sono il minore possibile e di facile attuazione ?
- E' possibile evitare al preparatore di dover usare tool specifici aggiuntivi a quelli già usati quotidianamente ossia il File Manager e l'editor di testo?

In sintesi è possibile realizzare il proprio pacchetto didattico con la minor spesa di tempo e denaro possibile per poi ottenere il risultato aspettato?

Dunque, usando il calcolatore si può scegliere come esporre informazioni fra diverse modalità e si hanno a disposizione vari strumenti o li si possono costruire. Alcuni di questi metodi di esposizione e strumenti si sono rivelati più efficaci di altri. È stato così possibile definire un elenco di oggetti che è opportuno inserire in un pacchetto di software didattico.

- Una base di informazioni contenente del materiale sotto forma di testo che normalmente viene trovato in un manuale.
- Una serie di simulazioni che possono essere utilizzate sia per illustrare gli eventi nella loro evoluzione temporale che per manipolazioni interattive da parte degli utenti.
- Commenti audio.
- Animazioni e filmati
- Esempi
- Strumenti per la manipolazione algebrica e per le stime numeriche
- Grafici.

Nell'ambito dell'ingegneria, dove il calcolatore elettronico è già da tempo strumento di supporto necessario per l'apprendimento e l'approfondimento, strumento di simulazione e di verifica delle conoscenze in molte materie didattiche, la situazione è leggermente diversa; anzi, nel particolare dell'ingegneria elettronica, gli strumenti utilizzati sono sostanzialmente di simulazione; i concetti esposti nei vari corsi, hanno un primo riscontro nelle simulazioni al calcolatore, certamente più pratico ed a volte più economico e meno pericoloso, rispetto l'attuazione reale, neanche proponibile in prima istanza.

Nel ambito del particolare corso di riferimento di Elettrotecnica, ma anche in altre varie materie di insegnamento dell'ingegneria del settore informazione, i software di simulazione a cui ci si appoggia sono sostanzialmente i pacchetti:

- MathWorks Matlab
- Spice.

Il primo, MATLAB in quanto “high-performance language for technical computing”, che integra nel proprio ambiente un sistema di calcolo, visualizzazione e programmazione dove i problemi e le loro soluzioni sono espresse in una notazione “familiarmente” matematica e permette di risolvere vari problemi di computazione tecnica, il secondo, Spice, diretto alla simulazione circuitale analogica e digitale.

Quindi, i “target” prefissati, i fruitori del pacchetto software a cui si vuole presentare il modello sono degli utenti già esperti e/o che dovranno prima o poi

affrontare l'uso di questi due software: I docenti che già lo usano ed i discenti che dovranno in ogni caso conoscerlo.

Nel progettare quest'applicazione, infine, si è voluto focalizzare l'attenzione su un altro punto fondamentale:

Normalmente, il primo passo da compiere, per il *course-designer* è quello di creare le risorse didattiche che s'intende usare per lo sviluppo del corso. Dato l'avanzato stato delle fonti informatiche già presenti e sviluppati dai docenti negli anni (programmi e listati Spice e Matlab), e delle varie risorse a cui si ha e si è avuto accesso fino ad oggi per presentare la documentazione, appunti, e materiale didattico in genere, (sostanzialmente file di testo Word e pdf), si è voluto dare la possibilità di recuperare tale materiale nel formato in cui si trova, senza doverli trasformare in formati più adatti o doverci "rimetterci le mani".

È stato quindi prevista la possibilità di creare il proprio corso accludendo le risorse didattiche preesistenti e dei formati più comuni:

- Documenti in formato Microsoft Word
- Documenti in formato Adobe Acrobat che a loro volta permettono di allegare risorse multimediali e hyperlink al suo interno
- Risorse didattiche multimediali in generale, quindi immagini, suoni e filmati creati usando i più diparati software di acquisizione
- Pagine Web

Tutto senza perdere di vista il fatto di voler realizzare un modello di ambiente il più possibile aperto, che permette allo studente un'accesso veloce e libero alle risorse didattiche disponibili ma contemporaneamente organizzate nella maniera più appropriata possibile.

II.2: Le scelte per la realizzazione del toolbox

II.2.1: MatLab come “shell”

Nel creare un progetto atto ad organizzare e rendere fruibile il materiale già sviluppato ed acquisito nel tempo, senza doverlo né riscrivere, né stravolgere o modificare, è stato pensato di costruire una sorta di contenitore all'interno del quale, il *course-designer*, docente o chi altri, possa semplicemente inserire il proprio materiale: esempi matlab, spice, documenti, e strutturarli a piacimento secondo una propria logica, dopodiché il programma provveda alla visualizzazione ed al suo interfacciamento con il fruitore in una sorta di ambiente integrato.

Date le potenzialità dei calcolatori come strumenti multimediali, si è aggiunta la possibilità di collegare al materiale, oltre alle documentazioni, anche immagini di insieme, commenti vocali, animazioni o filmati per il completamento dell'informazione che si vuole dare.

Dunque, si vuole dare al preparatore del pacchetto didattico, uno strumento semplice e di facile utilizzo, per creare autonomamente e senza grosse incombenze un sistema che possa contemporaneamente, dare una struttura ben definita al materiale didattico e renderlo il più possibile completo nella sua presentazione-esposizione.

Non solo, ma si vuole anche dare al discente la possibilità di avere non solo la stessa semplicità d'uso ma anche la completa accessibilità al materiale dimostrativo rilasciato, con la possibilità di poterlo editare, modificare, studiare, “metterci le mani”, così da stimolarlo a ragionarci sopra per aggiungere il suo contributo e in qualche modo “completarlo”.

Fondamentale importanza è stata data quindi alla simulazione.

L'ambiente presentato nasce proprio dall'esigenza di poter dare allo studente degli esempi di simulazioni (circuitali e non) mirati, da far “girare” a casa sul proprio calcolatore e quindi, come principio basilare di poter interagire su di esso. È stato così portato avanti l'idea del progetto “open”, ossia aperto completamente all'utente, ritenuto non totalmente inesperto e che dovrà interagire non in maniera passiva, ma che si attiverà anche nel testare delle proprie scelte.

Proprio per le specifiche richieste a cui si è voluto mirare, dunque è stato utilizzata una soluzione necessariamente *stand-alone* e non in tecnologia *web-style*. Non solo ma, dato che il Matlab è di per sé strumento didattico necessario per lo scopo prefissato di poter eseguire e costruire simulazioni, è stato usato lo stesso

MatLab come base per la costruzione dell'ambiente di interfacciamento per il materiale didattico, in quanto oltre che strumento prettamente di calcolo, MatLab offre un più che completo linguaggio di programmazione, che ne permette un uso anche come ambiente di sviluppo e di integrazione grafica.

Il motivo base, che spinge proprio per la soluzione non web è che anche se un sistema legato ad internet mette a suo favore la facilità di raggiungere gli studenti senza che questo “installi” sul proprio PC null'altro che il browser e di mantenere sempre aggiornati tutti i “Client” una volta aggiornato il Server, questo non permette le simulazioni dei listati MatLab e spice: il discente è comunque costretto ad avere la propria copia dei due software installati sul suo computer, scaricare dalla rete ciò che gli interessa ed eseguirli.

Non solo, ma il docente dovrebbe appoggiarsi necessariamente ad una struttura di accoglienza per il proprio spazio web, essere costretto a riorganizzarsi tutto il suo materiale didattico in un formato “internet-compatibile”, quindi: o ad affidare a terzi lo sviluppo e il progressivo aggiornamento del sistema, ovviamente guidandolo, o ad apprendere un certa conoscenza di base od usare tool di authoring più complessi per metter su le pagine necessarie per la pubblicazione. Non ultimo, nel caso di esempi MatLab, il docente sarebbe costretto lo stesso a costruire dei pacchetti software da far girare off-line, in quanto quasi tutti i programmi MatLab non sono costituiti da un unico sorgente, ma per semplicità e chiarezza di esposizione si tende a creare subroutine e funzioni in file separati che ovviamente dovranno essere scaricati dalla rete in un unico pacchetto, sì da evitare che il programma poi fallisca perché mancano dei “pezzi” necessari.

II.2.2: Modalità di “delivery” e fruizione

L' ambiente sviluppato ha più possibili strade di utilizzo e devoluzione, quali il CD-ROM o la diffusione Internet, per quanto riguarda la distribuzione, mentre ha una più vasta fruibilità dal punto di vista dell'utilizzazione.

Con il CD-ROM, all' utente (a cui è comunque demandato il possesso di una copia del Matlab) viene consegnato un pacchetto di per sé esaustivo, quale possa essere un intero corso universitario, dove però non sono solo memorizzate “video lezioni” o quale supporto al materiale didattico “tradizionale” quale il libro di testo o appunti così come attualmente avviene per diversi corsi e libri tradizionali (si veda,

ad esempio Sedra-Smith, Microelettronic Circuit ed. IV, dove però il software e gli esempi inclusi nel CD non permettono una vera iterazione con l'utente).

Ovviamente, la soluzione CD offre anche il sostanziale vantaggio della velocità, nel caso di materiale didattico multimediale, senza imporre "limitazioni di banda"

Non è comunque inimmaginabile una diffusione tramite i mezzi che Internet mette a disposizione. Difatti, considerando il pacchetto didattico a moduli, e che l'intero pacchetto base, ossia il necessario per lanciare l'ambiente creato, non è poi così tanto grande, nulla vieta ad una distribuzione via rete, così che l'utente-discente "scarichi", solo i moduli di interesse. Non solo, ma così facendo si può anche creare un vero corso di lezioni, e quindi con una sequenza prefissata dal docente, che provvede a mettere in rete il materiale un po' per volta, se non a spedire tramite mailing lists i moduli alla sua classe virtuale. È altresì chiaro, che con questo sistema, però, per non appesantire troppo i moduli didattici, bisogna provvedere nei singoli moduli didattici a snellire il materiale multimediale, che comunque assume la sua dimensione.

Un'altra modalità d'uso sta ovviamente nel mezzo rispetto le due precedenti, ossia è immaginabile una distribuzione tramite CD di un pacchetto base ed aggiornarsi poi tramite i più disparati mezzi telematici.

Il toolbox creato, ancora, non è solo uno strumento per la didattica a distanza, ma anzi prevede un largo utilizzo come strumento di apprendimento diretto in aula. Si prevede cioè che il docente lo possa utilizzare "in diretta", in aula per presentare e costruire con i suoi allievi, data anche la semplice e veloce modalità d'uso, delle "simulazioni integrate", così da carpire ancora di più l'interesse degli allievi, stimolati dalle applicazioni di ciò che stanno studiando.

II.2.3: Interazione con il Web

Dato il "target" a cui è finalizzato questo toolbox, ossia lo studente di ingegneria del settore informazione, non è stato né previsto, né implementato, nessun programma di comunicazione specifico tra docente o tutor ed i discenti, lasciando agli attuali strumenti di e-mailing e di browsing questi compiti.

Si immagina cioè che l'uso di questi software siano oramai ben noti agli utenti, per cui diviene non solo superfluo, ma anche inutilmente complicato usare altri strumenti per fare esattamente la stessa cosa!

Quindi, con il sistema sviluppato, l'interazione con il web diviene relativo, non è necessario metter su dei veri e propri server-web, in quanto, per la diffusione dei “moduli di aggiornamento”, basta l'uso della sola posta elettronica, o solamente di appoggiarsi a server ftp o web dove il docente lascia i “moduli” o pacchetti di aggiornamento, ed il discente provvedere al suo recupero.

Il pacchetto in questione, poi data la particolare filosofia adottata per lo sviluppo del materiale didattico, non è null'altro che una semplice copia della directory di lavoro del computer del docente, dove quest'ultimo ha costruito il materiale.

Attualmente in rete sono già presenti alcuni server web ed ftp che fanno semplicemente da “strutture di parcheggio”, ossia dove viene lasciato il materiale, documentazioni, immagini e listati, lasciando poi al discente di scaricarsi e guardarsi e studiarsi fuori linea.

Esempi di questo tipo sono www.dis.uniroma1.it/PUB/palagi/didattica e imagets10.univ.trieste.it/didattica/tesine/t33 .

A differenza di questi siti, dove però è lasciato al discente, che recupera i vari documenti, di ricostruirsi la struttura del modulo didattico, la soluzione presentata dal nostro tool è certamente più semplice e immediata: infatti si prevede che il docente, costruita la struttura ed i vari collegamenti tra i file del suo modulo, comprimi il tutto (come se fosse una copia di backup) in un unico file e lo rilasci ai discenti cosicché una volta installato ci si ritrovi con il sistema già pronto per l'utilizzo.

II.3: Il “Learning Environment” per i circuiti ed i campi

Per rispondere alle questioni precedentemente illustrate, e date le scelte progettuali di cui si è discusso nei paragrafi precedenti, si è costruito il toolbox per la piattaforma operativa Microsoft Windows 95/98/Me, in quanto attualmente la più diffusa e supportante i pacchetti software Matlab, spice ed i tool di visualizzazione di file multimediali e testo.

È chiaro, comunque, che essendo l'intero pacchetto del toolbox scritto in linguaggio MatLab, è facilmente esportabile anche su tutte le altre piattaforme ad

ambiente grafico, dove il MatLab è presente così come è presente una delle tante versioni e realizzazioni dello Spice (quindi Mac e Unix).

Il toolbox qui presentato, dunque, è stato creato immaginando di dover far da interfaccia e da collegamento tra programmi e materiale preesistente e l'utente finale, ovviamente non precludendo nulla ad altre e nuove produzioni.

E' stato così pensato di creare un software con cui il docente o più in generale il preparatore del pacchetto multimediale non debba direttamente collegarsi, ma debba solo organizzare il suo materiale in un percorso predefinito ed all' interno di questo avere la massima libertà.

È stata così messo a punto una semplice metodologia di regole e formati da seguire, che non creano di fatto nessuna sostanziale modifica al materiale didattico normalmente realizzato nei corrispondenti corsi di riferimento e che ne permettano la omogeneizzazione con il rimanente materiale multimediale a cui si vuol fare riferimento.

Dal punto di vista pratico, ciò si è potuto realizzare creando un “parser” che riconosce dei “comandi aggiuntivi” inseriti come commenti all' interno degli esempi dimostrativi MatLab e delle associazioni aggiuntive tra i file di vario tipo per quanto riguarda invece gli esempi Spice.

Tali comandi sono comunque opzionali, in quanto perché il toolbox mostri i file nella sua maschera di esplorazione, l'unica regola da seguire è quella di indicare la directory di lavoro.

In aggiunta a questi comandi aggiuntivi, che permettono di collegare direttamente ai singoli esempi file multimediali, immagini ed ipertesti, è stata prevista anche la possibilità di legare ad interi pacchetti di esempi dei documenti di testo (in formato MS-Word ed Adobe-Acrobat).

Andando nel particolare, il docente, partendo da una directory nominata “FILE” organizza al suo interno, il proprio materiale operando solo con il file manager del Windows, creando cartelle e sottocartelle ed inserendo all' interno di queste il suo materiale, creando così il suo personale percorso didattico.

Insieme ai file Matlab e Spice, poi, può inserire i documenti ed i file multimediali che più gli sembrano opportuno. Fatto questo, può, lanciando l'ambiente sviluppato con il giusto percorso, già visualizzare il proprio prodotto. Infatti l'ambiente ripresenterà direttamente la struttura data ed i file all' interno della directory FILE.

Per quanto riguarda i file in formato Matlab, a questo punto, se vuole stesso operando dall'ambiente, può completare i collegamenti alle risorse aggiuntive: legando ad ogni file Matlab, la figura più opportuna, il testo da voler far visualizzare, eventuali "link" a pagine web (locali e non), e nascondere i file che non devono essere direttamente visualizzati nell' ambiente, semplicemente editando (direttamente dall'ambiente o no) il file in questione ed inserendo all' interno di questo i comandi aggiuntivi, potendone così visualizzare immediatamente gli effetti.

Per quanto riguarda i file Spice, essendo essi semplicemente "ospitati" dall' ambiente Matlab, che provvedere al loro riconoscimento, esecuzione, editing e visualizzazione dei risultati, i collegamenti ad i singoli file in formato spice, vengono creati sempre utilizzando solo il file manager, semplicemente modificando i nomi delle risorse collegate con una sintassi ben precisa.

Ancora, se sono presenti nella struttura documenti Word ed Acrobat, questi verranno presentati all'utente come risorsa aggiuntiva. Non solo i documenti presenti nella cartella dove è presente il file in questione, ma anche quelli presenti nelle cartelle "padre", per un massimo di sette.

Infine, è possibile legare a tutto il pacchetto didattico, una risorsa multimediale ed una immagine di "default", che vengono presentati quando manca, o è ritenuto superfluo, una specifica.

Fatto questo, il docente, come già detto in precedenza, deve solo copiare (magari comprimere) la directory FILE o solo una sua sottocartella su un supporto per la distribuzione e rilasciarlo.

Da parte del discente, infine, una volta scompattato il file ricevuto, nella cartella giusta, non deve far altro che, lanciando l'ambiente, usarlo, studiarlo e farne profitto.

