

**PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA DE PRODUÇÃO**

**UM MODELO NEURO-DIFUSO DE
MULTIAGENTES PARA APOIO AO ENSINO DE
DISCIPLINAS DE CIÊNCIAS EXATAS**

TESE SUBMETIDA À UNIVERSIDADE FEDERAL DE SANTA
CATARINA COMO REQUISITO PARCIAL PARA A OBTENÇÃO
DO GRAU DE DOUTOR EM ENGENHARIA DE PRODUÇÃO

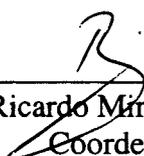
LORÍ VIALI

Florianópolis, 14 de outubro de 1999

Um modelo neuro-difuso de multiagentes para apoio
ao ensino de disciplinas de ciências exatas

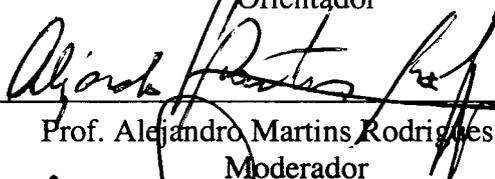
Lorí Viali

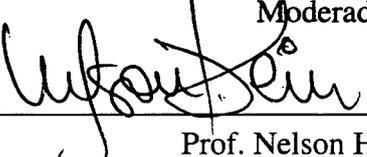
Tese aprovada como requisito parcial para a obtenção do
grau de Doutor em Engenharia de Produção

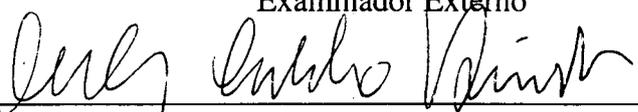

Prof. Ricardo Miranda Barcia, Ph.D.
Coordenador

BANCA EXAMINADORA

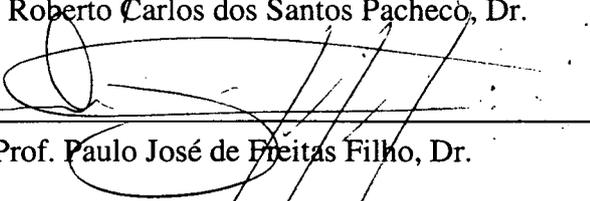

Prof. Rogério Cid Bastos, Dr.
Orientador


Prof. Alejandro Martins Rodrigues, Dr.
Moderador


Prof. Nelson Hein, Dr.
Examinador Externo


Prof. Carlos Eduardo da Cunha Pinent, Dr.
Examinador Externo


Prof. Roberto Carlos dos Santos Pacheco, Dr.


Prof. Paulo José de Freitas Filho, Dr.

AGRADECIMENTOS

Em especial à Dolores, a quem este trabalho é dedicado.

As filhas Lívia e Elisa.

A PUCRS pela liberação com vencimentos.

Aos diretores da FAMAT prof. Nilton e prof^a Alaydes pelo apoio.

Ao prof. Riboldi da UFRGS pelo apoio.

A UFRGS pela licença remunerada concedida.

Ao CNPq pelo suporte na forma de bolsa.

Ao prof. Ricardo Miranda Barcia coordenador do PPGEP.

Ao prof. Suresh Khator pela apoio e orientação em Tampa.

Aos colegas Roberto, Alejandro, Paulo e Rosina que compartilharam o tempo em Tampa e que contribuíram para que fosse uma experiência indelével.

Ao André e a Ana pela acolhida e amizade no exterior.

Aos colegas da PUCRS pelo incentivo e em especial ao José e o Luiz que compartilharam da mesma experiência e ao Guilherme e ao Carlos pela criteriosa revisão dos originais.

Ao professor Plinio Stange (im memoriam) meu orientador original e ao professor Rogério que gentilmente aceitou o encargo de seguir na orientação.

Ao professores Alejandro, Carlos, Paulo, Nelson e Roberto e que gentilmente aceitaram participar da banca.

Finalmente, mas não menos importante ao Tadeu e a Cida.

E a todos que de uma forma ou outra me incentivaram a prosseguir e que pelas circunstâncias não foram lembrados explicitamente.

RESUMO

Um modelo neuro-difuso de multiagentes para apoio ao ensino de disciplinas de ciências exatas

Autor: Lorí Viali

Orientador: Rogério Bastos

Lecionar disciplinas que requerem um alto grau de pré-requisitos matemáticos não é uma tarefa fácil. O motivo é que grande parte dos estudantes não possuem apreço por qualquer disciplina que envolva números de forma substancial.

As razões são variadas. Muitos estudos já foram elaborados tentando identificar as causas porque muitos estudantes de todas as áreas são reprovados em disciplinas como: Cálculo, Álgebra Linear, Geometria Analítica, Probabilidade e Estatística. Na área de Engenharia as razões não são exatamente a falta de apreço pelos números no entanto, o problema não é muito diferente.

Após o sucesso dos computadores em todas as áreas, inclusive no ensino, e o crescimento fantástico da Internet e da WWW, meios novos e mais eficientes estão atraindo os estudantes e despertando o entusiasmo para o estudo em todas as áreas. No entanto, só entusiasmo não é suficiente, uma vez que a maioria dos estudantes tem falta de pré-requisitos em matemática, mas é possível tirar vantagem do interesse e disposição para reforçar as deficiências na formação. A melhor maneira de se fazer isto, eu acredito, é através de um sistema que inclua a WWW e a Internet. A grande maioria dos estudantes estão familiarizados com a WWW e sua interface, desta forma seria uma perda de tempo utilizar algum outro sistema onde o estudante precisasse aprender tudo desde o início.

Este trabalho apresenta um modelo hipermídia multiagentes para auxiliar no ensino de disciplinas da área matemática, especialmente a Probabilidade e a Estatística, dirigido pela Internet e utilizando a WWW como recurso, onde o usuário pode acessar o conteúdo através de um navegador de uma forma dinâmica e atrativa. O sistema possui recursos associados para auxiliar o usuário a entender melhor os conteúdos através de um sistema multiagentes que orientam e monitoram seu desempenho. Faz uso de gráficos dinâmicos e a possibilidade de executar simulações, executar cálculos através de uma planilha e ainda ferramentas para permitir a comunicação com o professor (*e-mail*) e para discutir dúvidas com colegas (*chat*).

UNIVERSIDADE FEDERAL DE SANTA CATARINA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

Tese de Doutorado em Engenharia de Produção e Sistemas

Florianópolis, 14 de outubro de 1999

ABSTRACT

A multiagent neuro-fuzzy model to help teaching courses in the Mathematics area

Autor: Lorí Viali

Orientador: Rogério Bastos

Teaching courses that require a high mathematics background is not an easy task. This happens because the majority of the students do not like anything that includes a lot of numbers.

The reasons for this are multiples. Many studies have already tried to identify the causes why many students of all areas fail to get approved in undergraduates courses like Calculus, Linear Algebra, Analytic Geometry, Probability and Statistics. In Engineering areas the reasons are not exactly the lack of appreciation of Mathematics, however the problem seems to be the same.

After the success of the computers in all areas, including learning, and the fantastic growth of Internet and WWW, new and more effective ways are attracting students and arise their enthusiasm to study any subject. However, the enthusiasm is not enough (the majority of the students have a poor background in mathematics), but we can take advantage of their interest and disposition to reinforce the weakness of their math formation. The best way to do this, I believe, is to use a system that includes WWW and Internet. The majority of students are familiar with the WWW and its interface, thus it is a waste of time using some other system where the students need to know everything from the beginning.

This work presents a multiagent hypermedia model to help teach courses in the Mathematic area, especially Probability and Statistics, driven by Internet using WWW as a tool, where the user can take the subjects from a browser in a dynamic and attractive way. The system has associated auxiliary resources like an agent model to help the user understand better the contents by performing simulations, visualizing charts in a dynamic way, entering functions formulas, performing calculations using a spreadsheet with an intelligent interface and tools to communicating with the teacher or colleagues (by e-mail) and to share and discuss his or her doubts (by chat).

FEDERAL UNIVERSITY OF SANTA CATARINA
GRADUETE PROGRAM IN INDUSTRIAL ENGINEERING
Dissertation in Industrial Engeneering

Florianópolis, October 14 1999



SUMÁRIO

1. INTRODUÇÃO	1
1.1. O PROBLEMA.....	1
1.2. RELEVÂNCIA.....	3
1.3. OBJETIVO GERAL.....	4
1.4. OBJETIVOS ESPECÍFICOS	4
1.5. TECNOLOGIAS BÁSICAS	5
1.6. DELIMITAÇÃO DO ESTUDO	5
1.7. ESTRUTURA DO TRABALHO.....	6
2. A EDUCAÇÃO E O COMPUTADOR	7
2.1. INTRODUÇÃO	7
2.2. TEORIAS DE EDUCAÇÃO E O COMPUTADOR	8
2.3. COMO O COMPUTADOR PODE SER UTILIZADO NO ENSINO.....	10
2.3.1. Como tutor.....	10
2.3.2. Como assistente.....	12
2.3.3. Como auxílio.....	12
2.4. LINHAS DE DESENVOLVIMENTO	13
2.4.1. Linguagens de autoria	14
2.4.2. Sistemas de autoria.....	14
2.5. CONCLUSÃO	15
3. HIPERMÍDIA	16
3.1. INTRODUÇÃO	16
3.2. HISTÓRICO.....	16
3.3. SISTEMAS DE HIPERMÍDIA: UMA CLASSIFICAÇÃO.....	17
3.3.1. Principais sistemas pioneiros (primeira geração).....	17
3.3.2. A segunda geração	18
3.3.3. A idade adulta (terceira geração)	18
3.3.4. Padronizações (propostas) – modelos e estruturas	19
3.4. HIPERMÍDIA: CARACTERIZAÇÃO	20
3.4.1. Nodos, âncoras e ligações	20
3.4.2. Estrutura organizacional.....	21
3.4.3. Outros elementos ou características	22
3.5. APLICAÇÕES DOS SISTEMAS HIPERTEXTO/HIPERMÍDIA.....	23
3.6. ARQUITETURA DO HIPERTEXTO/HIPERMÍDIA.....	23
3.6.1. O nível de apresentação (interface com o usuário).....	24
3.6.2. O Nível da Máquina Abstrata de Hipertexto (MAH)	24

3.6.3. O nível da base de dados.....	24
3.7. O HIPERTEXTO VERSUS A COMPETIÇÃO	25
3.7.1. O hipertexto versus os sistemas especialistas.....	25
3.7.2. O hipertexto versus texto eletrônico.....	25
3.7.3. O hipertexto versus o texto tradicional.....	26
3.8. CONCLUSÃO	27
4. A HIPERMÍDIA NO ENSINO	28
4.1. INTRODUÇÃO	28
4.2. OS MODELOS DE APRENDIZAGEM E O HIPERTEXTO.....	29
4.3. HIPÓTESES SOBRE APRENDIZAGEM	30
4.4. O HIPERTEXTO E A TEORIA DA SUBSUNÇÃO	31
4.5. O HIPERTEXTO E CONSTRUTIVISMO	32
4.6. PROPRIEDADES DO HIPERTEXTO ÚTEIS À EDUCAÇÃO	32
4.6.1. Adaptação.....	33
4.6.2. Atualização.....	33
4.6.3. Audiovisualização	33
4.6.4. Conectividade.....	34
4.6.5. Dinamismo e interação.....	34
4.6.6. Intercomunicação	34
4.6.7. Redução.....	35
4.7. O HIPERTEXTO COMO RECURSO, TUTOR E ASSISTENTE	35
4.8. CONCLUSÃO	36
5. A WWW, A INTERNET E A EDUCAÇÃO.....	37
5.1. INTRODUÇÃO	37
5.2. A LINGUAGEM HTML.....	38
5.2.1. A html como um tipo mime	38
5.2.2. CGI.....	39
5.2.3. Java	40
5.2.4. A JavaScript	40
5.3. A INTERNET.....	40
5.4. O PROJETO WWW	41
5.4.1. O http	41
5.4.2. Arquitetura	42
5.4.3. O funcionamento da rede (WWW)	42
5.5. O NAVEGADOR	43
5.5.1. Recursos	44
5.5.2. O projeto dos navegadores	44
5.5.3. Aplicativos auxiliares	45
5.5.4. Controles ActiveX.....	45
5.6. A WWW E A INTERNET COMO RECURSOS INSTRUCIONAIS	46
5.6.1. Recursos necessários.....	47
5.6.2. Recursos de infra-estrutura.....	47
5.6.3. Recursos de publicação	47

5.6.4. Recursos de suporte.....	47
5.6.5. Recursos de serviço (o local)	48
5.7. LIMITAÇÕES.....	48
5.8. CONCLUSÃO	49
6. A INTELIGÊNCIA ARTIFICIAL NO ENSINO	50
6.1. INTRODUÇÃO	50
6.2. SISTEMAS INTELIGENTES DE ENSINO	50
6.2.1. Métodos de estruturar o conhecimento	51
6.2.2. Componentes de um sistema ICAI.....	52
6.2.3. ICAI: potencial e limitações.....	53
6.2.4. Uma alternativa aos sistemas tutores inteligentes	54
6.3. ALTERNATIVAS AOS MODELOS TRADICIONAIS	54
6.4. O HIPERTEXTO E A INTELIGÊNCIA ARTIFICIAL.....	55
6.5. DA IA AOS AI.....	55
6.6. AGENTES	56
6.6.1. Definições de agentes.....	56
6.6.2. Classificação dos agentes	57
6.7. INTELIGÊNCIA ARTIFICIAL DISTRIBUÍDA	58
6.8. SISTEMA MULTIAGENTES	59
6.9. ARQUITETURAS DE AGENTES	60
6.10. CONCLUSÃO	60
7. O MODELO MULTIAGENTES PROPOSTO.....	61
7.1. INTRODUÇÃO	61
7.2. A ESTRUTURA DO MODELO	62
7.3. CARACTERIZAÇÃO DOS AGENTES.....	64
7.3.1. O agente "Identificador do Usuário" (IU)	64
7.3.2. O agente "Cria Perfil" (CP).....	65
7.3.3. O agente "Perfil do Usuário" (PU).....	65
7.3.4. O agente "Sugere Exercício" (SE).....	66
7.3.5. O agente "Auxílio ao Usuário" (AU)	68
7.4. MODELOS NEURO-DIFUSOS	68
7.4.1. Arquitetura do modelo neuro-difuso	69
7.4.2. Classificadores neuro-difusos	71
7.5. CAPACIDADES DOS AGENTES PROPOSTOS	72
7.6. CONCLUSÃO	72
8. A IMPLEMENTAÇÃO DO MODELO	73
8.1. INTRODUÇÃO	73
8.2. CAMADAS DE SOFTWARE	73
8.3. LINGUAGENS ESPECÍFICAS	75
8.4. LINGUAGENS GERAIS	76
8.4.1. O Paradigma da orientação a objetos	76
8.4.2. Vantagens da orientação a objetos	76

8.4.3. A orientação a objetos e os agentes.....	77
8.4.4. Os componentes de software.....	77
8.4.5. A linguagem de implementação.....	78
8.4.6. Controles ActiveX.....	79
8.5. OUTRAS FACILIDADES	80
8.6. CONCLUSÃO	81
9. UMA APLICAÇÃO NA ESTATÍSTICA.....	82
9.1. INTRODUÇÃO	82
9.2. OS RECURSOS EXISTENTES.....	82
9.3. OS RECURSOS <i>ONLINE</i>	83
9.3.1. Os cursos (tutoriais) na rede.....	83
9.4. OS RECURSOS OFF-LINE	84
9.4.1. Os pacotes de propósito geral.....	85
9.4.2. Pacotes de propósitos específicos.....	87
9.4.3. Cursos.....	89
9.4.4. Planilhas	90
9.5. A APLICAÇÃO DO MODELO.....	92
9.6. O MÓDULO DE NAVEGAÇÃO (BROWSER)	94
9.7. O MÓDULO ESTATÍSTICO (PLANILHA)	95
9.7.1. Os Sistemas Especialistas e a Estatística.....	96
9.7.2. As formas tradicionais.....	97
9.7.3. A alternativa proposta	99
9.8. O MÓDULO GRÁFICO.....	101
9.8.1. Gráficos dinâmicos interativos.....	101
9.8.2. Conceito e métodos	102
9.9. O MÓDULO DE COMUNICAÇÃO.....	103
9.9.1. A correspondência eletrônica.....	103
9.9.2. A conferência <i>online</i>	103
9.9.3. A vídeo conferência	103
9.10. O EDITOR.....	104
9.11. CARACTERÍSTICAS ADICIONAIS.....	104
9.11.1. Interatividade	105
9.11.2. A simulação.....	105
9.11.3. Utilização educacional da simulação	106
9.11.4. Razões para utilização da simulação no ensino.....	106
9.12. CONCLUSÃO	106
10. CONCLUSÕES E RECOMENDAÇÕES.....	108
10.1. CONCLUSÕES.....	108
10.2. RECOMENDAÇÕES.....	110
11. GLOSSÁRIO	112
12. REFERÊNCIAS	122

LISTA DE FIGURAS

Figura 2.1 - Formato típico da técnica da repetição-e-prática.....	11
Figura 4.1 - Representação esquemática do modelo de Ausubel.....	31
Figura 5.1 - Como um arquivo html é e como é visto pelo navegador	44
Figura 6.1 - Rede semântica com conceitos de Estatística Descritiva	52
Figura 6.2 - Uma representação esquemática de um sistema ICAI.....	53
Figura 6.3 - Tipologia dos agentes [NWA96].....	58
Figura 7.1 - Estrutura do modelo multiagentes proposto	63
Figura 7.2 - Janela de observação das atividades dos agentes	64
Figura 7.3 - Avaliação difusa do perfil do usuário.....	65
Figura 7.4 - Janela de registro de atividades do agente "perfil do usuário"	66
Figura 7.5 - Classificação das aplicações.....	66
Figura 7.6 - Classificação das aplicações normalizada	67
Figura 7.7 - Interface do agente AU, obtida através do MSAgent.....	67
Figura 7.8 - Arquitetura simplificada de um sistema híbrido neuro-difuso	69
Figura 7.9 - Uma visão mais detalhada do sistema de inteligência dos agentes	70
Figura 8.1 - Relação entre os paradigmas estruturado e objeto orientado	76
Figura 8.2 - Relação entre agentes e objetos.....	77
Figura 9.1 - Uma visão do sistema, destacando o módulo de navegação	92
Figura 9.2 - Uma visão do módulo de navegação	94
Figura 9.3 - Planilha mostrando o menu flutuante com as opções ativas	100
Figura 9.4 - Uma visão do analisador sintático do sistema.....	101
Figura 9.5 - Uma visão do módulo gráfico do sistema	102
Figura 9.6 - Um exemplo de gráfico dinâmico [BEC88].....	103
Figura 9.7 - O módulo editor do sistema.....	104

LISTA DE TABELAS

Tabela 3.1 - Sistemas de hipertexto/hipermídia de primeira geração	17
Tabela 3.2 - Sistemas de hipertexto/hipermídia de segunda geração.....	18
Tabela 3.3 - Sistemas de hipertexto/hipermídia de terceira geração	19
Tabela 3.4 - Padronizações propostas para os sistemas de hipertexto	20
Tabela 5.1 - Alguns comandos html.....	39
Tabela 5.2 - Alguns aplicativos auxiliares mais comuns [HAL97]	45
Tabela 5.3 - Controles ActiveX mais comuns [HAL97].....	46
Tabela 6.1 - Caracterização dos agentes	59
Tabela 7.1 - Características dos agentes propostos	72
Tabela 8.1 - Camadas de software para a implementação de agentes.....	74
Tabela 8.2 - Linguagens específicas para a programação de agentes	75
Tabela 8.3 - Opções para utilizar controles ActiveX [COF96].....	79
Tabela 8.4 - Principais linguagens de comunicação de agentes.....	80
Tabela 8.5 - Linguagens de troca de conhecimentos	81
Tabela 9.1 - Alguns cursos (tutoriais) <i>online</i>	83
Tabela 9.2 - Resumo dos principais pacotes genéricos	86
Tabela 9.3 - Resumo dos principais pacotes específicos	88
Tabela 9.4 - Resumo dos principais cursos off-line	89
Tabela 9.5 - Resumo das principais extensões para a planilha Excel	91



SIGLAS UTILIZADAS

- ACM** (*Association of Computer Machinery*). Associação de sistemas computacionais.
- AI**. Agentes Inteligentes.
- ANSI** (*American National Standards Institute*). Instituto nacional americano de padronização.
- ARPA** (*Advanced Research Project Agency*). Agência avançada de pesquisa de projetos.
- ASCII** (*American Standard Code for Information Interchange*). Código padrão americano para a troca de informações.
- AVI** (*Audio Video Interleave*). Vídeo e áudio intercalados.
- BASIC** (*Beginners All-purpose Symbolic Instruction Code*). Código de instrução simbólico de propósito geral para iniciantes.
- CAD/CAM** (*Computer-Aided Design/Computer-Aided Manufacturing*). Projeto auxiliado por computador/Manufatura auxiliada por computador.
- CAI** (*Computer-Aided Instruction ou Computer-Assisted- Instruction*). Instrução assistida ou auxiliada por computador.
- CAL** (*Computer-Assisted Learning*). Aprendizagem assistida por computador.
- CASE** (*Computer-Aided Software Engineering*). Software de engenharia com base em computador.
- CBE** (*Computer Based-Education*). Educação baseada no computador.
- CBI** (*Computer Based-Instruction*). Instrução baseada no computador.
- CBL** (*Computer Based-Learning*). Aprendizagem baseada no computador.
- CBT** (*Computer Based-Training*). Treinamento baseado no computador.
- CCITT** (*Comite Consultatif Internationale de Telegraphique et Telephonique ou Consultative Committe on International Telephone and Telegraphy*). Comitê consultivo internacional de telegrafia e telefonia.
- CERN** (*Centre Européen pour la Recherche Nucleaire*). Centro europeu de pesquisas nucleares.
- CGI** (*Common Gateway Interface*). Interface comum de acesso.
- Class** (*Computer-based laboratory for automation of school systems*). Laboratório para automação de sistemas escolares baseado em computador.
- CMI** (*Computer Managed Instruction*). Instrução gerenciada por computador.
- DBMS** (*DataBase Management System*). Sistema de gerenciamento de base de dados.
- DDE** (*Dynamic Data Exchange*). Troca dinâmica de dados.
- DLL** (*Dynamic Link Library*). Biblioteca de conexão dinâmica.
- DNS** (*Domain Name System*). Sistema de nomes de domínio.
- DOS** (*Disk Operational System*). Sistema operacional em disco.

- DTD** (*Document Type Definition*). Definição do tipo de documento.
- ENIAC** (*Electronic Numerical Integrator And Calculator*). Integrador e calculador eletrônico numérico.
- ESS** (*Expert Statistical System*). Sistema especialista estatístico.
- FTP** (*File Transfer Protocol*). Protocolo de transferência de arquivos.
- GUI** (*Graphical User Interface*). Interface gráfico do usuário.
- HTML** (*HyperText Markup Language*). Linguagem de marcação de hipertexto.
- HTTP** (*HyperText Transfer Protocol*). Protocolo de transferência de hipertexto.
- HyTime** (*Hypermedia/Time-based Structuring Language*). Linguagem estruturada de hipermídia baseada no tempo.
- IA**. Inteligência Artificial.
- IACR** (*Integrated Approach to Crop Research*). Instituto de pesquisa de culturas aráveis.
- ICAI** (*Intelligent Computer Assisted Instruction*). Instrução inteligente assistida por computador.
- ILE** (*Intelligent Learning Environment*). Ambiente inteligente de aprendizagem.
- IP** (*Intenet Protocol*). Protocolo da Internet.
- ISDN** (*Integrated Services Digital Network*). Rede integrada de serviços digitais.
- ISO** (*International Standard Organization*). Organização internacional de padronização.
- ISS** (*Intelligent Statistical Software*). Software estatístico inteligente.
- ITS** (*Intelligent Tutoring System*). Sistema de tutoriamento inteligente.
- JPEG** ou **JPG** (*Joint Photographic Experts Group*). Grupo de especialistas em fotografia
- Mbps**. MegaBytes por segundo.
- Memex** (*Memory extender*). Ampliador da memória.
- MIME** (*Multipurpose Internet Mail Extensions*). Extensões multipropósito do correio eletrônico.
- MIT** (*Massachuset Institute of Technology*). Instituto de Tecnologia de Massachuset.
- MPEG** (*Moving Picture Expert Group*). Grupo especializado em vídeo.
- MHEG** (*Multimedia and Hypermedia Experts Group*). Grupo especializado em multimídia e hipermídia.
- MSIE** (*Microsoft Internet Explorer*). Explorador da Internet da Microsoft.
- NATAL** (*National Authoring Language*). Linguagem nacional de autoria.
- NCSA** (*National Center for Supercomputing Applications*). Centro nacional de Aplicações de Supercomputação.
- NFS** (*Network File System*). Sistema de arquivo em rede.
- NIST** (*National Institute of Standards and Technology*). Instituto Nacional de Tecnologia e Padronização.
- NNTP** (*Network News Transport Protocol*). Protocolo de transporte de notícias em rede.
- OCX** (*Object Linking and Embedding*). Inserção e conexão de objetos.
- ODA** (*Office Document Architecture*). Arquitetura de documentos de escritório.
- OLE** (*Object Lynking and Embedding*). Inserção e conexão dinâmica de objetos.

- ORMS** (*Operational Research/Management Science*). Pesquisa Operacional e Administração.
- PARC** (*Palo Alto Research Center*). Centro de pesquisa de Palo Alto - Califórnia (USA).
- PC** (*Personal Computer*). Sigla utilizada para denominar um computador pessoal, isto é, o microcomputador.
- PLATO** (*Programmed Logic for Automatic Teaching Operation*). Lógica programada para o ensino automatizado.
- PNG** (*Portable Network Graphics*). Gráficos de rede portáteis.
- POP** (*Point-of-presence*). Ponto de presença.
- Prolog** (*Programming in logic*). Programação em lógica.
- RAD** (*Rapid Application Development*). Desenvolvimento rápido de aplicações.
- RFC** (*Request For Comments*). Requisição para comentários.
- SAID** (*Speech Auto-Instructional Device*). Aparelho de fala autoinstrucional.
- SGML** (*Standard Generalized Markup Language*). Linguagem de marcação padrão generalizada.
- SMTP** (*Simple Mail Transfer Protocol*). Protocolo simples de transferência de mensagens.
- SOCRATES** (*Systems for Organizing Content to Review and Teach Educational Subjects*). Sistema para organizar conteúdos para revisar e lecionar assuntos educacionais.
- SOPHIE** (*Sophisticated Instructional Environment*). Ambiente instrucional sofisticado.
- SPSS** (*Statistical Package for the Social Sciences*). Pacote estatístico para as ciências sociais.
- TCP** (*Transmission Control Protocol*). Protocolo de controle de transmissão.
- TCP/IP** (*Transmission Control Protocol/Internet Protocol*). Protocolo de controle de transmissão/protocolo da Internet.
- TICCIT** (*Time-shared Interactive Computer Controlled Information Television*). Televisão interativa de tempo compartilhada controlada por computador.
- TLTP** (*Teaching and Learning Technology Programme*). Programa de ensino e aprendizagem tecnológico.
- UDP** (*User Datagram Protocol*). Protocolo de dados do usuário.
- URL** (*Uniform Resource Locator*). Localizador uniforme de recursos.
- VMS** (*Virtual Memory System*). Sistema de memória virtual.
- WAIS** (*Wide Area Information System*). Sistema de informação de área ampla.
- WWW** (*World Wide Web*). Grande rede mundial.
- XBM** (*X BitMap*). Mapa de *bits* estendido.



1. INTRODUÇÃO

1.1. O Problema

O ensino de disciplinas que envolvem raciocínio abstrato como as da área Matemática (Probabilidade, Cálculo, Álgebra Linear, Geometria Analítica, Equações Diferenciais, etc.) e as que envolvem e exigem modelagem, isto é, aplicações de modelos teóricos como as da área de Matemática Aplicada, representadas pela Estatística, Física, Química, etc. para os cursos universitários, é feito, apesar do desenvolvimento acelerado dos meios eletrônicos, especialmente dos computadores, quase que exclusivamente através de aulas expositivas. O esforço é quase que inteiramente exercido pelo professor/treinador, cabendo ao aluno ou a pessoa a ser treinada pouca ou nenhuma participação. Marcoulides [MAR90] coloca: "*os estudantes são essencialmente participantes passivos no processo pedagógico com suas atividades restritas a ouvir, tomar notas e revisar exercícios já feitos*". Isto gera desestímulo e baixa produtividade em virtude de que o receptor do conhecimento ou treinamento assume uma atitude passiva e não participativa. Muito do que o professor/treinador pretende transmitir não é aproveitado por não despertar o interesse do envolvido, pela quantidade de informações acima do que ele pode assimilar, pelo pouco tempo de reflexão sobre os conhecimentos sendo transmitidos, pelo pequeno número de exemplos e, muitas vezes, pelo despreparo ou pouca qualificação do professor/treinador. O aluno não dispõe de exercícios em quantidade suficiente, bem como não pode fazer experimentações por si próprio de forma a ver como "a coisa funciona". O ensino destas disciplinas, que exigem uma carga de pré-requisitos substanciais, se ressentem cada vez mais com o despreparo da população estudantil, demonstrada através da falta de pré-requisitos mínimos para absorver os novos conhecimentos sendo transmitidos.

Este não é um fenômeno restrito a certos locais ou países. Nicolson e Simpson [NIC92] do departamento de Psicologia da universidade de Sheffield observam que: "*o ensino de Estatística é um dos menos populares aspectos da vida acadêmica universitária, visto como um verdadeiro inferno ao invés de um desafio intelectual excitante. Além disso o ensino de Estatística é, muitas vezes, visto pelos próprios professores como uma das menos gratificantes tarefas universitárias. A falta de entusiasmo dos professores pelo seu trabalho (e, freqüentemente, a falta de conhecimento, também) pode conduzir todos os envolvidos a uma experiência educacional altamente insatisfatória*" (p. 520).

Além da problemática acima, a grande diversidade da clientela dos cursos de Matemática e de aplicações da Matemática na graduação exige que se tente uma nova abordagem. A heterogeneidade do público alvo de um professor destas disciplinas exige que ele seja um especialista nas diversas áreas do conhecimento como forma de adequar sua linguagem, seus exemplos e seus exercícios. Conteúdos de Matemática e de Estatística (Matemática aplicada) são lecionadas para a quase totalidade dos cursos universitários, envolvendo disciplinas que variam de 2 a 6 créditos em cursos de 1, 2 ou até 3 semestres (como é o caso do Cálculo para as Engenharias). Como o professor da disciplina possui uma formação específica (Matemática, Estatística, Engenharia, Economia,

etc.), além do conhecimento especializado dos cursos de pós-graduação, é natural que ele direcione seus exemplos e exercícios para a área de seu maior conhecimento tornando sua linguagem mais difícil ou ininteligível para boa parte dos alunos. Junte-se agora uma deplorável formação básica numérica de grande parte do público alvo que deve cursar estas disciplinas, juntamente com idéias pré-concebidas de uma matéria difícil e chata, enfrentando um professor falando outra língua e citando exemplos totalmente fora de sua área de interesse e poderemos imaginar um panorama dos possíveis resultados.

Devido a diversidade de clientela que freqüentam estes cursos, é necessário que se tente novas abordagens adequando o ensino ao aluno e não o aluno ao ensino como é normalmente feito hoje. Uma maneira de lidar com o problema é disponibilizar materiais instrucionais (apostilas, exemplos e exercícios) diferentes para cada público. Assim, alunos de Engenharia teriam um material que envolvesse e lidasse basicamente com problemas de Engenharia, enquanto que um aluno da área de Biologia ou Medicina teria uma abordagem direcionada a sua área. Evidentemente a conciliação de tal diversidade não é fácil e nem simples, mas com os recursos atualmente disponíveis de laboratórios computacionais e principalmente da Internet acredita-se que haja condições de o problema começar a ser encaminhado.

Considerando a situação descrita, o que este trabalho propõe é um modelo de auxílio ao ensino que objetiva aumentar o desempenho tanto do professor quanto do aluno. Esta abordagem propõe apresentar os conteúdos através de um sistema de hipermídia ao estilo WWW com o auxílio da Internet e de um sistema multiagentes, que coordene a apresentação dos conteúdos ao tipo de usuário (conforme curso), apresenta material de reforço quando necessário e gradua de forma inteligente a apresentação e a quantidade de exercícios como forma de promover o aprendizado e forçar a retenção na memória de longo prazo.

Disciplinas que envolvem modelos complexos como as citadas acima necessitam de uma prática constante e metódica de resolução de exercícios. Exigem a revisão dos exemplos e o estudo de casos como forma de promover a compreensão dos conceitos e permitir a aquisição de uma visão ampla de interações entre estes conceitos. Sem estes requisitos a transferência de conhecimentos dificilmente será alcançada, tornando o ensino uma mera aplicação de fórmulas que desestimulam o estudo com o conseqüente fracasso na aprovação e a sensação de que os conteúdos sendo vistos "não servem para nada".

A proposta não é apresentar um sistema CAI, ICAI ou similar, onde o aluno tenha que, além de assimilar conteúdos, se familiarizar com uma nova interface, um novo software, aumentando assim o tempo necessário para chegar ao que realmente interessa. O que se propõe é um modelo que utiliza a hipermídia que comprovadamente já deu certo e funciona através da WWW. O modelo proposto funcionará exatamente como qualquer outro software para Windows e agregará uma base de suporte (*site*) que fornecerá através de um agente identificador o material adequado ao curso de cada usuário.

O primeiro componente do modelo é um navegador multimídia, com capacidade de navegação e recursos semelhantes a softwares tradicionais do tipo Netscape, Internet Explorer e Mosaic. A diferença é que este navegador é projetado com o objetivo específico de apresentar material de ensino, possuindo portanto características e recursos específicos para o ensino deste tipo de disciplina e a possibilidade de acessar bases de conhecimentos diferenciadas, através de um agente, em função do tipo de usuário.

Um segundo componente do modelo é um sistema gráfico que suporta diagramas em 2D e 3D de forma dinâmica, permitindo a construção de gráficos e diagramas e sua

manipulação instantânea, seja pela alteração de um valor na base de dados ou de algum elemento no próprio diagrama. Agregado ao sistema gráfico existe um analisador sintático que transforma entradas em forma de texto em equações que são visualizadas de forma imediata na forma gráfica. Desta forma o usuário poderá visualizar imediatamente não apenas equações ou distribuições embutidas no sistema, mas qualquer outra que achar desejável.

Um terceiro componente do modelo é um conjunto de recursos adicionais que inclui: um editor de texto, que permite fazer pequenas modificações ou correções nos conteúdos do navegador e mediante a recarga do arquivo a correção ou modificação poderá ser visualizada instantaneamente; um nódulo de comunicação formado por um recurso de correio eletrônico (*e-mail*) e um módulo para conferência simultânea (*chat*).

Para a aplicação específica na área de Estatística, o modelo agrega uma planilha para a prática de exemplos apresentados no navegador. Esta planilha dotada de uma interface com conhecimento identificará o tipo de variável existente na base de dados, importada ou digitada pelo usuário, e em função do tipo de variável selecionada e de seu número (uma, duas ou mais) apresentará menus específicos que limitem ou mesmo inviabilizem a possibilidade de erros ou análises sem sentido por parte do usuário. A interface da planilha é baseada numa visão global de análise e não em tópicos ou técnicas específicas. Nela não será possível encontrar os tradicionais menus contendo um leque de técnicas a serem aplicadas.

A planilha funciona de maneira integrada ao navegador, estendendo o paradigma dos endereços URLs da WWW através de acréscimos específicos, como a URL exemplo `::/ exemplo-1`, que permite transferir o conteúdo do exemplo para a planilha. Da mesma forma é feita a transferência de uma equação escrita em html no navegador para o módulo gráfico através de uma URL equação `::/ equação-1`.

Os recursos apresentados acima serão integrados e gerenciados por um modelo multiagentes. O conjunto de agentes é formado por um agente com a função de identificar o usuário e verificar se ele está ou não cadastrado no sistema e também fornecer os conteúdos adequados obtidos da base de conhecimentos (páginas html). Um segundo agente se encarregará de traçar um perfil inicial do usuário e enviar este perfil para um agente responsável por manter e atualizar a base de perfis. Em função do seu perfil o usuário será abastecido com exercícios retirados de uma base de aplicações por um agente que possui a tarefa de manter e atualizar esta base. Finalmente caso o usuário encontre dificuldades um agente com a finalidade de monitorar o seu desempenho se encarregará de lhe fornecer subsídios através de dicas ou exemplos relacionados a aplicação sendo trabalhada. As dicas e exemplos serão buscadas numa base criada especificamente com esta finalidade.

1.2. Relevância

Disciplinas que envolvem conceitos abstratos e modelagem de dados exigem uma carga maior de dedicação e estudo para o seu entendimento e principalmente a prática de uma grande número de exercícios para o seu completo domínio. Este tipo de disciplina ocorre na maioria dos currículos universitários, sendo especialmente freqüente na área de Ciências Exatas e particularmente importante nas Engenharias.

Por envolverem um tipo de raciocínio não muito praticado nas séries iniciais (primeiro e segundo graus), elas são responsáveis por um número elevado de reprovações, constituindo a principal barreira para que um número substancial de alunos completem sua graduação no prazo normal. Assim, qualquer trabalho que vise melhorar

o ensino destas disciplinas e aumentar o rendimento dos alunos contribuirá para que este quadro possa ser amenizado e conseqüentemente evitará o desperdício de muitos recursos. Uma melhora no rendimento destas disciplinas permitirá também uma maior liberação de vagas nas séries iniciais destes cursos de graduação, contribuindo para que um maior número de candidatos entrem na universidade.

O modelo multiagentes proposto poderá ser utilizado como apoio ao ensino em várias disciplinas da área de Ciências Exatas, possibilitando que o aluno obtenha maior desempenho no seu estudo através de exercícios adequados que o estimulem a prosseguir a níveis crescentes de dificuldade. Entretanto sua aplicação ao ensino de Estatística requer, em virtude, das suas particularidades, o acréscimo de uma planilha. A proposta é acrescentar ao modelo uma planilha com interface inteligente (com conhecimento embutido na forma de redes semânticas).

1.3. Objetivo Geral

O trabalho tem como finalidade colocar à disposição de professores e alunos de disciplinas da área de Ciências Exatas um modelo multiagentes que integre as habilidades de navegar na rede, rodar programas, fazer simulações, construir gráficos e realizar exercícios adequados ao perfil de cada aluno e de acordo com os objetivos da disciplina sendo lecionada. Um modelo de ensino que utiliza agentes para apresentar exercícios e mediar sua carga de acordo com o usuário, tornando o ensino mais individualizado e colaborativo.

Como resultado, espera-se obter com a aplicação do modelo, um sistema interativo, de fácil manuseio, auto-adaptativo, expansível e reutilizável para o ensino e treinamento em disciplinas da área de Ciências Exatas (particularmente Estatística e Probabilidade). Espera-se que a aplicação do modelo possa aumentar o rendimento sobre o sistema tradicional de ensino/treinamento e também permitir que o aluno possa estudar no seu próprio ritmo, tendo ao mesmo tempo a sua disposição uma base de conhecimentos que possa suprir de forma imediata e automática lacunas existentes na sua formação.

1.4. Objetivos Específicos

Em termos específicos, esta tese tem por objetivos:

a) Apresentar o estado-da-arte em hipertexto/hipermídia

Os sistemas de hipertexto/hipermídia estão se expandindo rapidamente em todas as áreas do conhecimento, impulsionados pelas novas interfaces gráficas (como o Windows, por exemplo) e pelo rápido crescimento do mercado de computação multimídia, que adicionou aos computadores a capacidade de armazenar e reproduzir som e vídeo. A possibilidade de adicionar a um documento escrito som, vídeo, animação e mesmo programas abriu novas perspectivas para a apresentação de informações e desta forma ao ensino e treinamento tradicional.

No entanto, como em todas as áreas que experimentam uma rápida expansão, a profusão de conceitos, definições, sistemas, técnicas, arquiteturas, formas, etc. criam um clima de caos e torna-se necessário um minucioso estudo para filtrar e organizar estes novos conceitos e desta forma selecionar uma metodologia coerente e adequada ao objetivo proposto, de forma a optar pela melhor abordagem para o ensino e treinamento em Estatística e outras disciplinas semelhantes dos cursos de Engenharia.

b) Trazer contribuições ao ensino

Esta é a principal contribuição do trabalho. A integração dos modelos de navegação hipermídia, planilha, recursos de comunicação e um conjunto de agentes trará uma contribuição sobre o sistema tradicional de ensino/treinamento e especialmente sobre o ensino e treinamento mediado por computador.

c) Trazer contribuições ao ensino de disciplinas da área de Ciências Exatas

O modelo multiagentes proposto poderá ser utilizado em qualquer disciplina da área de Ciências Exatas e que exija portanto uma carga substancial de prática por parte do aluno, através da resolução de um grande número de exercícios;

d) Trazer contribuições ao ensino de Estatística e Probabilidade

A contribuição específica se dará no ensino de Estatística e Probabilidade, através de uma nova forma de abordar os conteúdos de Estatística que enfatize uma visão global e um entendimento dos propósitos e necessidades de se utilizar determinados recursos, ao invés de se colocar ênfase no manuseio de técnicas específicas.

1.5. Tecnologias básicas

O alcance dos objetivos propostos neste trabalho pressupõe a utilização das seguintes itens:

- Sistemas de hipertexto/hipermídia;
- WWW e Internet;
- Sistemas de ensino;
- Paradigma da orientação a objetos;
- Linguagem de programação Delphi;
- Interface com conhecimento;
- Sistemas multiagentes;
- Sistemas neuro-difusos;
- Analisador sintático;
- Simulação e gráficos dinâmicos.

1.6. Delimitação do estudo

O modelo de ensino proposto, poderá ser aplicado a muitas disciplinas da área de Ciências Exatas e principalmente nas Engenharias. Ele poderá ser útil como apoio ao ensino de Cálculo, Álgebra Linear, Geometria Analítica, Probabilidade e Estatística. Este ensino poderá ser off-line ou *online* (à distância). No entanto, para permitir a operacionalização, o protótipo (aplicação do modelo) está sendo feito ao ensino/treinamento em Probabilidade e Estatística Básica. O assunto foi escolhido por:

● Ter uma metodologia complexa. Metodologias complexas são volumosas, sem flexibilidade, difíceis de serem atualizadas e não são fáceis de usar. Hipertextos bem projetados aumentam o valor e a utilidade de metodologias complexas.

● Apresentar procedimentos e orientações complexas dos quais muitos profissionais não conhecem os detalhes que necessitariam conhecer. Estes procedimentos e orientações podem ser colocados sob a forma de hipertextos, levando ao usuário à informação certa e na hora certa. Hipertextos bem projetados podem diminuir sensivelmente os perigos da prática profissional equivocada.

- Normas, planos, gráficos e tabelas são normalmente volumosas e difíceis de manejar e podem ser projetados na forma de hipertexto para otimizar a ajuda que podem fornecer ao usuário.

- Exigir conhecimentos abstratos, aplicações de modelos matemáticos e probabilísticos e conseqüentemente necessitar de uma quantidade bastante grande de exercícios por parte do aluno.

1.7. Estrutura do trabalho

O trabalho foi dividido em doze capítulos, como forma de abordar o assunto em abrangência (computador no ensino, hipermídia no ensino, o papel da WWW e da Internet) e profundidade (a Inteligência Artificial no Ensino, o papel dos agentes, a aplicação pretendida).

No presente capítulo foi apresentado a motivação, a relevância, os objetivos (o geral e os específicos), a justificativa, a delimitação e a estrutura do trabalho.

No capítulo dois é feita uma abordagem de como o computador vem e está sendo utilizado no ensino, as principais formas e as linhas de desenvolvimento.

No capítulo três é feito um levantamento abrangente da hipermídia. Seu desenvolvimento, uma classificação dos sistemas de hipertexto que levaram ao desenvolvimento da WWW. Discute-se, também, os principais componentes da hipermídia, suas aplicações e sua relação com o texto tradicional e os sistemas especialistas.

O capítulo quatro apresenta um panorama de como a hipermídia foi e está sendo utilizada no ensino. Quais são as propriedades e características que a tornam útil para educação.

No capítulo cinco é feito um levantamento da WWW e da Internet como recursos instrucionais e verificado quais as características e necessidades para que elas alcancem o objetivo.

No capítulo seis é apresentado uma revisão sobre a Inteligência Artificial no ensino, teoria dos agentes, dos sistemas inteligentes tradicionais e do hipertexto inteligente.

O capítulo sete detalha o modelo multiagentes sendo proposto e apresenta os sistemas neuro-difusos que formarão a inteligência de alguns agentes.

O capítulo oito discute as formas de implementação do modelo proposto e apresenta os motivos da alternativa escolhida.

O capítulo nove discute a aplicação do modelo ao ensino de Probabilidade e Estatística e propõe a ampliação do modelo com o acréscimo de uma planilha com conhecimento para atender as particularidades do ensino destas disciplinas.

No capítulo dez são apresentados as conclusões do trabalho e feita recomendações sobre sua ampliação.

O capítulo onze apresenta um glossário de termos e conceitos utilizados.

O capítulo doze apresenta a bibliografia utilizada.



2. A EDUCAÇÃO E O COMPUTADOR

2.1. Introdução

O ensino com base no computador se desenvolveu paralelo à programação dos computadores iniciada no final dos anos 50 e recebeu denominações como: CAI (*Computer-Aided/Assisted Instruction*), CAL (*Computer-Assisted Learning*), CBE (*Computer Based-Education*), CBI (*Computer-Based Instruction*) CBL (*Computer Based-Learning*), CBT (*Computer Based-Training*) ou CMI (*Computer Managed Instruction*). O desenvolvimento destes sistemas ocorreu em fases distintas. Cada uma é caracterizada pelo seu comportamento diante das necessidades de hardware e software, frente à psicologia da aprendizagem e pelas barreiras enfrentadas para a sua ampla aceitação nas escolas e universidades.

A primeira fase do ensino computadorizado vai de 1958, quando as primeiras aplicações dos computadores ao ensino foram consideradas, até 1961, quando a primeira conferência em aplicações de computadores ao ensino automatizado foi realizada. Esta fase foi caracterizada principalmente por questões de hardware, e, em particular, em como conectar terminais, projetores de slides e outros aparelhos de entrada e saída aos computadores para uma interação prática [VEN91].

Nestes programas lineares, a informação era apresentada em etapas curtas e fáceis, a fim de que o reforço fosse dado a todas as respostas emitidas pelo aluno, minimizando assim o número de erros. Segundo os seguidores de Skinner, a graduação do conteúdo ajudava o estudante a dominar progressivamente aspectos mais avançados.

A instrução programada se desenvolveu e tomou diversas formas com o objetivo de alcançar um ensino mais individualizado. Uma das mais utilizadas, denominada de programação intrínseca, de Norman Crowder (1960), foi utilizada pela maioria dos programas CAI. Nela as respostas fornecidas pelo aluno encontravam várias explicações, permitindo que fossem seguidos caminhos diferentes e que cada aluno chegasse ao final do estudo em períodos de tempos diferenciados [NIE89].

A segunda fase estende-se de 1962 até 1967. Durante este período incentivados pelo aumentos dos fundos governamental e militar nos Estados Unidos, muitos projetos de CAI surgiram, caracterizados pelos suas siglas (por exemplo: CLASS, PLATO, SOCRATES, SAID) bem como pelas suas orientações computacionais e psicológicas.

A terceira fase foi de grandes projetos, em particular, PLATO e TICCIT, que receberam volumosos recursos, muita publicidade e poucas avaliações críticas durante este tempo. Avaliações posteriores, mais criteriosas, levadas a cabo pelo Serviço de Testes Educacionais Americano e publicados no final dos anos 70, marcaram o fim desta era de expectativas inflacionadas, dando origem a estudos mais sérios e produtivos do potencial do CAI na educação. Domínios como a Matemática que possibilitam a geração e solução automática de exercícios numéricos propiciaram, ainda nesta época, o surgimento de sistemas capazes de gerar automaticamente o material instrucional, no todo ou em parte, denominados de generativos. Estes sistemas eram capazes de corrigir os exercícios confrontando a resposta do estudante com a do sistema, mas mesmo os

melhores sistemas generativos, não foram hábeis para ir muito além de informar ao aluno a opção correta e não eram capazes de mostrar como a solução foi obtida [YAZ87].

A tentativa de solucionar as deficiências dos sistemas generativos levou ao desenvolvimento dos denominados sistemas tutores inteligentes ou ITS (*Intelligent Tutorial System*). Estes sistemas surgiram da aplicação de técnicas de inteligência artificial sobre os sistemas CAI [BAR86], originando, também, os ICAI (*Intelligent Computer Assisted Instruction*).

A quarta fase, começou em 1977 com a manufatura do primeiro microcomputador e vai até o surgimento da WWW. Ela é marcada por uma mudança de atitude em relação ao CAI e uma substancial recuperação do interesse industrial em produzir hardware e software CAI.

Estes novos sistemas proporcionam avisos de erro, oferecendo a alternativa correta de solução, e também podiam formular hipóteses sobre a história do aluno, e conhecer a origem da concepção incorreta. O diálogo entre o usuário e o programa é uma simulação em que o computador aparenta possuir uma compreensão do raciocínio, mas as advertências e explicações dadas são preparadas de antemão pelo programador e impressas na tela, quando certas contingências são satisfeitas [CAR92]. A maioria desses sistemas trabalha com enfoque tutorial e estão restritos a determinados domínios.

A última fase, a atual, começa em 1992 com o surgimento da WWW e se estende até hoje. Caracteriza-se pela descentralização do ensino. Neste caso, o material é colocado em um único computador, o servidor, ao invés de em cada computador das fases anteriores, e o usuário acessa o material através de um software cliente (o navegador). A manutenção do material é mais fácil de ser feita, pois a atualização e correção de erros pode ser feita até diariamente. No modelo anterior a manutenção do material era praticamente igual a do material impresso. Nesta nova fase o conhecimento, ou cursos, são distribuídos através da Internet e acessados através de WWW.

2.2. Teorias de educação e o computador

Para a epistemologia o conhecimento pode ser empírico, racional ou relativo. O empirismo acredita que o conhecimento é obtido através da experimentação e parte do princípio que a mente seria, inicialmente, uma tábula rasa na qual o conhecimento iria sendo gravado como fruto da experiência. O racionalismo rejeita a informação sensorial e defende a idéia de que tudo está sujeito à razão e supõem que esta capacidade é inata no indivíduo. Já para o relativismo o conhecimento não é resultante da experiência sensorial e nem da capacidade inata do sujeito, mas obtida através da interação entre o sujeito e o objeto sendo estudado.

Estas teorias do conhecimento na área da Psicopedagogia transformaram-se nos movimentos ou escolas:

- Behaviorista ou comportamentalista;
- Cognitiva;
- Interacionista ou construtivista.

O movimento **behaviorista** começou a ganhar corpo com a descoberta do psicólogo russo Ivan Petrovich Pavlov do reflexo ou resposta condicionada ao estímulo. Nele a aprendizagem é vista como uma resposta a estímulos. Esta linha de pensamento além do reflexo condicionado de Pavlov é formada pelo conexionismo de Edward Lee Thorndike e pelo condicionamento operante de F. B. Skinner.

O conexionismo acredita na "lei do efeito", considerando a recompensa como geradora de aprendizagem e que a punição não exerce efeitos na modificação de comportamentos. Para a teoria de Skinner, quanto mais a conexão estímulo-resposta for praticada mais forte ela se torna, quanto menos, mais fraca. O seu trabalho enfatiza a modificação de comportamentos através do condicionante operante, defendendo a instrução programada. O professor com seu livro texto é a fonte de informação. Os objetivos comportamentais são fornecidos, as aulas são planejadas, os conteúdos são transmitidos, uma orientação prática é fornecida, a retenção e a transferência de atividades de aprendizagem são encorajadas e o teste da informação ensinada é o meio padrão de avaliação.

A teoria **cognitiva** da aprendizagem é baseada no princípio da reorganização de estruturas de conhecimento através de um processo incremental de integração entre o velho e o novo conhecimento [JON88]. O papel do professor, de acordo com esta teoria, não é tentar reproduzir a estrutura da realidade externa dentro da mente do aluno, mas a de ajudar construir uma representação funcional e significativa. Nesta linha de pensamento podem ser enquadradas a teoria cumulativa de Robert Gagné e a teoria da assimilação de David Ausubel.

Para o modelo cumulativo de Robert Gagné, a organização de eventos de aprendizagem é baseado no pré-requisito de relacionamentos entre condutas aprendidas. O método identifica pequenos componentes de um assunto e organiza-os em uma determinada seqüência (tarefas), isto é, existe um conjunto de regras que devem ser seguidas no projeto de eventos instrucionais.

Já a teoria da assimilação do psicólogo, David Ausubel centrou-se na aprendizagem significativa. Seus princípios de subsunção, diferenciação progressiva e reconciliação integrativa formam as bases para a utilização instrucional dos sistemas de hipertextos.

Na linha **interacionista**, o expoente mais conhecido é Jean Piaget e sua teoria construtivista. Para ele, aprender é um ato inteligente, baseado na reorganização das estruturas do conhecimento, em função de interações entre o indivíduo e o ambiente. Neste caso o professor atua como um agente controlador do processo de ensino/aprendizagem. Os estudantes descobrem regras e conceitos através de interações em um ambiente que encoraja as estratégias de soluções de problemas, que são, por sua vez, desenvolvidas enquanto aprendem como pensar. A educação é considerada como uma viagem guiada de experiências preparatórias na qual o estudante pratica tomando decisões simulando situações do mundo real.

Estas teorias psicológicas influenciaram o uso do computador na educação, levando a criação de diferentes tipos de aplicações instrucionais. Um software pode refletir uma ou mais destas abordagens e poderá fazer hipóteses sobre o modelo de ensino e de promoção da aprendizagem que será utilizado quando, de fato, for utilizado. Técnicas distintas serão utilizadas para alcançar os mesmos objetivos em relação a perspectivas filosóficas/educacionais diferentes. **As conseqüências destas teorias sobre a tecnologia educacional determinam os vários modelos de transmitir conteúdos e incentivar a aprendizagem através do uso do computador. A partir dos anos 90, vários projetos [BAR91, MER90, WIN91] tentam ressaltar a diferença de um novo paradigma: promover aprendizagem e não ensinar conteúdos. Dentro deste enfoque o compromisso é procurar um ambiente educativo livre, onde o computador atue como instrumento de exploração e incentivo.**

2.3. Como o computador pode ser utilizado no ensino

O uso do computador como instrumento de ensino tem variado na dependência da teoria pedagógica envolvida. Talvez a abordagem pedagógica, atualmente, mais aceita seja a **abordagem cognitiva**. Esta abordagem enfatiza o conceito de estrutura - a cognitiva - como forma de explicar o processo de aprendizagem. Isto significa que cada pessoa tem uma representação mental (modelo) do seu ambiente, que é construído com base nas suas experiências. A aprendizagem ocorre com a reorganização do seu modelo cognitivo que incorpora uma informação nova num processo denominado de *insight*. Na reorganização da estrutura pré-adquirida para incorporar a nova informação tanto o estudante quanto o professor desempenham um papel ativo. O professor toma partido da experiência prévia do estudante e tenta estabelecer analogias entre a nova informação e a que o estudante já sabe. Neste modelo pedagógico, é do professor o papel principal, embora caiba ao estudante o papel de reorganizar seu conhecimento prévio.

Uma outra abordagem pedagógica é a centrada na escolha do aprendiz, que é, às vezes, denominada como **aprendizagem exploratória**. Esta abordagem é centrada na idéia de que o aprendiz é capaz de controlar com eficiência seu processo de aprendizagem. Nenhuma hipótese é feita sobre como ele incorporaria novas informações. Desta forma, pode-se interpretar esta abordagem como formalmente oposta a abordagem cognitiva, uma vez que esta deixa o aprendiz por si só, e a primeira centra-se no papel do professor.

Tradicionalmente na aprendizagem baseada em computadores, a teoria cognitiva tem estimulado o desenvolvimento de sistemas onde é do computador a responsabilidade de juntar novas informações para a estrutura cognitiva do aprendiz. Neste caso, o computador passa a desempenhar o papel que, em tese, caberia ao professor. Por outro lado, na teoria centrada no aprendiz, o computador desempenha um papel mais passivo, pois é deixado ao aprendiz a tarefa de reorganizar sua estrutura cognitiva.

Na hipermídia, e da mesma forma no modelo proposto, são mescladas as duas abordagens, formando-se um sistema híbrido, no qual as lições são preparadas pelo professor seguindo uma determinada metodologia, mas o aluno tem completa liberdade de como vai explorá-la, sendo entretanto orientado por um sistema multiagentes.

Com base nestas duas teorias pedagógicas pode-se classificar o uso do computador na educação em três diferentes formas [TAY80]: duas como resultado da aplicação direta das teorias pedagógicas ao ensino computadorizado, ou seja, uma tentando substituir o professor, o computador seria o tutor, e a outra, o oposto, deixando todo o esforço a cargo do aprendiz, o computador faria o papel do aluno (*tutee*). A terceira via, seria uma espécie de forma "neutra", nem uma nem outra, o computador seria apenas um auxílio, uma ferramenta (*tool*) [CHA89, ROM90, MER96].

2.3.1. Como tutor

O computador auxilia na apresentação de novos conteúdos que inclui a repetição-e-prática (*drill-and-practice*) e o tutorial. O computador é o professor que fornece o conhecimento ao estudante. O computador pode executar uma série de tarefas para treinar (*drilling*) o estudante, em assuntos onde a incorporação de conhecimento fatural seja necessário, levando-o através de um domínio de conhecimentos, tal como a Estatística, por exemplo, colocando questões apropriadas a cada passo e avaliando o entendimento do estudante pela checagem de suas respostas antes de prosseguir para assuntos mais com-

plexos. Este tipo de aplicação computacional é rotulada ainda como instrução baseada no computador (CBI), instrução assistida por computador (CAI) ou aprendizagem assistida por computador (CAL). O processo, em linhas gerais, é o seguinte:

1. O computador apresenta alguma informação;
2. O estudante é solicitado a responder a questão ou exercício relacionado à informação apresentada;
3. O computador avalia a resposta do estudante de acordo com algum critério específico;
4. O computador determina o que deve ser feito a seguir com base na avaliação da resposta.

Uma aplicação deste tipo pode ainda ser de simulação, de jogos e de soluções de problemas, além das aplicações de repetição-e-prática e tutoriais.

Repetição-e-prática. Esta técnica é utilizada pelos educadores para reforçar o aprendizado através de repetições necessárias para mover novas habilidades adquiridas para a memória de longo prazo. Um formato típico desta técnica é ilustrado na [FOR96]. Uma aplicação muito comum deste tipo inclui a repetição de exercícios matemáticos, onde o computador apresenta um problema e o aluno deve entrar com a resposta correta.

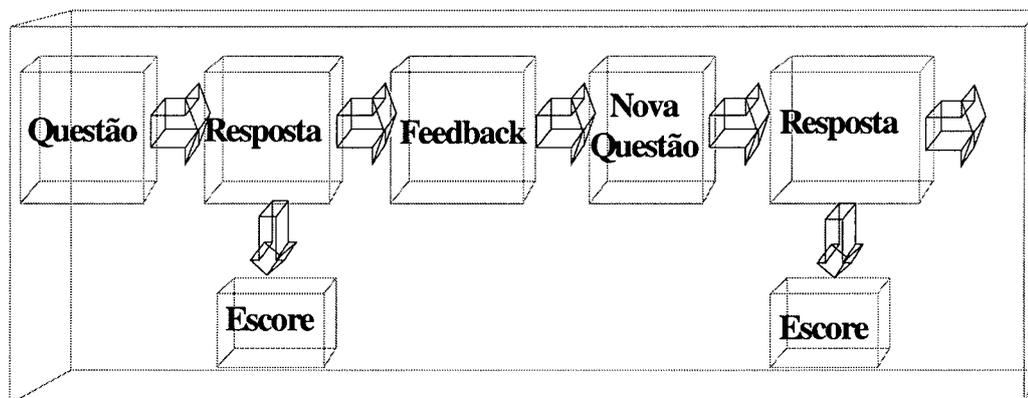


Figura 2.1 - Formato típico da técnica da repetição-e-prática

Aplicações tutoriais. O objetivo principal de uma aplicação deste tipo é apresentar novas informações. Um tutor computacional objetiva colocar o computador no papel de responsável por todas as tarefas de dirigir o aluno na obtenção de um conjunto específico de objetivos. Em contraste, outros tutoriais, tais como repetição-e-prática, simulação e jogos, são projetadas para executar somente uma pequena porção do papel instrucional. A natureza ampla de um tutorial torna-o mais complexo e mais difícil de ser projetado e também mais caro de desenvolver do que outras aplicações.

Simulações. Uma simulação é uma representação ou modelo de um sistema real ou fenômeno. Ela permite que o estudante possa vivenciar certos fenômenos com menos riscos e custos. Uma simulação de um mercado de ações permite que os estudantes compreem e vendam ações sem a necessidade de investir dinheiro real e ver o resultado de suas decisões de forma imediata.

Aplicações de soluções de problemas. São aplicações em que o computador apresenta situações (problemas) que devem ser resolvidas através de um processo de dedução lógica, síntese e implementação. O estímulo do pensamento analítico é uma das principais características deste tipo de software.

Jogos. São utilizados para despertar interesse e motivar o aluno para a situação de aprendizagem. Programas deste tipo envolvem um jogo competitivo entre o estudante e um ou mais oponentes que pode ser o próprio computador ou ele mesmo.

2.3.2. Como assistente

Neste tipo de aplicação, o computador é o assistente (*tutee*) ou o estudante, o usuário torna-se o professor. O usuário precisa ensinar ao computador como executar alguma tarefa. Para fazer isto, ele precisa aprender a como se comunicar com o computador em uma linguagem que o computador entenda, isto é, o usuário precisa aprender a como escrever programas computacionais. Um programa computacional é um conjunto de comandos, dispostos de uma forma particular, que informam ao computador como executar uma determinada tarefa ou resolver determinado problema. Antes que alguém possa ensinar ao computador como resolver algum tipo de problema, ele precisa primeiro entender como o problema pode ser resolvido, isto é, saber resolvê-lo. Isto requer o desenvolvimento de algumas habilidades específicas e muitos educadores concordam que o desenvolvimento de tais habilidades é um dos principais objetivos educacionais [MER96].