Data dunque la semplicità di uso e la modalità di interazione con il toolbox è addirittura consigliabile, una volta creata la struttura a directory, usare lo stesso ambiente per meglio amalgamare il tutto.

Quindi, in definitiva, il toolbox creato può essere visto contemporaneamente come ambiente che è di sviluppo, costruzione, debugging e fruizione del materiale didattico.

CAPITOLO III:

ASPETTI REALIZZATIVI

III.1: Strutturazione generale

Il problema fondamentale da risolvere, nella realizzazione del progetto, era quello di renderlo il più aperto possibile agli utenti, sia esso docente che discente, di immaginare come dover organizzare il materiale per renderlo facilmente fruibile e contemporaneamente facilmente costruibile senza dover interagire con specifici tool di preparazione, file di configurazione e/o ancor peggio, dover mettere mano ai listati sorgenti dell'ambiente di interfaccia.

Per eliminare quindi, ogni qualsiasi forma di configurazione del sistema, si è pensato di sfruttare una organizzazione a “directory”; ossia, data l'unica accortezza di partire da una directory predefinita, ossia con un nome ben preciso: nel nostro caso una directory nominata “FILE”, inserire al suo interno tutto il materiale,

organizzabile nel modo più libero possibile, fra listati e/o cartelle a sua volta contenenti altro materiale, senza nessuna regola particolare fissata a priori.

In questa maniera, l'ambiente diviene completamente trasparente al docente in fase di creazione, il quale dovrà interagire solo con il "file manager" del sistema operativo MS-Windows.

Altro problema affrontato, è stato quello di voler mostrare al fruitore dell'applicazione, solo il materiale utile e, nel frattempo, avere la massima libertà di organizzazione dello stesso materiale: in Matlab, infatti, molto spesso, se non praticamente sempre, ci si costruisce non un unico file di programma, le subroutine e le funzioni chiamate dal programma principale, sia per facilità operativa, che per leggibilità e strutturazione, vengono creati in file indipendenti. Ebbene, un sistema che legge e propone il contenuto della directory, mostrerebbe anche queste subroutine che non sono eseguibili direttamente.

Ancora, spesso e volentieri i nomi dati ad i vari sorgenti sono degli acronimi di ciò che si sta per eseguire e per giunta i nomi, sono strettamente legati al file manager, quindi non ammettono caratteri definiti "speciali", quali l'apice " ' ", le virgolette i segni di maggiore e minore e così via, quindi per renderne di più semplice visibilità si doveva trovare il modo di come nascondere i file "indesiderati" e di come legare dei nomi simbolici ai nomi dei file Matlab.

Infine, data l'idea di creare un interfaccia il più completa possibile, e che contenga collegamenti anche a spiegazioni testuali, figure, immagini e audiovisivi, l'ultimo problema da risolvere era il collegamento tra questi file, a loro volta indipendenti dal resto, e i listati visibili.

Si è allora pensato di creare un esiguo numero di "comandi" supplementari da inserire opzionalmente nei listati Matlab come semplici commenti, così da permettere all'ambiente di fare i relativi collegamenti:

```
% titolo:<nome esteso>
% bitmap:<nomefile>
% notefl:<nomefile>
% mmedia:<nomefile>
% htmlfl:<nomefile>
```

ed uno per nascondere il file dalla visualizzazione:

```
% HideMe:
```

dove *<nome esteso>* è null'altro che il testo che deve comparire nella lista e *<nomefile>* il file ASCII, la figura, il file multimediale e il link html associati al listato matlab in questione.

Per quanto riguarda invece i listati Spice, il sistema utilizzato è stato quello di collegare a questi, i file aggiuntivi utilizzando una semplice associazione di nomi; ossia, semplicemente dando al file multimediale che si vuole collegare per esempio al file spice di nome "xx.cir", il nome "xx.cir.mmf", cioè lo stesso nome ma con una estensione aggiuntiva specifica che ne permette l'individuazione ed il tipo di legame con il file principale.

Nel particolare le estensioni aggiuntive sono:

- “.fig”** per collegare immagini
- “.txt”** per collegare i testi di presentazione
- “.mmf”** per collegare file multimediali
- “.html”** per collegare ipertesti e/o file comunque visualizzabili con un browser.

L'interattività del programma è poi assicurata dalla possibilità di lanciare ed editare i vari file disponibili sia MatLab che Spice.

In ultimo, si è poi voluto dare la possibilità di collegare alla struttura dei file testo, scritti in formato MS-Word e Adobe-PDF senza fare nessun collegamento esplicito, ma sfruttando la stessa strutturazione a directory dove vengono lasciati i file di questo tipo.

III.2: Aspetti realizzativi

Data la volontà di voler costruire un'ambiente ad interfaccia grafica, al cui interno vengano lanciati e presentati i programmi, un'ulteriore problema si è presentato, in quanto, buona parte dei programmi matlab fanno largo uso della console testo offerta dal matlab per richiedere input e mostrare dei semplici output testuali a schermo. Per tale ragione, volendo evitare che la finestra grafica del toolbox si sovrapponga alla "Command Window" del Matlab facendone così perdere la visibilità, si è pensato di organizzare la visualizzazione in modo da avere entrambe le "finestre" visibili, lasciando all'ambiente integrato, però, il maggior spazio possibile per visualizzare le informazioni supplementari agli esempi.

Si è quindi, per scelta organizzativa, pensato di lasciare alla console del Matlab un 30% circa dello schermo così da occupare il restante con l'ambiente di dialogo del toolbox.

Dato che non esiste la possibilità di ridimensionare la "Command Window" direttamente da comandi del MatLab è stato necessario scrivere un piccolo programma esterno che provveda a ridimensionarla automaticamente.

In particolare, il programma di ridimensionamento della finestra di comando del Matlab è stato scritto il più generico possibile, infatti, scritto in linguaggio C, (WR.C) permette di spostare e ridimensionare qualsiasi "Finestra" presente sul desktop del MS-Windows sia essa riconoscibile dall'intestazione che dalla sua classe di appartenenza (sintassi: `wr [-c classname | -t caption] left top width height`). Questo, è l'unico programma strettamente legato al sistema operativo MS-Windows in quanto per ottenere il ridimensionamento è stato necessario utilizzare una "API" (Application Programming Interface) del Windows della libreria "user32". Dato che comunque, il toolbox si appoggia a questo programma esterno, nulla vieta di crearne il corrispondente per le altre piattaforme e sistemi operativi dove il MatLab è presente.

Risolto ciò si è preso a modello il programma demo fornito con il MatLab per creare l'interfaccia.

Anche se, il numero di controlli grafici supportati dalla GUI ("Graphical User Interface") del Matlab è discreto, sono stati necessari per l'implementazione dell'interfaccia solo due "uicontrol" tipo "ListBox" una per la visualizzazione della directory, l'altra per la visualizzazione del testo ad esso associato, quattro "uicontrol" tipo "PushButton" per i bottoni di collegamento, running e editing e un "uicontextmenu" con otto "uimenu" ad esso collegati per la creazione del menu a tendina a comparsa (cosiddetto menu di popup) per la visualizzazione delle risorse testuali.

È stato quindi creata una maschera che in una singola schermata presenti tutti i collegamenti ed i possibili comandi.

La realizzazione di questa maschera, (RunMask.m) è stata poi resa tutta in coordinate relative, così da potersi autodimensionare a secondo della risoluzione grafica del computer ospite e delle esigenze degli utenti.

La lettura della directory di base e la costruzione della lista da associare alla "listbox" di "navigazione" è stata poi implementata con al suo interno un primo

piccolo parser, in maniera tale da poter recuperare da i file matlab le informazioni aggiuntive necessarie per la visualizzazione, cioè il nome “libero” (se presente) ed il flag di invisibilità.

Queste informazioni sono state inserite in una lista e conservata nella stessa struttura del listbox, così da poterle poi recuperare senza creare in matlab delle variabili globali, cosa che poteva essere pericolosa, in quanto, lanciando programmi esterni indipendenti, queste ultime potevano essere involontariamente cancellate.

A tal ragione, sono stati presi anche altri accorgimenti per impedire interazioni involontarie, tra l’ambiente, che comunque viene eseguito dal matlab, ed i programmi esterni che l’ambiente stesso provvede a gestire: così si è provveduto anche a rendere invisibile al resto del matlab gli “handler” (puntatori simbolici agli oggetti) sia della “figure” (la maschera grafica con cui il toolbox creato dialoga con l’utente) che degli oggetti in esso contenuti.

Tornando alla stessa lista inserita nel controllo di “navigazione”, al suo interno sono state poi aggiunte altre informazioni supplementari riguardo la posizione dei file e delle directory visualizzate (il percorso assoluto) , del tipo di file, in quanto per scelta di progettazione si vuole dare visibilità solo a determinati tipi di questi (i file matlab, spice ed html, dove però è mostrato soltanto l’ index.htm o index.html) , il nome reale del file (e quindi il comando associato per la sua esecuzione), e nel caso delle directory, il suo stato (implosa, esplosa ed il livello di innestamento e quindi di indentazione) (ReadDir.m).

La seconda parte del parser relativo al recupero delle informazioni addizionali, ossia dei collegamenti ai file multimediali alle immagini ed ai testi collegati, sono invece stati implementati nel modulo principale del sistema, che è poi l’intero gestore degli eventi (MultiIO.m).

Per quanto riguarda i file in formato SPICE e html, sempre in quest’ultimo modulo è presente la ricerca ed il riconoscimento dei file ad essi collegati.

In tale modulo, è stato inserito il codice necessario per eseguire le richieste date dall’utente tramite l’interfaccia. Dunque a secondo dell’evento che si scatena può essere eseguito la funzione matlab, l’esecuzione dello Pspice, l’esplosione o il collassamento delle directory, la creazione del menu di popup legato al bottone dei collegamenti ai documenti, o l’avvio dei relativi programmi per la visualizzazione delle risorse aggiuntive e la loro apertura.

Ovviamente, tutte queste operazioni, sono previste in maniera dinamica; ossia, l' esecuzione del codice di esplosione di una directory, comporta anche la lettura della directory stessa (e quindi il rimando alla apposita funzione scritta), così da ottenere due benefici in contemporanea: la prima, di velocità, in quanto non dovendo essere mostrato il contenuto di directory ancora chiuse, ci si risparmia il tempo di doverle indagare, la seconda, di poter variare all' interno delle directory la strutturazione dei file in esso contenute, e vederne praticamente subito le variazioni nell'ambiente. Stesso vantaggio è dato dal fatto che solamente quando il singolo nome viene selezionato dalla lista di navigazione, allora vengono cercati e costruiti i riferimenti ad i file aggiuntivi collegati, cosicché le variazioni fatte al file vengono anch'esse mostrate nell' immediato.

III.3: Il nuovo "Toolbox" Matlab

Di seguito, sono mostrati e commentati i listati delle funzioni e le relative subroutine sviluppate per la costruzione dell' ambiente integrato.

III.3.1: EnvShell.m

```
function EnvShell(mydir)
%     envhell(mydir)
%
%     Punto di partenza del nuovo ambiente:
%     envshell lancia l' interfaccia grafica mostrando
%     la struttura delle directory e dei file presenti nella
%     directory di partenza "FILE" che deve essere
%     presente nel percorso "mydir" dato dall' utente.
%     Se omissso "mydir", il programma mostra la struttura
%     della directory "FILE"
%     nel percorso dove è presente il file envshell.m
%
%
a=which('EnvShell');           % restituisce il percorso
                               % ed il nome della funzione "etnshell"
b=max(findstr(a,'\'));         % Cerca per l'ultima occorrenza di "/"
a=a(1:b-1);                    % recupera il solo percorso completo
                               % della funzione "etnshell"
addpath (a) ;                  % aggiungilo al path di MatLab
                               % (solo per questa sessione)
if (nargin<1)                  % se etnshell non è chiamata con un
    UserDir=a;                 % argomento opzionale
                               % allora mostra la directory "FILE"
                               % nel path di default
else                            % altrimenti mostra la directory
    UserDir=mydir;% "FILE" presente in "mydir"
```

```

        addpath (mydir);      % ed aggiungi "mydir" al path di
MatLab                               % (solo per questa sessione)
    end

    multiio('initialize',UserDir);
%end function

```

III.3.2: RunMask.m

```

function RunMask(UserDir)
%   RunMask(UserDir)
%
%   funzione di inizializzazione del programma "envshell"
%   Crea l' interfaccia grafica di dialogo e provvedere a
%   ridimensionare la Command Window di Matlab in rapporto
%   70 - 30 % dello schermo
%
    ScreenSize=get(0,'ScreenSize'); %recupero delle
% dimensioni dello schermo [ 0 0 w h]
    Left  =ScreenSize(1);
    Top   =ScreenSize(2);
    Width =ScreenSize(3);
    Height=ScreenSize(4);
    Hc=Height*.70;           % 70% dello schermo all' Explorer
    %   il resto alla Command Window
    xx=19; % dimensione in pixel delle caption standard di Windows!
    CommandWindowSize=[ 1 1+Hc+xx Width Height-Hc-xx];
    MIOWindowSize=[1 1+Height-Hc-xx Width Hc];
    a=sprintf('%s\\wr.exe -t "MATLAB Command Window" %d %d %d
%d',UserDir, CommandWindowSize);
    clc;
    vms(a); % Riposiziona e ridimensiona
% la finestra di comando del MATLAB
% (esegue il comando presente nella variabile a)

    h_FormMain=1000;
    figure(h_FormMain); %creazione della finestra di dialogo

% Lettura Directory per la Lista
[ListBox_List,ListDir]=ReadDir([UserDir '/file'],0);

set(h_FormMain , 'Units','pixels', ...
    'Color',[0.84 0.84 0.84], ...
    'HandleVisibility','off',...
    'NumberTitle','off', ...
    'Name','EnvShell', ...
    'MenuBar','none',...
    'UserData',UserDir, ...
    'Position',MIOWindowSize, ...
    'Tag','FormMain');
%Settaggio delle proprietà della finestra di dialogo
% e creazione e organizzazione dei vari oggetti della finestra
    h_PictureMain = axes( ...
'Parent',h_FormMain, ...
    'Color',[0.84 0.84 0.84], ...
    'Box','off', ...
    'Position',[0 0 1 1], ...
'Tag','PictureMain', ...

```

```

'TickLength',[0 0], ...
    'Visible','on', ...
    'XTickLabel','',...
    'YTickLabel','',...
    'ZTickLabel','',...
    'DataAspectRatioMode','auto',...
    'PlotBoxAspectRatioMode','auto',...
    'XDir','normal',...
    'YDir','reverse',...
    'XLim',[1 size(CData,2)],...
    'YLim',[1 size(CData,1)],...
    'ZLimMode','auto',...
    'XColor',[0 0 0], ...
    'YColor',[0 0 0], ...
    'ZColor',[0 0 0]);
%Visualizzazione figura di sfondo principale
CData=imread('background2.jpg','jpg');
h_imageMain=image( ...
'Parent',h_PictureMain, ...
    'CData',CData, ...
    'Clipping','on', ...
    'BusyAction','cancel', ...
    'CDataMapping','direct', ...
    'Tag','imageMain', ...
    'Interruptible','off');
%Listbox per la visualizzazione del testo o del listato
h_TextBoxNote = uicontrol( ...
'Parent',h_FormMain, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'HorizontalAlignment','left', ...
    'Enable','off', ...
    'Position',[0.28 0.11 0.70 0.335], ...
    'String','', ...
    'Style','listbox', ...
    'Tag','TextBoxNote');
%Bottone di Run
h_ButtonRun = uicontrol( ...
'Parent',h_FormMain, ...
    'Units','normalized', ...
    'Enable','off', ...
    'ListboxTop',0, ...
    'Position',[0.28 0.038 0.20 0.055], ...
    'String','Run',...
    'Callback','multiio(''ButtonRun_Click''),' ...
    'Tag','ButtonRun');
%Bottone di editing
h_ButtonEdit = uicontrol( ...
'Parent',h_FormMain, ...
    'Units','normalized', ...
    'Enable','off', ...
    'ListboxTop',0, ...
    'Position',[0.50 0.038 0.20 0.055], ...
    'String','Edit', ...
    'Callback','multiio(''ButtonEdit_Click''),' ...
    'Tag','ButtonEdit');
%Bottone per i link a documenti ed ipertesti
% collegato ai menu' di popup
h_ButtonHtml = uicontrol( ...
'Parent',h_FormMain, ...
    'Units','normalized', ...
    'ListboxTop',0, ...
    'Position',[0.725 0.038 0.12 0.055], ...