Quando o computador é utilizado desta forma o estudante tem oportunidade de aprender ensinando. O processo de ensinar ao computador como fazer algo requer que o estudante, como professor, analise e entenda a tarefa sendo resolvida de uma maneira mais profunda do que normalmente seria necessário, pois a tarefa a ser feita precisa ser plenamente entendida, já que o computador necessita de uma lista precisa e logicamente coerente de instruções para executar ou resolver qualquer tarefa. O computador é dirigido pelo estudante. Ele pode desempenhar um papel mais passivo no sentido de que é o aluno quem decide a seqüência de passos que ele deve seguir de forma a entender o assunto em um domínio particular. Existe uma interação constante e uma completa liberdade de ação por parte do aprendiz.

O ensino de linguagens de programação aos estudantes em geral é defendido não com o intuito de torná-los programadores profissionais, mas sim com o objetivo de melhorar o entendimento de como o computador funciona. Isto reforçaria a habilidade de resolver problemas e forneceria um sentido de controle, aumentando, desta forma, a compreensão da utilização das ferramentas computacionais. Os estudantes certamente obterão um entendimento melhor dos conteúdos se tiverem a oportunidade de ensinar ao computador estes conteúdos [MER96].

Para muitos, o computador permanece um mistério que funciona tipo uma "caixa mágica". O conhecimento de programação ajudará, em muito, no entendimento de como a máquina funciona e este entendimento, por sua vez, vai remover muito do medo e dos mitos associados ao seu uso. Afinal, o computador está presente e estará cada vez mais em todas as tarefas cotidianas e um entendimento de como ele funciona é e será uma parte importante da educação geral. Papert [PAP80] enfatiza que a programação é um dos melhores recursos disponíveis para ensinar crianças a aprender fazendo e a pensar no que estão fazendo. Quem leciona a algum tempo há de concordar plenamente com esta afirmação ao verificar que grande parte dos erros cometidos pelos alunos, são resultados da má interpretação de questões ou de uma leitura desatenta das instruções.

2.3.3. Como auxílio

O computador ajuda o estudante no processo de aprendizagem, mas não dirige os seus esforços. O computador serve como um instrumento para auxiliar e para executar

tarefas, como na execução de um processador de texto ou planilha de cálculos. Além disso, permite que o estudante trabalhe de uma forma mais transparente e eficiente.

As ferramentas para aumentar nossa capacidade intelectual, principalmente a habilidade para realizar uma maior quantidade de cálculos e com menos erros, começaram com a régua de cálculo, passaram pelas calculadoras e chegaram ao computador. Como consequência de encarar o computador como uma ferramenta, as suas aplicações são também denominadas de **aplicativos**. Por aplicativos quer-se denominar todos aqueles recursos computacionais que não lidam diretamente com instrução (*tutoriais*) ou com programação (*tutee*). Se bem que, devido aos recursos de "ajuda" e de "programação" dos softwares de hoje, esta distinção nem sempre poderá ser feita sem um certo grau de dificuldade. No entanto, a idéia é de que como tutor o computador tenta desempenhar o papel de professor, como assistente (*tutee*) a de aluno e como ferramenta (*tool*) a de auxiliar.

Os softwares normalmente incluídos na categoria de ferramentas computacionais ou aplicativos são: os processadores de texto, as planilhas de cálculo, os programas de desenho e de tratamento de imagens, os gerenciadores de base de dados, os programas para a construção de gráficos, etc. Alguns aplicativos se tornaram tão sofisticados e com tal quantidade de recursos que já estão sendo utilizados como recursos didáticos, sem bem que ainda sem a idéia de tentar ensinar o conteúdo. Estes aplicativos atuam como calculadoras poderosas e sofisticadas, como é o caso da planilha Excel, que tem incorporado uma grande número de funções Estatísticas e Probabilísticas e que, hoje, pode ser usada como um razoável recurso instrucional em aulas dessas disciplinas. O mesmo vem acontecendo com softwares do tipo, Derive, Maple, MathCad, Mathematica e Matlab que estão sendo cada vez mais utilizados como recursos instrucionais em disciplinas de Cálculo e mesmo em Álgebra Linear e Geometria Analítica. Softwares mais sofisticados e específicos, como o SPSS, o Statistica, o StatGraphics, o Minitab, também já se tornaram ferramentas comuns em muitas disciplinas que envolvam análise de dados com algum grau de complexidade.

2.4. Linhas de desenvolvimento

No passado, a abordagem mais comum para qualquer educador que desejasse desenvolver seus próprios programas era valer-se de uma das muitas linguagens disponíveis. No início o Basic era a escolha mais popular, provavelmente porque esta linguagem fosse fornecida junto com a compra de muitos micros [MAD92]. O Pascal era outra linguagem, também, freqüentemente utilizada. Estas linguagens eram utilizadas para escrever softwares pouco sofisticados e geralmente para consumo próprio, isto é, para utilizar com os próprios educandos. A situação atual é bastante diferente. Hoje algumas dessas linguagens ainda podem estar sendo utilizadas, mas a tendência mudou em direção ao uso de linguagens mais poderosas, como o C, e também mais específicas e adequadas a certas áreas, como o Lisp e o Prolog, para programas que envolvam sistemas especialistas. Entretanto, essas linguagens ainda apresentam as restrições de um paradigma de construção de software (a abordagem estruturada) que já esgotou seu ciclo de aperfeiçoamento e apresenta uma série de problemas, como um baixo reaproveitamento de código já existente, baixa produtividade e dificuldade de manutenção. A tendência dominante nos anos 90 é em direção a um novo paradigma de programação, a orientação a objetos, que parece encaminhar soluções de problemas que a abordagem estruturada não conseguiu resolver.

Uma ferramenta que auxilia o usuário a escrever seu próprio programa instrucional é denominado de "software de autoria". Existem muitos tipos de softwares de autoria,

cada qual com diferentes objetivos e capacidades. O leque varia desde os altamente prescritivos e fáceis de usar, mas com um limitado número de opções, até os totalmente equipados que são somente limitados pela habilidade do programador e pela configuração de equipamentos do usuário. Em cada caso, um software de autoria oferece protocolos para criar instruções computacionais e muitos podem ser utilizados para criar instruções multimídia interativas. Estes softwares podem ser classificados em: (a) linguagens de autoria, (b) sistemas de autoria e/ou auxílio.

2.4.1. Linguagens de autoria

Uma linha de pesquisa importante, além dos sistemas inteligentes de ensino, foi o desenvolvimento das linguagens de autor e ambientes de autoria. As linguagens de autoria são linguagens computacionais de propósito especial que são projetadas expressamente para facilitar a escrita de programas educacionais. Para Maddux [MAD92], as linguagens de autoria são mais simples e podem ser apreendidas mais facilmente que as linguagens tradicionais em virtude de serem desenvolvidas especificamente para educação, sendo ideais para escrever programas conversacionais ou interativos. Das muitas linguagens de autoria surgidas uma das mais interessantes foi a Natal [VEN91], cujo projeto foi financiado pelo Conselho Nacional de Pesquisa do Canada. A Natal é uma linguagem estruturada em blocos, semelhante ao Algol, que requer declaração de tipos, mas com facilidades para programadores, mesmo os mais inexperientes.

Um tipo especial de linguagem de autoria que pode e é utilizada para escrever softwares educacionais é a linguagem de autoria de hipertexto, cuja finalidade é a de produzir sistemas educacionais de hipertexto ou hiperímia. Um dos mais conhecidos representantes deste tipo de software é a linguagem denominada HyperTalk. Esta linguagem permite a escrita de programas instrucionais nos computadores do tipo Macintosh, que rodam o HyperCard.

2.4.2. Sistemas de autoria

Os sistemas de autoria foram projetados com a intenção de permitir a escrita de software educacional sem a necessidade de programação, ou seja, seriam um passo acima na escala de automatização da construção de software além das linguagens de autoria. A grande desvantagem deste tipo de sistema é a falta de flexibilidade, pois normalmente o usuário precisa se adaptar ao projeto do sistema, uma vez que a maioria das decisões já foi tomada pelo construtor do software. Em resumo, os sistemas de autoria traduzem as escolhas do autor no código, enquanto que o usuário de uma linguagem de autoria ou tradicional precisa tomar todas as decisões e gerar o código linha por linha.

Um exemplo de um sistema deste tipo é o Socratic, que permite ao usuário a entrada de dados por menu e permite, também, que usuários avançados do sistema incorporem, ao mesmo, seus próprios programas escritos em linguagens tradicionais, como o Basic e o Pascal

Um outro exemplo deste tipo de sistema é o Plato, que foi expandido para um ambiente de autoria pleno pela incorporação não somente da linguagem de autoria Tutor, mas também de um editor de texto, um corretor de erros, de documentação *online*, de um sistema de gerenciamento, de uma biblioteca e de correio eletrônico, entre outras características.

Sistemas auxiliares de autoria incluem ainda os modelos (*templates*), os programas geradores de código e os softwares editáveis. Os dois primeiros requerem que o usuário

seja um programador competente e eles são, de fato, ferramentas de programação. O softwares editáveis possuem menos poder e flexibilidade, mas são também mais simples de usar. Tais recursos permitem que o educador tome certas decisões que são normalmente deixadas para o projetista. As escolhas normalmente referem-se ao nível de dificuldade, a velocidade de execução ou ao tipo de reforço [MAD92].

2.5. Conclusão

O desenvolvimento e a larga aplicação dos sistemas computacionais instrucionais só começaram com a popularização do computador, que aconteceu após a invenção do microcomputador. Com o computador transformando-se de um bem escasso e extremamente caro para um eletrodoméstico acessível a grande parte da população, sua aplicação ao ensino tornou-se não só viável, mas inevitável.

As atividades de aprendizagem são baseadas em teorias (ou crenças) de como os estudantes aprendem. Das muitas doutrinas filosóficas, duas se destacam claramente como exemplos relacionados ao desenvolvimento, seleção e uso do software: o comportalismo e o cognitivismo. Os comportamentalistas (behavioristas) vêem o professor como o manipulador do ambiente que é vivenciado pelo aprendiz, enquanto que os da linha cognitiva vêem o professor como um auxiliar na construção de uma representação funcional e significativa.

Os softwares educacionais evoluíram de simples repetições e prática ou tutoriais altamente dirigidos para sistemas abertos que pretendem incorporar inteligência ou até mesmo a linguagem natural, para tentar um diálogo com o usuário. Estes avanços foram feitos em várias áreas; no entanto, na área de Matemática e Matemática Aplicada os resultados obtidos são essencialmente a eliminação do trabalho "braçal". Existe ainda uma necessidade muito grande de softwares instrucionais.

Os computadores podem cobrir esta nova demanda de flexibilidade, tendo um papel de destaque como meio de ajuda neste novo processo educativo. Ao contrário de outras mídias utilizadas na educação, o computador tem a característica de processar e manipular a informação que recebe. Ele pode, a partir de uma informação fornecida, transformá-la, traduzi-la, usá-la em cálculos, ordená-la, arrumá-la e mesmo fazer inferências [KOZ87]. A flexibilidade pode ser exercida no tempo, no espaço e no conteúdo. Flexibilidade temporal significa que o aluno não está mais restrito ao horário da aula, podendo estipular seus próprios horários de acordo com suas conveniências. Já a flexibilidade no espaço significa que o aluno decide onde estudar. Se na sala de aula, no laboratório, na biblioteca ou em casa. Os conteúdos são flexibilizados pois eles serão apresentados no ritmo de cada usuário, que decidirá a quantidade que será acessada de cada vez e com que profundidade cada tópico será abordado. Todas estas alternativas significam que o ensino poderá ser adaptado a cada um, levando ao ensino personalizado, de forma a atender às diferenças individuais.



3. HIPERMÍDIA

3.1. Introdução

O texto é uma das formas mais populares de armazenar conhecimento e transmitir idéias. Entretanto, em uso normal ele é um meio de comunicação estritamente linear (ou seqüencial). Quando o texto é organizado e processado em uma forma não linear ele é denominado de hipertexto. Um sistema de hipertexto é constituído de fragmentos de texto interligados. O termo hipertexto está relacionado ao termo "hiperespaço", popularizado no século 19 pelo matemático Felix Klein, que o usou para descrever uma geometria com muitas dimensões [RAD91]. Ted Nelson acreditava que os sistemas de texto deviam refletir os hiperespaços de conceitos implícitos no texto, e cunhou o termo hipertexto em 1965 [NEL87].

O paradigma da hipermídia é essencialmente a generalização do conceito de hipertexto onde, além do texto, outras formas de mídia, como gráficos, imagens estáticas ou animadas, som, vídeo ou até mesmo programas, são adicionados, ou seja, **um sistema de hipermídia é um hipertexto multimídia**. Entretanto, a recíproca nem sempre é verdadeira, isto é, nem todo programa multimídia se utiliza necessariamente de um hipertexto, assim **nem todo sistema multimídia é hipermídia**.

3.2. Histórico

Vannevar Bush é normalmente considerado o criador, ou pelo menos o lançador, da idéia de hipertexto uma vez que propôs um sistema com estas características em 1945. O sistema Memex [BUS45] nunca chegou a ser implementado, o que não impediu que fosse considerado como o primeiro sistema de hipertexto proposto. Bush, na realidade, desenvolveu algumas das idéias do sistema Memex em 1932 e 1933 e finalmente escreveu o rascunho do artigo em 1939. Por vários motivos, o manuscrito só foi publicado em 1945 na revista *Atlantic Monthly* sob o título *As We May Think* [NIE90].

O motivo principal que levou Vannevar Bush a sua proposta para o Memex foi sua preocupação sobre a explosão da informação científica, que tornaria impossível, mesmo para um especialista, seguir o desenvolvimento em uma disciplina. É claro que a situação é bem pior agora, mas em 1945 Bush discutia a necessidade de as pessoas encontrarem a informação de uma forma mais fácil do que em papel.

Após o artigo de 1945, nada mais aconteceu na área de hipertexto por aproximadamente 20 anos. As pessoas se ocuparam em melhorar os computadores ao ponto de tornar viável o seu uso interativo, mas eles eram tão caros que muitas agências financiadoras de projetos consideravam completamente irresponsável a sugestão de que os recursos computacionais fossem desperdiçados em tarefas não numéricas, como o processamento de texto. O termo hipertexto só foi cunhado em 1965, por Ted Nelson, 20 anos após aquele que é considerado como o primeiro sistema de hipertexto.

3.3. Sistemas de hipermídia: uma classificação

Os sistemas de hipermídia podem ser classificados de várias formas. Uma delas é considerar a época, a plataforma e o propósito com que foram criados e desenvolvidos. Sob esta ótica existiriam três gerações.

A primeira geração seria formada por aqueles sistemas cuja plataforma básica era o *mainframe* e cuja interface era apenas textual. Estes sistemas foram criados antes de 1980 e tinham suporte multiusuário. Podem ser incluídos, nesta fase, sistemas como: *Augment/NLS*, *Dynabook*, *FRESS*, *HES*, *Xanadu* e *ZOG*.

A segunda geração pode ser formada por aqueles sistemas essencialmente com as mesmas características dos de primeira geração, mas baseados em estações de trabalho e que já incluíam suporte para gráficos e animações e possuíam interfaces mais elaboradas. Possuíam ainda suporte para um único usuário ou um pequeno grupo de usuários. Podem ser incluídos na segunda geração os seguintes sistemas: *Symbolic Document Examiner*, *Intermedia*, *KMS*, *Notecards*, *Writing Environment*.

Estes sistemas foram seguidos por outros que rodavam em PCs ou Macs, como o *Guide*, *HyperTies*, *HyperCard*, que eram mais limitados em funcionalidade e escopo, mas que, em geral, tinham as mesmas capacidades que os sistemas rodando em estações de trabalho. Entre os sistemas já estabelecidos que podem ser incluídos na terceira geração está a *WWW* e o sistema de ajuda (*Help*) do Windows.

3.3.1. Principais sistemas pioneiros (primeira geração)

A principal característica destes sistemas é que eles foram propostos como mecanismos para armazenar e recuperar toda a capacidade humana de ler e escrever, como mecanismos naturais para refletir a mente, como ambientes para aumentar o conhecimento e para substituir a leitura e a escrita de texto tradicionais. Eles eram dirigidos primeiramente a autoria e desta forma suas ferramentas de navegação eram bastante limitadas. Eles não forneciam qualquer mecanismo para estender o ambiente ou personalizá-lo. Os nodos não eram tipificados e não suportavam mais do que um tipo de mídia. As conexões tinham uma única direção e um único destino. A única estrutura suportada era a hierárquica. Não existiam navegadores gráficos e a interface com o usuário era baseada em simples monitores de texto. A tabela 3.1 apresenta um resumo destes sistemas. Uma pequena descrição de cada um é encontrada no glossário.

Tabela 3.1 - Sistemas de hipertexto/hipermídia de primeira geração

Ano	Descrição	Autor
1945	<i>Memex (Memory extender)</i>	Vannevar Bush
1962/76	<i>Augment/NLS</i>	Douglas Engelbart
1965	<i>Xanadu</i>	Ted Nelson
1967	<i>HES (Hypertext Editing System)</i>	Andries van Dam
1968	<i>FRESS (File Retrieval and Editing System)</i>	Andries van Dam
1970	<i>Dynabook</i>	Alan Kay
1972	<i>ZOG</i>	Donald McCracken, Robert Akscyn

3.3.2. A segunda geração

A segunda geração de sistemas foi desenvolvida nos anos 80 e caracterizava-se principalmente por ter como plataforma de hardware as estações de trabalho ao invés dos *mainframes*. Alguns sistemas posteriores já rodavam em PCs ou Macs. Estes sistemas não eram sensivelmente diferentes dos de primeira geração e caracterizavam-se por apresentar interfaces mais elaboradas e por dar suporte a um único usuário ou pequenos grupos de usuários. A tabela 3.2 apresenta os principais sistemas e um pequeno resumo poderá ser visto no glossário.

Tabela 3.2 - Sistemas de hipertexto/hipermídia de segunda geração

Ano	Descrição	Autor
1978	Aspen Movie Map	Andrew Lippman
1983	KMS	Donald McCracken, Robert Akscyn
1983	HyperTies	Ben Shneiderman
1985	NoteCards	Xerox PARC
1985	Symbolics Document Examiner	Janet Walker
1985	Intermedia	Andries van Dam
1986	Textnet	Randy Trigg
1986	Guide	Peter Brown
1986	Writing Environment	Universidade da Carolina do Norte.
1987	HyperCard	Brian Atkinson

3.3.3. A idade adulta (terceira geração)

Vários sistemas de hipertexto foram anunciados em 1985 e depois desta época o seu uso se alastrou. Em 1986 a OWL (*Office Workstations Limited*) lançou o Guide, como um produto comercial. Ele foi o primeiro hipertexto disponível comercialmente a rodar em computadores pessoais. Foi primeiramente lançado para Macintosh e mais tarde apareceram as versões para o IBM PC. De alguma forma pode-se dizer que este produto marcou a transição do hipertexto como um produto exótico e apenas de pesquisa para o conceito de uma técnica computacional de uso real em aplicações comerciais [NIE95].

A etapa final para a verdadeira popularização surgiu em 1987 quando a empresa Apple (fabricante do computador Macintosh) introduziu o HyperCard, com uma estratégia de marketing inovadora de fornecer o produto "gratuitamente" com cada computador da empresa vendido após esta época.

Um acontecimento, da época, que marcou a transição do hipertexto como objeto de estimação de uns poucos dedicados para a popularidade foi a realização da primeira conferência ACM em hipertexto (*Hypertext87*) realizada na universidade da Carolina do Norte de 13 a 15 de novembro de 1987. Em 1989 apareceu o primeiro periódico dedicado ao hipertexto o *Hypermedia*. A tabela 3.3 apresenta alguns destes sistemas. Um pequeno resumo de cada um poderá ser visto no glossário.

Tabela 3.3 - Sistemas de hipertexto/hipermídia de terceira geração

Descrição	Referências
ExperText	RAD92 e JON94
StratchTutor	
Modelo Canto de hipertexto	NIC93
Modelo de Hipermídia Harmonia	
WWW	Tim Berners-Lee
Ajuda do Windows	Microsoft

3.3.4. Padronizações (propostas) – modelos e estruturas

Tanto o processamento de texto quanto a editoração tiveram um impacto muito grande na prática de escrever e, da mesma forma que o hipertexto, elas não existiriam sem o computador. O que as separa do hipertexto, no entanto, é que ambas objetivam a produção de documentos impressos, que não é finalidade do hipertexto. Não existe uma cópia impressa de um hiperdocumento inteiro. Uma impressão de todos os nodos não faria sentido, pois não existiria nem a estrutura provisória criada pelo leitor ao utilizar do hipertexto. Da mesma forma, uma impressão do material visto durante uma sessão, em uma dada seqüência, não seria uma cópia do hipertexto [SLA91]. Pode-se dizer que o objetivo principal do hipertexto, além, é claro, de conectar o conhecimento, é o de permitir que o texto seja lido no próprio computador, dispensando, desta forma, a versão impressa.

As tentativas de estabelecer padrões para o hipertexto, geralmente, propõem os seguintes objetivos [DEV91]:

- ⇨ embutir suporte, para o maior número possível de padrões de fontes de texto, de formatos gráficos e de formatos de som e vídeo;
- ⇨ permitir a importação de objetos das aplicações mais conhecidas;
- ⇨ tornar viável o uso de recursos padronizados, tais como processadores de texto e pacotes gráficos, para modificar os hiperdocumentos e
- ⇨ permitir que os hiperdocumentos sejam compartilhados entre diferentes plataformas.

O problema com estes objetivos é que não existem formas específicas para alcançá-los. Os sistemas de hipertexto não foram ainda totalmente aceitos como recursos básicos para o aumento do conhecimento porque muitos deles são pacotes isolados e monolíticos. A grande maioria são sistemas fechados, isto é, o material criado em um hipertexto não pode ser transferido ou integrado em outro, devido aos formatos proprietários dos documentos e dos mecanismos de armazenagem. Programas de conversão são também difíceis de serem escritos, pois estes formatos não são tornados públicos pelas organizações [FOU90]. Para tornar estes sistemas abertos e também integrar as suas funcionalidades, os pesquisadores vêm propondo vários modelos e padronizações para o hipertexto. A tabela 3.4 apresenta alguns destes sistemas com um pequeno resumo de cada um sendo apresentado no glossário.

Tabela 3.4 - Padronizações propostas para os sistemas de hipertexto

Sistema proposto	Referências
HAM	CAG88
HDM	GAR93
Modelo Dexter	HAL90
Modelo Trellis	STO89
Modelo Formal	LAN90
Estrutura Geral de Hipertexto	RAO90

A WWW resolveu boa parte destes problemas. Quanto aos demais modelos, as restrições acima continuam ainda válidas. Este é um dos motivos para que o modelo que está sendo proposto, neste trabalho, utilize como plataforma o modelo de hipertexto da WWW. Os documentos que são apresentados no sistema podem lidos por qualquer outro navegador, praticamente sem nenhuma modificação.

3.4. Hipermídia: caracterização

Os sistemas de hipermídia podem ser caracterizados por alguns componentes básicos. Isto não quer dizer que, necessariamente, cada sistema deste tipo possua cada uma destas características.

Existem 4 componentes básicos no hipertexto: **elementos de informação, abstrações, âncoras e ligações**. Os elementos de informação são os objetos criados e manipulados pelo usuário. Eles incluem: pensamentos, diagramas, desenhos, idéias, discussões, planos, argumentos, algoritmos, etc. Suas manifestações físicas podem ser: textos, imagens, gráficos, sons, animações, processos, etc. Uma abstração é um objeto que permite que os elementos de informação e outras abstrações sejam estruturadas, agrupadas ou relacionadas, podendo ou não ser possível que um usuário as manipule diretamente. Alguns exemplos de abstrações nos hipertextos existentes incluem: cartões (*cards*) e caixa de arquivos (*fileboxes*) no NoteCards, quadros (*frames*) e conjuntos de quadros (*framesets*) no KMS, artigos e enciclopédias no HyperTies e documentos e artigos no Augment/NLS. O nodo é um termo genérico para uma abstração. Uma âncora representa elementos de informação ou abstrações que podem ser a fonte ou o destino de uma ligação. Uma ligação é uma conexão entre duas âncoras [LEG90].

3.4.1. Nodos, âncoras e ligações

A terminologia empregada na bibliografia é variada. A nomenclatura mais citada inclui: nodo ou nó; âncora ou botão e ligação (*link*), referência ou enlace. Cada nodo, expressão física de uma abstração, está associado a outro ou outros através das ligações ou conexões. O número de ligações pode variar dependendo do conteúdo de cada nodo. As ligações são geralmente direcionadas, possuindo uma origem e um destino. Enquanto as ligações vinculam pares de nós, formando a estrutura do hipertexto, as âncoras executam procedimentos, alguns dos quais fornecem a capacidade de exploração de nodos.

Nodo (*node*) ou unidade básica de armazenagem de informação ou ainda **quadro** (*frame*). É talvez o mais comum de todos os elementos que compõe um sistema de hipertexto. Um **nodo** consiste de fragmentos de texto, gráfico, vídeo ou outra informação e pode ser visto como uma transação com o leitor. O tamanho de um nodo, sua **granulosidade**, varia a partir de uma simples figura ou umas poucas palavras até incluir o tamanho de um documento completo.

Com base nos nodos pode-se fazer distinção entre sistemas baseados em quadros ou em janelas. Os quadros utilizam uma quantidade fixa de espaço na tela do computador, não importando quanta informação ele contenha. Exemplos típicos são os quadros do KMS e os cartões do HyperCard. A vantagem do quadro é que a navegação pode ser feita usando qualquer tipo de mecanismo de hipertexto que seja fornecido pelo sistema. Uma desvantagem do quadro é que, normalmente, se torna necessário particionar a informação em vários quadros quando ela não se adaptar em apenas um. Em contraste, os sistemas baseados em janelas requerem que o usuário use o mecanismo de rolagem além do mecanismo do hipertexto.

Ligações (*links*). A conexão entre dois nodos de informação é feita pela **ligação**. As ligações nos sistemas de hipermídia definem uma relação associativa entre o par de nodos que conectam e são elas que permitem a navegação através da base de conhecimento do sistema. Elas também fornecem informação organizacional pela declaração explícita da natureza da relação entre os nodos interconectados. Muitos sistemas também permitem que o usuário modifique, elimine ou crie novas ligações. O destino de uma ligação pode ser: uma linha de texto, um segmento de texto, outro documento, uma figura, um vídeo ou animação, um programa executável, um som ou uma referência bibliográfica.

Âncora (*button*). O ponto inicial de uma ligação é denominado de âncora. Uma âncora constitui uma região sensível ao mouse e é destacada de alguma forma na tela do computador. Geralmente o destaque é feito através da alteração da cor da fonte ou pelo uso do sublinhado e também pela mudança do ponteiro do *mouse* quando passa sobre a região. A âncora pode ser formada por uma palavra, grupo de palavras, uma área (em um gráfico ou diagrama), um ícone ou um rótulo sobre uma região de uma imagem.

As ligações no seu ponto de partida estão, quase sempre, ancoradas; a âncora fornece ao usuário algum objeto explícito para ativar de forma a seguir a ligação. Muitas vezes estes pontos de ancoragem tomam a forma de menus embutidos, onde parte do primeiro texto ou gráfico tem dupla função de ser informação por si só e ser a âncora da ligação. O resultado de ativação de uma âncora é seguir a ligação até seu nodo destino.

3.4.2. Estrutura organizacional

Os nodos são interconectados através de vários tipos de estruturas. A estrutura do nodo e a estrutura de ligação formam uma rede de idéias. Uma rede é um grupo ou sistema de idéias interligados ou interconectados. A estrutura organizacional de qualquer sistema de hipermídia determina o modelo de informação da base de conhecimentos. Este por sua vez facilita a localização e a recuperação dos conteúdos. As estruturas usadas pelos sistemas de hipertexto são a hierárquica, de rede, linear, tabela indexada, de regras e de base de dados [JON90].

● **Hierárquica.** É o tipo de estrutura mais comum. Ela possibilita a exploração de nodos através de vários níveis ou categorias, favorecendo a visibilidade e facilitando a identificação de nodos aninhados. O nó inicial pode ser usado para especificar a idéia geral e os de níveis inferiores na hierarquia para explicitar ou particularizar o conceito.

A implementação de uma estrutura hierárquica pode ser feita através de tabelas, caixas embutidas, diagramas de colchetes, lista dentada, árvores "de cima-para baixo" (*top-down*) e árvores esquerda-direita (*left-to-right*). Dificilmente um hiperdocumento será constituído apenas de ligações hierárquicas. Pode ser necessário a conexão entre níveis da hierarquia tornando a sua estrutura mista.

- **Rede.** É uma estrutura onde um nodo está ligado com vários outros num mesmo nível ou em níveis acima ou abaixo. Quando um nodo é referenciado em mais de um local, seu conteúdo não é duplicado, pois existe uma ligação que leva a ele. Este tipo de organização permite que se alcance um nodo a partir de qualquer outro, pois toda a estrutura se encontra relacionada. Uma estrutura deste tipo pode também ser hierarquizada através de conexões cruzadas. No entanto, a representação hierárquica não será única.

- **Linear.** É o tipo mais simples de estrutura, mas isto não significa que seja fácil estruturar um conteúdo desta forma. Na estrutura linear os nodos estão referenciados dentro de um mesmo plano, em uma organização linear única, possibilitando um acesso de forma seqüencial.

- **Tabela indexada.** Neste tipo de estrutura os nodos são apresentados através de colunas de uma tabela relacional. Cada nodo pode aparecer em mais de uma tabela e estas estão relacionadas a uma base de dados com poder de álgebra relacional. A recuperação da informação será feita através de termos indexados.

- **De regras.** Em uma estrutura de regras, os nodos são conectados através de regras que devem ser satisfeitas. Desta forma, um nodo A será conectado a todo nodo que satisfazer determinada condição, por exemplo possuir um determinado texto ou palavra, figura, vídeo, etc.

3.4.3. Outros elementos ou características

Interatividade (controle dinâmico). A hipermídia permite que o usuário determine a seqüência em que acessa a informação (*browsing*), adicione ou emende a informação de forma a torná-la pessoalmente mais significativa (colaboração) ou construa e estructure sua própria base de conhecimentos. O nível de controle do usuário varia com o sistema e com o seu objetivo. O importante é que o usuário está interagindo e controlando dinamicamente a informação de uma forma significativa. Ele pode acelerar ou desacelerar, mudar de direção ou expandir seus horizontes de informação. Diferente de muitos outros sistemas de informação, os usuários dos sistemas de hipermídia devem estar mentalmente ativos enquanto interagem com a informação. A hipermídia permite um alto nível de controle dinâmico pelo usuário.

Caminhos. Os caminhos recebem, na literatura, diversas denominações, entre elas: rotas (*tours*), rotas orientadas (*guided tours*) e roteiros (*scripts*). Os caminhos podem ser determinados pelo autor, pelo usuário/leitor ou por ambos. Os caminhos determinados pelo autor são aquelas ligações pré-fixadas, dentro do espaço de informação (rotas dirigidas), através da base de conhecimentos. Os usuários ou leitores podem também criar seus próprios caminhos. Alguns sistemas permitem que o leitor grave em disco estes caminhos para uma revisão posterior ou anotação.

Anotação. Alguns sistemas permitem que o usuário faça anotações enquanto visita os nodos. Isto pode ser feito através de um tipo de ligação especial denominada de **ligação de anotação**. Esta ligação leva a um nodo que seria o equivalente a uma nota de rodapé de um livro. Ela pode ser implementada na forma de uma janela automática que seria fechada tão logo o botão do mouse fosse liberado [NIE95]. Uma outra forma de anotação, que dispensa conexões, é permitir que o usuário destaque partes do texto atra-

vés do uso de brilho ou cores e o sistema permita registrar e posteriormente visitar apenas os nodos em que foram feitas tais anotações.

Ambiente autoral. Os sistemas de hipermídia são tipicamente ferramentas flexíveis e adaptativas para autoria e navegação. Alguns sistemas de hipermídia tornaram-se populares e são comumente usados para criarem instrução com base no computador, isto é, eles oferecem um ambiente com recursos para que o autor (ou autores) do hipertexto possa criar o conteúdo da aplicação, assim além de desempenhar o papel de navegador e/ou base de dados alguns sistemas também são ambientes de autoria.

Navegador (browser). É um componente do sistema hipertexto que recupera a informação contida nos nodos. No início, os sistemas geralmente incorporavam tanto a base de dados quanto a ferramenta de navegação. Hoje é mais comum a construção de navegadores e bases de dados independentes, a exemplo do modelo WWW.

Deve-se salientar que estes componentes são uma síntese dos muitos sistemas de hipertextos até hoje construídos. Como cada um destes sistemas foi construído com um propósito específico, existem recursos de todos os tipos e dificilmente um sistema atual englobaria todas estas características, mesmo porque os recursos são oferecidos em função dos objetivos de projeto do sistema de hipermídia.

3.5. Aplicações dos sistemas hipertexto/hipermídia

Os sistemas de hipertexto/hipermídia são aplicados às mais diferentes áreas do conhecimento. Contudo, nem todas as aplicações devem ser feitas em hipertexto. Para determinar se uma aplicação é apropriada para hipertexto, Ben Shneiderman, propôs o que ele denominou de as três regras de ouro do hipertexto:

- ⇒ Existe grande quantidade de informação organizada em muitos fragmentos;
- ⇒ Os fragmentos se relacionam entre si;
- ⇒ O usuário precisa de uma pequena quantidade em um determinado tempo.

Uma vez que o hipertexto é uma ferramenta computacional, então uma quarta regra do hipertexto poderia ser: não utilize hipertexto se a aplicação não exige que o usuário esteja na frente de um computador. Aliás, esta regra poderia ser revisada, uma vez que as três regras acima foram claramente orientadas a aplicações que envolviam apenas texto. No entanto, com a generalização do hipertexto para a hipermídia o leque de aplicações que podem tirar proveito do hipertexto cresceu substancialmente. Assim, situações que envolvam muitos gráficos, diagramas ou simulações (como as disciplinas da área Matemática e a Estatística) são apropriadas para a apresentação através de um sistema hipermídia.

3.6. Arquitetura do hipertexto/hipermídia

Várias são as arquiteturas encontradas nos sistemas de hipertexto/hipermídia. Estabelecer uma classificação não é uma tarefa fácil, pois a variedade de objetivos e a diversidade de plataformas envolvidas é muito grande. Um sistema de hipertexto/hipermídia possui dois tipos de estrutura. A estrutura lógica imposta pelo autor, que inclui as relações entre as ligações (*links*) e a base de dados (que pode ser em árvore, rede, grafo, etc.) e a relação das ligações aos nodos e o conteúdo e projeto dos nodos individuais e rede de nodos. Os autores normalmente tem muito menos liberdade na definição da estrutura física do hipertexto. O layout, a interface, os mecanismos de entrada e outras considerações estéticas são sempre fortemente influenciadas pela plataforma onde o hipertexto será utilizado [SHN89].

Teoricamente, podem-se distinguir três níveis em um sistema de hipertexto/hipermídia:

- **Nível de apresentação.** Caracterizado pela interface com o usuário, que é um ambiente interativo baseado em janelas ou quadros.
- **Nível da Máquina Abstrata do Hipertexto (MAH).** Caracterizado por um mecanismo que gerencia toda a informação sobre o hipertexto, é constituído pelos nodos e pelas conexões.
- **Nível da base de dados.** Caracterizado pela forma de armazenagem dos nodos, dos dados compartilhados e pelo acesso à rede.

Na realidade os sistemas atuais de hipertexto não seguem esta estrutura interna, eles são uma mistura mais ou menos confusa de características. Mesmo assim este modelo mostra uma direção importante para uma padronização futura [NIE95].

3.6.1. O nível de apresentação (interface com o usuário)

A interface com usuário lida com a apresentação da informação contida na base de dados e inclui questões como: que comandos devem ficar disponíveis para o usuário, como apresentar nodos e ligações e se serão incluídos diagramas globais ou não. Vamos supor que o nível MAH (Máquina Abstrata de Hipertexto) defina as ligações sendo feitas. O nível de interface com o usuário pode decidir não apresentar a informação a todos os novos usuários e permitir que a informação seja digitada somente num nodo de autoria.

3.6.2. O Nível da Máquina Abstrata de Hipertexto (MAH)

A MAH fica no meio do sanduíche entre a base de dados e a interface do usuário. É onde o sistema de hipertexto determina a natureza básica dos nodos e ligações e onde ele mantém as ligações entre eles. A MAH deve ter conhecimento da forma dos nodos e ligações e deve saber que atributo está relacionado com qual. Por exemplo, um nodo, pode ter um atributo próprio, que informe ao usuário quem o criou e quem tem autorização para modificá-lo. As ligações podem ser digitadas como no NoteCards ou podem ser simples ponteiros como no Guide. A MAH é a melhor candidata para padronização, uma vez que a base de dados é altamente dependente da plataforma computacional e a interface com o usuário é normalmente diferente de um sistema para outro, pois depende do sistema operacional sendo utilizado. Isto deixa somente a MAH como totalmente independente das plataformas computacionais e, como se quer a habilidade de poder transferir a informação de um sistema para outro, é necessário padronizar um formato de intercâmbio a este nível.

3.6.3. O nível da base de dados

A base de dados na realidade não tem nada a ver com o hipertexto. É necessário armazenar grandes quantidades de informações em vários mecanismos de armazenagem, como discos rígidos, CDs, etc., e alguma informação pode ser armazenada em um servidor remoto que deve ser acessado através de uma rede. Não importa como e onde a informação esteja armazenada, deve ser possível recuperá-la em partes especificadas em um tempo muito curto. Isto soa como a especificação de uma base de dados, o que de fato é. Além disso, o nível da base de dados deve apresentar algumas características

próprias tais como: permitir o acesso compartilhado (multiusuário), requerer senhas de acesso para proteger as informações e construir cópias de segurança.

3.7. O hipertexto versus a competição

O hipertexto pode ser comparado com outras técnicas tradicionais de apresentação de informações. Abaixo são relatados alguns experimentos conduzidos, comparando o hipertexto com os sistemas especialistas e com os textos tradicionais.

3.7.1. O hipertexto versus os sistemas especialistas

Alguns hipertextos são muito semelhantes a sistema especialistas. Peper et al [PEP89] compararam um sistema de hipertexto interno da IBM com um sistema especialista comercial para a representação de informações necessárias para diagnosticar problemas em uma rede mundial de computadores. A mesma informação foi representada nas duas formas e testada com um conjunto amostra de problemas para serem resolvidos por 12 usuários, que estavam envolvidos na determinação de problemas com rede de computadores ou tinham tais responsabilidades anteriormente em suas carreiras. Os usuários testados resolveram 81% dos problemas corretamente quando utilizaram o sistema de hipertexto comparado com somente 67% quando utilizaram o sistema especialista. A utilização do sistema especialista, no entanto, foi mais rápida, com quatro minutos por problema, enquanto que, com o hipertexto foram gastos cinco minutos por problema. Quando perguntados qual sistema seria preferível para uso no trabalho, 50% escolheram o sistema de hipertexto e 25% o sistema especialista. Do ponto de vista da autoria, Peper et al afirmam que é uma tarefa simples atualizar a informação em um sistema de hipertexto enquanto que é difícil manter a base de conhecimento de um sistema especialista.

3.7.2. O hipertexto versus texto o eletrônico

Talvez a comparação mais importante é entre a implementação de um hipertexto completo de um conjunto de informação com a mesma informação na forma de um livro ou um artigo. Shneiderman [SHN89] comparou duas versões de um conjunto de artigos históricos com mais ou menos 138 páginas na versão impressa. A versão eletrônica foi implementada no Hyperties. Quando foram respondidas questões para os quais as respostas podiam ser encontradas no início de um artigo, o Hyperties teve um desempenho inferior ao da versão impressa (42 segundos versus 22 segundos), mostrou-se um pouco melhor quando a resposta estava no corpo do artigo (58 segundos versus 51 segundos) e apresentou o mesmo desempenho quando a resposta devia ser encontrada por um combinação de fatos (107 segundos nas duas situações). Estes dados parecem indicar que o hipertexto é de alguma ajuda quando a resposta tem de ser encontrada saltando através da informação e que o hipertexto aumenta o tempo de procura quando a informação pode ser encontrada de imediato em uma página.

Egan et al [EGA89; EGR89] conduziram testes da utilização de um livro de estatística, em uma versão de hipertexto no sistema SuperBook da sua tradicional versão impressa. O livro foi originalmente escrito com a publicação impressa em mente. Então, é possível projetar uma versão especificamente para hipertexto que pode ser melhor do que a que foi a testada por eles. Os usuários foram solicitados a encontrar respostas para certas questões sobre Estatística que poderiam ser respondidas do livro e os seus tempos foram medidos. Quando as palavras chaves das questões eram palavras dos títulos do li-

vro ,os usuários do livro levaram vantagem (3,5 minutos versus 4,4 minutos). A situação, no entanto, foi o inverso quando as questões utilizaram palavras que não ocorriam nos títulos. Neste caso, a versão *online* foi mais rápida (4,3 minutos versus 7,5 minutos). Foi verificado, também, a correção das questões e concluído que a versão em hipertexto apresentou um desempenho melhor do que a impressa.

Uma segunda forma de avaliação foi feita através da escrita de um ensaio que foi corrigido por um avaliador competente que não foi informado sobre qual versão ele foi baseado (se a impressa ou a computacional). Os graus foram significativamente melhores para os usuários que utilizaram a versão em hipertexto (5,8 versus 3,6 em uma escala de sete pontos).

3.7.3. O hipertexto versus o texto tradicional

A maneira mais simples de caracterizar o hipertexto é contrastá-lo com um livro. Todo texto tradicional, não importa se na forma impressa ou computacional, é sequencial, isto é, existe uma única sequência em que o texto pode ser lido. Primeiro é lida a página um, então a dois e assim sucessivamente.

Não existe uma evidência clara de que o hipertexto é de fato superior ao papel. Para algumas aplicações ele se mostrou melhor, enquanto que para outras o papel foi melhor do que a atual geração de hipertextos. Mas para muitas aplicações será necessário esperar pela melhora da tecnologia e por melhores projetos de interface para se ter esperança de se superar o texto impresso. Além disso, não se tem ainda experiência real com hipertextos realmente grandes. Quase todos os hipertextos que foram testados em experimentos formais de utilização eram pequenos e continham aproximadamente um milhar de nodos ou menos [EGR89].

A afirmação acima foi feita no final da década de 80, agora, no entanto, com a Internet pode-se ter uma idéia do que é um hipertexto realmente grande. Na realidade, a Internet é uma base de dados tão grande que é difícil até mesmo medi-la. E a despeito do seu tamanho pode-se dizer que ela já foi suficientemente testada e aprovada pelo sucesso alcançado.

Vantagens do hipertexto quando comparado com a versão em papel

- ⇨ Pode mostrar imagens, animações e filmes;
- ⇨ É facilmente atualizável, permitindo alterações automaticamente;
- ⇨ Pode ser enviado através de uma rede;
- ⇨ Ocupa muito pouco espaço de armazenagem;
- ⇨ Pode ser compartilhado por muitas pessoas;
- ⇨ A leitura é orientada pelo usuário;
- ⇨ Potencialidade: todo o conhecimento disponível a um clique do mouse.

Desvantagens do hipertexto quando comparado com a versão em papel.

- ⇨ A velocidade de leitura é 30% menor nos monitores atuais.
- ⇨ Necessidade de se ter acesso a um computador e ao software e saber usá-los.
- ⇨ Sem romantismo: primeira edição, capa dura, etc.

3.8. Conclusão

Após sua idealização em 1945, o hipertexto enfrentou um período em que nada foi feito, principalmente em virtude da ausência de base para fazê-lo. Idealizado como um equipamento mecânico para a manipulação de microfichas, sua efetiva implantação começou a acontecer quando migrou para o computador, com o seu desenvolvimento ocorrendo paralelo à indústria da computação. A fase de maior crescimento ocorreu com a invenção do microcomputador, particularmente após ter sido popularizado através da linha Apple Macintosh, que oferecia o HyperCard junto com a máquina adquirida. Em resumo, pode-se dizer que o hipertexto foi idealizado em 1945, nasceu em 1960, cresceu lentamente nos anos 70 e finalmente se desenvolveu nos anos 80, com um crescimento particularmente rápido após 1985, tendo se firmado definitivamente durante o ano de 1989.

O texto convencional tem um único ponto de partida, que é o parágrafo de abertura. O hiperdocumento oferece múltiplos pontos de entrada. O leitor escolhe por onde entrar no sistema. Da mesma forma o hiperdocumento oferece não uma única saída, mas um bom número de pontos de saída e novamente fica à escolha do leitor onde e quando deixar o sistema. A característica mais radical de todas é que o hiperdocumentos fornece muitos caminhos através do seu conteúdo e novamente o leitor escolhe que caminhos quer seguir e quais podem ser deixados para outra visita. O texto seqüencial é um recurso extremamente poderoso, mas exige que o leitor se submeta a sua estrutura.

A multimídia cria novas oportunidades para a apresentação de documentos, pois aqueles que contém uma combinação de gráficos, áudio, vídeo e animação podem expressar melhor e mais claramente certas idéias e conceitos dos que aqueles que contém somente texto. Enquanto alguns pesquisadores acreditam que o hipertexto pode ser padronizado, existem outros que argumentam que a noção ainda não está sólida o suficiente. Existe ainda um terceiro grupo que acredita que apenas as características atuais e bem estabelecidas do hipertexto podem ser padronizadas. Acredito que para se falar em padronização é necessário primeiramente especificar o tipo de modelo desejado. Se é mono ou multiusuário. Se é para um computador só ou para rede. Provavelmente nunca se chegue a um padrão único. Afinal, por que isto deveria acontecer com o hipertexto em particular, se não aconteceu nem com o hardware apesar dos muitos esforços feitos? Pode-se ter, como o terceiro grupo acima acredita, características padrões de maneira que ocorre hoje com outros sistemas computacionais, como os planilhas e processadores de texto, por exemplo.

No entanto, como já aconteceu em outras áreas, a padronização de fato acaba ocorrendo quando um sistema ou tecnologia se torna popular a ponto de sufocar ou inviabilizar outras concorrentes que ocuparão apenas fatias pequenas do mercado. Com o hipertexto parece estar acontecendo algo deste tipo. O sistema da WWW, aliado com a Internet, está se tornando o padrão de fato de hipertexto e troca de informações. Em vista disto, acredito que será muito mais prático tentar fazer uso eficiente deste sistema, já que ele tem se mostrado como o que o usuário de fato prefere, sem, é claro, significar com isto que ele seja o melhor ou mesmo o mais eficiente.



4. A HIPERMÍDIA NO ENSINO

4.1. Introdução

Bush, Engelbart, e Nelson estão entre os pioneiros a propor sistemas que aumentam o aprendizado através do manuseio de informações armazenadas no que hoje é denominado de hipertexto. Eles tinham uma forte sensação de que as conexões no hipertexto eram idéias relacionadas e que associações podiam ser fácil e arbitrariamente estabelecidas, que a informação podia ser personalizada, livremente anotada, vista e prontamente acessível. Eles propuseram sistemas que ofereciam uma abordagem de manipulação direta da informação e dependiam fortemente da visualização, raciocínio espacial e pensamento associativo. O conteúdo revolucionário de suas idéias foi e continua sendo a profundidade com que estes sistemas engajam o usuário como um participante ativo nas interações com a informação.

Nas últimas décadas os sistemas de hipertexto tornaram-se tecnologicamente viáveis para a comunidade educacional. A partir de 1991, com a criação da WWW, o sistema se popularizou de tal modo que não é exagero afirmar que seja, hoje, talvez o paradigma computacional mais utilizado no mundo. A disponibilidade de hardware poderoso a baixo custo e as redes locais tornaram o uso do hipertexto não só viável, mas também atraente para o ensino dentro da comunidade universitária, onde o uso da Internet se alastrou de tal forma que parece inconcebível que a apenas alguns anos ela não existisse.

Dentro da comunidade educacional, o interesse pelo hipertexto cresceu rapidamente a partir do ano de 1987 (ano da primeira conferência sobre o tema). Isto pode ser verificado pela literatura sobre o tema e pela oferta de softwares de autoria para hipermídia lançados e pelo número de navegadores para a WWW. Isto aconteceu porque foi percebido que o hipertexto é uma ferramenta versátil que pode adaptar-se a diferentes usos educacionais. Ele pode ser tratado como CAI, atuar como ferramenta de ensino aprendizagem e ser utilizado, ainda, para que o aluno efetue programação. Dentro dos recursos computacionais existentes atualmente ele parece ser o mais completo para apoiar o processo de ensino aprendizagem [SME91].

Um dos principais fatores para a popularização da hipermídia é que as ligações nos documentos despertam novos interesses e atratividade ao que poderia ser apenas um conjunto de páginas de um assunto sem interesse. O que é novo e atraente, principalmente para a construção de sistemas de ensino/treinamento, é a possibilidade de misturar texto, áudio, vídeo e programas com o poder de processamento de um computador.

A hipermídia junta as mídias: texto (palavras e números), áudio (música, voz, discurso), vídeo (imagens, fixas e animadas, televisão), gráficos (diagramas fixos ou dinâmicos) e programas. A hipermídia pode gerar produtos como jogos, simulações, tutores e também sistemas de treinamento e ensino. Um sistema deste tipo teria as seguintes vantagens sobre sistemas tradicionais de ensino e transmissão de informações:

- Seleção de informações por associação, e não por indexação, que é bastante artificial. A idéia é associar as vantagens de um livro, documento ou norma ao poder de

processamento do computador para calcular, traçar gráficos, otimizar e imprimir resultados, coisas que nenhum livro faz.

- Poder interagir, voltar atrás, pular e recomeçar.
- Utilizar a simulação para ilustrar conteúdos, repetindo exemplos com novos dados, para fazer projeções, ilustrar tendências ou mostrar alternativas.
- Ensino praticamente individualizado, pois o aluno pode acessar qualquer informação através de critérios pessoais como: relevância, interesse, curiosidade, experiência anterior e objetivos.
- O usuário poder repetir os exemplos quantas vezes quiser ou achar necessário e sempre com valores diferentes. Por exemplo, suponha-se que se queira ilustrar uma distribuição de probabilidades. Num editor de texto não é possível e num livro teríamos uma ou duas ilustrações. Mas o que se quer é uma interação dinâmica. O usuário do sistema lê ou ouve os conceitos e pode interagir com o sistema para produzir quantos diagramas (curvas) quiser. O sistema ativa então um programa que se encarrega de desenhar a distribuição, podendo calcular seus momentos, redesenhá-la com novos parâmetros ao comando do usuário e ao final imprimir os resultados.

Um aspecto que se pode enfatizar é que estes sistemas não apenas fornecem um ambiente para armazenar grande quantidade de informação interconectada, mas oferecem, também, um ambiente de escolha do processo educacional por parte do aluno. Segundo Stanton [STA90] o hipertexto, e a idéia de conhecimento entrelaçado de várias formas, determina uma estrutura que permite a exploração de domínios. Com a intenção de acomodar a informação ao leitor, a estrutura do conteúdo é posta sob controle do aluno que pode especificar que informação acessar, em que ordem e velocidade. Marchionini [MAR88] observa que os sistemas de hipertexto não apenas fornecem uma nova maneira de aprender, mas, também, um novo caminho para aprender a estudar. E acrescenta que estes ambientes levam a uma constante tomada de decisões e uma avaliação contínua por parte do usuário acarretando, desta forma, o desenvolvimento de habilidades cognitivas.

4.2. Os modelos de aprendizagem e o hipertexto

A aprendizagem, dentro do contexto do conhecimento humano, pode ser descrita através de teorias competitivas que podem ser classificadas em racionalismo, empirismo e holismo.

O racionalismo estabelece que a mente, ao invés dos sentidos físicos, é a fonte do conhecimento. Mente e aprendizagem são completamente independentes do senso de percepção e experiência. Pensamento e lógica, ao invés de observação, criam conhecimento.

O empirismo, por outro lado, vê o oposto como verdade: os sentidos são a fonte do conhecimento. Mente e aprendizagem são dependentes dos sentidos e a observação é a base construtiva do conhecimento. A abordagem holística combina o racionalismo e o empirismo, argumentando que a mente e os sentidos interagem para criar o conhecimento.

Métodos de ensino holísticos orientados estruturam a aprendizagem pela ênfase em conexões entre informações, conceitos e princípios. Estratégias baseadas em computador, utilizando modelos cognitivos orientados, podem auxiliar o estudante a explorar seu processo de pensamento, lidar com a ambigüidade e reconhecer padrões subjacentes comuns entre eventos. A hipermídia, do ponto de vista holístico, é útil por permitir ex-

periências de aprendizagem por descoberta e a transferência de conhecimentos e habilidades inter-disciplinar. Diferenças individuais na motivação do aprendiz podem representar um papel mais importante com o hipertexto do que com qualquer outra mídia. Por exemplo, alguns estudantes obtêm mais conhecimentos e habilidades de experiências de descobertas que outros estudantes que tendem a se basear na memorização mecânica. Desta forma, os educadores precisam expandir seu conhecimento de estratégias auto-dirigidas e centradas no aprendiz se quiserem obter o potencial pleno do sistema de hipermissão.

A característica diferenciadora da pesquisa cognitiva é a idéia de que a instrução afeta o aprendizado através do processo do pensamento do estudante; isto influencia o pensamento, que por sua vez, influencia a aprendizagem e o desempenho. A abordagem cognitiva, além disso, assume que a instrução é mediada pelo processo de pensamento do estudante. A mente pode adaptar esquemas conceituais internos existentes para absorver novas informações. Muitos sistemas de hipertexto aceitam a premissa de que os aprendizes constroem significados através de um processo ativo de inquirir, explorar e de gerar e testar hipóteses. A abordagem cognitiva coloca a maior responsabilidade pela aprendizagem no estudante. Este modelo enfatiza que pensar envolve uma transação ativa entre um indivíduo e a informação sob estudo. Fornecendo-se condições apropriadas de aprendizagem, todos os estudantes são capazes de pensamentos analíticos abstratos. Táticas instrucionais apropriadas podem ser projetadas para melhorar, de forma individualizada, as habilidades de pensamento de alta ordem do aprendiz [FRA90].

Estratégias cognitivas são básicas para a aplicação de muitos projetos instrucionais e teorias de aprendizagem. As estratégias podem mostrar como investigar um problema. Elas podem auxiliar como aprender, como lembrar, como executar o pensamento analítico e reflexivo que leva a mais aprendizagem. E também, aprender um modelo subjacente ao processo de raciocínio utilizado, para gerar uma solução para um domínio específico, auxilia a formalizar o conhecimento e a aquisição de conceitos.

Conhecimento conceitual significa mais do que armazenar e recuperar fatos. Envolve o entendimento da estrutura operacional de um conceito e idéias correlacionadas. O hipertexto e a hipermissão são úteis para mostrar o contexto, a estrutura e a associação de idéias. Convém lembrar que na área em que o modelo está sendo proposto este tipo de raciocínio é fundamental. Um conceito emergente da literatura reconhece que muitos estudantes preferem experiências ativas e auto direcionadas. Incentivar esta preferência resulta em tanto uma melhor aquisição quanto uma atitude mais positiva em relação ao conteúdo estudado. Técnicas de hipermissão vão mais além do que simplesmente apresentar informação enciclopédica, encorajando a aprendizagem generativa e a motivação intrínseca. Programas de hipermissão que são cuidadosamente construídos com sensibilidade para itens motivacionais e fatores humanos potencializarão a formação de conceitos, transferência, curiosidade do estudante e a satisfação com as atividades de aprendizagem [FRA90].

4.3. Hipóteses sobre aprendizagem

O que distingue um sistema de aprendizagem hipermissão de um é que o sistema de aprendizagem, normalmente, fornece um leque de ferramentas que estendem consideravelmente as facilidades básicas de navegação. Uma das tentativas mais bem sucedidas para desenvolver o hipertexto nesta direção é fornecida pelo projeto Intermedia da universidade Brown. De fato, neste sistema, mesmo quando os aprendizes simplesmente seguem as conexões oferecidas de uma forma passiva existe uma pequena razão para

supor que a aprendizagem é de alguma forma mais efetiva do que aquela oferecida pelos programas na tradição da instrução programada. O *Intermedia* demonstrou que um uso efetivo e extensivo do hipertexto, com uma interface cuidadosamente planejada para o autor e o leitor, é capaz de fornecer um ambiente favorável para vários tipos de atividades de aprendizagem [MAY90].

Vários modelos de aprendizagem podem ser usados num sistema de hipertexto. Por exemplo o modelo novato/especialista limita a liberdade que um estudante possui na exploração e construção do conhecimento em um ambiente de hipertexto. Jonassen [JON90] defende este modelo e sugere várias maneiras de estruturar o hipertexto de forma a obter resultados convergentes. Ele vê o hipertexto como um método de transferir o conhecimento de um especialista em uma área particular para o aprendiz. Nesta abordagem o aprendiz deve seguir caminhos específicos e as associações projetadas no sistema. Por outro lado Nelson [NEL87] vê o hipertexto como uma forma de liberar o estudante do acesso controlado ao conhecimento e obter resultados de aprendizagem divergentes. Ele acredita que o estudante deve ir além do modelo novato/especialista. Que o estudante precisa de liberdade para criar sua própria visão do conteúdo através da direção de suas próprias experiências em um ambiente de hipertexto aberto.

4.4. O hipertexto e a teoria da subsunção

A teoria da aprendizagem de David Ausubel [AUS78] serve como fundamentação teórica para a estrutura hierárquica do hipertexto como aplicação de ensino aprendizagem, pois mostra a importância deste tipo de estrutura enfatizando que, em um determinado domínio, os conceitos gerais devem ser introduzidos primeiramente sendo seguidos posteriormente pelos conceitos mais específicos.

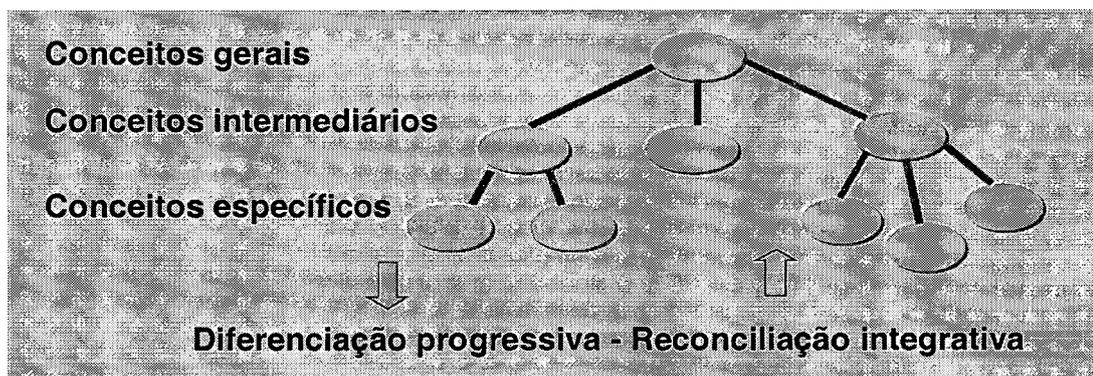


Figura 4.1 - Representação esquemática do modelo de Ausubel

Na teoria da subsunção de Ausubel, o armazenamento de idéias no cérebro é altamente organizado, com relações ocorrendo entre elementos mais antigos e mais recentes, produzindo uma hierarquia conceitual, na qual os elementos mais específicos do conhecimento estão ligados a conceitos mais gerais. O ensino deve ser efetuado programando os assuntos de forma hierárquica, com a estrutura lógica explicitando as relações entre as idéias, ressaltando as similaridades comuns e levando em consideração o conhecimento anterior do aluno. As idéias mais gerais de um conteúdo devem ser apresentadas primeiro e então serem progressivamente diferenciadas em termos de detalhe e especificidade. A figura 4.1 ilustra o modelo da diferenciação progressiva de Ausubel.

4.5. O hipertexto e construtivismo

Existem certas áreas do conhecimento que não podem ser representadas de forma estruturada devido a sua complexidade. A maioria dos recursos computacionais existentes para o desenvolvimento de sistemas educacionais não apresentam flexibilidade suficiente, acarretando em construço de ambientes simplificados, onde apenas alguns aspectos so destacados. Uma ferramenta com a flexibilidade do hipertexto permite a representaço de contedos por uma rede semntica, oferecendo diferentes perspectivas, onde os conceitos so individualizados atravs de atributos. No processo de aprendizagem por descoberta, estas redes podem ser identificadas com a estrutura cognitiva do aluno, permitindo que o aprendiz participe na determinaço dos temas sendo analisados, buscando, investigando, elaborando, organizando e integrando conhecimentos. Jones [JON89] coloca que para domnios no-estruturados, estratgias exploratrias, de busca de informaço, atravs de navegadores so as mais adequadas. Ele acredita que a melhor forma de promover a aprendizagem incidental  pela utilizaço de recursos de hipertexto.

O construtivismo fornece uma fundamentaço terica para o desenvolvimento de sistemas baseados em computador, tais como os hipertextos estruturados em rede. A alegaço principal  a de que a aprendizagem  um processo ativo na qual novas idias ou conceitos so constridos baseados no conhecimento atual ou passado. O aprendiz seleciona e transforma a informaço, constri hipteses e toma decises, baseado em uma estrutura cognitiva para fazer isso.  esta estrutura (modelo mental) que fornece significado e organizaço s experincias e permite que se v alm da informaço dada [MER91].

Para Bruner, o instrutor deve encorajar o estudante a descobrir princpios por ele mesmo, que  exatamente o objetivo do hipertexto. A tarefa do instrutor  apresentar a informaço num formato apropriado para o seu grau de entendimento atual, que no hipertexto  alcançado pela forma com que o contedo  elaborado.

Aprender consiste em reorganizar estruturas de conhecimento. Tais estruturas podem ser modeladas por redes semnticas (idias ligadas por associaçes), que possibilitam que o leitor combine idias, infira e extrapole. O aprendizado ocorre quando novas estruturas so agregadas s existentes atravs de novos nodos. Quanto mais ligaçes puderem ser estabelecidas entre o conhecimento antigo e o novo, melhor ser a integraço da informaço e mais facilmente se dar o aprendizado.

Pessoas diferentes possuem mapas cognitivos diferentes e o hipertexto, com sua flexibilidade, fornece liberdade para que cada um estabeleça ligaçes que mais se adaptem a estes mapas cognitivos, modelando o processo de aprendizagem individual. O controle do sistema hiperfídia  exercido pelo aprendiz, de acordo com suas necessidades, objetivos, interesses e num ritmo prprio e de acordo com seu conhecimento anterior. Ao exercer o controle sobre o material instrucional, o estudante fica livre de presses e ansiedades, ganhando mais motivaço, que  a mola propulsora do conhecimento.