```

```

        'String','More...', ...
        'Enable','off', ...
        'Callback','multiio(''ButtonHtml_Click''),' ...
        'Tag','ButtonHtml');
    %Bottone per i link ai file multimediale
    h_ButtonMedia = uicontrol( ...
'Parent',h_FormMain, ...
    'Units','normalized', ...
    'ListboxTop',0, ...
    'Position',[0.860 0.038 0.12 0.055], ...
    'String','Media', ...
    'Enable','off', ...
    'Callback','multiio(''ButtonMedia_Click''),' ...
    'Tag','ButtonMedia');
    %Rettangolo predisposto per la visualizzazione
    % delle immagini collegate ai file
    h_PictureView = axes( ...
'Parent',h_FormMain, ...
    'Units','normalized', ...
    'Box','on', ...
    'Color',[0.9 0.9 0.9], ...
    'DataAspectRatioMode','auto', ...
    'PlotBoxAspectRatioMode','manual', ...
    'XLimMode','auto', ...
    'YLimMode','auto', ...
    'ZLimMode','auto', ...
    'XDir','normal',...
    'YDir','reverse',...
    'Box','on', ...
    'GridLineStyle','none', ...
    'Position',[ 0.28 0.47 0.69 0.5 ], ...
    'Tag','PictureView', ...
    'TickDirMode','manual', ...
    'TickLength',[0 0], ...
    'Visible','on', ...
    'XTickLabel','',...
    'YTickLabel','',...
    'ZTickLabel','',...
    'XColor',[0 0 0], ...
    'YColor',[0 0 0], ...
    'ZColor',[0 0 0]);
    h_imageView=image( ...
'Parent',h_PictureView, ...
    'CData',[], ...
    'BusyAction','cancel', ...
    'CDataMapping','direct', ...
    'Tag','imageView', ...
    'Interruptible','off');
    %ListBox per il "File Browsing"
    % dove viene proposto la strutturazione ed i file matlab e spice
    h_ListBox = uicontrol( ...
'Parent',h_FormMain, ...
    'Units','normalized', ...
    'BackgroundColor',[0.753 0.753 0.753], ...
    'Position',[0.043 0.038 0.20 0.70], ...
    'String',ListBox_List, ...
    'Style','listbox', ...
    'Tag','ListBox', ...
    'Callback','multiio(''ListBox_Click''),' ...
    'UserData',ListDir);
    %PopUpMenu: menu a comparsa per la visualizzazione
    %dei documenti e dei link ad ipertesti
    h_PopUpMenu=uicontextmenu( ...

```

```

'Parent',h_FormMain, ...
    'Tag','popupmenu');
    %Il primo e' riservato al collegamento diretto
    % fatto con l' apposito comando aggiuntivo
    % o con l'associazione dell'estensione dei file
    h_Menu1=uimenu( ...
'Parent',h_PopUpMenu, ...
    'Tag','menu1', ...
    'Callback','multiio(''menuH_Click''),' ...
    'Visible','off');
    %dal secondo all' ottavo utilizzati per mostrare
    % i documenti Word e pdf presenti nella directory
    % dove e' presente il File in questione e in tutte le
    % sue directory genitrici fino alla "FILE"
    h_Menu2=uimenu( ...
'Parent',h_PopUpMenu, ...
    'Tag','menu2', ...
    'Callback','multiio(''menu_Click''),' ...
    'Visible','off');
    h_Menu3=uimenu( ...
'Parent',h_PopUpMenu, ...
    'Tag','menu3', ...
    'Callback','multiio(''menu_Click''),' ...
    'Visible','off');
    h_Menu4=uimenu( ...
'Parent',h_PopUpMenu, ...
    'Tag','menu4', ...
    'Callback','multiio(''menu_Click''),' ...
    'Visible','off');
    h_Menu5=uimenu( ...
'Parent',h_PopUpMenu, ...
    'Tag','menu5', ...
    'Callback','multiio(''menu_Click''),' ...
    'Visible','off');
    h_Menu6=uimenu( ...
'Parent',h_PopUpMenu, ...
    'Tag','menu6', ...
    'Callback','multiio(''menu_Click''),' ...
    'Visible','off');
    h_Menu7=uimenu( ...
'Parent',h_PopUpMenu, ...
    'Tag','menu7', ...
    'Callback','multiio(''menu_Click''),' ...
    'Visible','off');
    h_Menu8=uimenu( ...
'Parent',h_PopUpMenu, ...
    'Tag','menu8', ...
    'Callback','multiio(''menu_Click''),' ...
    'Visible','off');

    %Mostra se presente il logo di presentazione.
    Filename=FileExist(UserDir,'start.jpg');
    if (length(Filename)>0)
        [CData,map]=imread(Filename);
        [Y,X,C]=size(CData);
        set (h_imageView,'CData',CData);
        set (h_PictureView,'YLim',[1 Y]);
        set (h_PictureView,'XLim',[1 X]);
        set (h_PictureView,'DataAspectRatioMode','auto');
        set (h_PictureView,'PlotBoxAspectRatio',[X Y 1]);
    end;

%end function

```

III.3.3: MultiIO.m

```
function multiio(action,UserDir)
% multiio(action,UserDir)
%
% Gestore degli eventi della maschera di dialogo
%
% recupera i dati e i collegamenti dei vari file
% li mostra nella finestra envshell
% e li esegue.
%

if nargin<1, %nargin =number of argoment in input
    action='initialize';
end;

[h_obj,h_fig]=gcbo; % Recupero dell' handler della form e
                    % dell' oggetto che ha prodotto l'evento

switch action
case 'initialize'
    RunMask(UserDir);

case 'ListBox_Click' % Click sulla listbox delle directory
    ListBox=get(h_obj);
    ListIndex=ListBox.Value;
    ListItem=ListBox.UserData(ListIndex);
    % La riga selezionata e' un .m .html (.htm) .pdf o .cir
    if (ListItem.DirState(1)==0) % carica e mostra
        bitmap,note,Media e HTML file
        AppPath=strrep(ListItem.AppPath,'\','/'); % Tutti i Path
        con lo "\" in "/"!
        cd (AppPath);
        Command=ListItem.Command;
        %valori di Default
        BitMap=FileExist(AppPath,[Command,'.fig']);
        NoteFl=FileExist(AppPath,[Command,'.txt']);
        HtmlFl=FileExist(AppPath,[Command,'.html']);
        MMedia=FileExist(AppPath,[Command '.mmf']);
        Command=FileExist(AppPath ,Command);
        switch ListItem.FileType
        case 'isMatLab'
            [titolo,BitMap,NoteFl,HtmlFl,MMedia]=
recuperadati(Command);
        case 'isIndexHtl'
            %case {'isIndexHtml','isPDF'}
            HtmlFl=Command
            Command='';
        case 'isSpice'
            if (isempty(NoteFl))
                NoteFl=Command;
            end %if
        end % switch-case

    % Mostra il disegno se presente il collegamento
    Filename=FileExist(AppPath,BitMap);
    if (length(Filename)==0)
        CurrDir=get(h_fig,'UserData');
        Filename=FileExist(CurrDir,'/def.bmp');
```

```

end; %if
Setfigure(h_fig,Filename);

% Mostra il testo di Commento se presente il collegamento
h_TextBoxNote=findobj(h_fig,'Tag','TextBoxNote');
set(h_TextBoxNote,'String','');
set(h_TextBoxNote,'Enable','off');
Filename=FileExist(AppPath,NoteFl);
if (length(Filename)>0)
    txt=LoadText(Filename);
    set(h_TextBoxNote,'Value',1)
    set(h_TextBoxNote,'String',txt)
    set(h_TextBoxNote,'Enable','on');
end;

% Attiva i tasti Run e Edit se esiste un comando da
% eseguire e salvane il comando in 'UserData'
h_ButtonRun=findobj(h_fig,'Tag','ButtonRun');
h_ButtonEdit=findobj(h_fig,'Tag','ButtonEdit');
if (~isempty(Command))
    set(h_ButtonRun,'Enable','on');
    set(h_ButtonRun,'UserData',Command);
    set(h_ButtonEdit,'Enable','on');
    set(h_ButtonEdit,'UserData',Command);
else
    set(h_ButtonRun,'Enable','off');
    set(h_ButtonRun,'UserData',[]);
    set(h_ButtonEdit,'Enable','off');
    set(h_ButtonEdit,'UserData',[]);
end;

% Attiva il tasto Link al html se presente il
% collegamento e salvane il comando in 'UserData'
h_ButtonHtml=findobj(h_fig,'Tag','ButtonHtml');
h_Menu1=findobj(h_fig,'Tag','menu1');
ClearPopUpMenu(h_fig)
if (~isempty(HtmlFl))
    Name=HtmlFl;
    if (HtmlFl(1)=='*')
        CurrDir=get(h_fig,'UserData');
        HtmlFl=['file://', CurrDir,HtmlFl(2:length(HtmlFl))];
        %allo stesso livello della directory FILE
    elseif (HtmlFl(1)=='.')
        HtmlFl=['file://', AppPath,HtmlFl(2:length(HtmlFl))];
%nella directory corrente
    end %if
    set(h_Menu1,'label',Name);
    set(h_Menu1,'visible','on');
    set(h_Menu1,'UserData',[HtmlFl]);
else
    set(h_Menu1,'label','');
    set(h_Menu1,'visible','off');
    set(h_Menu1,'UserData',[]);
end;
FindDocs(h_fig,AppPath);
%Cerca tutti i documenti Word e pdf dalla
% directory corrente a risalire fino alla
% genitrice FILE.
set(h_ButtonHtml,'Enable',ExistDocs(h_fig));

% Attiva il tasto Media se presente il collegamento
% e salvane il comando in 'UserData'
h_ButtonMedia=findobj(h_fig,'Tag','ButtonMedia');

```

```

Filename=FileExist(AppPath,MMedia);
if (length(Filename)==0)
    CurrDir=get(h_fig,'UserData');
    MMedia='/def.mpg';
    Filename=FileExist(CurrDir,MMedia);
end; %if
if (length(Filename)>0)
    set(h_ButtonMedia,'Enable','on');
    set(h_ButtonMedia,'UserData',Filename);
else
    set(h_ButtonMedia,'Enable','off');
    set(h_ButtonMedia,'UserData',[]);
end;

% La riga selezionata e' una directory da espandere
elseif (ListItem.DirState(1)==1) %Espandi la directory
    DisableButtons(h_fig); %disabilita i bottoni
    OldLabelList=ListBox.String;
    OldUserData=ListBox.UserData;
    NewLabelList=OldLabelList(1:ListIndex-1,:);
    if (length(NewLabelList)==0)
        NewLabelList=strrep(OldLabelList(ListIndex,:),'+','-');
    else
        NewLabelList=str2mat(NewLabelList,strrep(
OldLabelList(ListIndex,:),'+','-'));
    end %if
    NewUserData=OldUserData(1:ListIndex);
    NewUserData(ListIndex).DirState(1)=2; % la directory e'
% espansa
    set(h_obj,'Enable','off');
    [XListBox_List,XListDir]=ReadDir(ListItem.Command,
ListItem.DirState(2)+1);
    set(h_obj,'Enable','on');
    if (length(XListBox_List)~=0)
        NewLabelList=str2mat(NewLabelList,XListBox_List);
        NewUserData=[NewUserData,XListDir];
    end;
    x=size(OldLabelList);
    if (ListIndex<x(1))
        NewLabelList=str2mat(NewLabelList,
OldLabelList(ListIndex+1:x,:));
        NewUserData=[NewUserData,OldUserData(ListIndex+1:x)];
    end;
    set(h_obj,'String',NewLabelList);
    set(h_obj,'UserData',NewUserData);

% La riga selezionata e' una directory da collassare
elseif (ListItem.DirState(1)==2) % Collassa la
% directory Corrente
    DisableButtons(h_fig); %disabilita i bottoni
    OldLabelList=ListBox.String;
    OldUserData=ListBox.UserData;
    NewLabelList=OldLabelList(1:ListIndex-1,:);
    if (length(NewLabelList)==0)
        NewLabelList=strrep(OldLabelList(ListIndex,),'-','+');
    else
        NewLabelList=str2mat(NewLabelList,strrep(
OldLabelList(ListIndex,),'-','+'));
    end %if
    NewUserData=OldUserData(1:ListIndex);
    NewUserData(ListIndex).DirState(1)=1; % la directory
% e' collassata
    IndentLevel=OldUserData(ListIndex).DirState(2)+1;

```

```

        x=size(OldLabelList);
        for i=ListIndex+1:x(1)
            if (OldUserData(i).DirState(2)<IndentLevel)
                break; %for
            end; %if
        end; %for
        if (i<x(1))
            NewLabelList=str2mat(NewLabelList,
OldLabelList(i:x(1),:));
            NewUserData=[NewUserData,OldUserData(i:x(1))];
        elseif(i==x(1))
            if (OldUserData(i).DirState(2)<IndentLevel)
                NewLabelList=str2mat(NewLabelList,
OldLabelList(i:x(1),:));
            NewUserData=[NewUserData,OldUserData(i:x(1))];
            end; %if;
        end; %if-elseif
        set (h_obj,'String',NewLabelList);
        set (h_obj,'UserData',NewUserData);
    end; %if

    case 'ButtonRun_Click' % Click sul Bottone di Run
        Command=get(h_obj,'UserData');
        a=max(findstr(Command,'/')); % Cerca l'ultima occorrenza di "/"
        AppPath='.';
        if (~isempty(a)) % in Name è presente un percorso
% oltre che al nome del file
            AppPath=Command(1:a); % recupero del Path
            Command=Command(a+1:length(Command)); % e del comando
        end;
        cd (AppPath);
        set (h_obj,'String','Running...');
        MatLabCommand=Command(1:max(findstr(lower(Command),'.m'))-1);
        if isempty(MatLabCommand), % non e' un .m allora
            % deve essere uno SPICE
            % esegui come comando esterno
            Command=get(h_obj,'UserData');
            vms(['start /w "' Command '"']);
            c=[Command(1:length(Command)-3) 'out'];
% se lo Spice ha prodotto un output mostralo con
% l'editor interno
            if length(FileExist(AppPath,c))>0
                edit (c);
            end% if
            c=[Command(1:length(Command)-3) 'dat'];
% se lo Spice ha prodotto un datafile mostralo con
% il probe dello spice
            if length(FileExist(AppPath,c))>0
                vms(['start /w "' c '"']);
            end% if
        else
            feval(MatLabCommand); % esegui il programma Matlab
        end % if
        [h_obj,h_fig]=gcbo; % Ripeto il recupero dell' handler
    caso
% mai ci fosse stato un "clear" nella
% routine appena eseguita!
        set (h_obj,'String','Run');
        h=msgbox('Esecuzione Completata','','error');
        set(h,'unit','normalized');
        pos=get(h,'position');
        pos(1,1)=0.67;
        pos(1,2)=0.07;

```

```

        set(h,'position',pos);

    case 'ButtonEdit_Click' % Click sul Bottone di Editing
        edit(PatchString(get(h_obj,'UserData')))

    case 'ButtonHtml_Click' % Click sul Bottone di collegamento
% ai documenti
        h_pop=findobj(h_fig,'Tag','popupmenu');
        set(h_obj,'Units','pixels');
        pos=get(h_obj,'position');
        set(h_obj,'units','normalized');
        set(h_pop,'position',pos(1,1:2));
        set(h_pop,'Visible','on');

    case 'menu_Click'% Click sul menu di collegamento ai doc e pdf
        vms(get(h_obj,'UserData'));

    case 'menuH_Click'% Click sul menu di collegamento a ipertesti
        web(get(h_obj,'UserData'));

    case 'ButtonMedia_Click'% Click sul Bottone di collegamento
% ai file multimediali
        vms(['mplayer.exe -play -close ' get(h_obj,'UserData')]) ;

    end; %Switch-case

% end function multiio

%*****
%
% SubRoutine interne:
%
%*****

function Txt=LoadText(FileName)
% Txt=LoadText(FileName)
%     Copia il file ASCII Filename
%     nella stringa Txt
%     per essere inserita in una ListBox

    Txt='';
    f=fopen(FileName,'rt');
    if (f>0)
        while (~feof(f))
            s=fgetl(f);
            s=strrep(s,' ',' '); % Rimpiazza i TAB con quattro
% spazi (la ListBox non li visualizza)
            Txt=[Txt,s,'|'];
        end %while
        if (length(Txt)>1)
            Txt=Txt(1:length(Txt)-1);
        end %if
        fclose(f);
    end % if
% end function LoadText

function ret=PatchString(s)
%     ret=PatchString(s)
%     Corregge l' incompatibilita' delle stringhe
%     contenenti al proprio interno
%     apici duplicandoli cosi' da consentirne l'editing.