4.6. Propriedades do hipertexto teis  educaço

Yankelovich et al [YAN88] observam, sobre a utilizaço do hipertexto na educaço:

"estudos quantitativos mostram que o uso da hiperfídia habitua o estudante a fazer suas prprias conexes assim como tambm melhora a discusso em

aula, o hábito de leitura e a habilidade geral de manejar grandes quantidades de informações difíceis. Da mesma forma, a experiência nos convenceu que o hipertexto também encoraja algumas formas de trabalho cooperativo. Sistemas educacionais que empregam grandes bases de dados de múltiplos autores têm outras capacidades para mudar a educação fundamentalmente, uma delas reside na habilidade para encorajar o ensino e aprendizagem multidisciplinar. Outro potencial do hipertexto que afeta a mudança educacional reside em sua habilidade para mudar nossa concepção básica sobre o currículo universitário" (p.84).

Landow [LAN90] acrescenta:

"se verdadeiro ou não que os leitores retêm menos informação quando lêem um texto em uma tela de computador do que quando eles lêem numa página impressa, o fato é que texto eletronicamente conectado e texto impresso possuem tipos diferentes de vantagens. Então ao usar esta tecnologia deve-se levar em conta sua principal vantagem que é a conectividade" (p.42).

Além da conectividade, outras características dos hipertextos que os tornam úteis como ferramentas educacionais são: a adaptação, a atualização, a audiovisualização, o dinamismo e a interação, a intercomunicação e a redução.

4.6.1. Adaptação

Um sistema de hipertexto pode ser adaptado, através de filtros a determinados tipos de arquivos, a diferentes usuários sem que seu projeto fique demasiadamente complexo. Neste sentido, pode dizer-se que tais ambientes aceitam mais de um enfoque sobre um mesmo domínio. Os leitores, através de mecanismos fornecidos para este fim, podem criar seus próprios modelos de conhecimento. Um curso pode ser preparado com vários enfoques. Por exemplo, pode ser feita uma diferenciação entre especialistas e novatos ou entre alunos da área técnica e da humanas. Ele também poderá ser usado por pessoas com motivações e conhecimentos prévios variados. No caso de educandos iniciantes, os autores podem definir caminhos que sirvam de guia [JON88]. Pode-se, ainda, estabelecer diferenças segundo a funcionalidade pretendida, utilizando estruturas de conhecimento distintas.

4.6.2. Atualização

O processo de modificação de documentos impressos é lento pois, além do tempo necessário para ser rescrito e revisado, precisa de tempo para ser impresso e distribuído. Isto acarreta altos custos com o novo material e além disso provoca o desperdício de todo o material impresso anteriormente que não terá mais utilidade. Com o uso de sistemas de hipertextos, um software educativo pode ser fácil e rapidamente atualizado com custos muito mais baixos do que o de um livro. Um texto colocado na Internet para ser visualizado com um navegador ou sistema semelhante estará disponível instantaneamente após ser modificado. Além disso, as modificações podem ser feitas a qualquer tempo e hora. Vejam-se as páginas (*sites*) de empresas ou pessoas colocadas na rede, que são atualizadas, às vezes, diariamente.

4.6.3. Audiovisualização

Recursos instrucionais que utilizam várias mídias se mostram mais adequadas ao ensino. A apresentação de informações através de multimeios incrementa o interesse e o

entendimento, criando um ambiente de aprendizagem mais ativo, possibilitando uma melhor percepção de fenômenos complexos. As pessoas podem melhorar a aprendizagem quando uma mesma informação é apresentada através de múltiplos canais, via diferentes meios. O hipertexto, com sua capacidade de apresentar informações multimídia, pode manter a motivação e o interesse do usuário por períodos mais longos de tempo, permitindo, desta forma, que o processo de compreensão seja acelerado.

4.6.4. Conectividade

Pode-se dizer que a principal característica da hipermissão é a possibilidade de ligação entre diversos conceitos através da utilização de conexões. Segundo Landow [LAN92], as conexões são cruciais na educação, sendo particularmente interessante para ajudar o aluno a observar as relações entre o tema sendo estudado e outros conceitos. A existência de conexões implícitas ou explícitas a dicionários e referências, constituem ajudas de navegação que permitem aos leitores acesso imediato à informação necessária, sem interrupção nem procura adicional fora do ambiente [MAR88]. Este tipo de recurso é extremamente importante no ensino de disciplinas da área de Ciências Exatas, uma vez que elas requerem uma carga substancial de pré-requisitos que podem ficar acessíveis a um clique de mouse. Num livro, a referência a grande parte dos pré-requisitos é bastante limitada ou mesmo inexistente, mas num hipertexto ela pode ser explorada em profundidade, pois não existe a limitação de espaço dum livro e nem a possibilidade de fuga do assunto sendo tratado.

4.6.5. Dinamismo e interação

O dinamismo está relacionado à forma de apresentar a informação. Ao invés de uma imagem estática (como ocorre num livro) o hipertexto permite que um modelo matemático possa ser visualizado de muitas formas e sob o controle do usuário (interatividade). Pode-se fornecer uma imagem padrão (*default*) e inserir controles deslizantes ou, então, caixas de edição para que o usuário possa fornecer ou manipular os parâmetros e verificar o comportamento do modelo quando eles são modificados. A interatividade é exercida, também, através da flexibilidade oferecida ao aluno na forma de maior controle individual sobre o material em estudo, ao proporcionar condições que lhe permitem utilizar o material no seu próprio ritmo e na profundidade desejada [JAS91], eliminando através desta adaptação individual muito da angústia associada às pressões de tempo imposta na maioria dos cursos tradicionais. O fato do aluno poder perguntar, sem platéia, diretamente ao instrutor ou a um colega suas dúvidas também facilita bastante o aprendizado, já que numa sala de aula muitos se sentem inibidos em formular perguntas tanto por timidez quanto por medo de se mostrarem ridículos na frente dos colegas e do professor.

4.6.6. Intercunicação

O intercunicação de informações entre pessoas localizadas em lugares distantes é valiosa. Enviar e receber dados sem a necessidade de abandonar o local do trabalho ou estudo é uma possibilidade que pode ser exercida através da hipermissão. A maioria dos atuais sistemas de hipertexto funcionam em ambientes de redes, permitindo que os usuários possam comunicar-se de diferentes modos [CHU91]. A utilização da Internet e seu protocolo http possibilita a qualquer software que siga o padrão o acesso a informações em qualquer ponto em que ela estiver disponível. Na educação este tipo de capacidade

pode ser implementada para a elaboração de ambientes de trabalho ou estudo colaborativos entre colegas de estudo. Um aluno pode utilizar um arquivo que não é proprietário e sobre o qual tem permissão de uso para escrever comentários, estabelecendo um intercâmbio de idéias. Este método é muito usado para melhorar a aprendizagem da escrita. Podem trocar idéias para a solução de exercícios que não podem ser resolvidos isoladamente.

4.6.7. Redução

A utilização do computador no ensino reduz o volume do material tradicional utilizado, como os livros, os dicionários e as enciclopédias. Ao invés de muitos volumes de uma enciclopédia que ocupam alguns metros quadrados de estante em uma biblioteca, pode-se utilizar um ou dois CDs, ocupando uma porção irrelevante de espaço quando comparado com o texto impresso. Um dicionário pode ser substituído por um CD, reduzindo o volume do material manuseado e permitindo a consulta de forma muito mais rápida. Isto sem mencionar que a introdução de meios como o som e o vídeo, que não são possíveis com o material impresso, enriquecem o conteúdo, sem no entanto aumentar significativamente o volume armazenado. Pode-se gastar mais bits em um CD; no entanto, com o avanço da tecnologia eles estão cada vez mais disponíveis, isto é, pode-se cada vez armazenar mais informações no mesmo espaço e com um custo sempre relativamente menor. Com o texto impresso o custo é proporcional ao volume de material utilizado.

4.7. O hipertexto como recurso, tutor e assistente

Analisando-se os sistemas de hipertexto, percebe-se que eles não são sistemas instrucionais no sentido pleno do termo, mas sim sistemas de informação, projetados para facilitar a recuperação da informação ou criá-la como uma rede de documentos interconectados. Eles podem, no entanto, servir como sistemas instrucionais, se projetados para promover a obtenção de objetivos educacionais específicos, suportados por componentes suplementares necessários (aqueles que definem os objetivos, fornecem condições que permitam alcançá-los e fornecem meios apropriados para medir se foram atingidos) e avaliados para verificar o quão bem estes objetivos foram alcançados.

Pode ser útil, nesta situação, usar a classificação primeiramente utilizada por Taylor [TAY80] em relação ao uso de computadores na educação. Ele classificou os programas de computadores como: ferramenta (*tool*) (por exemplo os processadores de texto), tutor (por exemplo os sistemas CAI) e assistente (*tutee*) (programação em Pascal, por exemplo). Esta classificação pode ser expandida para a área dos sistemas de conhecimento inteligentes: a classe de ferramentas exemplificadas pelo uso de sistemas inteligentes de aconselhamento como auxiliar no desempenho de tarefas, a classe dos tutores ICAI e a classe dos assistentes pelo uso de projetos que ensinam ou reforçam o conhecimento de um dado domínio pela construção de bases de conhecimento e sistemas especialistas simples capazes de resolver problemas no domínio [ROM90].

No contexto dos sistemas de hipertexto, a mesma classificação pode ser útil. A ferramenta (*tool*) pode ser exemplificada pela mais comum das aplicações de hipertexto: seu uso como um sistema de informação para pesquisa e referências. No outro extremo, o uso como assistente (*tutee*) é exemplificado pela utilização de uma casca (*shell*) hipertexto vazia para criar, tanto individual como colaborativamente, uma rede estruturada de informações em um domínio ou para debater entre os participantes múltiplos pontos de vista em um assunto complexo. Uma vez mais, o engajamento apenas em tal

exercício não é explicitamente instrucional; no entanto, sem dúvida, mesmo uma aprendizagem implícita ocorrerá como resultado da dinâmica do processo [ROM90].

No meio das duas abordagens acima, tem-se o uso do hipertexto como tutor. Conforme já mencionado, um sistema deste tipo, com total independência do usuário e sem *feedback*, não é um sistema instrucional, assim como programas de televisão educacionais, por si só, não são sistemas completamente instrucionais.

O sistema de informações deve ser suplementado por [ROM90]:

- ✓ Um sistema que conduza o aprendiz em direção aos objetivos (com estes sendo selecionados pelo sistema ou pelo aprendiz) e que seja capaz de fornecer orientação em como alcançá-los;

- ✓ Um sistema que forneça prática apropriada para desenvolver os conhecimentos e habilidades necessários para alcançar os objetivos;

- ✓ Um sistema que possa monitorar, medir e avaliar o progresso em direção aos objetivos e, quando se tornar necessário, possa tomar ou recomendar ações corretivas de forma a reduzir a lacuna entre os objetivos e a realidade.

4.8. Conclusão

As aplicações de hipertexto no ensino começaram praticamente com o seu desenvolvimento. Embora a maioria dos sistemas de hipertexto inicialmente criados não tenha sido desenvolvida com propósitos educacionais, os que foram criados para tal se mostraram efetivos e cumpriram seus objetivos. Pode-se observar que hoje todo o software computacional vem com um sistema de hipertexto agregado com o objetivo de ensinar o seu uso. No início eram apenas instruções na forma de texto, hoje estes sistemas estão muito mais sofisticados e incluem recursos multimídias, como som, vídeo e animações.

O hipertexto tem sido aplicado a praticamente todas as áreas do conhecimento e no ensino a sua utilização está crescendo rapidamente, como pode ser verificado pela grande quantidade de material existente na Internet, oferecida nos formatos de *CD-ROM*, cursos *online*, enciclopédias, bases de dados, simulações, *applets* Java, etc.



5. A WWW, A INTERNET E A EDUCAÇÃO

5.1. Introdução

Um documento de hipertexto é diferente de um documento de um processador de texto, que é estático, genérico e estruturado. Documentos de hipertexto não são estruturados e podem ser dinâmicos. Desta forma, os padrões usados nos documentos estruturados não são adequados para representar documentos de hipertexto.

As formas atuais da ODA (*Office Document Architecture*) e SGML (*Standard Generalized Markup Language*) não são ainda o bastante para a representação e troca de documentos de hipertexto, pois elas precisam extensões para representar apropriadamente o mecanismo de ligações. A SGML não especifica a forma e a apresentação da informação e também como manejar gráficos e imagens. A ODA tem como fazer isso mas ainda não é o suficiente. Ao mesmo tempo um único padrão pode não ser suficiente para a diversidade de utilização das aplicações de hipertexto. Desta forma, podem necessitar de padrões diferentes. A SGML é um sistema complexo para definir tipos de documentos estruturados e para definir linguagens de marcação (*markup*) que podem representar estes tipos de documentos. Entretanto ela apresenta as seguintes limitações:

1. o mecanismo de referência cruzada só pode ser usado para se referir a elementos dentro do mesmo documento e não a outros documentos e
2. não suporta dados dependentes do tempo tais como áudio e vídeo e também não suporta gráficos e imagens.

Já o padrão ODA apresenta os seguintes problemas:

1. o processo de layout é seqüencial e baseado em páginas e desta forma não fornece completa interatividade;
2. não suporta edição *online*, tal como a rolagem do documento;
3. não pode apresentar informações adicionais por demanda, como as janelas *pop-up*, necessárias em glossários e notas de rodapé;

Para superar estas limitações alguns padrões tem sido propostos como o HyTime (*Hypermedia/Time-based Structuring Language*). É uma proposta de linguagem para representar a estrutura de documentos hipermídias como uma extensão da SGML, de onde herda a habilidade para definir:

- modelos múltiplos de documentos;
- a estrutura do documento independente da apresentação do mesmo.

Ela define os seguintes aspectos da estrutura de um documento hipermídia:

- endereços de documentos objetos;
- relações entre documentos objetos;

Um segundo padrão é o MHEG (*Multimedia and Hypermedia Experts Group*) que foi proposto pelo CCITT, como o padrão internacional futuro para objetos de informação multimídias e hipermídias. Os objetivos iniciais do padrão MHEG incluem: fornecer abstrações para apresentações em tempo-real, incluindo sincronização e interativi-

dade multimídia; fornecer abstrações para troca em tempo-real com um mínimo de armazenagem (*buffering*) utilizando comunicações de dados de velocidade normal; fornecer abstrações para manipulação direta da informação sem qualquer processamento adicional. A despeito deste novos padrões sugeridos e sendo estudados, o que de fato está valendo e funcionando como padrão de trocas de informações, atualmente, é a html.

5.2. A linguagem html

A **html** é o acrônimo de *HyperText Markup Language*. A linguagem foi projetada para especificar a organização lógica de documentos de texto, com extensões para incluir imagens, formulários, e conexões (*links*) para outros documentos ou recursos na Internet. O objetivo foi torná-la independente de plataforma particulares, representando a estrutura lógica do documento e não sua apresentação. Desta forma, os interessados em visualizar um documento podem usar as mais diferentes plataformas de hardware e diferentes programas navegadores (*browsers*). Por exemplo, a linguagem não especifica que um determinado texto deva ser visualizado em uma fonte *Times Roman* com um tamanho de 14 pontos, pois, dependendo de que hardware esteja sendo utilizado, isto não teria nenhum valor. Por este motivo a linguagem html não especifica detalhes datilográficos de um documento, mas, ao invés, marca a estrutura lógica do documento, definindo parágrafos, cabeçalhos, títulos, listas, tabelas, etc. Os detalhes da apresentação dos documentos são deixados para os navegadores. A linguagem foi definida em termos da ISO SGML.

5.2.1. A html como um tipo mime

A linguagem html foi projetada como sendo uma extensão do correio eletrônico, isto é, como uma extensão mime. Este, por sua vez, foi projetado para mandar mensagens de correio eletrônico multimídia (gráficos, texto e outros formatos) utilizando o protocolo padrão do correio eletrônico. Este padrão utiliza cabeçalhos tipo de conteúdo, descrevendo o dado sendo enviado. Na rede os tipos são utilizados pelo protocolo http para informar que documento está sendo enviado. Quando o servidor manda um arquivo para um cliente ele inclui como cabeçalho, um "cabeçalho tipo de conteúdo MIME", indicando que dados estão sendo enviados. No padrão mime são definidos sete tipos de conteúdo.

Por exemplo, um arquivo, contendo uma imagem no formato GIF, sendo enviado de um servidor para um cliente, terá a seguinte linha:

Content-Type: image/gif,

como parte do cabeçalho http que precede a imagem de fato.

Os servidores de http tratam documentos em html como sendo um documento mime. Quando um documento é enviado ele é precedido pelo cabeçalho:

Content-type: text/html,

que informa ao navegador que o documento é html e não meramente texto, que seria enviado como text/plain, que é o tipo padrão. Assim se um documento não tiver um campo "tipo de conteúdo", deve ser considerado do tipo text/plain.

Um documento html é um arquivo ASCII que contém comandos da linguagem html. A tabela 5.1 mostra uma lista de alguns comandos da linguagem.

Tabela 5.1 - Alguns comandos html

Comando html	Finalidade
<html>	Deve ser o primeiro comando de um arquivo html.
<head>	Apresenta a seção de cabeçalho.
<title> título </title>	Fornece o título do documento.
<p>	Parágrafo.
 texto 	Exibe o texto em negrito.
 texto 	Declara um vínculo ativo. A palavra texto aparece para o usuário em uma cor diferente. Um clique sobre ela faz com que o navegador abra o arquivo "NomeUrl".
< Applet code = "nomeclasse" width = w, height = h>	Começa a definição de um applet. Nomeclasse é o nome da classe do applet. Width e height são as dimensões iniciais da janela na qual o applet será executado.
<param name = "nomeparam" value = "valorparam">	Define um parâmetro para o applet. Nomeparam representa o nome do parâmetro e valorparam o seu valor.
</applet>	Finaliza a definição do applet.
</body>	Finaliza a seção do corpo.
</html>	Finaliza o arquivo html.

5.2.2. CGI

Um dos mais poderosos aspectos da html é a CGI (*Common Gateway Interface*). Seria ótimo se existissem alguns comandos que permitissem ao usuário rodar programas através de um navegador e processar os resultados no documento sendo apresentado. Ao invés de criar uma linguagem própria, os projetistas da html construíram a CGI [ROW96]. Um *gateway* é uma conexão para um sistema operacional externo. A CGI fornece um meio para que programadores de páginas html possam chamar programas externos, escritos em qualquer linguagem, e apresentar os seus resultados.

Rodar um programa CGI a partir de um navegador é simples para o usuário. Para o programador nem tanto. Algumas etapas devem ser seguidas [ROW96]:

1. o usuário invoca o programa CGI clicando uma conexão;
2. o navegador contata o servidor solicitando permissão para rodar o programa;
3. o programa servidor verifica os arquivos de configuração e de acesso para ter certeza de que o solicitante está autorizado a rodar o programa;
4. o servidor verifica se o programa CGI solicitado existe;
5. se ele existir, o programa CGI é executado;
6. qualquer saída produzida pelo programa CGI é retornada ao navegador;
7. o navegador mostra as saídas do programa CGI.

A informação pode ser passada do navegador da rede para o programa CGI de várias formas e o programa pode retornar o resultado como um comando html embutido, apenas texto ou como uma imagem.

5.2.3. Java

A CGI é um ótimo recurso. No entanto, ela é adequada se a comunicação for muito rápida, o que não é verdade para a maioria das pessoas que se utilizam de linhas de telefones convencionais. Uma solução é transferir o programa para o cliente e executá-lo localmente. O problema desta abordagem é o mesmo anterior, com o agravante de que seria necessário ter certeza de que o programa sendo transferido é apropriado para ser executado na máquina do usuário e, segundo, ter garantia de que o programa não irá executar alguma ação nociva no computador cliente. A linguagem Java pode resolver estas questões. A maioria do código necessário para executar um programa reside no navegador utilizado pelo cliente, e o que é transferido do servidor são programas compactos, denominados de *applets*, embutidos nas páginas escritas em html.

Um aplicativo em Java é um programa independente, que é executado no sistema (cliente) através de um interpretador Java. Um *applet* em Java é, geralmente, um programa em Java remoto executado por um navegador da rede. O navegador localiza o *applet* através do comando

```
<applet code="Mine" width=larg_janela height= alt_janela></applet>
```

embutido em uma página html.

Uma desvantagem desta abordagem, além de funcionar de forma mais lenta por ser interpretada, é que ela só funciona se o navegador possuir um interpretador de códigos de bytes Java.

5.2.4. A JavaScript

A JavaScript é uma linguagem de *script* que complementa os recursos do Java. Ela apresenta uma sintaxe semelhante e alguns recursos do Java sem exigir, no entanto, que o usuário aprenda programação objeto orientada. A linguagem pode reconhecer e responder a vários eventos gerados pelo navegador, inclusive cliques e movimentos de *mouse*, abertura de páginas e saídas de formulários. Os roteiros (*scripts*) da JavaScript estão incluídos nos documentos e são executados pelo navegador quando o documento é carregado.

As linguagens de *script* são colocadas nos documentos através da marca (*tag*) ou comando `<script>`. Esta marca pode ser colocada em qualquer lugar do corpo do documento ou no cabeçalho. Se ela for colocada no cabeçalho, o *script* será executado antes mesmo do documento ser totalmente recebido pelo navegador. Por isto, é conveniente colocar as funções e o código de inicialização no cabeçalho. A marca deve ser finalizada por um segundo comando `</script>`, que indica o fim do bloco *script*. Esta marca apresenta o parâmetro opcional "*language*", que especifica a linguagem em que *script* foi escrito.

5.3. A Internet

No início dos anos 60 a ARPA (*Advanced Research Project Agency*), agência do departamento de defesa americano, começou a estabelecer uma rede de computadores experimental que cobriria os Estados Unidos, denominada de ARPANet. O objetivo inicial da rede era permitir que os contratados do governo pudessem compartilhar os recursos computacionais então caros e escassos. Desde o início os usuários da rede a utilizaram para serviços colaborativos. Estes serviços incluíam o compartilhamento de arqui-

vos e a troca de correspondência eletrônica para o desenvolvimento conjunto de pesquisas utilizando computadores remotos compartilhados.

O protocolo TCP/IP foi desenvolvido no início dos anos 80 e rapidamente tornou-se o padrão da ARPANet que se tornou a via principal (*backbone*) de uma confederação de redes locais e regionais baseadas neste protocolo denominada de **Internet**. Seguindo o sucesso da ARPANet, a NSF (*National Science Foundation*), criou em 1981, a CSNet. Um acordo entre a ARPA e a NSF permitiu a interconexão das duas redes. A CSNet evoluiu e foi rebatizada de NSFNet integrando as redes dos campi da maioria das universidades americanas e outras grandes redes. Em 1988 o departamento de defesa americano começou a desmantelar a ARPANet, deixando a NSFNet como a principal rede (*backbone*) dos Estados Unidos, do que é hoje conhecido como a Internet.

Deve-se fazer uma diferença entre a **Internet** e uma internet. A Internet, grafada com **I** maiúsculo, refere-se à rede que substituiu a ARPANet e constitui a confederação de todas as redes que compartilham o protocolo TCP/IP, direta ou indiretamente conectadas a NSFNet. Por outro lado uma internet, grafada com letra minúscula, é simplesmente uma rede formada de outras pequenas redes compartilhando o mesmo protocolo e que não está necessariamente conectada na Internet e que também não necessariamente se utiliza do protocolo TCP/IP.

5.4. O projeto WWW

A rede mundial (*World Wide Web*), normalmente abreviada por WWW, algumas vezes por W3 ou simplesmente a **rede** (*the Web*) é o mais novo serviço a integrar a Internet. A rede é baseada na tecnologia do hipertexto. Muito do seu desenvolvimento foi realizado no CERN (*Centre Européen pour la Recherche Nucleaire*), mas isto não significa que ela foi projetada para ser de uso exclusivo de físicos. Segundo a definição fornecida pelo seu criador, Tim Berners-Lee, ela: "É o universo da informação acessível através de uma rede". É uma, talvez a mais flexível, ferramenta para explorar a Internet. Trata-se de um mecanismo extremamente dinâmico que continua a evoluir rapidamente e que foi responsável pelo desenvolvimento acelerado da Internet. A WWW é formada, basicamente, por dois softwares. Um programa cliente denominado de navegador (*browser*) que o usuário utiliza para acessar um programa provedor. O servidor é o responsável pelo fornecimento de serviços, isto é, arquivos espalhados ao redor do mundo. Os dois softwares (servidor e cliente) trocam informações através do protocolo http.

5.4.1. O http

O http (*HyperText Transfer Protocol*) original foi definido em 1991, e estabelece o método de transferência de informações através da Internet. O protocolo se utiliza do estilo convencional da Internet de transferência de arquivos, o telnet, em uma ligação TCP/IP. O computador cliente solicita uma conexão TCP-IP¹ com o computador servidor (*host*) utilizando o DNS (*Domain Name System*) ou um número IP e o número da porta dado no endereço. Se o número da porta não for especificado, então o número 80 é assumido como padrão. O servidor aceita a conexão. O cliente envia uma requisição consistindo de uma linha de caracteres ASCII terminada pelo par CR LF (*Carriage Return, Line Feed*), que nem sempre é necessário. A requisição consiste da palavra "get", um espaço, o endereço do documento, o servidor e a porta. A resposta é uma mensagem

¹ O HTTP atualmente roda sobre o TCP, mas pode rodar sobre qualquer serviço semelhante.

em html, isto é, é um fluxo de bytes de caracteres ASCII. A mensagem é terminada pelo fechamento da conexão com o servidor. A conexão é fechada pelo servidor assim que a totalidade do documento é transferido.

5.4.2. Arquitetura

Essencialmente a WWW segue a arquitetura dos três níveis da maioria dos sistemas de hipertexto. O nível mais baixo, o da base de dados, é formado pela Internet, isto é, por todos os computadores através do mundo conectados a ela e que fornecem material para a WWW. Estes computadores agem como servidores, e o usuário (em princípio) não precisa saber onde eles estão localizados, de que tipo de hardware ou software eles estão constituídos e que mecanismo interno de armazenamento de dados eles estão utilizando. Todos eles fornecem seus dados aos clientes em um formato padrão através de um protocolo também padrão. Cada servidor tem um endereço ou URL (*Uniform Resource Locator*), de modo que ele possa ser acessado pelos usuários da Internet. Cada URL aponta para um local da rede (*site*) particular, que é um diretório contendo os arquivos que formam as páginas visíveis da rede. Esta combinação entre o html e o http, formando o segundo nível, constitui a máquina abstrata do hipertexto e é o único ponto em que os computadores cliente e servidor devem concordar. O computador usuário pode usar uma grande variedade de softwares, rodando em variadas plataformas de hardware, desde que aceite o http e entenda arquivos formatados em html. A grande vantagem destes protocolos é que eles permitem a conexão de dois computadores, não importando se eles são mainframes, estações de trabalho, PCs ou mesmo Macs. O terceiro nível, o de apresentação do modelo (a interface com o usuário) é gerenciado pelo visualizador cliente, rodando na máquina do usuário, isto é, pelo navegador.

5.4.3. O funcionamento da rede (WWW)

A rede funciona segundo o modelo **cliente-servidor**. Um **servidor** é um programa cujo propósito é fornecer documentos para os clientes quando requerido. Um cliente é um programa que faz a interface com o usuário e realiza pedidos de documentos a um servidor especificado pelo usuário. O modelo funciona da seguinte forma:

1. através do **cliente** (o navegador), o usuário seleciona uma conexão de hipertexto (URL), que aponta para um documento (texto, imagem, som, programa, etc.);
2. o cliente conecta-se ao computador (servidor) especificado pelo endereço contido na URL (em algum lugar da Internet) e solicita o documento.
3. o servidor responde enviando o documento e qualquer outro elemento de mídia existente no documento (figuras, sons ou filmes) para o usuário.

A linguagem que clientes e servidores da rede mundial usam para se comunicarem é o http. Todos os clientes e servidores precisam utilizar este protocolo para poderem enviar e receber documentos hipermídia. Os documentos ou serviços disponíveis na Internet são acessados através de endereços denominados de URL (*Uniform Resource Locator*). É possível representar quase que todos os serviços da Internet através destes endereços.

Um URL possui o seguinte formato geral:

serviço://nome_do_computador/nome_do_caminho.

O serviço indica o protocolo usado para obter os dados do local sendo acessado. Por exemplo, o ftp é um serviço possível. O serviço mais comum é o **http**. O serviço **file** é utilizado quando se carrega um arquivo do próprio computador, ao invés de um da

Internet. Este é o único serviço local, todos os demais são remotos.

Nome_do_computador indica o endereço do local que contém o arquivo. Por exemplo, **www.pucrs.br** é o endereço da PUCRS (Pontifícia Universidade Católica do Rio Grande do Sul). Esse campo é necessário para todos os serviços remotos.

Nome_do_caminho é o nome do arquivo a ser carregado. Quando ele não estiver presente, o navegador carregará o arquivo padrão, que geralmente possui o nome **index.html**. Esse arquivo é chamado de "página da casa" (*home page*) do local.

A primeira parte do URL (antes das duas barras) especifica o método de acesso. A segunda parte é tipicamente o endereço do computador, os dados ou a localização do serviço. Outras partes podem especificar nomes de arquivos, a porta a ser conectada, ou o texto para se buscar em um banco de dados. Abaixo alguns exemplos de URLs.

 **file://endereco.do.local/imagem.gif** – recupera uma imagem do disco rígido e a exibe, em um programa separado ou dentro de um documento.

 **http://endereco.do.local/diretorio/book.html** – recupera o arquivo **book.html**.

 **ftp://endereco.do.local/pub/file.txt** – abre uma conexão **ftp** em **endereço do local** e recupera o arquivo texto **file.txt**.

 **telnet://endereco.do.local:1234** – conecta-se através do serviço **telnet** em **endereço do local** através da porta lógica 1234.

 **nntp://alt.hypertext** – conecta-se a um servidor de notícias em **alt.hypertext**.

As conexões podem fazer referências não somente a outros textos e mídias, mas também a outros serviços da rede. Os navegadores não são simplesmente clientes da rede, mas são também clientes **ftp**, **gopher** e **telnet**. As conexões são realizadas através de uma porta lógica do computador. Cada computador possui muitas portas e em cada uma ele fornece um serviço diferente.

O **html+** também inclui um URL de correspondência eletrônica (*e-mail*). Assim, selecionando o endereço **viali@pucrs.br**, por exemplo, inserido em um documento, um programa de correspondência eletrônica é aberto e fica disponível para enviar uma mensagem para o endereço referenciado.

A conexão de um software cliente a um servidor é feita através de uma tomada (*socket*), abstração desenvolvida na universidade da Califórnia em Berkeley, para tornar a programação em rede mais fácil. Teoricamente um *socket* pode ser independente dos protocolos de rede existentes, no entanto, na prática, ele foi atrelado ao **TCP/IP**, tendo conseguido muito sucesso como **API**. Pode-se encontrar implementações de tomadas **API** em quase todo sistema operacional que suporte o **TCP/IP**. No começo dos anos 90 foi criada uma versão para o ambiente **MSWindows**, que foi denominada de **WinSock** e que contribuiu para a migração de programas de rede das estações (utilizando **Unix**) para o **PC**.

5.5. O navegador

O navegador (*browser*) é o software que foi responsável pela verdadeira explosão no crescimento da Internet. Muitos dos recursos da Internet já existiam, só que para acessá-los era preciso conhecer vários programas diferentes e basicamente todos dependentes de plataformas específicas. Assim, para acessar um ambiente rodando **Unix** era necessário conhecer comandos deste sistema, para acessar um *mainframe* **IBM** era necessário conhecer detalhes do seu sistema operacional. Os navegadores e o **http** padronizaram e incorporaram num único software estes recursos, colocando ordem no então

aparente caos reinante. Desta forma, para utilizar ou acessar a rede não é mais necessário possuir conhecimentos avançados das ferramentas a serem utilizadas. Basta apontar, clicar e pronto.

Os navegadores interpretam arquivos escritos em html e apresentam os comandos da linguagem de forma própria. Cada navegador possui particularidades e um mesmo arquivo html pode parecer bem diferente quando vistos com diferentes navegadores. A figura 5.1, mostra o que um arquivo html é e como ele é visto por um navegador.

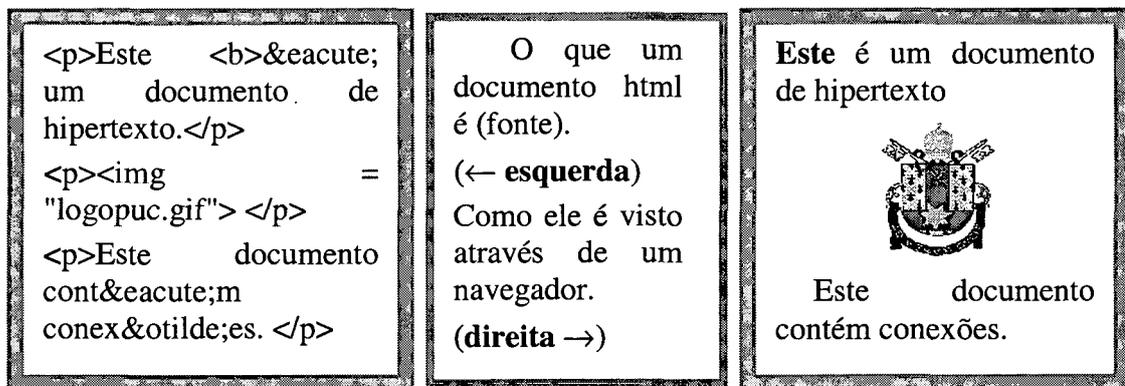


Figura 5.1 - Como um arquivo html é e como é visto pelo navegador

5.5.1. Recursos

Entre as características que os navegadores oferecem podem-se destacar:

- interface gráfica orientada por mouse;
- formatação de documentos com elementos como listas e tabelas;
- suporte para sons (os formatos dependem dos *plug-ins* instalados);
- suporte para vídeo (os formatos dependem dos *plug-ins* instalados);
- exibição de caracteres definidos no conjunto ISO 8859;
- suporte para formulários eletrônicos interativos;
- suporte para imagens estáticas (Gif e Jpeg);
- suporte para os serviços de rede: [ftp](#), [gopher](#), [telnet](#), [nntp](#), [wais](#).

5.5.2. O projeto dos navegadores

Os navegadores são pacotes projetados para usar de forma eficiente as informações da rede. Eles são bastante semelhantes na aparência, diferindo apenas em características particulares. Em geral todos apresentam os seguintes pontos em comum:

As opções de **avançar** (*forward*) e **voltar** (*back*) que permite que o usuário avance ou retroceda através das conexões já percorridas. A opção **parar** (*stop*) que é utilizada para interromper uma transferência corrente. É usada tipicamente quando o usuário desiste de carregar o arquivo completo por alguma razão. A opção **recarregar** (*refresh ou reload*) que força o navegador a recomençar o carregamento do arquivo do local original. A opção **local** (*home*) que é usada normalmente para iniciar a navegação e apresenta uma página padrão, geralmente do fabricante do software. A opção **procura** (*search*) que é usada para conectar a mecanismo de procura de informações na rede como o

Yahoo, Lycos, Cadê, Zaz, etc.

Os navegadores são eficientes para recuperar e apresentar páginas html, mas não possuem recursos para preparar e apresentar material didático, por isto este trabalho propõe um modelo de navegador, construído com as opções clássicas acima mas acrescido de novos recursos voltados ao ensino, através da extensão de algumas das habilidades existentes e com a integração de um modelo de multiagentes.

5.5.3. Aplicativos auxiliares

Arquivos de animação, de áudio, de vídeo e de realidade virtual podem ser executados numa janela de um navegador normalmente com o auxílio de aplicativos adicionais específicos denominados de auxiliares ou *plug-ins*. São programas especializados que aumentam a capacidade do navegador. O objetivo é traduzir para o navegador arquivos com formatos particulares e que de outra forma não seriam entendidos. Este tipo de operação não é visível para o usuário de modo que para ele é como se o próprio navegador estivesse interpretando aquele formato específico.

Esses programas também denominados de executores (*players*) estão disponíveis para um grande número de formatos de arquivos e alguns estão incorporados ao navegador enquanto que outros não. A tabela 5.2 lista alguns destes aplicativos.

Tabela 5.2 - Alguns aplicativos auxiliares mais comuns [HAL97]

Aplicativo	Empresa	Aplicação
Acrobat Reader	Adobe	Mostra arquivos no formato pdf.
CMX Viewer	Corel	Mostra arquivos gráficos vetoriais de alta resolução.
Neuron	Asymetrix	Executa animações e programas produzidos com o ToolBook II.
RealAudio	Progressive Networks	Executa arquivos de áudio por demanda em tempo real.
Shockwave	Macromedia	Executa arquivos Authorware e Director.
VDOLive	VDONet	Comprime imagens de vídeo em função da velocidade de conexão do usuário.

5.5.4. Controles ActiveX

São programas semelhantes aos *plug-ins*, só que na versão da empresa Microsoft. Com o navegador Netscape os arquivos são visualizados com objetos "externos" ou *plug-ins*. Com o Microsoft Internet Explorer são utilizados objetos "controles" ou "ActiveX". A principal diferença é que o ActiveX é uma tecnologia exclusivamente para o sistema operacional Windows, enquanto que os auxiliares podem ser escritos para qualquer plataforma. A tabela 5.3 relaciona alguns controles ActiveX mais comuns.

Tabela 5.3 - Controles ActiveX mais comuns [HAL97]

Aplicativo	Empresa	Aplicação
Button Maker	FarPoint	Controla o tamanho, a forma e a cor de botões que substituem os padrões do Windows.
ClearFusion X	Iterated Systems	Executa arquivos de vídeo do formato AVI, à medida que vão sendo baixados, sem a necessidade de esperar que sejam totalmente transferidos.
Control Pak 1	Kwery	Conecta com a base de dados Access.
Power Point Animation Player	Microsoft	Executa apresentações multimídias no formato PowerPoint.
RealAudio	Progressive Networks	Executa arquivos de som no formato RealAudio.
ShockWave	Macromedia	Executa animações interativas nos formatos dos pacotes Authorware e Director.
Surround Video	Black Diamond	Apresenta imagens de 360° no formato SDK.

5.6. A WWW e a Internet como recursos instrucionais

As razões para se utilizar a tecnologia da Internet podem ser administrativas, culturais ou pedagógicas. A Internet fornece uma maneira extremamente flexível de distribuir informação e permite que os interessados cheguem a ela através de uma grande variedade de plataformas de software e hardware a qualquer tempo. O material pode ser acessado instantaneamente, sem a demorada etapa intermediária da impressão e distribuição que ocorre com o impresso. Afinal, hoje, todo e qualquer conteúdo impresso é inicialmente digitado e formatado em um computador, para depois passar por outros processos intermediários e então ser levado a uma gráfica onde será finalmente impresso. Apresentar o material através da Internet significa cortar etapas e desta forma fazer chegar ao usuário de uma forma mais rápida.

Os problemas enfrentados para corrigir e atualizar a informação impressa são, sem dúvida, uma das suas principais desvantagens. Uma vez efetuada a tiragem de um livro, panfleto, etc. a correção só será efetuada se houver uma nova edição ou tiragem. Com o material apresentado de forma virtual, esta correção pode ser efetuada no mesmo momento em que for detectada.

A Internet pode aumentar a interação entre alunos, entre professor e aluno e entre professores através de várias formas. Isto permite realimentação (*feedback*) imediata de modo que o professor ou instrutor pode acrescentar ou melhorar o material sendo apresentado. A capacidade multimídia fornece tanto uma apresentação mais entusiasmadora quanto formas de adequação a diferentes estilos de aprender de alunos individuais. Alguns aprendizes são mais visualmente orientados do que outros e alguns precisam uma combinação de estimulação audiovisual para aprenderem bem.

A retenção, de acordo com Begley [BEG93], é de 10% do que é visto, 20% do que é ouvido, 50% do que se vê e ouve (hipermídia) e 80% do que é visto, ouvido e feito (hipermídia interativa). Por isto é que o sistema, que está sendo proposto, enfatiza a interatividade e agrega recursos e diversos programas adicionais para que o usuário possa praticar os conceitos sendo lidos, vistos e/ou ouvidos.

5.6.1. Recursos necessários

Os recursos necessários para se usar a Internet como ferramenta educacional variam em função do propósito da aplicação pretendida. Pode-se classificar estas necessidades nas seguintes categorias.

- De infra-estrutura;
- De publicação;
- De suporte e
- De serviços.

5.6.2. Recursos de infra-estrutura

A infra-estrutura necessária para que se possa usar a Internet como ferramenta educacional com sucesso, deverá incluir:

Conexão a uma rede. Esta conexão pode ser feita diretamente (placa de rede) ou indiretamente através de um modem.

Acesso ao equipamento. Os alunos devem ter à disposição acesso suficiente tanto em recursos quanto em tempo.

Suporte técnico. Deve haver um suporte técnico disponível para a manutenção dos micros sendo utilizados, do servidor ou servidores e da própria rede.

O principal obstáculo, atualmente, para a utilização ampla dos serviços de rede e da Internet é a disponibilidade de equipamentos. Quando, num futuro próximo, os micros portáteis (*notebooks*) forem tão comuns quanto as calculadoras o são hoje, pode-se mudar radicalmente a forma de ensino, com a utilização massiva do computador. Neste ponto será importante uma ferramenta que possa se valer da Internet e que também possa manejar e trabalhar com arquivos e recursos locais. O sistema sendo proposto pode trabalhar tanto remotamente, baixando arquivos no momento em que for necessário, mas poderá trabalhar também localmente, com conjuntos de arquivos, já armazenados previamente no disco rígido.

5.6.3. Recursos de publicação

Os recursos de publicação necessários poderão incluir;

Ferramentas de conversão. Caso o material já se encontre na forma digital, escrito em processador de texto, deve-se ter pelo menos um software para converter o material escrito no processador de texto específico para o formato html.

Software de autoria. Estes softwares incluem editores html, programas para gerar tabelas e gráficos e até mesmo ferramentas de editoração para a WWW.

Existe hoje uma grande quantidade de softwares disponíveis para realizar as tarefas acima. Os processadores de texto também já estão incorporando recursos para a transformação de arquivos armazenados em formatos próprios para arquivos html.

5.6.4. Recursos de suporte

Os recursos de suporte incluem ferramentas para a instalação e o suporte de provedores (*sites*) WWW. Estas ferramentas incluem:

Softwares de pesquisa (*Search Engines*). Para colocar à disposição um critério para investigar todas as páginas de um determinado local (*site*) à procura de uma determinada frase ou palavra.

Bibliotecas de gráficos, imagens e ícones. Coleções de gráficos, imagens e ícones que podem ser utilizados para construir as páginas.

Bibliotecas de vídeos e sons. Coleções de arquivos de vídeo e sons que podem ser usados nas páginas.

Softwares visualizadores. Um software que possa visualizar os vários tipos de arquivos utilizados nas páginas (texto, gráfico, imagem, vídeo, som) e os vários formatos de cada um destes arquivos.

5.6.5. Recursos de serviço (o local)

Incluem ferramentas para preencher necessidades específicas como, por exemplo, um recurso de correio eletrônico (*e-mail*) para a comunicação professor-aluno.

Dentre todos os recursos, talvez, o mais importante seja o local (*site*). De nada adianta uma ferramenta sofisticada de navegação se não existir um conjunto de recursos bem estruturado para fornecer material didático. Para que qualquer curso *online* tenha sucesso é necessário que exista, à disposição do usuário, um local (*site*) que forneça informações atualizadas, adequadas ao nível do aluno e de uma forma dinâmica e atrativa. Por isto é necessário um conjunto de conhecimentos armazenados em linguagem apropriada, neste caso o html, para que o aluno tenha acesso ao material a qualquer tempo.

O local fornece, além dos arquivos html com o conteúdo estruturado por áreas, conforme a identificação fornecida ao sistema, base de aplicações com classificação difusa, base de perfis difusa dos alunos usuários, base de casos (exemplos), base de conhecimentos retroativos (pré-requisitos). No caso do local para o modelo proposto, a adequação ao aluno se dará pela carga de pré-requisitos matemáticos embutidos nos conteúdos, pelos exemplos apresentados e principalmente pelas listas de exercícios. Um agente se encarregará de verificar o tipo de usuário que está se conectando ao sistema e adequará o material sendo acessado a este tipo de usuário.

5.7. Limitações

A grande vantagem da rede para o ensino é que o material pode ser divulgado para uma grande número de pessoas que podem estar situadas em qualquer lugar do Globo. No entanto, para que ela se torne um instrumento adequado para ensinar ou auxiliar no ensino são necessários algumas ferramentas adicionais. O hipertexto é uma ferramenta extremamente útil para ser recuperada à distância através de um computador, mas as conexões embutidas no documento que tornam possível esta característica podem ser uma desvantagem na hora de imprimir ou trocar, pois pode-se cortar informações. Ou seja, o documento em si é impresso, mas suas ligações não. Desta forma as conexões em si não devem fazer parte do material de ensino.

Aprender fazendo é um dos melhores métodos de ensino. Desta forma, o material didático de alto nível deve ser multimeio e interativo. Simulações e apresentações multimídias são excelentes meios de fornecer material de ensino. No entanto, na WWW, o suporte a imagens embutidas não é amplo o suficiente para incluir a variedade de formatos existentes. Normalmente os únicos formatos aceitos para imagens é o GIF ou JPG. Já para texto somente o formato ASCII. A falta de vídeo, áudio e animação

embutidas em documentos requerem a utilização de programas adicionais para rodar, os denominados visualizadores externos (*external viewers*). A desvantagem desta abordagem é a perda de uniformidade na interface com o usuário e a funcionalidade do hipertexto para estes materiais.

A principal desvantagem desta dependência em programas externos para executar funções avançadas é que a capacidade de conexão do hipertexto é perdida no momento em que o nodo de um dos programas externos é executado. Quando o software passa o controle para um programa externo, o usuário não pode mais seguir qualquer conexão do nodo que chamou o programa que está sendo executado. O modelo proposto deverá manter a última conexão de modo que ao término da execução do programa ele possa retomar o caminho a partir daí.

5.8. Conclusão

Os sistemas de hipermídia têm sido construídos como sistemas fechados com mecanismos de armazenagem proprietários e com pouca ou nenhuma interoperabilidade. Um grande número de arquiteturas, modelos ou mecanismos e estruturas têm sido propostos e desenvolvidos por pesquisadores no esforço de tornar os sistemas de hipertexto sistemas mais genéricos e integrados ao ambiente de desktop. Ferramentas de desenvolvimento foram construídas para auxiliar os programadores na tarefa de adicionar funcionalidade aos sistemas existentes. Com o objetivo de tornar os sistemas de hipertexto totalmente abertos e integrados, os seguintes problemas têm sido trabalhados: interoperabilidade, controle de concorrência para acesso multiusuário em um ambiente compartilhado, nodos e tipos de conexões, sistemas operacionais de suporte, protocolos de conexões, suporte multimídia, consistência da interface e controle de versões. Muitos destes problemas foram encaminhados através do uso de técnicas objeto orientadas. Independente destas dificuldades menores, a WWW foi criada, cresceu e é hoje, sem nenhuma dúvida, o sistema de hipermídia mais utilizado no mundo. Ela poderá sofrer modificações, melhorias, mas acredito que, pelo menos por enquanto, dificilmente as coisas mudarão a ponto de se adotar um padrão totalmente diferente.

A maioria das melhorias no que diz respeito à tecnologia do computador é incremental ou evolucionária. Contudo, algumas vezes surge uma nova tecnologia que mexe com toda a indústria. A WWW é certamente um destes tipos de tecnologia. Ela representou para a Internet o que o telefone representou para a telegrafia. Antes dela a Internet possuía um meio de trocas de informações internacional poderoso, mas que não era fácil de usar e por isto acessível a poucas pessoas. Os usuários iniciais da Internet tinham que dominar vários programas, cada um com uma interface diferente e portanto com particularidades e comandos próprios. A WWW padronizou e colocou ordem na confusão antes reinante. Com os navegadores o usuário pode acessar os locais (*sites*) sem ter que se preocupar com o tipo de computador que está sendo acessado, isto é, sem ter que se preocupar se o comando seguinte é para Unix, ou Vax ou para um ambiente IBM.

Uma das características da WWW é que muitos navegadores diferentes estão disponíveis, fornecendo diferentes habilidades e atendendo a diferentes necessidades dos usuários. No entanto, esses navegadores, apesar de apresentarem capacidades poderosas de navegação, não são ou não foram construídos com o objetivo específico de servirem como base de ensino. Por isto, não fornecem capacidades para tal.



6. A INTELIGÊNCIA ARTIFICIAL NO ENSINO

6.1. Introdução

Aplicações tradicionais de computação, como bases de dados, planilhas, programas gráficos e processadores de texto, são fundamentalmente ferramentas de desenvolvimento que objetivam economizar tempo, ou seja, aumentar a produtividade de tarefas tradicionais tais como calcular, classificar, desenhar, datilografar, etc. Elas não tentam imitar atividades humanas como, por exemplo, resolver problemas, diagnosticar, ou planejar. Além disso, tais programas empregam algoritmos verdadeiros, não apresentando inteligência *per se*. Estes algoritmos tipicamente criam uma solução, requerem um conjunto completo de dados e usam um conjunto previsível de passos determinado pelo programador. Eles não estão usando conhecimento ou inferência para obter o resultado. Ao invés, utilizam um limitado conjunto de instruções que não variam, mesmo se as condições ambientais mudarem [BIE91].

Sistemas inteligentes, por sua vez, são mais flexíveis e adaptativos no sentido de que eles se baseiam no conhecimento e no poder da associação e inferência para encontrar a direção que leve a um resultado útil. Eles lidam com interações complexas de muitos fatores que devem ser considerados como um todo, ao invés de uma série de passos, num processo geralmente denominado processamento simbólico [BIE91]. Eles funcionam através de regras que dirigem a manipulação dos símbolos. Desta forma, mesmo trabalhando com informações limitadas, os sistemas inteligentes podem alcançar resultados úteis. Esses sistemas trabalham com quasi-algoritmos.

6.2. Sistemas Inteligentes de Ensino

Os sistemas inteligentes de ensino tiveram como origem a aplicação da Inteligência Artificial ao CAI. No meio dos anos 60, algumas tentativas já tinham sido feitas para desenvolver sistemas generativos para CAI [VEN91]. De fato, a primeira aplicação de inteligência artificial aconteceu no programa SCHOLAR [CAR70], que utilizava uma complexa, porém bem-definida, estrutura de informação na forma de uma rede de fatos, conceitos e procedimentos, como uma base de dados. Ou seja, o sistema descrevia seus conceitos utilizando uma rede semântica do seu domínio: a Geografia. O sistema podia tanto ensinar quanto testar um estudante, ou permitir ao estudante formular questões. O estilo Socrático de diálogo era utilizado. O sistema primeiro tentava diagnosticar os mal-entendidos do estudante e então apresentava material que forçava o estudante a ver seus erros [ROB90]. O sistema SCHOLAR fez sucesso, não tanto pelo que fez enquanto sistema CAI em si, mas sim pela influência de suas técnicas em sistemas posteriores.

Muitos sistemas inteligentes de ensino foram projetados mas não saíram da fase de protótipo [YAZ87]. Existem, no entanto, algumas exceções que devem ser notadas. Criado em 1975, o SOPHIE é o mais antigo e mais bem conhecido. Este sistema foi patrocinado pelo Departamento de Defesa dos Estados Unidos. A intenção dos pesquisadores no seu desenvolvimento foi o de explorar um ambiente interativo de aprendizagem que

encorajasse a formulação explícita de hipóteses pelo estudante, envolvido na solução de problemas, que seriam então passadas para a máquina de modo que pudessem ser criticadas, ou seja, a idéia era de que o estudante adquirisse habilidades para resolver problemas através de suas próprias idéias, ao invés de receber instruções do sistema [ROB90]. O sistema ficou limitado como tutor e treinador laboratorial. Muitos dos benefícios do sistema são resultado do fornecimento de jogos experimentais, bem como de sua ação como livro interativo, onde o material escrito descrevendo uma peça particular de equipamento pode ser impresso sempre que for relevante para a tarefa do estudante.

Desenvolvido em Stanford, o sistema especialista MYCIN contém o conhecimento dos principais especialistas no campo das doenças infecciosas do sangue. Dado um conjunto de sintomas para diagnosticar, o MYCIN utiliza seu conhecimento para chegar à conclusão mais razoável e então recomenda o melhor tratamento alternativo [BIE91].

6.2.1. Métodos de estruturar o conhecimento

Os sistemas tradicionais de ensino com base no computador têm sido desenvolvidos por pesquisadores educacionais para resolver seus problemas práticos na aplicação da tecnologia computacional. Educadores e treinadores estão principalmente interessados na melhoria da eficácia e eficiência instrucional pela aplicação de vários tipos de técnicas de software e estratégias instrucionais no desenvolvimento de lições. Em contraste, os ICAI têm sido iniciados basicamente por cientistas computacionais para explorar a capacidade de técnicas de IA no processo de ensino aprendizagem. Por este motivo, o foco dos projetos ICAI reside nos aspectos técnicos do sistema (como, por exemplo, representação do conhecimento, diálogos em linguagem natural, mecanismo de inferência, etc.), ao invés de nas características instrucionais [PAS90].

Um dos métodos que parece ser adequado ao domínio dos tutores são os denominados **sistemas de produção**. Estes sistemas são baseados em **regras de produção**, isto é, pares de expressões consistindo de uma condição e uma ação. Foram criados nos anos 70 como ferramenta para a modelagem comportamental e psicológica. Eles também podem ser utilizadas como método de representação de conhecimento em sistemas especialistas [BIT96]. A estrutura simples de uma regra de produção pode englobar uma grande variedade de formas de conhecimento. Além das regras de produção, o conhecimento pode ser representado ainda através dos seguintes métodos:

Lógica. A lógica é a base da maioria dos formalismos de representação do conhecimento, seja na forma explícita, como nos Sistemas Especialistas baseados na linguagem Prolog, ou disfarçada em formas específicas como a utilizada no sistema MYCIN.

Redes semânticas. Uma rede semântica consiste de um conjunto de nodos conectados por um conjunto de arcos. Os nodos em geral representam objetos e os arcos, relações binárias entre estes objetos. A herança de propriedades através de caminhos formados por arcos é uma das características mais importantes da representação do conhecimento através de redes semânticas. Isto permite que as propriedades de um nodo sejam especificadas uma única vez, no nível mais alto da hierarquia e então herdadas por todos os nodos derivados. A figura 6.1 ilustra uma rede com conceitos de estatística descritiva.

Quadros. Os quadros (*frames*) e sua variação, os roteiros (*scripts*) foram criados para permitir a expressão das estruturas internas dos objetos, mantendo a possibilidade de representar herança de propriedades, como as redes semânticas. As idéias básicas deste método foram estabelecidas por Minsky [MIN75]. Um quadro consiste de um conjunto de atributos que descrevem as características do objeto sendo representado. Os

valores que os atributos podem assumir podem ser outros quadros, criando desta forma uma rede de quadros, que podem ser organizados em uma hierarquia de especialização.

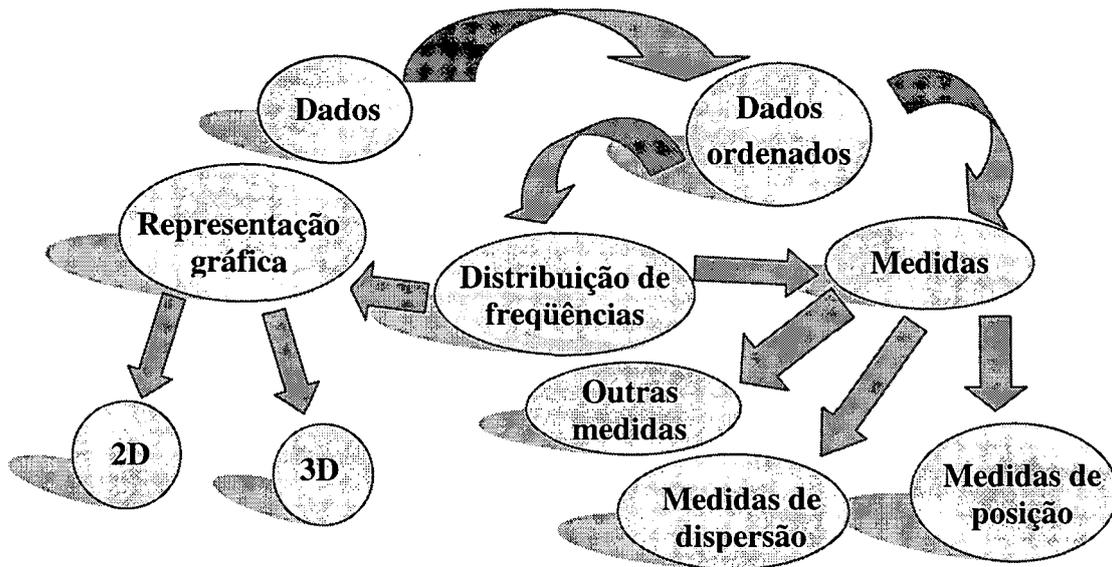


Figura 6.1 - Rede semântica com conceitos de Estatística Descritiva

6.2.2. Componentes de um sistema ICAI

A linha de desenvolvimento que começou com *SCHOLAR* continuou a se expandir, envolvendo cada vez mais áreas do conhecimento. Estes programas não são sistemas totalmente instrucionais mas sim ambientes práticos em que um maior ou menor grau de tutoriamento é oferecido [VEN91]. Eles apresentam algumas características em comum. Entre elas, destacam-se:

Modelo do Especialista ou Base de Conhecimentos. Todos os sistemas tutores englobam de uma forma ou outra o conhecimento de um especialista na matéria ou disciplina sendo ensinada. Este módulo tem a responsabilidade de gerar questões e avaliar a solução dada pelo estudante [ROB90].

Módulo do Estudante. Uma representação das habilidades do estudante, geralmente baseada em algum isomorfismo com o modelo especialista, é capturada e atualizada como parte do sistema de interação com o aluno. Este módulo contém as estratégias usadas para lecionar o conhecimento do domínio. Para Yazdani [YAZ87], este módulo é a interface do usuário, ou seja, é o módulo que administra a interação entre o sistema e o estudante.

Módulo de detecção de erros. Este módulo é constituído de uma relação de erros (*bugs*) e possíveis más interpretações, e integra quase a totalidade dos sistemas inteligentes de ensino. Para Robertson [ROB90] e O'Neil [O'NE91], este módulo está integrado no módulo tutor.

Módulo Tutor. Este módulo contém as estratégias usadas para ensinar o conhecimento no domínio pretendido. Em muitos sistemas não vai além do que informar ao usuário o que ele fez de errado e qual seria a resposta correta [VEN91, YAZ87]. A estratégia do módulo tutor é baseada em um dos seguintes métodos: (a) um diagnóstico ou abordagem corretiva na qual o sistema corrige os mal-entendidos através de tarefas e avalia as respostas obtidas; (b) o método sócrático, que envolve questionar o estudante de uma forma que o encoraje a raciocinar sobre o que ele sabe e desta forma modificar

suas concepções e (c) um método de treinamento no qual o estudante é engajado em alguma atividade tal como um jogo computacional que encoraje o aumento de suas habilidades na solução de problemas gerais [ROB90].

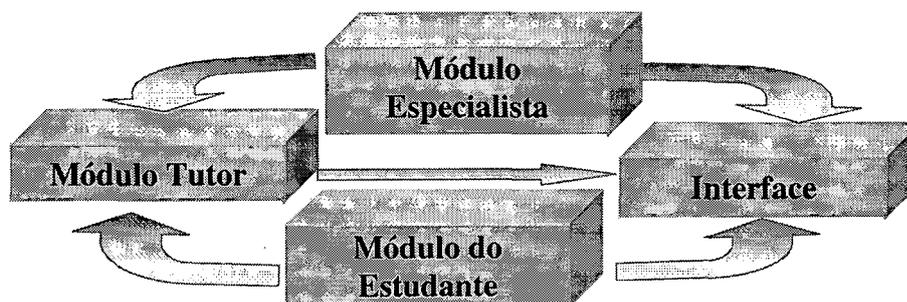


Figura 6.2 - Uma representação esquemática de um sistema ICAI

6.2.3. ICAI: potencial e limitações

Os sistemas inteligentes de ensino (ICAI ou ITS) apresentaram os mesmos problemas que a IA apresentou no seu desenvolvimento, mas continuam a evoluir e possuem uma grande gama de pesquisadores dedicados ao seu aprimoramento. A estrutura destes sistemas fornece um modelo para sistemas computacionais futuros e, como tal, oferecem boas oportunidades de explorar como os estudantes aprendem e como se pode ser mais efetivo ao lecionar. Como quaisquer outros sistemas os ICAI apresentam vantagens e limitações.

A pesquisa instrucional na sala de aula apresenta grandes limitações, mas provavelmente a mais significativa seja a dificuldade em isolar estratégias instrucionais que possam ser repetidas com outros professores e com diferentes alunos sobre várias áreas do conhecimento. Os sistemas CBI tradicionais têm oferecido um ambiente de pesquisa que pode superar alguns destes obstáculos, mas, mesmo aqui, a distinção entre conteúdo, estratégias de ensino e características do estudante muitas vezes torna-se difícil. Talvez a maior contribuição que os sistemas ICAI tenham a oferecer é a habilidade de, sem ambigüidades, isolar cada um dos quatro componentes dos sistemas inteligentes de ensino vistos na figura 6.2 acima [ROB90].

Uma outra vantagem destes sistemas reside em áreas específicas com o um exemplo fornecido pelo sistema GUIDON. Estes sistemas podem simular a resolução de problemas de especialistas em áreas como: análises químicas, diagnóstico médico e diagnóstico de falhas de computadores. Fornecidas as informações relevantes, eles podem, muitas vezes, resolver problemas como um especialista na área o faria. Além disso, uma vez que o problema tenha sido resolvido, eles podem explicitar o método de raciocínio utilizado para chegar àquela conclusão. Isto levou alguns pesquisadores a proclamar que "era apenas mais um pequeno passo" para colocar esta habilidade "para explicar e colocá-la num sistema de ensino" [ROB90]. Na verdade este "pequeno passo" não se mostrou tão pequeno, mas alguns passos significativos têm sido obtidos.