```

```

        ret='';
        for i=1:length(s)
            if (s(i)==' ')
                ret=[ret, ' ']; % aggiunge un apice per ogni
            % apice presente nella stringa s
            end % if
            ret=[ret,s(i)];
        end % for
    % end function PatchString

function DisableButtons(h_fig)
% Disabilita i bottoni e elimina l' immagine e il testo mostrato

h=findobj(h_fig,'tag','ButtonRun');
set(h,'Enable','off');
h=findobj(h_fig,'tag','ButtonEdit');
set(h,'Enable','off');
h=findobj(h_fig,'tag','ButtonHtml');
set(h,'Enable','off');
h=findobj(h_fig,'tag','ButtonMedia');
set(h,'Enable','off');
h=findobj(h_fig,'Tag','TextBoxNote');
set(h,'String','');
    set(h,'Enable','off');
h=findobj(h_fig,'Tag','imageView');
CurrDir=get(h_fig,'UserData');
Filename=FileExist(CurrDir,'start.jpg');
Setfigure(h_fig,Filename);
% end function DisableButtons

function [titolo,bitmap,notefl,htmlfl,MMedia]=recuperadati(FileName)
% Recupera dal file matlab "FileName" i comandi
% aggiuntivi per il titolo, l' immagine da visualizzare,
% il file testo da mostrare,il file multimediale
% e la pagina html collegata nel particolare
%
% BitMap = Collegamento ad un file immagine (nei formati
% accettati dal Matlab) recuperato nel file .m
% dalla linea "% bitmap:"
% se non esistente viene posto a []
%
% NoteFl = Collegamento ad un file testo in formato ASCII
% recuperato anch'esso nel file .m dalla linea
% "% notefl:"
% se non esistente viene posto a []
%
% HtmlFl = Collegamento ad un file html recuperato
% anch'esso nel file .m dalla linea "% htmlfl:"
% se non esistente viene posto a []
%
% Mmedia = Collegamento ad un file audio (.wav o similari)
% recuperato anch'esso nel file .m dalla linea
% "% mmedia:"
% se non esistente viene posto a []

titolo=FileName;
bitmap='';
htmlfl='';
MMedia='';
notefl=FileName;
hideme=0;

```

```

        f=fopen(FileName,'rt');
        while (~feof(f))
s=fgetl(f);
            if (length(s)>=9)
                if (strcmpi(s(1:9),'% HideMe:'))
                    hideme=1;
                end %if
                if (strcmpi(s(1:9),'% titolo:'))
                    titolo=s(10:length(s));
                end %if
                if (strcmpi(s(1:9),'% bitmap:'))
                    bitmap=s(10:length(s));
                end %if
                if (strcmpi(s(1:9),'% notefl:'))
                    notefl=s(10:length(s));
                end %if
                if (strcmpi(s(1:9),'% htmlfl:'))
                    htmlfl=s(10:length(s));
                end %if
                if (strcmpi(s(1:9),'% mmedia:'))
                    MMedia=s(10:length(s));
                end %if
            end %if
        end %while
        if (hideme==1)
            titolo='';
            bitmap='';
            notefl='';
            htmlfl='';
            MMedia='';
        end %if
        fclose(f);
    %end function recuperadati

```

```

function FindDocs(h_fig,AppPath);
% Cerca a retroso a partire dalla directory corrente fino
% alla dorectory genitrice FILE, tutti file in formato
% MS-Word (*.doc) e Adobe Acrobat (*.pdf) e li aggiunge
% al menu popup collegato al bottone "More..."
% recupera fino a 7 documenti

k=2;
Name=strrep(AppPath,'\','/'); %Tutti i Path con lo "\" in "/"
for j=2:8
    c=length(Name);
    if strncmpi(Name(c-4:c),'file',4)
        break
    end;
    a= dir ([Name '*.doc']);
    len=size(a);
    for i=1:len(1) %len non è una funzione ma una variabile!!
c=a(i).name;
        h_mn=findobj(h_fig,'Tag',['menu' k+48]);
        set(h_mn,'Visible','on');
        set(h_mn,'label',c);
        set(h_mn,'UserData',['start "' Name c '"']);
        k=k+1;
    end %for i
    a= dir ([Name '*.pdf']);
    len=size(a);
    for i=1:len(1) %len non è una funzione ma una variabile!!
        c=a(i).name;

```

```

        h_mn=findobj(h_fig,'Tag',['menu' k+48]);
        set(h_mn,'Visible','on');
        set(h_mn,'label',c);
        set(h_mn,'UserData',['start "' Name   c "']);
        k=k+1;
    end %for i
    c=max(findstr(Name(1:length(Name)-1),'/')); % Cerca l'ultima
% occorrenza di "/"
    if (c>0) % in Name è presente un percorso
%oltre che al nome del file
        Name=Name(1:c); %Accorcia il Path
    end;
end; %for j
%end function FindDocs

```

```

function y=ExistDocs(h_fig)
% Controlla se almeno una voce del menu e' attiva
    ok=0;
    for (i=1:8)
        ok=ok | strcmp(get(findobj(h_fig,'Tag',['menu'
i+48]),'Visible'),'on');
    end % for
    if (ok==1),
        y='on';
    else
        y='off';
    end % if
%end function ExistsDocs

```

```

function ClearPopUpMenu(h_fig)
% Svuota il Menu di PopUp del bottone "More..."
    for k=1:8
        h_mn=findobj(h_fig,'Tag',['menu' k+48]);
        set(h_mn,'Visible','off');
        set(h_mn,'label','');
        set(h_mn,'UserData',[]);
    end %for
%end function

```

```

function Setfigure(h_fig,Filename)
% Carica l'immagine Filename nell' apposito
% spazio dell' ambiente
%
    h_PictureView=findobj(h_fig,'Tag','PictureView');
    h_imageView=findobj(h_fig,'Tag','imageView');
    if (length(Filename)>0)
        [CData,map]=imread(Filename);
        [Y,X,C]=size(CData);
        set (h_imageView,'CData',CData);
        set (h_PictureView,'YLim',[1 Y]);
        set (h_PictureView,'XLim',[1 X]);
        set (h_PictureView,'DataAspectRatioMode','auto');
        set (h_PictureView,'PlotBoxAspectRatio',[X Y 1]);
    else
        set(h_imageView,'CData',[]);
    end;
%end function

```

III.3.4: ReadDir.m

```
function [ListNomi,ListDir]=ReadDir(CurrDir,IndentLevel,Titolo);
% [ListNomi,ListDir]=ReadDir(CurrDir,IndentLevel);
% Crea una Array di "blank padded" stringhe per essere inserita
% in una proprieta' "String" di una ListBox e una Lista
% Strutturata per la proprieta' "UserData" della stessa
% la prima contenente i "titoli" recuperati dalle "% titolo:"
% dei file contenuti nella direcotry da leggere (CurrDir) o dal
% nome del file se la linea "% titolo:" non e' presente la
% seguente contenente alcune delle informazioni necessarie al
% recupero dei dati da parte del programma "multio":
%
% ListNomi = struct
%
% AppPath= Percorso dove si trova il file
%
% Titolo = Titolo recuperato nel file .m dalla linea "% titolo:"
% se non presente titolo viene posto al nome del file
%
% Command= il nome del file vero e proprio compreso di percorso
%
% DirState= doppio flag [a,b]:
% a= 0 se la linea e' un file
% a= 1 se la linea e' il titolo di una directory Implosa
% a= 2 se la linea e' il titolo di una directory Esplosa
%
% b e' il livello di indentazione per le directory innestate
%
% FileType= stringa contenente il tipo di file attualmente puo'
% essere solo di quattro tipi autoesplicativi:
% 'isMatLab'
% 'isIndexHtml'
% 'isSpice'
% 'isPDF'
%
% IndentLevel e' il livello di indentazione
% richiesto dall' utente

ListDir=struct('AppPath',[],'Titolo',[],'Command',[],'DirState',
',[0 0],'FileType',[]);

nti=' ';
for (i=1:IndentLevel)
    nti=[nti,'  '];
end; %for
odr='- ';
cdr='+ ';
nti=[nti,'  '];
OldDir=cd;
labellist='';
j=1;
cd (CurrDir); %Inserisce il percorso relativo
CurrDir=cd; %Recupera il percorso completo
a=dir;
len=size(a);
titolo='';
for i=1:len(1) %len non è una funzione ma una variabile!!
len=size(a)
c=a(i).name;
b=length(c);
```

```

if (a(i).isdir==0)
    type=wathis(c);
    if (~isempty(type))
        switch type
            case 'isMatLab'
                titolo=RecuperaTitolo(c);
            case 'isIndexHtml'
                titolo='html pages';
            case 'isSpice'
                titolo=c(1:b-4);
            case 'isPDF'
                titolo=c;
        end % switch-case
        if (length(titolo)>0)
            ListDir(j).AppPath=[CurrDir, '/'];
            ListDir(j).Titolo=titolo;
            ListDir(j).Command=c;
            ListDir(j).FileType=type;
            ListDir(j).DirState=[0,IndentLevel]; % 0 non è una
directory (=>file!)
            if (length(labelist)>0)
                labelist=str2mat(labelist , [nti, titolo] );
            else
                labelist=[nti,titolo];
            end; %if (length(labelist)>0)
            j=j+1;
        end; %if (length(titolo)>0)
    end %if (~isempty(type))
else %if (a(i).isdir==0)
    if ( ~( strcmpi(c, '..') | strcmpi(c, '.') ) )
        ListDir(j).AppPath=[CurrDir, '/'];
        ListDir(j).Titolo=c;
        ListDir(j).Command=[CurrDir, '/', c];
        ListDir(j).DirState=[1,IndentLevel]; % 1 directory
Implosa (non visibile il contenuto)
        ListDir(j).FileType=[];
        j=j+1;
        if (length(labelist)>0)
            labelist=str2mat(labelist , [nti, cdr, c]);
        else
            labelist=[nti, cdr, c];
        end; %if (length(labelist)>0)
    end % if ( ~( strcmpi(c, '..') | strcmpi(c, '.') ) )
end %if (a(i).isdir==0)
end %for
[ListNomi,ListDir]=SortDir(labelist,ListDir);
cd (OldDir);

```

```

%*****
%
% SubRoutine interne:
%
%*****

```

```

function titolo=RecuperaTitolo(FileName)
    titolo=FileName;
    hideme=0;
    f=fopen(FileName, 'rt');
    while (~feof(f))
        s=fgetl(f);
        if (length(s)>=9)
            if (strcmpi(s(1:9), '% HideMe:'))

```

```

        hideme=1;
    end %if
    if (strcmpi(s(1:9),'% titolo:'))
        titolo=s(10:length(s));
    end %if
    end %if
end %while
if (hideme==1)
    titolo='';
end %if
fclose(f);
%end function RecuperaTitolo

function [ListNomi,ListDir]=SortDir(LNomi,LDir)
%       [ListNomi,ListDir]=SortDir(LNomi,LDir)
%       Ordina alfabeticamente la directory letta da ReadDir

    ListDir=struct('AppPath',[],'Titolo',[],'Command',[],'DirState
',[0 0],'FileType',[]);

    a=size(LNomi);
    TmpNomi='';
    ListNomi='';
    for i=1:a(1)
        TmpNomi=[TmpNomi ;[LNomi(i,:), i] ];
    end
    TmpNomi=sortrows(TmpNomi);
    for i=1:a(1)
        Idx=TmpNomi(i,a(2)+1)+0;
        ListNomi=[ListNomi;LNomi(Idx,:)];
        ListDir(i)=LDir(Idx);
    end
%end function

%*****
%versione switch-case
function y=wathis(Filename)
    y='';
    b=length(Filename);
    if (b>1 & strcmpi(Filename(b-1:b),'.m'))
        y='isMatLab';
        return;
    elseif (b>3 & strcmpi(Filename(b-3:b),'.cir'));
        y='isSpice';
        return;
    elseif (strcmpi(Filename,'index.html') | strcmpi(Filename,
'index.htm'))
        y='isIndexHtml';
        return;
    % elseif (b>3 & strcmpi(Filename(b-3:b),'.pdf'))
    %     y='isPDF';
    %     return;
    end %if
%end function

%versione single-if
function y=isMatLab(Filename)
    b=length(Filename);
    y=(b>1 & strcmpi(Filename(b-1:b),'.m'));
    return;
%end function

function y=isPDF(Filename)

```

```

    b=length(Filename);
    y= (b>3 & strcmpi(Filename(b-3:b),'.pdf'));
    return;
%end function

function y=isSpice(Filename)
    b=length(Filename);
    y= (b>3 & strcmpi(Filename(b-3:b),'.cir'));
    return;
%end function

function y=isIndexHtml(Filename)
    y=(strcmpi(Filename,'index.html') |
strcmpi(Filename,'index.htm'));
return;
%end function

%Versione con ricerca del comando .end da un ascii qualsiasi (senza
controllo dell' estensione) [più lenta]
%function y=isSpice(Filename)
% f=fopen(Filename,'rt');
% y=0;
% if (fseek(f,-30,'eof')===-1),           %cerca fra gli ultimi 30
caratteri!
%     return;
% end %if
% while (~feof(f))
%     s=strtok(fgetl(f));
%     if ~ischar(s)                       % se non sono caatteri allora
esci
%         return;
%     end;
%     if strcmpi(s, '.END')
%         y=1;
%         return;
%     end %if
% end %while
% fclose(f);
%end function

```

III.3.5: FileExist.m

```

function Filename=FileExist(Path,Name)
%     Filename=FileExist(External_Path,Name)
%
%     Controlla se il file "Name" esiste
%
%     Prima nel percorso corrente
%     Altrimenti nel percorso Alternativo
%     specificato da "Path"
%
%     In Name è possibile specificare anche
%     un percorso completo che verrà eliminato
%     per la ricerca con il percorso alternativo
%
%     Ritorna il nome del file dove trovato
%     con il percorso completo
%     altrimenti matrice vuota.

```

```

Filename='';
Name=strrep(Name,'\','/'); % Tutti i Path con lo "\" in "/"!
a=max(findstr(Name,'/')); % Cerca l'ultima occorrenza di "/"
if (length(a)>0) % in Name è presente un percorso
% oltre che al nome del file
    StrippedName=Name(a+1:length(Name));
else
    StrippedName=Name;
end;
Path=strrep(Path,'\','/');
a=max([0 findstr(Path,'/')]); % Cerca l'ultima occorrenza di "/"
if (a~=length(Path)) % se non e' presente nel Path
    Path=[Path,'/']; % aggiungilo
end;
if (length(Name)>0)
    [a,void]=size(dir(Name));
    [b,void]=size(dir([Path,StrippedName]));
    if (a>0)
        Filename=Name;
    elseif (b>0)
        Filename=[Path,StrippedName];
    end; %if
end; % if
%end function

```

III.3.6: Listato del programma WR.C

```

/*
Programma per il "resizing" di una generica finestra presente sul
desktop di Windows (versione 4.0 e superiori)
E' possibile ridimensionare qualsiasi finestra, riconoscibile o per
il nome (Caption) della finestra (opzione -t) o per la classe cui
appartiene (opzione -c)
*/

#include <windows.h>
#include <wtypes.h>
#include <stdio.h>
#include <string.h>

void main(int argc, char *argv[])
{
    long *h;
    int x,y,cx,cy;
    /*char buf[100];*/
    if (argc<7)
    {
        printf("Usage: %s [-c classname | -t caption] left top
width height\n",argv[0]);
        return;
    }
    x=atoi(argv[3]);
    y=atoi(argv[4]);
    cx=atoi(argv[5]);
    cy=atoi(argv[6]);

    if (strcmpi(argv[1],"-c")==0 )
    {
        h=(long *)FindWindow(argv[2],NULL);

```

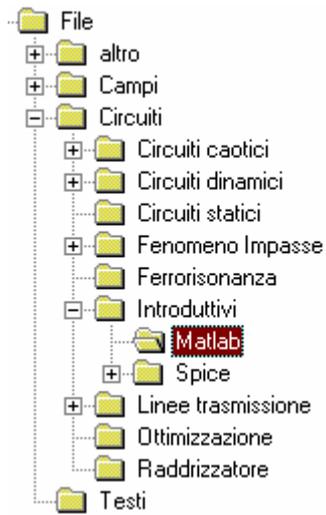
```
    }
    else
{
        h=(long *)FindWindow(NULL,argv[2]);
    }

    /*sprintf(buf,"Handler: %p\n",h);
    MessageBox(NULL, buf, "System information",
MB_ICONINFORMATION); */
    if (h!=NULL)
    {
        SetWindowPos(h,NULL,x,y,cx,cy,0);
    }
}
```


III.4: Un “dimostrativo”

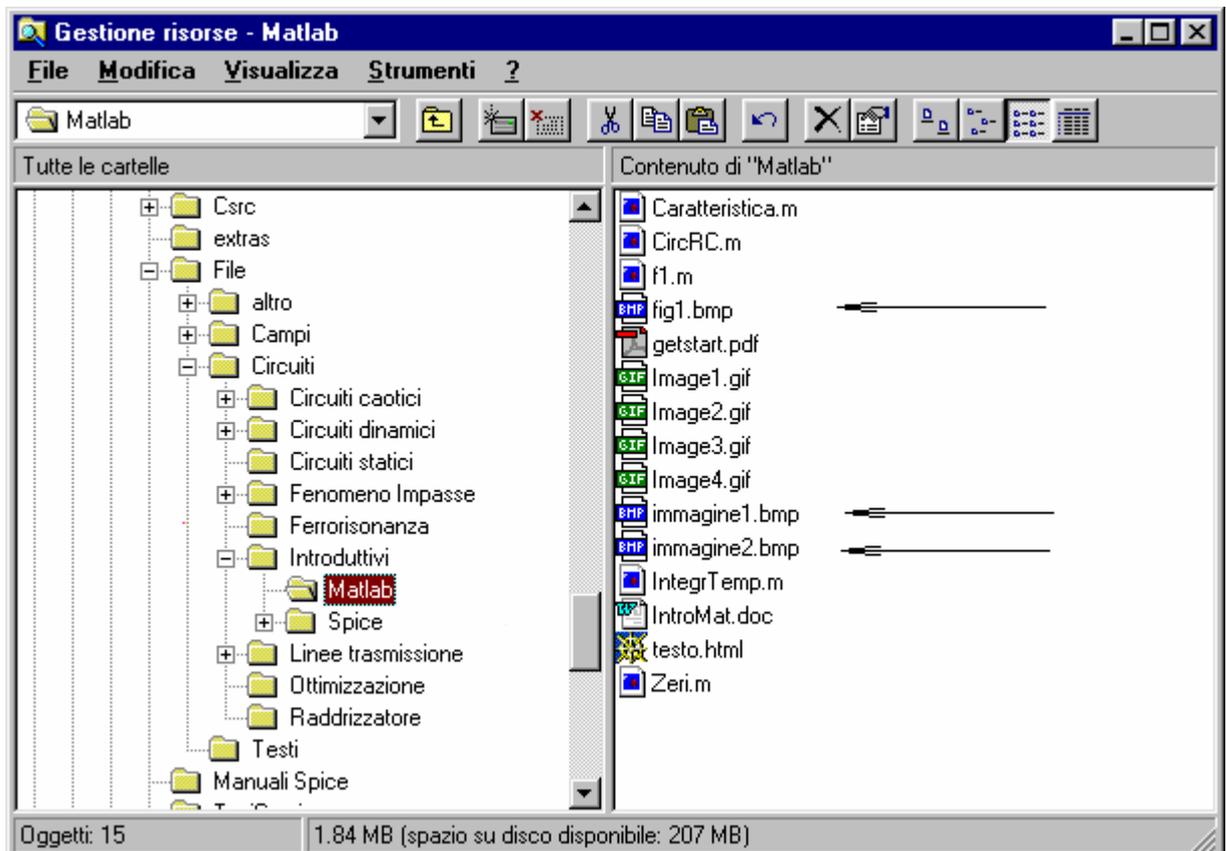
Il risultato finale, per provare, testare e controllare l'utilizzabilità dell'ambiente, è stata la realizzazione di un esempio d'uso concreto, utilizzando del materiale già presente ed in uso nei corsi di riferimento dell'ambito dell'Elettrotecnica.