Uma das mais sérias limitações dos sistemas ICAI reside na natureza dos diálogos do estudante com o computador. Enquanto muitos sistemas ICAI recomendam o diálogo em linguagem natural entre o estudante e o computador, muitos dos existentes são consideravelmente mais limitados do que ambientes naturais. Entender a linguagem natural é uma tarefa extremamente complexa e está sendo intensamente pesquisada pela comunidade de IA.

Uma segunda limitação é encontrada na hipótese inerente de que nós podemos entender o que o estudante sabe pela comparação do modelo do estudante com o correspondente modelo do especialista. O problema aqui é que não se sabe realmente muito sobre como as pessoas raciocinam e o modelo especialista pode não ser apropriado para todos os estudantes.

Uma terceira limitação é o desenvolvimento extremamente trabalhoso destes sistemas. A quantidade de tempo e esforço necessários para desenvolver um sistema inteligente de ensino, mesmo um destinado a ensinar uma pequena quantidade de conhecimento, é muito grande.

Uma quarta limitação tem sido os conteúdos escolhidos para implementação. Muitos sistemas ficam restritos a áreas com conteúdos altamente estruturados, como Matemática, Eletrônica e Jogos. Embora o sistema de ensino de Geografia sul-americana de Carbonell [CAR70] tenha mostrado que este não precisa ser necessariamente o caso, a aplicabilidade ampla destes modelos precisa ser demonstrada em outros domínios de conhecimento.

Segundo Grandbastien e Gavignet [GRA94], pode-se dizer que, após anos de pesquisa e de construções de protótipos, nenhum ambiente autoral parece ser largamente aceito ou mesmo estar disponível. Muitos dos protótipos existentes não fornecem algum tipo de ambiente autoral e nenhum deles foi bem sucedido em criar uma casca (*shell*) ITS que fosse amplamente aceita.

6.2.4. Uma alternativa aos sistemas tutores inteligentes

Recentemente uma nova categoria de sistemas inteligentes de ensino foi desenvolvida e ficou conhecida como ambientes inteligentes de aprendizagem ou ILE (*Intelligent Learning Environment*). A diferença entre um ITS tradicional e um ILE está no estilo de interação do tutor. Enquanto no ITS ele é rígido, com a iniciativa e o controle sendo exercidos pelo sistema, no ILE ele é baseado num estilo de interação flexível, com base num paradigma de aprendizagem cooperativa, onde a iniciativa é compartilhada em função de um diálogo de negociação [COS96].

Os ambientes inteligentes de aprendizagem apresentam algumas características que evidenciam a mudança de paradigma em relação aos sistemas tutores inteligentes que é a de promover a aprendizagem através da liberdade de interação do aluno com o conteúdo, onde ele promove o seu próprio conhecimento. Algumas destas características são [McA93]:

- ☐ o estudante aprende construindo o seu próprio conhecimento;
- ☐ o estudante é livre para explorar o ambiente, o tutor é guia e não professor;
- ☐ a personalização do processo é feita pelo usuário e não pelo computador e
- ☐ o conhecimento é obtido pela interação com o sistema e não com o tutor.

Contrastando as características destacadas acima e as do ensino através da hipermídia pode-se perceber uma clara convergência de propósitos. Pode-se desta forma, concluir que um ambiente deste tipo poderá ser implementado com os conteúdos apresentados na forma de um sistema de hipertexto, que é a proposta do presente trabalho.

6.3. Alternativas aos modelos tradicionais

Os sistemas inteligentes de ensino evoluíram substancialmente nos últimos anos. No entanto, o paradigma utilizado não se alterou substancialmente, que é o de tentar

modelar o aluno. Este tipo de sistema pode funcionar bem com certos tipos de conteúdo, mas em muitas disciplinas da área de Ciências Exatas, eles apresentam desvantagens pois não conseguem contemplar a diversidade de níveis de pré-requisitos dos alunos envolvidos. Para este tipo de ensino um sistema aberto em que o aluno tenha liberdade de seguir a seu próprio ritmo é mais adequado. Neste trabalho está se propondo um modelo que contemple esta opção e ao mesmo tempo procure de forma inteligente guiar os esforços do aluno, através da oferta de aplicações dos conteúdos estudados, do reforço conceitual guiado por um sistema multiagente, de arquitetura híbrida.

6.4. O hipertexto e a inteligência artificial

Existe um elo entre os sistemas de hipertexto e os especialistas. Os sistemas especialistas apresentam um mecanismo de inferência e uma base de conhecimentos preenchida por um especialista. Já um sistema de hipertexto é formado por conexões e um mecanismo de navegação e uma base de dados que é produzida pelo especialista. No entanto, o sistema especialista precisa consultar o usuário, isto é, obter dados externos, para fornecer uma resposta, controlando, desta forma, a situação na tomada de decisões. No hipertexto, o controle permanece com o usuário, que é o responsável pela tomada de decisões. Desta forma, o sistema especialista cria um modelo de aluno, enquanto que na hipermídia o aluno cria seu próprio modelo.

As duas abordagens no processo de tomada de decisões requerem um nível diferenciado de rigor na formalização da representação da informação. A representação usada na hipermídia é geralmente a das redes semânticas, enquanto que nos sistemas especialistas é mais comum a representação baseada em regras.

As semelhanças entre os dois tipos de sistemas sugere que uma união entre eles poderia ser proveitosa e tentativas estão feitas para uni-los, de forma a criar um ambiente integrado de ensino/aprendizagem [JON93, JON94, RAD91]. A combinação de um sistema de hipertexto com um sistema especialista foi denominada de hipertexto inteligente (*ExpertText*) [RAD91]. As diversas formas de acrescentar inteligência ao hipertexto são examinadas em [SAN95].

No modelo proposto a adaptação ao usuário é realizada através de um agente que identifica o usuário e decide qual base será a base de conhecimentos a ser utilizada. Assim, se o usuário for um aluno da Engenharia, então ele terá o conteúdo de Probabilidade, por exemplo, proveniente de uma determinada base de conhecimentos. Caso o aluno curse Administração, Economia ou mesmo Ciências Contábeis, então sua base de conteúdos será outra.

Procedimentos mais especializados na forma de lista de exercícios e conteúdos de revisão serão sugeridos com base em outros agentes. Nesta situação pode-se optar por um ensino mais individualizado. Por exemplo, o aluno tem seu perfil determinado inicialmente por um agente, então, com base no seu perfil determinado inicialmente receberá uma proposta inicial de trabalho (exercícios) e dependendo do seu desempenho poderá receber uma lista de conteúdos que devem ser revisados.

6.5. Da IA aos AI

As primeiras idéias de Inteligência Artificial (IA) possuem hoje aproximadamente 40 anos. Nos anos iniciais muito trabalho na área foi dedicado a tentar resolver problemas formais que eram bem estruturados e tinham limites bem definidos [BIG98]. Com a evolução da computação, as áreas consideradas de domínio da IA foram mudando. Uma

nova fase iniciou com o reconhecimento de que muitos projetos foram bem sucedidos quando aplicados a domínios específicos e bem delimitados. Esta abordagem de modelar conhecimento específico resultou numa das mais bem sucedidas áreas da IA: os sistemas especialistas. A partir dos anos 80 pode ser identificado um novo rumo de trabalho, onde o objetivo agora está voltado para a solução de problemas como o processamento e tradução da linguagem natural, controle de robôs, visão e fala de máquinas e raciocínio por senso comum. Um ramo da IA denominado de conexionismo ganhou popularidade e expandiu o leque de opções através do uso das redes neurais, modelagem e controle adaptativo. Algoritmos, como o genético, e sistemas alternativos, como os de lógica difusa, têm sido utilizados para dinamizar o campo da inteligência artificial. O crescimento explosivo da Internet e da computação distribuída levaram a idéia de pequenos programas móveis que podem ser utilizados para executar tarefas úteis para os seus usuários, dando origem aos Agentes Inteligentes (AI).

6.6. Agentes

O termo agente é utilizado com vários significados tanto na IA quanto fora dela. O termo é utilizado na computação tradicional pelas comunidades de base de dados, sistemas operacionais e de rede. Um agente é essencialmente para estes especialistas uma entidade computacional, uma transação, um processo ou um roteador de rede, que permite a realização de uma tarefa. Agentes são, desta forma, uma interface formal para um sistema arbitrário. Além dos vários significados o próprio termo "agente", não é utilizado de forma única. Um outro termo que é encontrado na literatura e que é utilizado no mesmo sentido é "*bot*", que é uma contração da palavra "*robot*" (robô).

6.6.1. Definições de agentes

O que pode ser percebido da literatura na área é que existem quase tantos conceitos quantos são as pessoas que trabalham no assunto. Assim, pode-se definir agente segundo tal ou qual autor. Alguns dos conceitos encontrados são:

"Um **agente** é qualquer coisa que pode vista como **percebendo** seu ambiente através de **sensores** e **agindo** sobre este ambiente através de **modificadores**" [RUS95].

"**Agentes autônomos** são **sistemas computacionais** que habitam algum **ambiente dinâmico complexo**, sentem e agem autonomamente neste ambiente e em fazendo isto **realizam** o conjunto de **objetivos** e **tarefas** para os quais foram projetados"[MAE95].

"Um **agente** pode ser definido como uma **entidade de software persistente** dedicada a um **propósito específico**. A 'persistência' distingue um agente de uma subrotina; agentes possuem suas próprias idéias sobre como desempenhar tarefas, suas próprias agendas. 'Propósito específico' distingue um agente de uma aplicação multifuncional; agentes são tipicamente muito menores" [SMI94].

"**Agentes inteligentes** executam continuamente três funções: **percepção** das condições dinâmicas do ambiente, **ações** para afetar as condições no ambiente e **raciocínio** para interpretar as percepções, resolver problemas, fazer inferências e determinar ações" [HAY95].

"**Agentes inteligentes** são **entidades de software** que executam um **conjunto de operações em benefício de um usuário** ou outro programa com algum grau de **independência** ou autonomia e em fazendo isto, empregam algum **conhecimento** ou representação dos objetivos ou desejos do usuário". [GIL96].

A definição acima é a que mais se aproxima dos objetivos propostos para este trabalho, onde os agentes tem uma tarefa específica que é a de orientar e aconselhar o usuário, bem como registrar o seu desempenho e classificar ou determinar o grau de dificuldade do exercício sendo proposto.

Uma última definição que, na realidade, não define exatamente mas coloca algumas propriedades que um agente deve satisfazer pode ser encontrada em Wooldrige e Jennings [WOO95]:

"Um agente é um sistema de hardware ou mais geralmente de software que apresenta as seguintes propriedades:

- autonomia: agentes operam sem a intervenção humana direta ou outra qualquer e possuem algum tipo de controle sobre suas ações e estados internos;
- habilidade social: os agentes interagem com outros agentes (e possivelmente com o homem) através de alguma linguagem de comunicação;
- reatividade: os agentes percebem seu ambiente (que pode ser o mundo físico, um usuário através de uma interface gráfica, um conjunto de outros agentes, a Internet, ou talvez todos estes fatores combinados) e responde de uma forma temporal a mudanças que ocorrem nele;
- pró-atividade: agentes não somente agem em resposta a seu ambiente, eles também podem exibir comportamento dirigidos por objetivos pela tomada de iniciativas".

6.6.2. Classificação dos agentes

A classificação de agentes também não é uma tarefa simples. Parecem que existem tantas classificações quantas definições. Uma classificação bastante completa pode ser vista em GIESE [GIE98, KER98].

Uma maneira possível maneira de se classificarem os agentes é de acordo com suas características ou dimensões [NWA96].

Uma primeira forma de classificação pode ser dividir os agentes de acordo com suas habilidades para trafegar em uma rede em **estáticos** ou **móveis**. Um estudo mais aprofundado dos agentes móveis poderá ser visto em [HAR95, KNA96].

Uma segunda maneira de classificar agentes é dividi-los quanto a forma de aquisição de inteligência entre **reativos** e **deliberativos**. Um agente reativo, seria segundo Goodwin [GOO93] aquele que responde rapidamente e apropriadamente a alterações em seu ambiente. Já para Nwana [NWA96] um agente reativo caracteriza-se por não possuir um modelo simbólico interno e atuaria segundo o modelo estímulo/resposta a alterações no seu ambiente. Pode-se perceber que o conceito dos dois autores é bastante semelhante. A única diferença é que o primeiro conceito não especifica a forma de reação a alterações do ambiente enquanto que o segundo teoriza que o modelo seria o do estímulo/resposta. Para Goodwin um agente reativo de Nwana seria um agente reflexivo. Já um agente deliberativo seria aquele que possui um modelo de raciocínio e um modelo simbólico interno que é utilizado para planejar e negociar de forma coordenada com outros agentes.

Nwana utiliza, ainda, um terceira forma de classificação que seria de acordo com os atributos mais importantes que ele identifica como sendo a autonomia, a aprendizagem e a cooperação. A **autonomia** seria o princípio de que os agentes podem operar por si só sem a necessidade de um instrutor humano. Os agentes teriam estados individuais internos e objetivos e agiriam de forma a alcançar estes objetivos. A **aprendizagem** seria caracterizada como um aumento do desempenho sobre o tempo. Por fim um agente será

cooperativo se possuir habilidade social, isto é, habilidade para interagir com outros agentes e com o agente humano através de alguma linguagem de comunicação [WOO95]. Os três atributos principais colocados por Nwana quando interseccionados dão origem a quatro tipos de agentes, conforme figura 6.3, que seriam os agentes: colaborativos, inteligentes, de interface e colaborativos de aprendizagem.

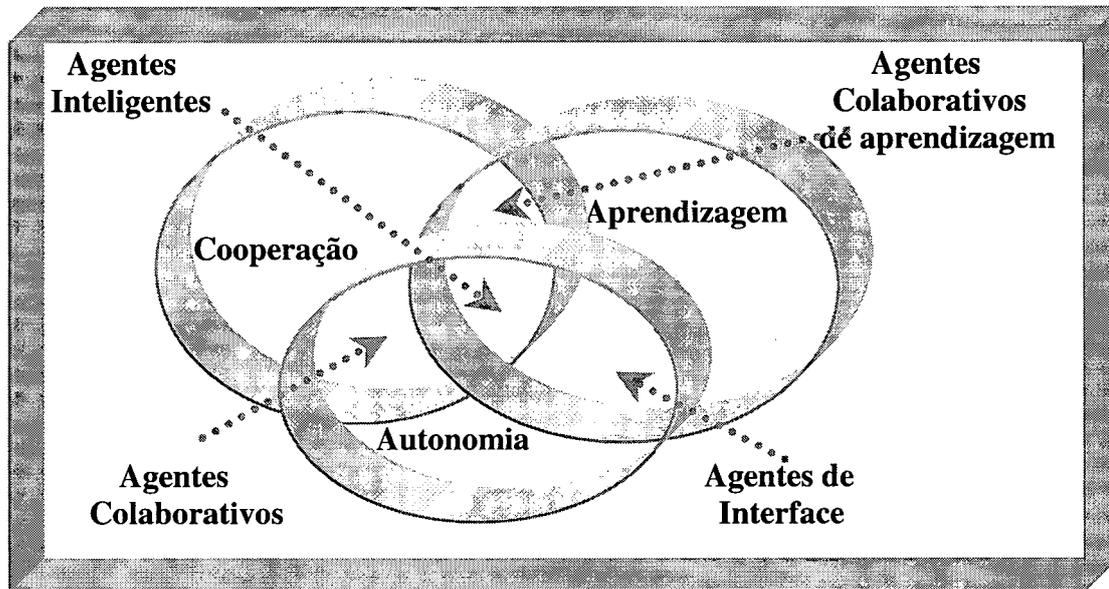


Figura 6.3 - Tipologia dos agentes [NWA96]

Além dos atributos de cooperação, aprendizagem e autonomia considerados por Nwana para definir os tipos de agentes, existem várias outras propriedades que de acordo com diversos autores, caracterizam um agente. A tabela 6.1, relaciona algumas das principais propriedades, ou pelo menos as mais frequentemente encontradas na literatura, bem como suas características e os artigos (ou autores) onde elas foram propostas.

6.7. Inteligência Artificial Distribuída

A Inteligência Artificial Distribuída (IAD) é um campo da Inteligência Artificial que se preocupa com a coordenação e distribuição do conhecimento e ações em ambientes multiagentes [O'HA96]. Ou de acordo com Moulin e Chaib-Draa [MOU96] os pesquisadores da IAD estão procurando entender e modelar ações e conhecimentos em projetos colaborativos. A área da inteligência artificial distribuída pode ser dividida em duas sub-áreas: resolução distribuída de problemas e sistemas multiagentes.

A **Resolução Distribuída de Problemas (RDP)** preocupa-se em de que forma a resolução de um problema, em particular, pode ser subdividida entre um certo número de módulos (ou nodos) cooperativos que compartilhem conhecimento sobre o problema e sobre como chegar a sua solução.

A pesquisa na área de sistemas multiagentes está direcionada a coordenação e o comportamento inteligente de uma coleção de agentes. Em como eles podem coordenar seu conhecimento, objetivos, habilidades e planos de forma conjunta para tomar ações ou para resolver problemas.

Os agentes em um sistema multiagentes podem trabalhar com objetivos individuais, de forma separada, mas interativa ou podem trabalhar com um único objetivo comum.

Da mesma forma que os módulos num sistema RDP os agentes em um sistema multiagentes devem compartilhar o conhecimento sobre o problema e a solução, mas devem, também, raciocinar sobre o processo de coordenação entre eles.

Tabela 6.1 - Caracterização dos agentes

Propriedade	Características	Referências
Aprendizagem ou Adaptabilidade	Adaptação a novas situações pela acumulação de conhecimentos. Um agente pode aprender através de conjunto de regras, por um sistema difuso, por um rede neural ou ainda através de sistemas híbridos neuro-difusos.	[AUE95] [JEN96] [BEL95]
Autonomia ou Independência	Habilidade para executar a maioria das tarefas sem a intervenção direta de humanos ou outros agentes, controlar suas próprias ações internas e seus estados internos.	[JEN95] [JEN96] [FON97]
Colaboração ou Cooperação	Habilidade para trabalhar em conjunto com outros agentes com o objetivo de executar uma tarefa complexa.	[WOO95] [SYC96]
Comunicabilidade ou Habilidade Social	Habilidade para interagir com outros agentes ou humanos como forma de completar suas tarefas ou auxiliar outros agentes quando requerido.	[JEN95] [JEN96] [FRA96]
Flexibilidade	É a habilidade demonstrada por um agente para escolher ações de uma forma dinâmica, bem como escolher a execução destas ações.	[JEN95] [FRA96]
Inteligência	Habilidade do agente para decidir sobre qual ou quais ações tomar, quando confrontado com mais de uma opção viável. O raciocínio pode ser efetuado por regras, conhecimento ou evolução artificial.	[GIL96] [NIS95]
Mobilidade	Possibilidade de que o agente se mova entre máquinas em uma rede, preservando seu estado interno.	[AUE95] [BEL95]
Pró-Atividade	Habilidade para reagir não apenas ao ambiente, mas a de exibir comportamento oportunista e orientado a objetivos, tomando a iniciativa quando apropriado.	[JEN96] [WOO95] [BEL95]
Reatividade ou Sentir e agir	Habilidade para perceber alterações no ambiente (o usuário, outro agente, a Internet, etc.) e responder prontamente a alterações que estejam ocorrendo.	[JEN96] [FRA96] [GOO93]

6.8. Sistema multiagentes

Muitas vezes os agentes são mais úteis agindo em conjunto, como partes de um Sistema MultiAgentes (SMA). As razões para isto são inteiramente práticas e seguem a estratégia de fracionar um problema em subproblemas como uma opção para resolvê-lo.

Um SMA pode ser definido como "uma rede fracamente agrupada de solucionadores de problemas que trabalham de uma forma integrada para resolver um problema que estaria além de suas capacidades individuais". Os solucionadores de problemas são denominados "agentes", são autônomos e podem ser "heterogêneos por natureza", isto é, caracterizados por vários graus de capacidade para a resolução do problema [MOU96].

Em sistemas multiagentes práticos, os agentes podem ser caracterizados por vários graus de capacidade da resolução do problema. A distinção pode ser feita entre agentes reativos, intencionais e sociais.

Um agente **reativo** é aquele que reage a alterações no seu ambiente ou a mensagens vindas de outros agentes. Ele não é capaz de raciocinar sobre suas intenções (manipulação de objetivos). Suas ações são executadas como resultado do disparo de regras ou atualização da base de conhecimentos (ou espaço de crenças) e de enviar mensagens para outros agentes ou para o ambiente.

Um agente **intencional** é capaz de raciocinar sobre suas intenções e crenças, para criar um plano de ações e executar este plano. Em um SMA os agentes intencionais se coordenam pela troca de informações sobre suas crenças, objetivos e ações. Esta informação é incorporada em seus esquemas.

Um agente **social** possui modelos explícitos de outros agentes e é capaz de manter estes modelos (atualizando planos crenças, objetivos e eventualmente planos)

6.9. Arquiteturas de agentes

A estrutura de um sistema multiagentes pode ser classificada em função das ações que serão executadas pelos agentes. Se a ação é feita por deliberação explícita sobre um conjunto de diferentes opções, através da utilização de um modelo interno do ambiente, então a estrutura é denominada de arquitetura **deliberativa**. Se a ação a ser tomada está diretamente ligada a ocorrência de um conjunto de eventos no ambiente, isto é, a ação depende de um conjunto de pré-condições então a arquitetura é dita **não-deliberativa**. Se, por outro lado, as ações forem tomadas de qualquer uma das duas formas acima então a arquitetura é dita do tipo **híbrido**.

6.10. Conclusão

A inteligência nos sistemas de ensino foi acrescentada aos tradicionais sistemas CAI, formando os ICAI ou então os ITS. Devido a sua falta de flexibilidade, recentemente foram criados os ambientes inteligentes de aprendizagem (ILE) em que o tutor oferece maior flexibilidade. Um dos paradigmas que atualmente está encontrando maior aceitação é baseado no conceito de "aprender a aprender", significando que o aluno faz o seu próprio caminho e decide quando, quanto e como estudar. Os sistemas de hipermídia permitem esta flexibilidade.

Na área das disciplinas que envolvam forte base de abstração é imperativo que os conceitos sejam praticados através da resolução de uma grande quantidade de exercícios. Para que o aluno não seja desestimulado, tentando resolver questões para os quais ele ainda não possui habilidades suficientes, é que se está propondo um sistema multiagentes para coordenar e dirigir os seus esforços no sentido de que ele obtenha maior produtividade e que pela graduação do nível de dificuldade se sinta estimulado a prosseguir para níveis mais altos de abstração e dificuldade.



7. O MODELO MULTIAGENTES PROPOSTO

7.1. Introdução

Há diversos tipos de documentos que podem ser encontrados na rede. Um documento pode conter texto, imagem, som, etc. Os navegadores são capazes de exibir apenas textos, mas com várias mídias, normalmente através de recursos adicionais (*plugin*). Entretanto, novos formatos de arquivos são agregados a cada dia. Os navegadores possuem softwares internos para manipular os tipos que conhecem. Mas e os novos tipos? Muitos navegadores suportam a noção de aplicativos auxiliares em que se pode associar um tipo particular de conteúdo ao aplicativo. Por exemplo, a Microsoft definiu um formato de intercâmbio de documentos denominado RTF (*Rich Text Format*). É possível associar o tipo de conteúdo deste arquivo ao MSWord. Assim, quando for feito a baixa (*download*) de um documento contendo texto em RTF, o navegador irá ativar o MSWord com o documento carregado. Isso é bastante cômodo. No entanto, ele abre um aplicativo totalmente novo em uma janela diferente.

Os navegadores não são softwares inteligentes, eles simplesmente solicitam, recuperam e exibem o conteúdo de documentos localizados em vários servidores. Se quiser algo diferente de algum local (*site*), será necessário criar um programa no servidor que modifique o documento antes que ele seja enviado ao navegador. Estas mudanças podem ser feitas por programas executados no próprio servidor ou por *scripts*.

Embora os navegadores sejam programas sofisticados que apresentem muitas funções, eles são essencialmente limitados. Eles recuperam e apresentam páginas da rede e então são encerrados. Os navegadores podem interagir com diversos tipos de servidores: http, ftp, gopher, etc. No entanto, eles são basicamente ferramentas de exibição multimídia. E mesmo esta característica é conseguida graças a softwares adicionais (*plug-ins*) que o navegador normalmente invoca quando é necessário.

Para tornar o navegador mais inteligente, é necessário acrescentar alguns recursos no cliente que permitam que um documento recuperado modifique a si próprio. Desta forma, a ênfase dos programas no servidor recai sobre o cliente, basicamente fortalecendo o navegador e tornando-o mais inteligente.

Um recurso disponível no cliente é uma linguagem de *script*. Desta maneira, um navegador pode lidar de maneira inteligente com situações que de outra forma iria exigir um programa no servidor. Além disso o usuário pode perceber que a situação é tratada com muito mais rapidez, porque o navegador não precisará enviar a solicitação ao servidor e esperar para exibir a resposta.

No entanto, esta maneira de resolver o problema requer que o professor que irá elaborar as páginas com os conteúdos sendo transmitidos conheça, além da html, uma linguagem de *script*. Além disso, o que pode ser feito através de uma linguagem de *script* é bastante limitado, em termos de programação e controle sobre a forma de apresentação. As particularidades de muitas disciplinas da área de Ciências Exatas, especialmente a Estatística e a Probabilidade, exigem programas bastante complexos que estão além do que pode ser obtido por este recurso. Uma outra abordagem seria a programação através

da linguagem Java com a utilização de applets. Novamente recai-se na exigência da competência em programação de quem precisa elaborar os conteúdos instrucionais. Apesar da propalada facilidade, a linguagem Java requer um conhecimento que poucos profissionais possuem. Em virtude destas dificuldades foi que se optou por um modelo que permita embutir no próprio navegador os recursos básicos para se lecionar este tipo de disciplina.

Apresentar páginas html interativas e mesmo com variados recursos de multimídia não é o suficiente para garantir o aprendizado efetivo nestes tipos de disciplinas. Para que a aprendizagem seja efetiva é necessário que o aluno solidifique os conceitos, praticando os mesmos através da aplicação a outras situações, ou seja, transferindo o conhecimento adquirido a casos não vistos. A maneira tradicional de se fazer isto é através da resolução de exercícios. Quanto mais a prática for efetivada maior a retenção e o entendimento dos conceitos, uma vez que resolver exercícios, neste caso, significa na realidade demonstrar as seguintes habilidades:

- dominar a simbologia específica de cada disciplina (linguagem)
- dominar os termos, conceitos e definições envolvidos (jargão);
- desconhecer o modelo subjacente, isto é, identificar qual ou quais são os conceitos envolvidos;
- transformar a linguagem comum (ordinária) na simbologia apropriada e
- manipular simbolicamente o modelo para obter uma solução;

As duas primeiras habilidades podem ser obtidas de forma satisfatória com a leitura atenta dos conteúdos, evidentemente com maior ou menor facilidade conforme as mídias e as formas de apresentação utilizadas. A terceira habilidade poderá ser atingida com uma gama variada de exemplos e com a interatividade fornecida através do computador. No entanto, as duas últimas habilidades só serão efetivamente desenvolvidas com a prática contumaz da resolução de exercícios; quanto mais esta prática for estimulada maior será a competência adquirida.

Entretanto, a resolução de exercícios poderá ser uma tarefa frustrante e mesmo desestimuladora se os mesmos estiverem em descompasso com a habilidade do aluno. Para que esta prática seja efetiva, é necessário uma dosagem cuidadosa para que as primeiras tentativas sejam bem sucedidas e desta forma estimulem o prosseguimento para novos patamares de dificuldades que requerem graus crescentes de abstração, envolvendo novas competências através de modelos mais complexos.

Nas disciplinas da área de Ciências Exatas que requerem uma formação anterior sólida para que novos conceitos e novos níveis de competência sejam alcançados é particularmente importante que seja detectado o nível de conhecimento inicial do aluno, isto é, que seja diagnosticado o estado dos pré-requisitos. Em função deste diagnóstico poderá ser oferecido uma série apropriada de exercícios que estejam em correspondência com o que ele efetivamente possa resolver, além, é claro, de ser possível sugerir conteúdos que devam ser revisados para um melhor aproveitamento dos novos conhecimentos.

7.2. A estrutura do modelo

A arquitetura do modelo é composta de cinco tipos de agentes. O primeiro tipo é um agente denominado de "Identificador de Usuários" (IU). Ele é o encarregado de identificar o tipo de usuário do sistema, verificando se o mesmo já está cadastrado e, portanto, já apresenta um perfil ou se é um usuário novo que precisa ser cadastrado e ter seu perfil definido. Se o agente IU identificar um usuário novo ele passa a informação

para o agente "Cria Perfil" (CP) que realiza o cadastramento do usuário e se encarrega de criar um perfil inicial do novo usuário. O agente IU também realiza a tarefa de selecionar as páginas html que o usuário deve receber de acordo com o curso identificado.

Uma vez que o agente IU tenha identificado o usuário ou o agente CP tenha criado um novo perfil, é acionado o agente "Perfil do Usuário" (PU), que é o encarregado de manter e atualizar o perfil de cada usuário do sistema. Este é um agente do tipo difuso que a partir do perfil inicial do usuário será encarregado de registrar as alterações ocorridas a cada vez que o usuário utilizar o sistema. O agente utilizará um modelo neuro-difuso, tipo NEFCLASS [NAU95] para a classificação dos usuários, isto é, para manter e atualizar o perfil de cada usuário.

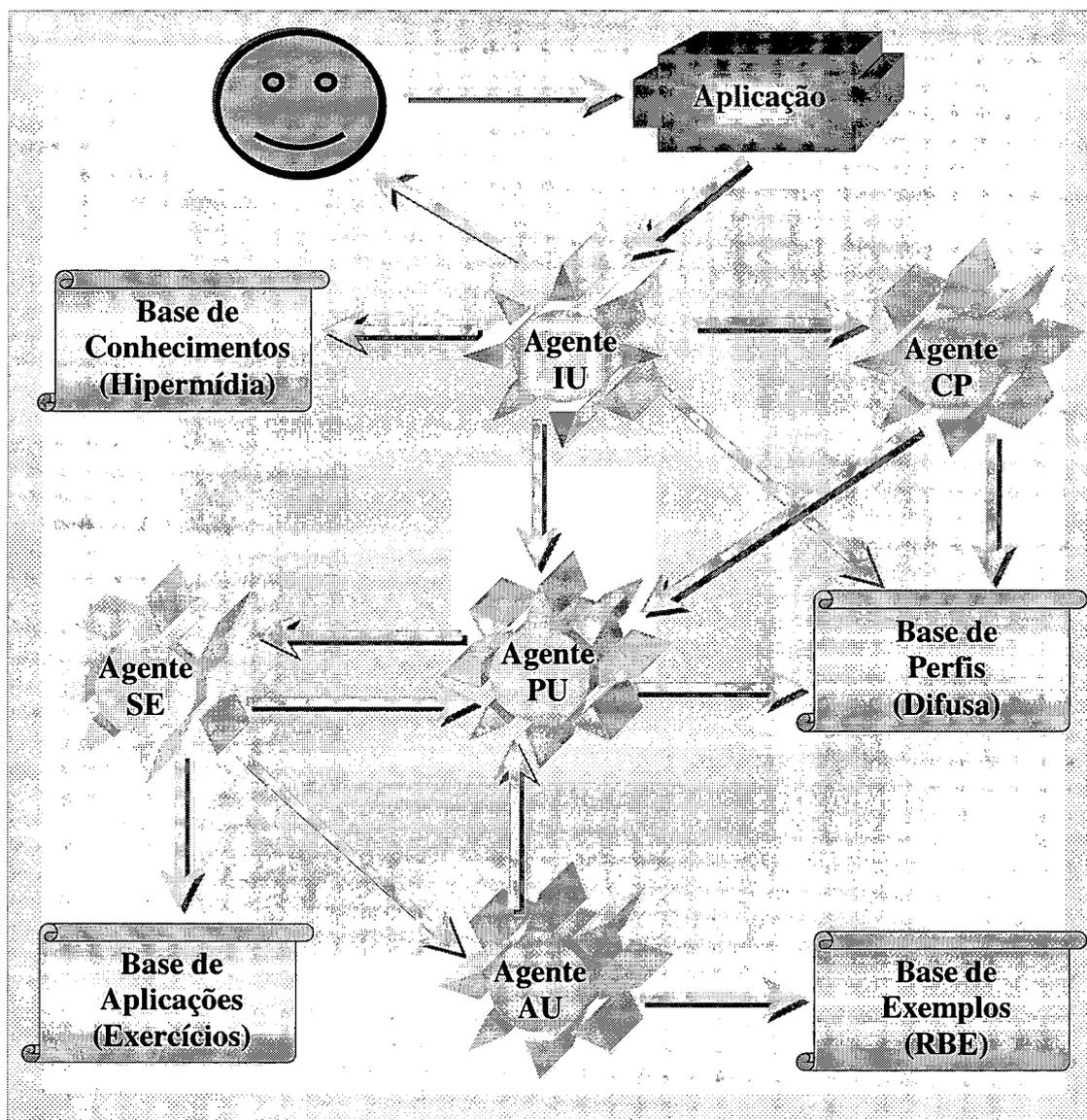


Figura 7.1 - Estrutura do modelo multiagentes proposto

O agente PU interage de forma bidirecional com a agente "Sugere Exercício" (SE), que é o responsável por manter um registro da dificuldade de resolução de cada exercício e sugerir o adequado, de acordo com o perfil do usuário. Este agente também utilizará um classificador neuro-difuso para manter e atualizar a base de aplicações.