La realizzazione del demo è stata quindi organizzata in questa maniera:

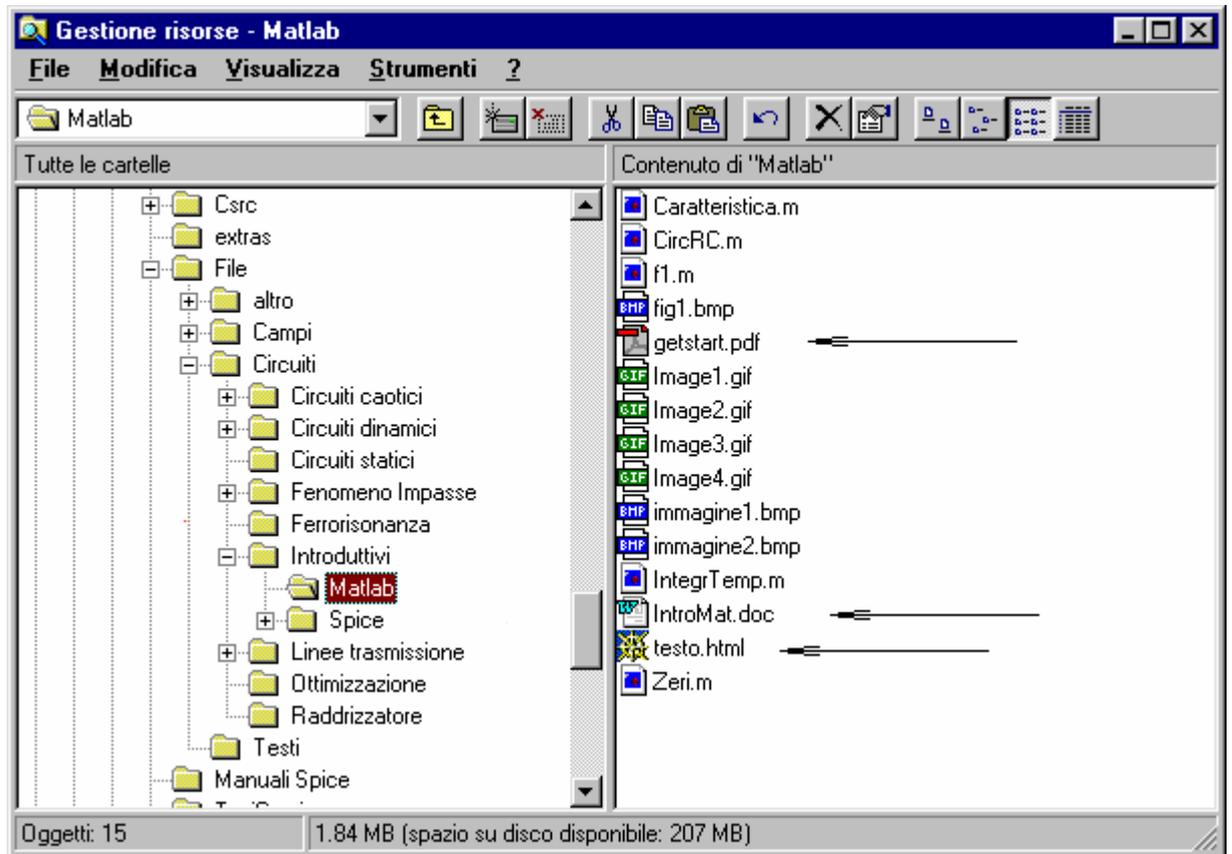


Utilizzando semplicemente l'explorer del MS-Windows, è stata costruita, partendo da una directory nominata FILE la struttura delle directory che dovevano ospitare il materiale didattico in formato SPICE e Matlab, ed all'interno di ognuna di queste è stato inserito il relativo materiale, così da ottenere in maniera ordinata, ciò che si voleva presentare.

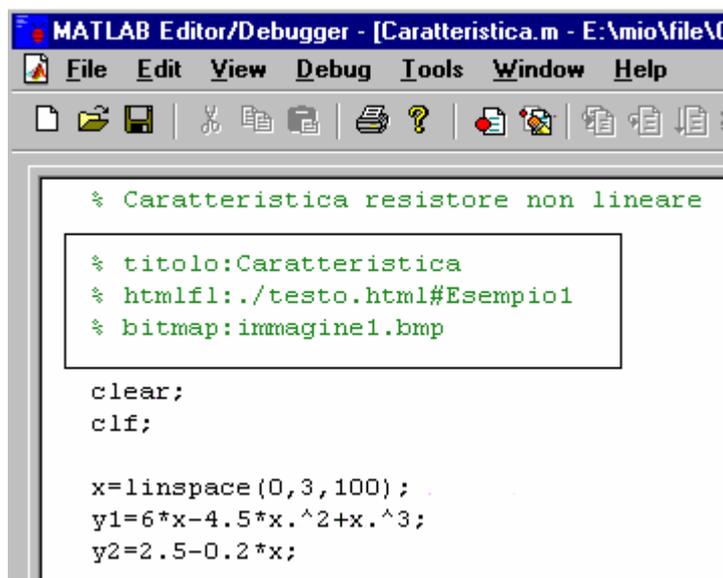
Una volta creata la struttura, sempre e solamente utilizzando l'explorer sono state riempite le directory con gli esempi appropriati e insieme a questi sono stati inseriti anche i file multimediali a cui si voleva far riferimento.



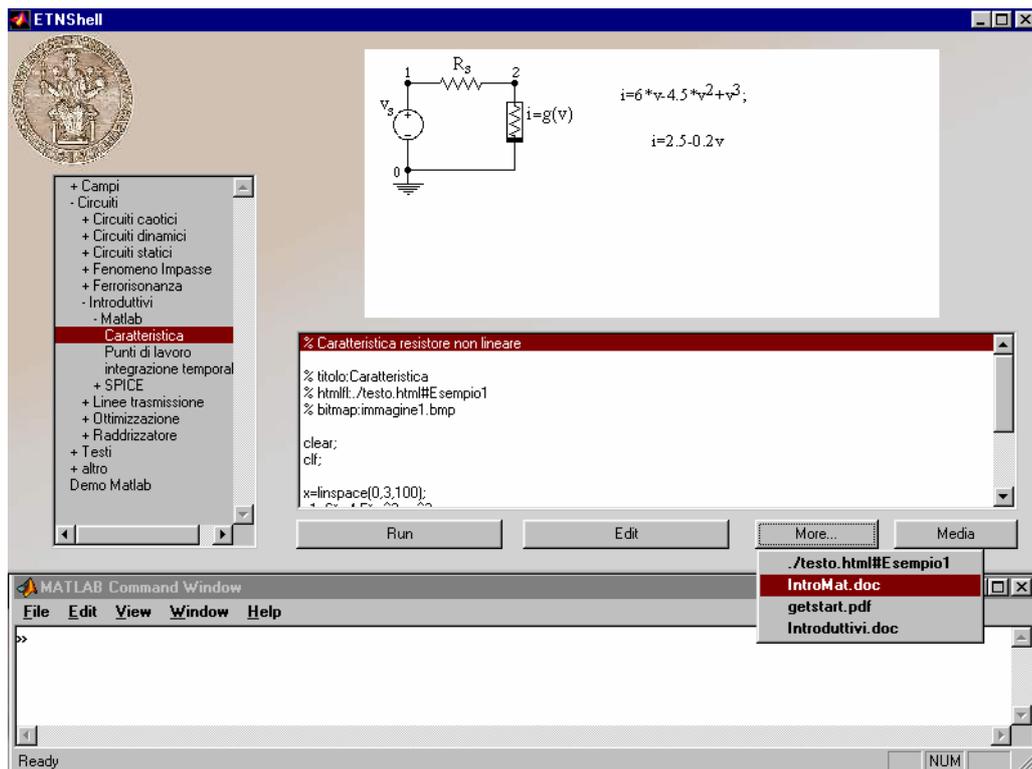
Non solo, ma sono state inserite all'interno delle singole cartelle, nella strutturazione, anche le relative documentazioni che si volevano rendere visibili per l'esempio in particolare:



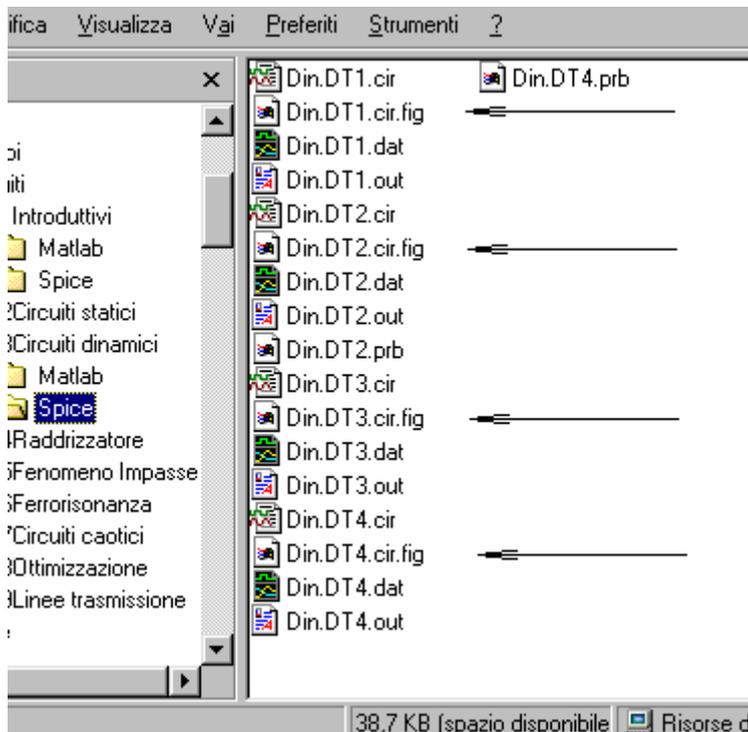
A questo punto, lanciando l'ambiente realizzato, e già possibile vedere ed



utilizzare gli esempi proposti, non solo, ma utilizzando l'ambiente stesso sono stati editati i singoli listati per inserire al loro interno le istruzioni necessarie ai collegamenti diretti delle risorse aggiuntive, così da ottenere il risultato finale:



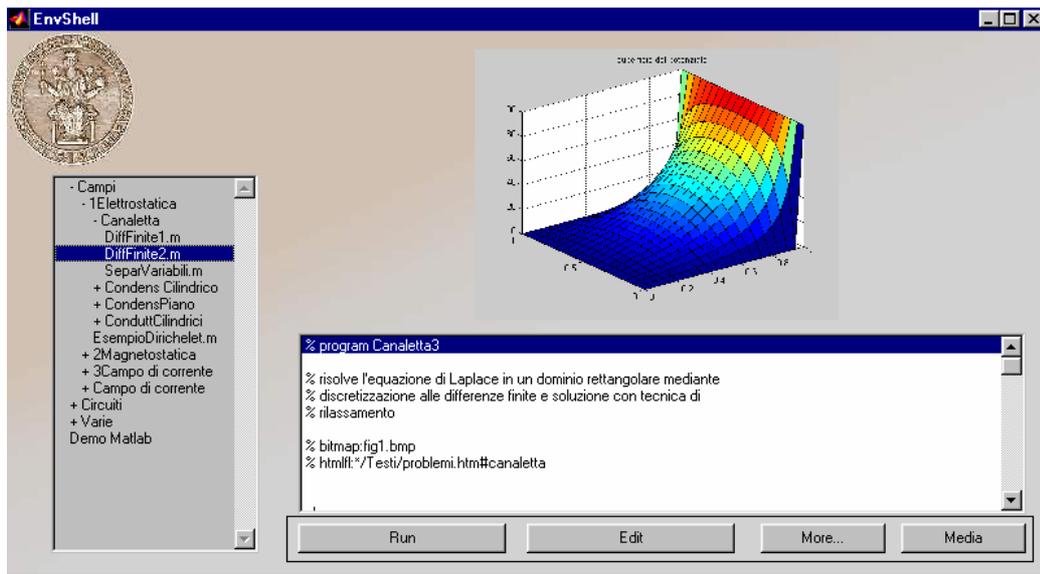
Da notare che oltre al link appositamente inserito nel file matlab ad una pagina html, il sistema a proposto in automatico tutti documenti word ed acrobat trovati a partire dalla directory corrente a retroso.



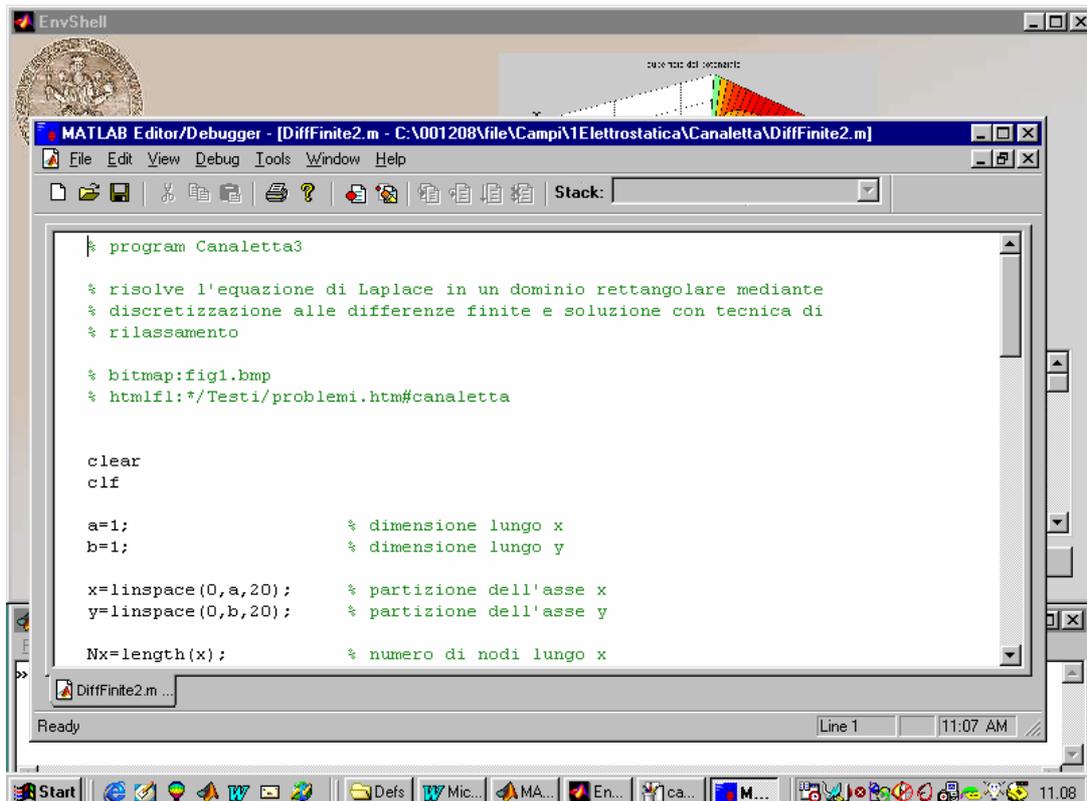
Per quanto riguarda i listati in formato spice, la sequenza è stata esattamente la stessa, con la differenza di dover modificare, invece che il listato, i nomi dei file che si volevano far comparire collegati.

Ciò fatto il pacchetto è pronto per essere distribuito agli utenti i quali una volta lanciato l'ambiente possono lanciare, editare, visualizzare i documenti collegati. Negli esempi fatti, comunque, non sono state utilizzate tutte le potenzialità dell' ambiente, per cui il multimediale resta collegato a quello di "default" fissato nella directory dove è installato il toolbox.

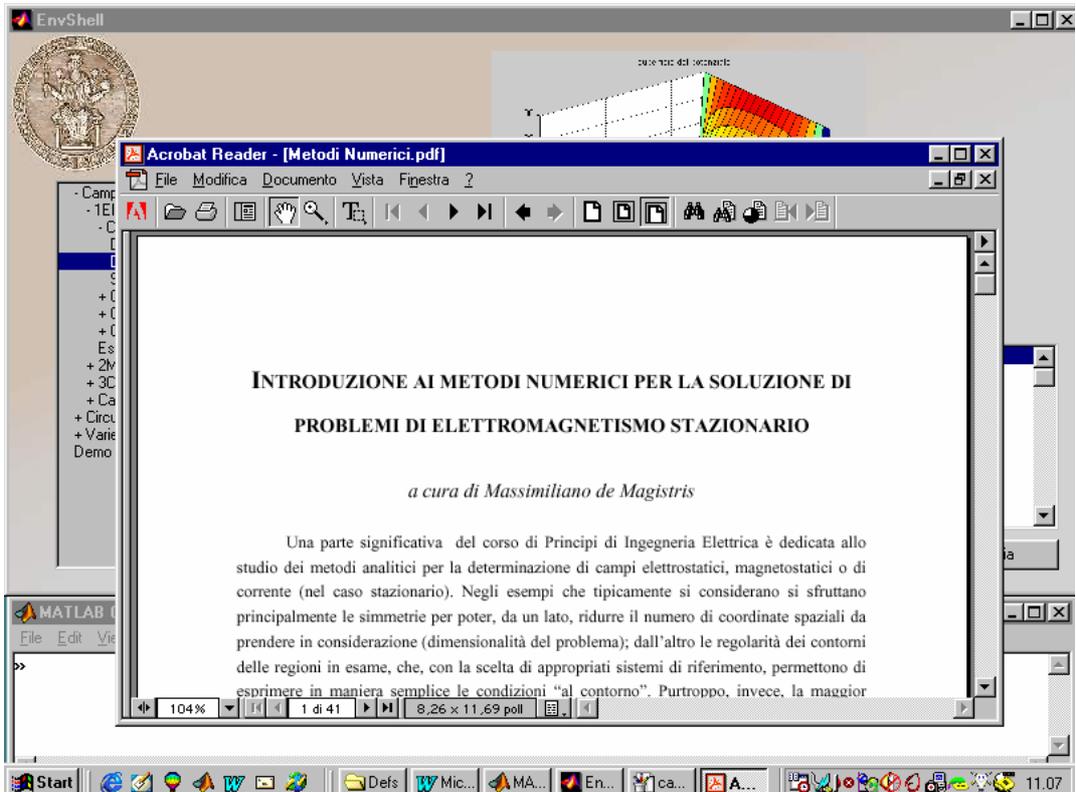
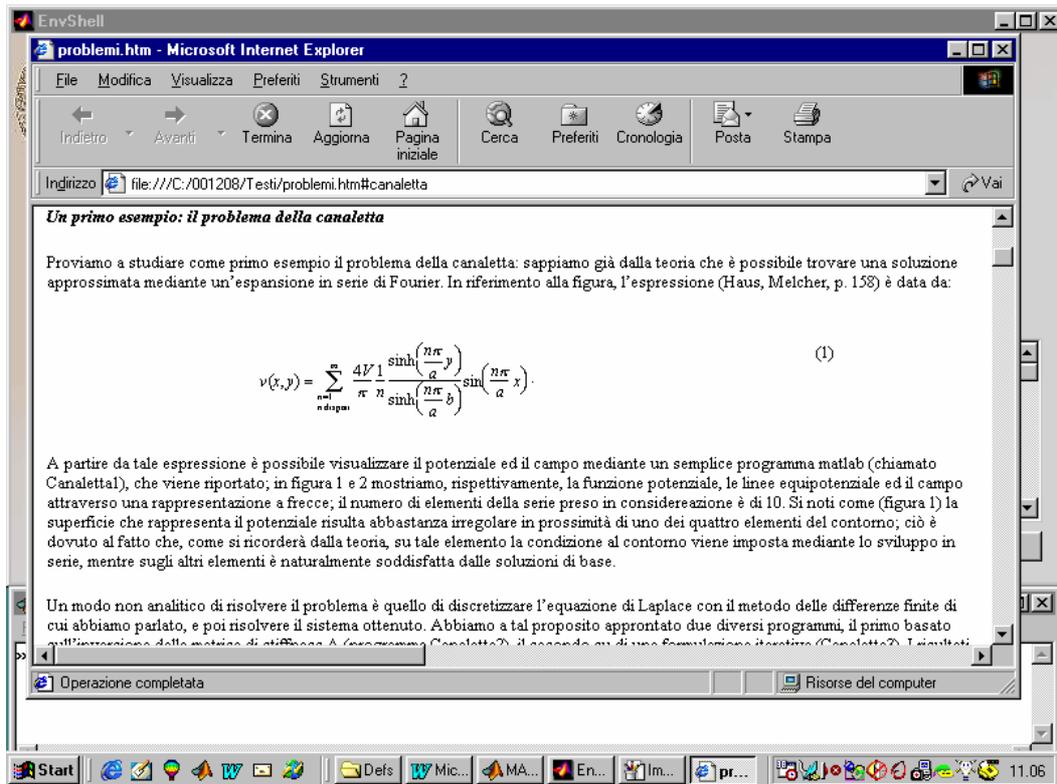
Sia per gli esempi Matlab,



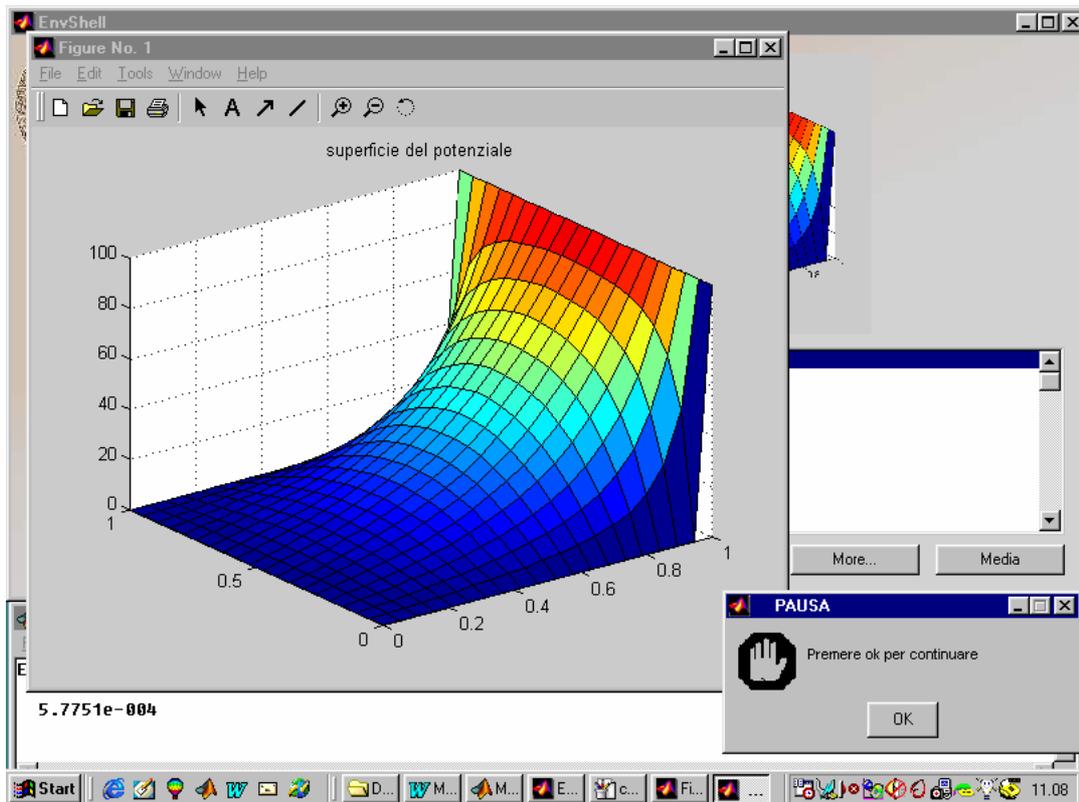
editing:



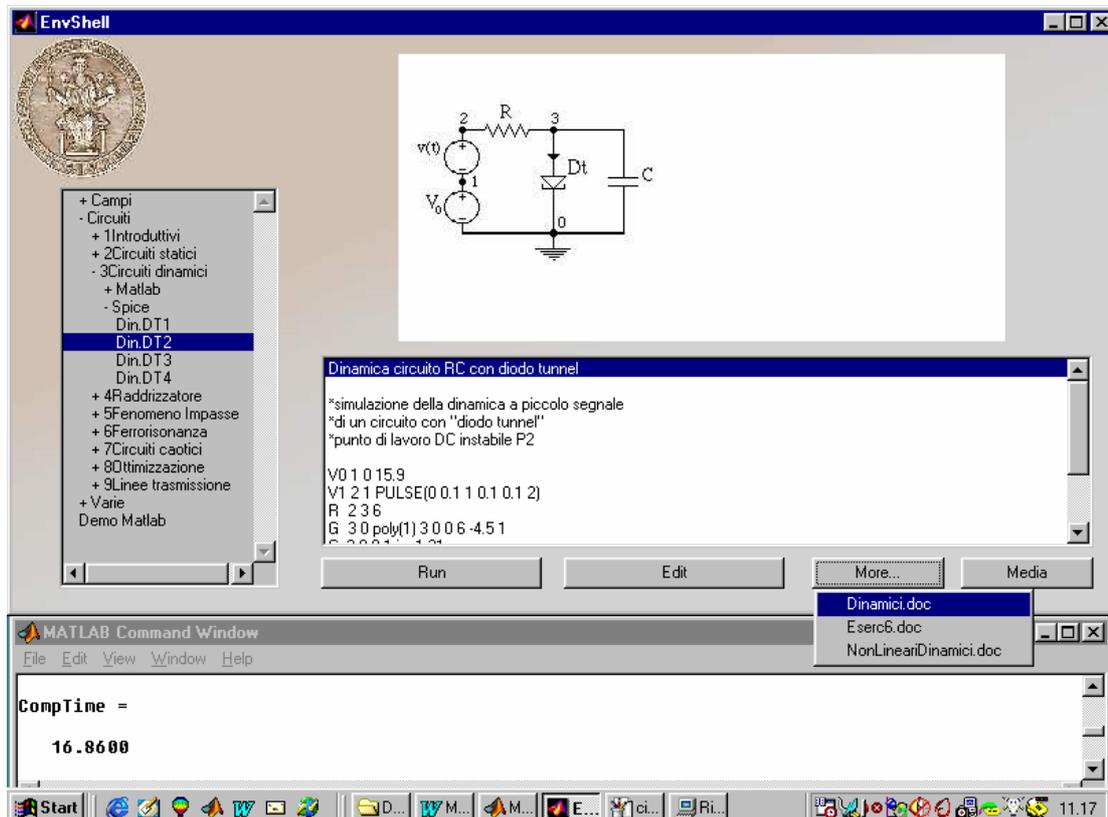
visualizzazione delle risorse nei formati più comuni:

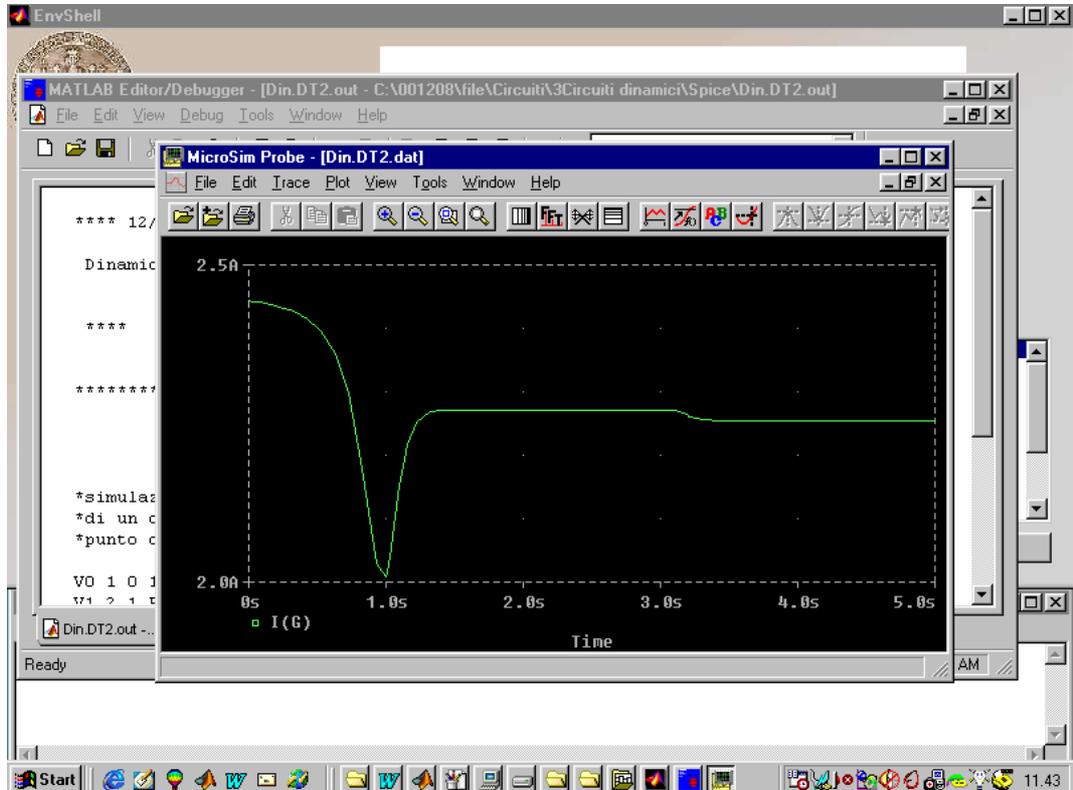
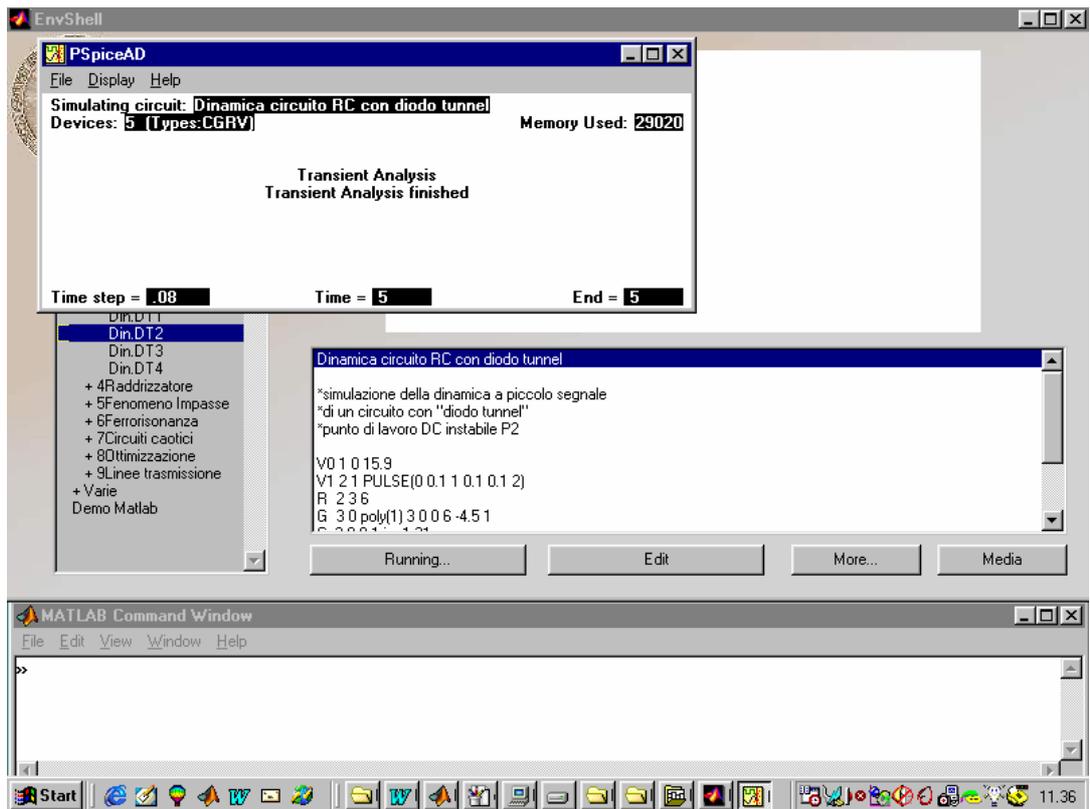


e sua esecuzione



così come per quelli SPICE, dove, una volta lanciata l'esecuzione, in automatico, l'ambiente propone se l'elaborazione è riuscita, sia l'output testuale dello SPICE che quello grafico se richiesto.





CONCLUSIONI

In questa tesi è stato affrontato il problema della realizzazione di un ambiente per la didattica assistita dal calcolatore, rivolto al settore dell'Ingegneria dell'Informazione e specificamente all'area didattico - disciplinare dei campi e dei circuiti.

Da una rassegna delle molteplici soluzioni proposte nella letteratura per la creazione e la gestione di "Learning Environments", sono stati evidenziati alcuni punti essenziali nella definizione delle specifiche di un tale sistema. In particolare, si è lavorato partendo dal presupposto di una stretta correlazione tra il tool "ambiente didattico" ed i suoi contenuti, scegliendo nel caso specifico di utilizzare Matlab come linguaggio per realizzarlo.

Un secondo punto qualificante riguarda la scelta di organizzare tutto il materiale didattico all'intero di una stessa "shell", così da permettere un accesso estremamente facilitato. Naturalmente, in un secondo momento, lo stesso materiale didattico potrà essere utilizzato anche all'esterno della shell di presentazione.

Ancora, una caratteristica importante che si è scelta di implementare è stata la facilità d'uso per il docente nella realizzazione del materiale didattico, che infatti viene realizzato interamente con strumenti tradizionali e non specifici (Word, Matlab, Spice etc.) e viene organizzato semplicemente in cartelle. L'integrazione all'ambiente avviene, in parte immediatamente con la sola strutturazione in cartelle e sottocartelle, e per una minima parte con l'aggiunta di semplici commenti ai listati.

Infine si è scelta, come modalità di "delivery", quella del CD auto installante nel quale, oltre all'ambiente, sono aggiunti tutti quei tools software di pubblico dominio che permettono la fruizione del materiale.

Il "toolbox" Matlab realizzato in questo modo si presta altrettanto bene ad un utilizzo dimostrativo per lezioni in aula multimediale. Una possibile estensione delle sue potenzialità potrebbe derivare dalla sua realizzazione per il WEB.

APPENDICE I :

***MANUALE D'USO* (Regole, strutture e formati)**

A.I 1: Regole generali

La sintassi del comando envshell è:

```
envshell [<usedirectory>]
```

dove <usedirectory> è una directory al cui interno è presente la directory "FILE" che è il punto di partenza da cui l'ambiente Envshell mostra i file Matlab (*.m), Spice (*.cir) e Html (index.htm o index.html) e le sottodirectory in esso presenti

Se il comando envshell viene inserito senza il percorso <usedirectory>, sarà usata di "default" la directory in cui è presente il comando "Envshell" stesso.

Es.: >envshell 'C:\mieifile';

In tal caso, si presuppone che esista la directory C:\mieifile\file al cui interno sono presenti le directory ed i file da visualizzare:

Nel particolare:

Per i file Spice verranno mostrati il nome dei file senza l'estensione .cir

Per i file Html solo il file index.htm o index.html verrà visualizzato con il generico nome di "html pages"

Per i file Matlab verranno visualizzati solo i file al cui interno non è presente il comando speciale "% HideMe:" e saranno mostrati con il nome completo del file (compreso di estensione .m) o con il nome scelto dall'utente con il comando "% Titolo:<titolo>" all'interno del listato Matlab.

La struttura da seguire delle varie directory e file all'interno della genitrice "FILE" è completamente libera, non vi sono limitazioni (tranne quelle impartite dal sistema operativo Windows su cui l'ambiente si appoggia) né nel numero di innestamenti né di file e directory presenti all'interno delle varie directory.

L'ambiente, infine mostrerà, legandoli ad i singoli file mostrati (oltre a quello legato direttamente con il comando apposito) fino a sette files, siano essi in formato pdf (*.pdf) che documenti MS-Word (*.doc) presenti, a partire dalla directory in cui si è fino a retrocedere alla directory di partenza "FILE".

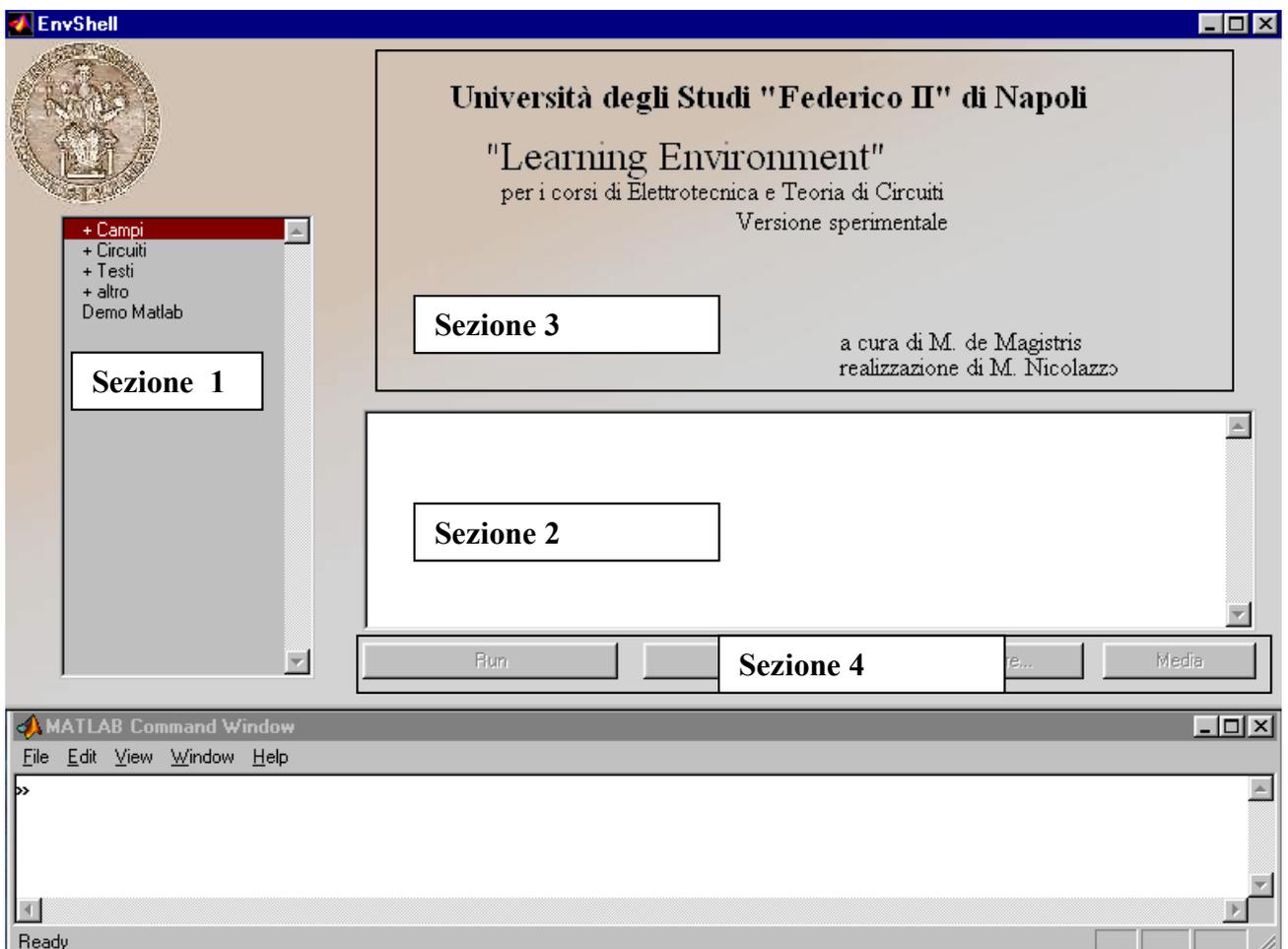
A.I 2: Formati e Regole per l'organizzazione del materiale didattico nell'ambiente "Envshell"

All'esecuzione del comando, La Command Window di Matlab si ridimensionerà e si posizionerà ad occupare il 30 % circa dello schermo in basso e si aprirà sul restante 70 % di schermo superiore la "finestra di navigazione".

La maschera di navigazione è suddivisa in quattro sezioni:

- 1) Sezione di esplorazione delle directory; dove è presente la struttura delle directory ed i file presenti nelle varie subdirectory presenti in quella genitrice "FILE"

- 2) Sezione testo; dove e' possibile far comparire un testo legato al file matlab (.m) ,pspice (.cir) o .html in questione.
- 3) Sezione di visualizzazione immagine; dove e' possibile far comparire una figura (immagine sintetica) legata sempre al file in questione.
- 4) Sezione di interazione; serie di quattro bottoni dove e' possibile eseguire il file, editarne il contenuto per visualizzarlo e/o modificarlo, lanciare i link a pagine html e file multimediali se presenti.



A.I 2.1: Istruzioni Aggiuntive ai File MatLab.

Per il generico file "XX.m", è possibile legare un immagine, un testo un multimediale audio e/o video ed una pagina ipertestuale di formato html.

E' anche possibile nascondere il file dalla visualizzazione.

Le istruzioni supplementari da inserire all' interno del file XX.m sono:

“% titolo:<titolo>”

Nella finestra di esplorazione viene visualizzato <titolo> invece del nome del file XX.m presente nella directory corrente.

Tutto il set di caratteri ASCII è permesso.

Se non e' presente l' istruzione % titolo: viene visualizzato il nome del file compreso di estensione: XX.m

”% bitmap:<nomefile>”

Nella Sezione di visualizzazione immagine viene, una volta selezionato il file XX.m visualizzato il file immagine <nomefile>.

Il formato del file deve essere uno di quelli supportati dalla routine imread() del MATLAB, quindi 'bmp' (Windows Bitmap (BMP)), 'hdf' (Hierarchical Data Format (HDF)), 'jpg' or 'jpeg' (Joint Photographic Experts Group (JPEG)), 'pcx' (Windows Paintbrush (PCX)), 'tif' or 'tiff' (Tagged Image File Format (TIFF)), 'xwd' (X Window Dump (XWD))

La dimensione massima della figura è circa 460x210 pixel. Oltre tale dimensione l' immagine viene ridimensionata (mantenendo il rapporto x:y) per rientrare in tale rettangolo.

”% notefl:<nomefile>”

Nella Sezione di visualizzazione testo, una volta selezionato il file XX.m viene visualizzato il file testo <nomefile>.

Il file deve essere semplicemente in formato testo ASCII.

Se il file <nomefile> e' inesistente o l' istruzione %notefl: non e' presente nel file XX.m, viene visualizzato nella finestra preposta il listato del file.

”% mmedia:<nomefile>”

Se presente l’istruzione, ed il file <nomefile> esiste, viene attivato il Bottone “Media”, ed è così possibile lanciare un qualsiasi file multimediale fra quelli supportati dal Windows Media Player fornito con i vari Windows.

”% htmlfl:[*.]<nomefile>”

Se presente l’istruzione, la prima voce del menu di popup collegato al Bottone “More...” sarà collegato a <nomefile>.

“Cliccato” questo viene lanciato il NetScape (od il browser predefinito) e da lì aperto il link al <nomefile>.

<nomefile> può essere un URL completo (http://www.unina.it), oppure assoluto (file://c|programmi/matlab/help/helpdesk.html), oppure relativo alla directory in cui è presente il file matlab ad esso associato (./esempio3.htm#cap1) oppure relativo al livello della directory genitrice ‘FILE’ in tal caso va’ inserito l’ asterisco “*” come primo carattere (* /Testo/problemi.htm#canaletta.)

”% HideMe:”

Con tale istruzione (se presente), il file XX.m non viene mostrato nella finestra di navigazione. Per non far comparire subroutine e funzioni necessari per l’ esecuzione del XX.m ma non eseguibili direttamente.

A.I 2.2: Regole per i formati PSpice.

Per il generico file “XX.cir” è possibile legare un immagine, un testo, un multimediale audio e/o video ed una pagina ipertestuale di formato html.

Relativamente alla directory in cui è presente “XX.cir”, possono essere aggiunti i file con le seguenti estensioni:

“.fig”

Nella Sezione di visualizzazione immagine viene, una volta selezionato il file *XX.cir* visualizzato il file immagine “*XX.cir.fig*” se presente.

Il formato del file deve essere uno di quelli supportati dalla routine `imread()` del MATLAB, quindi 'bmp' (Windows Bitmap (BMP)), 'hdf' (Hierarchical Data Format (HDF)), 'jpg' or 'jpeg' (Joint Photographic Experts Group (JPEG)), 'pcx' (Windows Paintbrush (PCX)), 'tif' or 'tiff' (Tagged Image File Format (TIFF)), 'xwd' (X Window Dump (XWD))

La dimensione massima della figura è circa 460x210 pixel. Oltre tale dimensione l'immagine viene ridimensionata (mantenendo il rapporto x:y) per rientrare in tale rettangolo.

“.TXT”

Nella Sezione di visualizzazione testo, una volta selezionato il file *XX.cir* viene visualizzato il file testo “*XX.cir.txt*” se presente.

Il file deve essere semplicemente in formato testo ASCII.

Se il file “*XX.cir.txt*” e' inesistente viene visualizzato nella finestra preposta il listato del file.

“.mmf”

Se presente il file “*XX.cir.mmf*”, il Bottone “Media” è attivo ed attiva il lettore multimediale di Windows con il file “*XX.cir.mmf*”.

Ovviamente è possibile utilizzare un qualsiasi file multimediale fra quelli supportati dal Windows Media Player.

“.HTML”

Se presente il file “*XX.cir.html*”, la prima voce del menu di popup collegato al Bottone “More...” sarà collegato ad esso.

“Cliccato” questo viene lanciato il NetScape (od il browser predefinito) e da li aperto il file “*XX.cir.html*”.

A.I 2.3: Regole per i file in formato Html (index.html o index.htm)

Per i file in formato HTML, l'unico file che viene riconosciuto come a se stante, è "index.htm" o "index.html" che quindi deve essere la pagina di partenza per tutte le html presenti.

I file verranno segnati nella sezione di esplorazione come "**html pages**".

In questo caso il Bottone "Run" è attivo in quanto questi verranno aperti sotto NetScape (o browser predefinito).

Per il file "index.htm[l]" è possibile legare un'immagine, del testo ed un multimediale audio e/o video..

Relativamente alla directory in cui è presente "index.htm[l]", possono essere aggiunti i file con le seguenti estensioni:

“.fig” (index.htm[l].fig)

Nella Sezione di visualizzazione immagine viene visualizzato, una volta selezionato "html pages", il file immagine "*index.htm[l].fig*" se presente.

Il formato del file deve essere uno di quelli supportati dalla routine imread() del MATLAB, quindi 'bmp' (Windows Bitmap (BMP)), 'hdf' (Hierarchical Data Format (HDF)), 'jpg' or 'jpeg' (Joint Photographic Experts Group (JPEG)), 'pcx' (Windows Paintbrush (PCX)), 'tif' or 'tiff' (Tagged Image File Format (TIFF)), 'xwd' (X Window Dump (XWD))

La dimensione massima della figura è circa 460x210 pixel. Oltre tale dimensione l'immagine viene ridimensionata (mantenendo il rapporto x:y) per rientrare in tale rettangolo.

“.txt” (index.htm[l].txt)

Nella Sezione di visualizzazione testo, una volta selezionato “html pages” viene visualizzato il file testo “*index.htm[l]..txt*” se presente.

Il file deve essere semplicemente in formato testo ASCII.

“.mmf” (index.htm[l].mmf)

Se presente il file “*index.htm[l].mmf*”, il Bottone “Media” è attivo ed attiva il lettore multimediale di Windows con il file “*index.htm[l].mmf*”.

Ovviamente è possibile utilizzare un qualsiasi file multimediale fra quelli supportati dal Windows Media Player.

A.I.2.4: Collegamenti di File in formato MS-Word e Adobe Acrobat

Nel creare la struttura di file e directory, è possibile collegare al Bottone “More...”, dei file in formato doc e pdf, semplicemente inserendoli nella struttura.

L’ ambiente, una volta selezionato un nome dalla sezione di navigazione, provvederà a scandagliare, a partire dalla directory in cui si trova, a ritroso fino alla directory genitrice “FILE”, per tutti i file doc e pdf in essi contenuti, presentandone i collegamenti nel menu di popup collegato al bottone “More...” in ordine di recupero per un massimo di sette file.

L’ unico caso in cui il bottone “More...” è disabilitato, è quando al file selezionato non è collegato nulla direttamente, tramite le apposite sintassi, e quando non vengono inseriti documenti nella sua struttura.

A.I 3: Programmi di utilità acclusi all'ambiente

L’ambiente “EnvShell” necessita del MathWorks MATLAB.

All’ interno del CD-ROM, sono presenti altri tool di appoggio, l’Adobe Acrobat Reader versione 4.05 per la visualizzazione dei file in formato pdf, il MicroSoft WordViewer 97, per la visualizzazione dei documenti Word, il NetScape

versione 4.76 per gli Html, e la versione dimostrativa del MicroSim Pspice 8.0 necessario per le simulazione dei sorgenti Spice.

L' ambiente "Envshell", comunque, tranne che per lo PSpice, dove è strettamente collegato, utilizza le associazioni fatte in ambiente Windows per lanciare i vari files, quindi se sono già installati sul computer dell'utente altri pacchetti software che permettono la visualizzazione dei formati necessari, non è necessario altro.

APPENDICE II :

FUNZIONI AGGIUNTE AL MATLAB

Per far sì che i programmi Matlab abbiano una interfaccia più visuale rispetto al classico stile terminale della Command Window di Matlab, per posizionare le schermate grafiche di output e richiedere gli input delle funzioni, in maniera “tunable”, sono state create allo scopo delle funzioni aggiuntive e/o sostitutive rispetto quelle classiche Matlab (sempre in linguaggio MatLab) , tali da non essere troppo invasive e quindi imporre eccessive modifiche ai sorgenti originali e ottenere una migliore visualizzazione.

A.II 1: `figureposition(handle,position)`:

Permettere di creare un semplice “Cascade” di finestre grafiche aperte dalla funzione “XX.m” creata dall’ utente durante il run, (così da non averle sovrapposte completamente) senza andare a modificarne singolarmente le varie “figure”. Basta conservarne l’ handler restituito dalla funzione matlab “figure” e passarlo con l’ordine cui si vuol far apparire alla funzione “figureposition”.

Ciò comporta modificare il sorgenti matlab da:

```
.....
figure (1);
.....
figure (2);
in
.....
h=figure(1);
figureposition(h,1) ;
.....
h=figure(2);
figureposition(h,2) ;
.....
```

Es.:

```
.....
h=figure(1); %rappr. curve di livello del potenziale
figureposition(h,1);

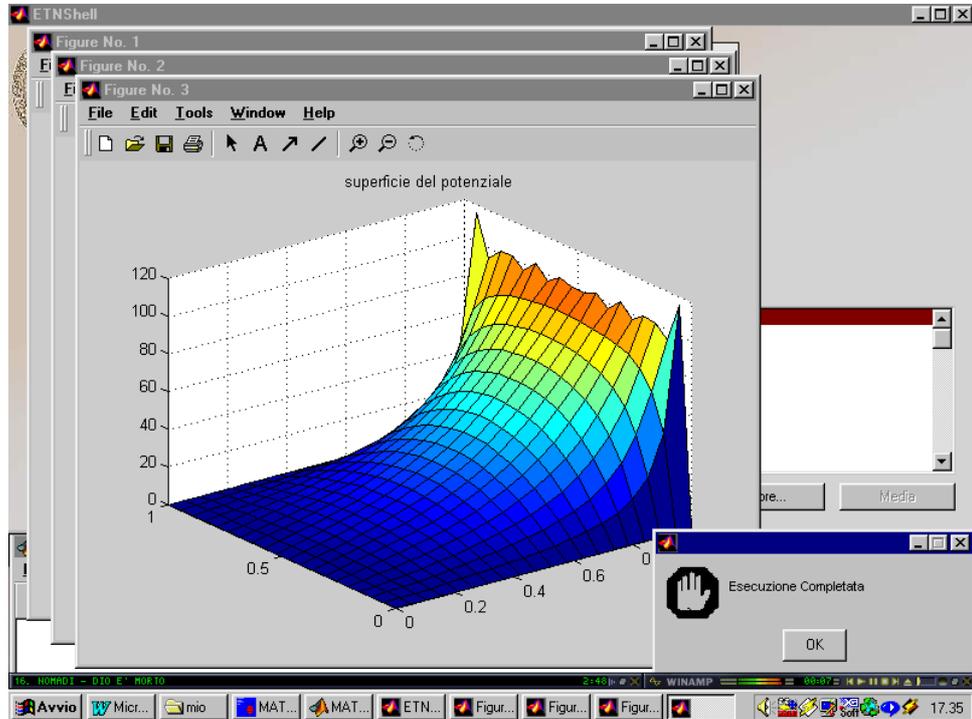
CF=contour(x,y,v,30);
clabel(CF);
title 'curve di livello del potenziale'
uipause

h=figure(2); % rappr. campo vettoriale
figureposition(h,2);
contour(x,y,v);
title 'campo vettoriale'
hold on
[Ex,Ey]=gradient(v,hx,hy); %calcolo del campo mediante la funzione gradiente
quiver(x,y,-Ex,-Ey,5);
uipause
```

```

h=figure(3);% rappr. superficie del potenziale
figureposition(h,3);
surf(x,y,v);
title 'superficie del potenziale'
.....

```



A.II 2: uipause (secs):

Pausa visibile graficamente (con una finestra di dialogo).

Si può sostituire le funzioni “pause” utilizzate all’interno del sorgente con la funzione “uipause”

Cio’ comporta modificare il sorgenti matlab da:

```

.....
pause
.....
pause(1);
in
.....
uipause
.....

```

```
uipause(1);
```

Nel secondo caso, la funzione uipause si comporta esattamente alla stessa maniera della pause, senza far apparire la finestra di dialogo ed attendendo 1 secondo prima di proseguire.

Es.:

```
.....  
uipause;
```



A.II 3: Paramwindow(np,

titolo,min,max,def,

titolo,min,max,def,

titolo,min,max,def,

.....)

Permette di richiedere graficamente, con una maschera di dialogo un numero (quasi) indefinito di parametri

Nel particolare:

np : numero di parametri da visualizzare

<

titolo : Intestazione.

min : valore minimo di selezione

max: valore massimo di selezione

def: valore di partenza per la selezione

>

la sezione tra “<” e “>” ripetuta per np volte.

I valori min e max possono essere modificati dall' utente nella maschera di dialogo stessa.

Es.:

```
[R,C1,L2,x0,TS]=ParamWindow(5, ...  
    'Resistenza [ohm]',1,1000,10, ...  
    'Capacità [Farad]',1e-6,1,1e-3, ...  
    'Autovalore 2 [s^-1]',1,10,1, ...  
    'Tensione iniziale di C1 [volt]',0,12,3, ...  
    'Durata della simulazione [s]',1,1000,30)
```



A.II.4: Listati delle functions

A.II.4.1: figureposition.m

```
function figureposition(h,x);
% figureposition(handler,position);
%
%   sposta in "Cascade" la finestra grafica di handler dato
%   (restituito dalla funzione figure).
%   position=0 corrisponde all' angolo in alto a sinistra
dello schermo
%   position=n trasla di n*20 pixel piu' in basso ed a
destra rispetto position=0

pos=get(h,'pos');
pos(1,1)=x*20;
pos(1,2)=pos(1,2)-(x-1)*20;
set(h,'pos',pos);
%end function
```

A.II.4.2: uipause.m

```
function uipause(n)
% uipause(secs)
%
%   comando grafico sostitutivo al normale pause()
%   vedere anche
%   pause

if nargin>0,
    pause(n)
else
    h=msgbox('Premere      ok      per      continuare','
PAUSA','error');
    set(h,'unit','normalized');
    pos=get(h,'position');
    pos(1,1)=0.67;
    pos(1,2)=0.07;
    set(h,'position',pos);
    uiwait;
end %if
%end function
```

A.II.4.3: ParamWindow.m

```
function varargout=ParamWindow(n,varargin)
% varargout=ParamWindow(n,varargin)
%
%   Crea una maschera di dialogo dove e' possibile
%   interagire per inserire dei dati numerici
```

```

%
% Paramwindow(np,
%         titolo,min,max,def,
%         titolo,min,max,def,
%         titolo,min,max,def,
%         .....)
% np : numero di parametri da visualizzare
% <
%         titolo : Instestazione.
%         Min : valore minimo di selezione
%         Max: valore massimo di selezione
%         Def: valore di partenza per la selezione
% >
% la sezione tra "<" e ">" ripetuta per np volte.

if ((nargin-1)~=n*4)
    msgbox ('Errore nell'inserimento dei parametri', '',
'error');
    return
end %if

h_ParamWindow=1001;
set(0, 'Units', 'pixels');
ScreenSize=get(0, 'ScreenSize');
sp=18; % spessore controlli
lt=30; % lunghezza testi max e min
sv=130; % start x-point del testo valore corrente
sb=20; % spessore bottone
nc=n*sp*2.5+sb*1.2; % altezza finestra
lv=80; % lunghezza testo valore corrente
lc=240; % lunghezza finestra
offs=20; % offset per tener conto della testata della
% finestra
st=nc-sp;% Start x-point dei controlli

figure(h_ParamWindow);
set(h_ParamWindow, 'Units', 'pixels', ...
'Color', [0.8 0.8 0.8], ...
'HandleVisibility', 'on', ...
'NumberTitle', 'off', ...
'Name', 'Scelta Parametri', ...
'Position', [5 ScreenSize(4)-nc-offs lc+20 nc], ...
'MenuBar', 'none', ...
'Tag', 'FormParam', ...
'WindowStyle', 'modal');
for i=1:n,
    j=(i-1)*4;
    yd=(i-1)*sp*2.5;
    %TextMin: Text del Min della Slider
    h_textMin(i) = uicontrol('Parent',h_ParamWindow, ...
'Units', 'pixels', ...
'BackgroundColor', [1 1 1], ...
'Callback', 'ParamOK(''TextMin_Change'')', ...
'ListboxTop', 0, ...
'Position', [0 st-yd-sp lt sp], ...
'Style', 'edit', ...
'String', varargin{j+2}, ...
'Tag', ['TextMin', i]);
    %Label della slider
    h1 = uicontrol('Parent',h_ParamWindow, ...
'Units', 'pixels', ...
'BackgroundColor', [0.753 0.753 0.753], ...

```

```

        'ListboxTop',0, ...
        'Position',[0 st-yd lc-lt-lt sp], ...
        'HorizontalAlignment','left', ...
        'String',varargin{j+1}, ...
        'Style','text', ...
        'Tag',['LabelSlider',i]);
    %Text del valore corrente della Slider
    h_textValue(i) = uicontrol('Parent',h_ParamWindow , ...
        'Units','pixels', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','ParamOK(''TextValue_Change'')', ...
        'ListboxTop',0, ...
        'Position',[sv st-yd lv sp], ...
        'String',varargin{j+4}, ...
        'Style','edit', ...
        'Tag',['TextValue',i]);
    %Slider
    h_Slider(i) = uicontrol('Parent',h_ParamWindow , ...
        'Units','pixels', ...
        'BackgroundColor',[0.753 0.753 0.753], ...
        'ListboxTop',0, ...
        'Callback','ParamOK(''Slider_Change'')', ...
        'Position',[lt st-sp-yd lc-lt-lt sp], ...
        'Style','slider', ...
        'Value',varargin{j+4}, ...
        'Min',varargin{j+2}, ...
        'Max',varargin{j+3}, ...
        'Tag',['Slider',i]);
    %TextMax: Text del Max della Slider
    h_TextMax(i) = uicontrol('Parent',h_ParamWindow , ...
        'Units','pixels', ...
        'BackgroundColor',[1 1 1], ...
        'ListboxTop',0, ...
        'Callback','ParamOK(''TextMax_Change'')', ...
        'Position',[lc-lt st-sp-yd lt sp], ...
        'String',varargin{j+3}, ...
        'Style','edit', ...
        'Tag',['TextMax',i]);

    end; %for
    command = 'ParamOK(''ButtonOK_Click'');uiresume;';
    h_ButtonOK = uicontrol('Parent',h_ParamWindow, ...
        'Units','pixels', ...
        'ListboxTop',0, ...
        'Position',[sv 5 lv+20 sb], ...
        'String','OK', ...
        'Callback',command, ...
        'Tag',['ButtonOK ',n]);

    uiwait(h_ParamWindow);
    var=get(h_ParamWindow,'UserData');
    for j=1:n
        varargout(j)={var(j)};
    end
    delete(h_ParamWindow);
%end function

```

A.II.4.4: ParamOk.m (Subroutine per ParamWindow)

```
function ParamOK(action)
% ParamOK(action)
%
%     funzione di ParamWindow
%     controlla gli input dall' esterno
%

[h_obj,h_fig]=gcbo;
q=get(h_obj,'Tag');
i=q(length(q))+0;

switch action
case 'Slider_Change'
    h0=findobj(h_fig,'Tag',{'TextValue',i});
    set(h0,'String',get(h_obj,'Value'));
case 'TextMax_Change'
    h0=findobj(h_fig,'Tag',{'Slider',i});
    max=get(h0,'Max');
    min=get(h0,'Min');
    val=get(h0,'Value');
    str=get(h_obj,'String')
    value=eval(str);
    if (value<min)
        set(h_obj,'String',min);
    end %if
    if (value<=val)
        set(h0,'Value',value)
    end %if
    set(h0,'Max',value);
case 'TextMin_Change'
    h0=findobj(h_fig,'Tag',{'Slider',i});
    set(h0,'Min',eval(get(h_obj,'String')));
case 'TextValue_Change'
    h0=findobj(h_fig,'Tag',{'Slider',i});
    min=get(h0,'Min');
    max=get(h0,'Max');
    str=get(h_obj,'String')
    value=eval(str);
    if (value<min),
        h1=findobj(h_fig,'Tag',{'TextMin',i});
        set(h1,'string',str)
        set(h0,'Min',value);
    end %if
    if (value>max),
        h1=findobj(h_fig,'Tag',{'TextMax',i});
        set(h1,'string',str)
        set(h0,'Max',value);
    end % if
    set(h0,'Value',value);
case 'ButtonOK_Click'
    for j=1:i,
        h0=findobj(h_fig,'Tag',{'Slider',j});
        var(j)=get(h0,'Value');
    end %for
    set(1001,'UserData',var);
end
%end function
```

BIBLIOGRAFIA

Abruzzese A. (1997), Nuove tecnologie nuove modalità d'insegnamento, Venezia

Bagnara (1993), conferenza a "DIDAMATICA 1993"

Building GUIs with MATLAB (Jun 1997), The MathWorks Inc.

Berlo D. (1968), The process of Communication, Holt, Rinehart & Winston, New York

Beaufils D., Le Touzè J.C., Blondel F.M. (Apr. 1994), Computer and camera: tools for solving physics problems, Physics Education

Clarck A. (1993), Investigating school physics laboratory software and hardware, Physics Education, Vol 28

Collins A., Brown J.S., Newman S.E (1989), Cognitive apprenticeship: Teaching the craft of reading, writing, and mathematics. Knowing, learning and instruction: Essays in honor of Robert Glaser. Hillsdale, New Jersey: Lawrence Erlbaum Associates

Cyberprof manual: <http://www.howhy.com/home/>

Dale E. (1969), Audiovisual Methods in Teaching, Dryden Press, Hionsdale

Donnelly D. (1990), CIP's First Annual Software Contest: The Winners. Computer in Physics, Sept/Oct 1990

Donnelly D. (1993), CIP's Fourth Annual Software Contest: The Winners. Computer in Physics, Vol.7 N.6 Nov/Dec 1993

Ferraris M., Midoro V., Olimpo G. (1984), Instructional Systems Design and Software System Design: a Unifying Approach, Journal of Structural Learning, vol.8, pp 55-61

Gagné R.M., Briggs L.J. (1979), Principles of Instructional Design (sec. ed.), Holt, Rinehart & Winston, New York

Hoban C.F. (1974), A Current View of the Future of Theory and Research in Educational Communication, Audiovisual Instruction

Hubler A.W., Assad A.M. (1995), CyberProf: an Intelligent Human-Computer Interface for Asynchronous Widearea Training and Teaching. Fourth International World Wide Web Conference December 11-14, 1995

Kass A. (1994), integrazione fra pratica ed apprendimento con il computer, Multimedia and distance learning for science and technology, Garamond

Klessig R.W., Tesink K. , "SMDS Wide-Area Data Networking with Switched Multi-megabit Data Service"

Lave J. Wenger E. (1991), Situated learning: Legitimate peripheral participation. New York: Cambridg University Press.

Merril M.D. (1971), Instructional Design: Readings, Educational Technology Publ., Englewood Cliffs

Miller M.A., P.E., "Analyzing Broadband Networks, II Ed."

Myers B.A. (July 1993), Why are Human-Computer Interfaces Difficult to Design and Implement ?

Olimpo G. (Aprile 1993), Nascita e sviluppi delle Tecnologie Didattiche Istituto Tecnologie Didattiche, Tecnologie Didattiche, n. 1

Pressey S. (1960), A Simple Apparatus which gives tests and scores and teaches, in Teaching Machines and Programmed Learning, Lumsdane A.A. and Glaser R. eds, N.E.A., Washington

Saettler P. (1968), A History of Instructional Technology, Mc Graw Hill, New York

Skinner B.F. (1954), The science of learning and the art of teaching, Harvard Education Review, vol.24, Spring

The MathWork, Inc: <http://www.mathworks.com>,

Newsgroup: comp.soft-sys.matlab

Twigg C.A. (1995), The need for a national learning infrastructure. Educom Review, vol. 29 Numbers 4,5,6

WebCT : Sito web ufficiale <http://www.webct.com/webct/>.