Caso o usuário não consiga solucionar uma determinada aplicação após um certo tempo ou após um determinado número de tentativas o agente PU invoca o agente "Auxílio ao Usuário" (AU), que deverá fornecer alguma pista ou dica adicional para a solução do problema. Caso a base de pistas (dicas) para esta aplicação em particular esteja vazia, então o agente irá procurar na base de exemplos um que seja o mais próximo possível do exercício sendo proposto e apresenta ao usuário. Se mesmo assim o usuário não conseguir acertar o exercício então o agente irá sugerir conteúdos de revisão ao usuário ou um conteúdo específico do domínio da aplicação que deva ser relido ou estudado novamente.

7.3. Caracterização dos agentes

Os agentes do sistema multiagentes proposto acima agem em conjunto no auxílio do usuário, executando as seguintes ações: identificação do usuário do sistema, sugestão de roteiro de leitura, determinação do perfil do usuário, promovendo e demovendo o usuário em função do nível de acertos, hierarquização da base de conhecimento (exercícios) em função da dificuldade de resolução e, por fim, sugestão de casos resolvidos (exemplos) para facilitar o entendimento e a resolução do problema. A seguir é feito o detalhamento de cada agente e a função que cada um desempenha no modelo.

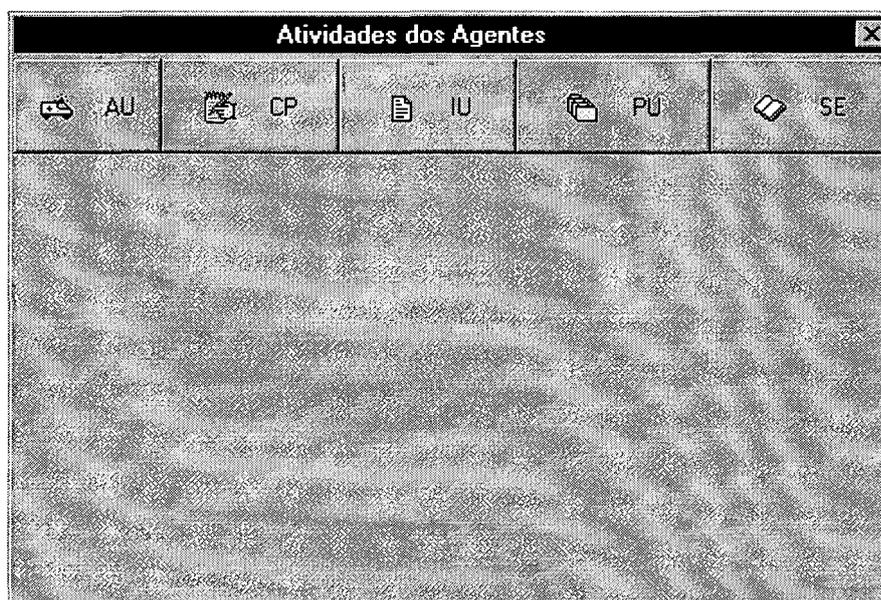


Figura 7.2 - Janela de observação das atividades dos agentes

7.3.1. O agente "Identificador do Usuário" (IU)

Quando o usuário executar a aplicação, ela invocará o agente IU que solicitará sua senha de acesso e seu curso. Com base na senha e no curso, o agente verifica se o usuário já está cadastrado no sistema. Caso isto ocorra, o agente então com base na informação sobre o curso do usuário fornece o conteúdo adequado (páginas html). Por exemplo, se o usuário estiver matriculado num curso de Engenharia, então o agente envia determinado tipo de arquivo, se ele for de um curso da área de humanas, então os arquivos serão de outro tipo. Existe a possibilidade de que este agente mantenha um

histórico das páginas visitadas pelo usuário e possa sugerir novas páginas a serem visitadas, de acordo com a estrutura do curso do aluno. Se o agente identificador do usuário não encontrar o nome do usuário no cadastro, então ele presume que seja um novo usuário e envia uma mensagem para o agente "Cria Perfil" (CP).

7.3.2. O agente "Cria Perfil" (CP)

O agente "Cria Perfil" (CP) é encarregado de cadastrar o usuário no sistema, criando uma senha, registrando o nome e o curso no banco de dados de usuários e, como função principal, obter o perfil inicial do aluno. Este perfil poderá ser traçado através de um teste inicial de questões básicas, envolvendo pré-requisitos que o aluno deva possuir para ser bem sucedido no curso e resolver os exercícios com certa competência. Com base nos acertos obtidos, que podem variar numa escala de "0" a "10", o usuário poderá receber uma classificação difusa do tipo: "fraco" (0 a 4 acertos), "médio" (de 3 a 6 acertos), "bom" (de 5 a 8 acertos) e "ótimo" de (7 a 10 acertos). A figura 7.3 fornece uma idéia desta classificação.

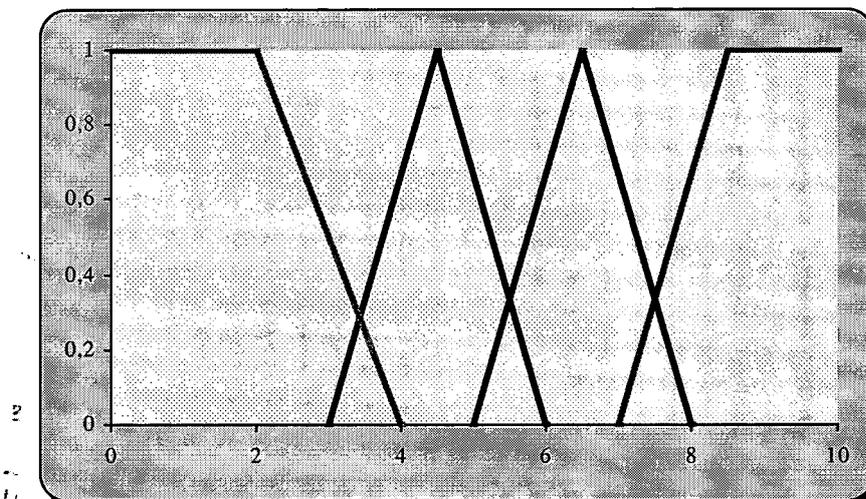


Figura 7.3 - Avaliação difusa do perfil do usuário

7.3.3. O agente "Perfil do Usuário" (PU)

O agente "Perfil do Usuário" obtém o perfil inicial através do agente CP e a cada seção se encarregará de atualizar o perfil do usuário em função do seu desempenho, caso este realize uma seção prática (exercícios). Assim, se o usuário inicia com um perfil igual a "fraco" e no conjunto de exercícios para este nível ele apresenta um desempenho "bom", então o seu perfil será promovido para "médio", digamos, através da operação difusa de "interseção" de perfis. Este agente utilizará um modelo neuro-difuso de inteligência (seção 7.4). Inicialmente será utilizado o modelo encontrado em [NAU95] e posteriormente um modelo próprio deverá ser proposto e implementado.

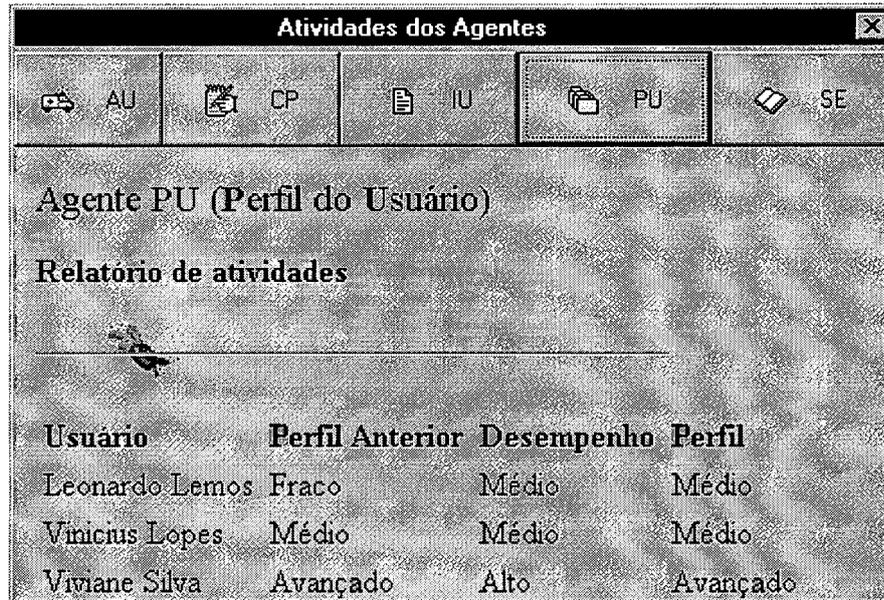


Figura 7.4 - Janela de registro de atividades do agente "perfil do usuário"

7.3.4. O agente "Sugere Exercício" (SE)

Em função do perfil do usuário, este agente apresenta um conjunto de aplicações adequados ao este usuário. A adequação da aplicação será feita através de um índice de dificuldade I_D que será definido da seguinte forma:

$$I_D = 1 - \frac{\sum S}{\sum T}, \text{ onde } S = \text{Número de sucessos (tentativas bem sucedidas) e}$$

$T = \text{Total de tentativas. Assim por exemplo se 10 usuários tentarem resolver a aplicação 1 e os 10 tiverem sucesso o } I_D \text{ será igual a } 1 - 10/10 = 0. \text{ Por outro lado uma aplicação terá o grau de dificuldade máximo se de 10 usuários que tentarem resolvê-la nenhum tiver sucesso, neste caso, } I_D = 1 - 0/10 = 1.$

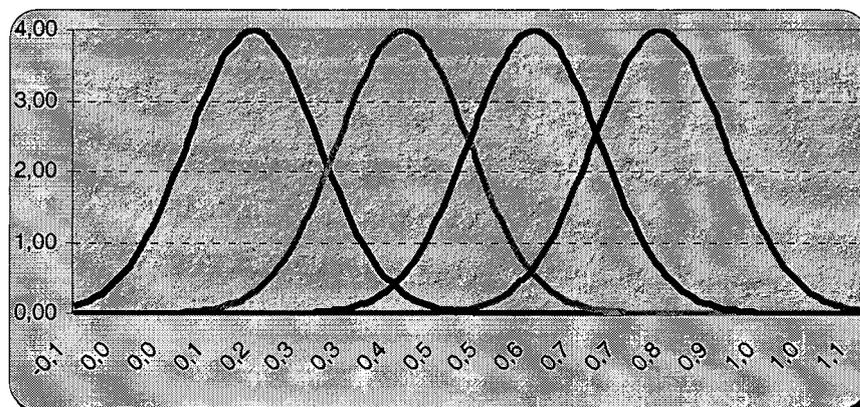


Figura 7.5 - Classificação das aplicações

A alteração do grau de dificuldade das aplicações será feita de forma difusa, utilizando-se uma curva normal de média igual a 0,2 para os exercícios considerados "fáceis", de média igual a 0,4 para os considerados "médios", de média igual a 0,6 para

os considerados "difíceis" e de média igual a 0,8 para os considerados "avançados". O desvio padrão de cada uma das curvas será igual "um". A figura 7.5 fornece uma idéia de como a classificação será executada.

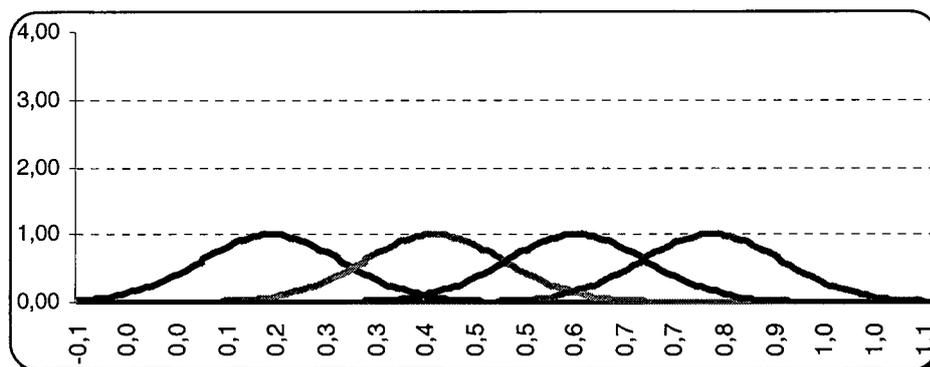


Figura 7.6 - Classificação das aplicações normalizada

O conjunto difuso acima não está normalizado. Na figura 7.6 pode-se ver este conjunto já normalizado.

As regras e alterações dos limites iniciais propostos acima deverão ser ajustadas a cada área de ensino. Assim, por exemplo, alunos de Engenharia terão limites e regras diferentes de alunos de Ciências Contábeis e Administração. Como cada turma possui um perfil próprio, a adequação dos limites definidos acima, isto é, das regras de classificação, serão executadas por um modelo difuso. Só que modelos difusos não possuem mecanismos de aprendizagem, então ele será agregado de um mecanismo de aprendizagem composto de uma rede neural de 3 camadas, formando um modelo neuro-difuso.

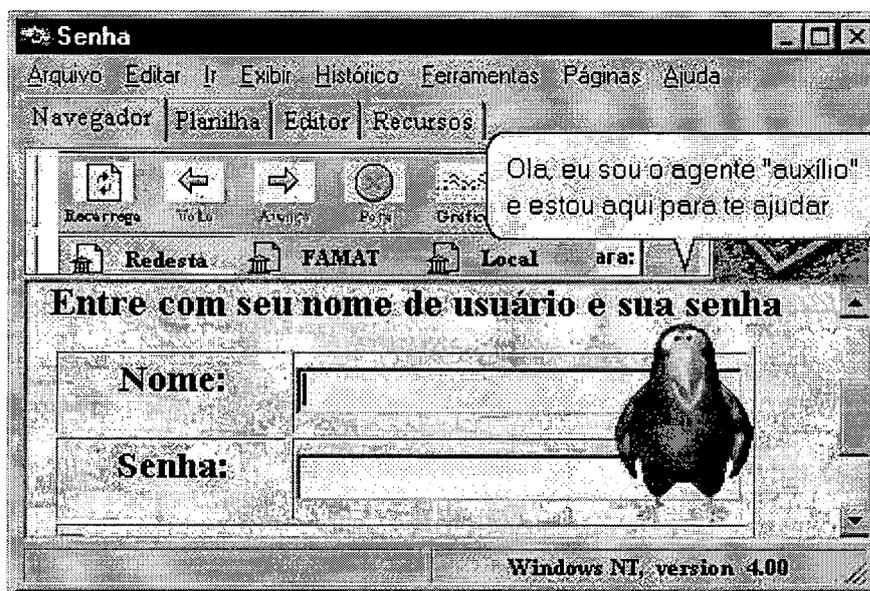


Figura 7.7 - Interface do agente AU, obtida através do MSAGENT

7.3.5. O agente "Auxílio ao Usuário" (AU)

O agente "Perfil do Usuário", PU, acionará o agente AU (Auxílio ao Usuário) sempre que o usuário tentar um determinado número de vezes uma certa aplicação e não encontrar a resposta correta. Neste caso o agente AU pesquisará a sua base de conhecimentos e verificará se existe alguma sugestão (dica) para a resolução da aplicação. Caso exista, ele a recuperará e enviará ao usuário. Caso não exista, ele então pesquisará numa base de exemplos um que esteja mais próximo da aplicação ativa. Este agente será implementado através da técnica RBE (**R**aciocínio **B**aseado em **E**xemplos) [AAM94].

7.4. Modelos neuro-difusos

As pesquisas em redes neuronais e sistemas difusos avançaram de forma independente nos últimos 30 anos [MAG97]. Isto é explicado pelas diferentes origens e motivações dos dois paradigmas. Os sistemas difusos adquirem conhecimento a partir de especialistas num determinado domínio e isto é representado em termos de conjuntos de regras Se-Então. Os sistemas difusos empregam esta abordagem baseada em regras e raciocínio interpolativo para responder a novas entradas. Em contraste, as redes neuronais oferecem uma arquitetura altamente estruturada, com capacidades de aprendizagem e generalização. Uma rede neural armazena seu conhecimento de uma forma distribuída dentro de suas ponderações que foram determinadas pelo treinamento (aprendizagem) com amostras conhecidas. A habilidade de generalização para novas entradas é então baseada na estrutura algébrica inerente. As duas tecnologias possuem pontos em comum e arquiteturas híbridas neuro-difusas estão crescendo substancialmente.

Esses sistemas são geralmente representados como redes neuronais multi-camadas *feedforward*. Nestes modelos, os pesos de conexão e as funções de ativação diferem das redes neuronais comuns. Embora existam muitas abordagens diferentes, o termo neuro-difuso é restrito a sistemas que possuem as seguintes propriedades [NAU97]:

- ☐ treinamento por um algoritmo de aprendizagem derivado da teoria das redes neuronais. O procedimento (heurístico) de aprendizagem opera em informações locais e causa somente modificações locais no sistema difuso subjacente. O processo de aprendizagem é dirigido por dados e não baseado em conhecimento;
- ☐ poder ser visto como uma rede neuronal de três camadas *feedforward*. As unidades da rede utilizam t-normas e t-conormas ao invés das funções de ativação comuns das redes neuronais. As variáveis de entrada constituem a primeira camada, as regras difusas formam a camada (oculta) intermediária e as variáveis de saída formam a terceira camada. Os conjuntos difusos são codificados como pesos de ligação (difusos);
- ☐ poder sempre (isto é, antes, durante e depois da aprendizagem) ser interpretado como um sistema de regras difusas sendo tanto possível criar o sistema de saídas a partir do treinamento por dados quanto inicializá-lo através de conhecimentos prévios na forma de regras difusas;
- ☐ o processo de aprendizagem leva em consideração as propriedades semânticas do sistema difuso subjacente que resulta em restrições nas modificações possíveis aplicáveis sobre os parâmetros do sistema e
- ☐ aproximar uma função n-dimensional (desconhecida) que é parcialmente fornecida pelos dados de treinamento. Os dados difusos codificados no sistema

representam amostras vagas e podem ser vistos como protótipos vagos dos dados de treinamento.

A lógica difusa tem sido aplicada com sucesso em muitas áreas onde abordagens através de modelos convencionais apresentam dificuldades ou custos elevados. No entanto, à medida que a complexidade do sistema aumenta, regras difusas confiáveis e funções de pertinência utilizadas para descrever o comportamento do sistema tornam-se difíceis de determinar. Além disso, devido ao dinamismo de muitas aplicações, as regras e as funções de pertinência devem ser adaptativas para se adequarem ao ambiente de modo a continuarem sendo úteis. Um modelo neuro-difuso é um modelo difuso que utiliza um algoritmo de aprendizagem supervisionado ou por reforço derivado ou inspirando por um rede neuronal para determinar seus parâmetros (os conjuntos difusos e as regras difusas), através do processamento de dados amostrais.

Um sistema híbrido neuro difuso combina as vantagens dos sistemas difusos, que tratam com conhecimento explícito que pode ser explicado e entendido, com as redes neurais, que lidam com conhecimento implícito que pode ser obtido por aprendizagem. As redes neuronais podem ajustar o conhecimento especializado e gerar automaticamente regras difusas adicionais e funções de pertinência de forma a cumprir especificações e reduzir o tempo e custos de projeto. Por outro lado a lógica difusa aumenta a capacidade dos sistemas de redes neuronais fornecendo saídas mais confiáveis quando é necessário se fazer extrapolação além dos valores utilizados no treinamento.

A capacidade de aprendizagem das redes neuronais fizeram dela um recurso bastante apropriado para serem combinadas com sistemas difusos, com o propósito de automatizar ou fornecer suporte ao processo de desenvolvimento de um sistema difuso para uma tarefa específica. As primeiras aplicações dos modelos denominados de neuro-difusos foram feitas na área do controle difuso [NAU94], mas atualmente esta abordagem é aplicada em outros áreas além da de controle, como na análise e classificação de dados [NAU95], na medicina [TEO98], na área financeira [CON96], só para citar algumas.

7.4.1. Arquitetura do modelo neuro-difuso

Um sistema deste tipo agrupa componentes de um sistema difuso tradicional, só que cada estágio é executado por uma camada de neurônios ocultos e a capacidade de aprendizagem da rede é utilizada para melhorar o conhecimento do sistema. A figura 7.8 apresenta um diagrama de um sistema com estas características.



Figura 7.8 - Arquitetura simplificada de um sistema híbrido neuro-difuso

Camada de fuzificação. Cada neurônio nesta camada representa uma entrada da função de pertinência do antecedente da regra difusa. Uma forma de implementar esta camada é expressar cada função de pertinência como pontos discretos. Desta maneira a regra difusa "Se X_1 é difícil e X_2 é avançado então Y é avançado". Este método pode aproximar muitas funções contínuas e o erro depende do número de pontos utilizados.

No entanto, uma maneira melhor é utilizar uma combinação de funções normais, conforme proposto em Mizumoto [MIZ97], e uma função linear para representar cada função de pertinência nas camadas de *fuzificação* e *defuzificação*. Os parâmetros dos neurônios (retângulos sombreados na figura 7.9) podem ser treinados para ajustar a formá e a localização final de cada uma destas funções. O número de neurônios nesta camada pode ser fixo. No entanto, é possível adicionar ou remover neurônios durante o treinamento, de acordo com as saídas produzidas pelas amostras de treinamento.

A coluna "Grau" na figura 7.9 representa o grau de certeza das afirmações: "X₁ é fácil" que, no caso, é 0,2, "X₂ é médio", que aqui, vale 0,4, e assim por diante. As saídas destas funções de pertinência estão conectadas a camada de regras difusas através de conexões com pesos fixos e todos iguais a unidade.

Camada de regras difusas. Esta camada é formada pela base de regras difusas e seu propósito é executar as operações lógicas difusas. Cada neurônio representa uma regra difusa e calcula a certeza de cada proposição do tipo "Se X₁ é fácil e X₂ é médio então Y é fácil", que avalia até que ponto os requisitos de cada regra difusa estão satisfeitos. Os neurônios podem ter funções lineares, por simplicidade, e suas saídas estão conectadas à camada de *defuzificação* através de conexões ponderadas. Estes pesos representam a significância relativa das regras associadas aos neurônios. Os valores podem ser pré-fixados pelo especialista ou inicializados para assumirem o valor 1,0 e então treinados para refletir a importância real da função de pertinência contida na camada de *defuzificação*.

Camada de defuzificação. O objetivo desta camada é a avaliação das regras. Cada neurônio nesta camada representa uma proposição conseqüente e a função de pertinência pode ser implementada da mesma forma que na camada de *fuzificação*, através de uma ou duas funções gaussianas. A certeza de cada proposição conseqüente é calculada com respeito a aderência a aquelas regras que possuem as mesmas proposições conseqüentes. Os pesos de cada conexão de saída destes neurônios representam o centro de gravidade de cada função de pertinência da saída do conseqüente e podem ser treinados. O valor final da saída é calculado, então, utilizando o método do centro de gravidade.

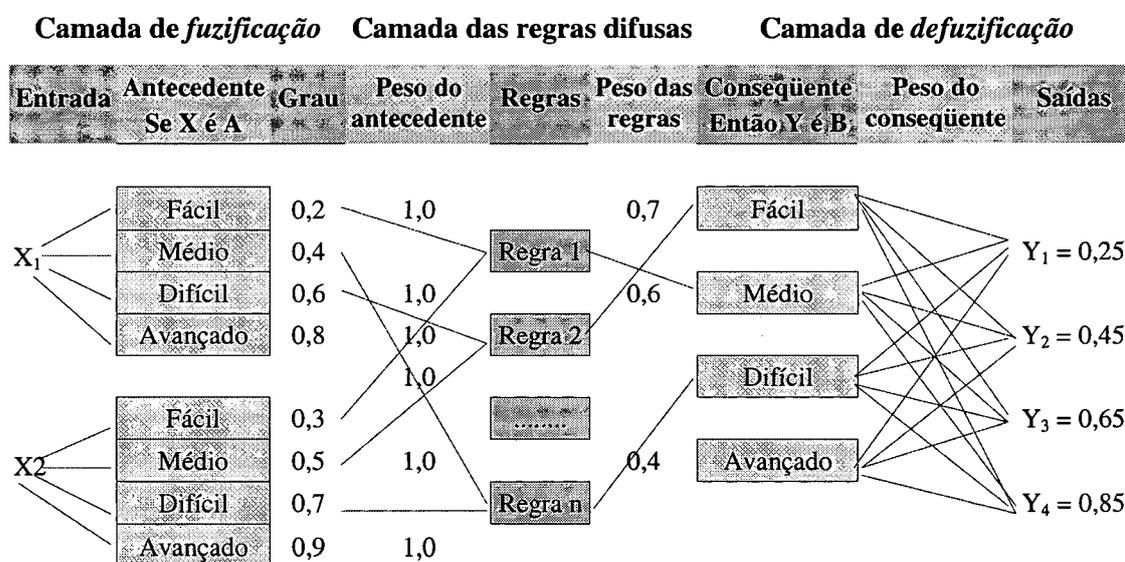


Figura 7.9 - Uma visão mais detalhada do sistema de inteligência dos agentes

A opção vista é uma das muitas maneiras de se combinar redes neuronais e técnicas difusas como forma de melhorar e ajustar velocidade de aprendizagem e pode variar dependendo dos propósitos da aplicação. Novas tecnologias como os algoritmos genéticos, também podem e estão sendo integrados com o objetivo de melhorar o desempenho dos sistemas híbridos.

7.4.2. Classificadores neuro-difusos

As abordagens tradicionais para a classificação de dados envolvem algumas técnicas estatísticas (análise de conglomerados e regressão logística) e as redes neuronais. No entanto, estas técnicas são limitadas pois sofrem da falta uma forma efetiva de definir os limites entre as categorias obtidas. Por outro lado, a classificação difusa assume que duas categorias vizinhas possuem uma área contínua de sobreposição dentro da qual um objeto é potencialmente membro de cada uma das categorias. Este ponto de vista não apenas reflete a realidade de muitas aplicações nas quais as categorias possuem limites difusos, mas fornece também uma representação simples da potencialmente complexa partição do espaço de características. As vantagens da utilização de um modelo neuro-difuso na classificação de dados podem ser enumeradas como:

- conhecimento vago pode ser utilizado;
- o classificador é interpretado na forma de regras lingüísticas e
- o classificador é fácil de implementar, usar e entender.

Um sistema neuro-difuso pode ser utilizado para a classificação com regras do tipo se x_1 é μ_1 e x_2 é μ_2 e x_n é μ_n então o padrão (x_1, x_2, \dots, x_n) pertence a classe i , onde $\mu_1, \mu_2, \dots, \mu_n$ são conjuntos difusos.

A base de regras de um classificador difuso que usa regras do tipo acima, representa uma aproximação de uma função (desconhecida) $f: \mathcal{R}^n \rightarrow [0, 1]^m$ que representa a tarefa de classificação, onde $f(x) = (c_1, c_2, \dots, c_m)$ tal que $c_i = 1$ e $c_j = 0$, onde $j \in [1, \dots, m], j \neq i$, isto é, x pertence a classe C_i .

Os classificadores são geralmente derivados dos dados e não são especificados diretamente. Existem dois métodos utilizados pelos classificadores difusos:

- o conglomerado difuso e
- a aprendizagem neuro-difusa.

No caso de conglomerados difusos o espaço de entrada é vasculhado com o intuito de formar grupos. O número de grupos é determinado por uma medida de avaliação e o tamanho e a forma do grupo (*cluster*) é determinado pelo algoritmo de aglomeração. O desempenho de um classificador por conglomerados é reduzida uma vez que ele é expresso na forma de regras difusas [NAU97].

Uma alternativa para se obter um classificador difuso a partir de dados é utilizar a abordagem neuro-difusa. Isto significa que o classificador é obtido a partir dos dados por um procedimento de aprendizagem heurístico. Esta abordagem é computacionalmente mais efetiva que a abordagem de conglomerados em virtude de sua simplicidade [NAU97]. Este modelo servirá de inteligência aos agentes "PU - Perfil do Usuário" e ao "SE - Sugere Exercício".

Inicialmente, será utilizado o classificador NEFCLASS (*Neuro Fuzzy Classification*) proposto por NAUCK [NAU95], nos dois agentes classificadores definidos como forma de diminuir o tempo de desenvolvimento do protótipo. Mais tarde poderá ser desenvolvido algoritmos próprios e específicos para cada um dos agentes.

Tabela 7.1 - Características dos agentes propostos

Propriedades	Agentes				
	IU	CP	PU	SE	AU
Aprendizagem		✓		✓	
Autonomia	✓	✓	✓	✓	✓
Colaboração	✓	✓	✓	✓	✓
Comunicabilidade	✓	✓	✓	✓	✓
Flexibilidade		✓		✓	
Inteligência	✓	✓		✓	✓
Mobilidade					✓
Pró-Atividade	✓				✓
Reatividade	✓	✓	✓	✓	✓

7.5. Capacidades dos agentes propostos

No modelo de agentes proposto existem agentes com arquiteturas e capacidades variadas. Alguns dos agentes são bastante simples enquanto outros bastante sofisticados, como, por exemplo, os agentes PU (Perfil do Usuário) e SE (Sugere Exercício), que utilizam um modelo neuro-difuso de aprendizagem e classificação dos usuários e do nível dos exercícios sugeridos aos usuários. A tabela 7.1 apresenta a relação das propriedades de cada um dos agentes sendo propostos.

7.6. Conclusão

Existem muitos sistemas que objetivam apoiar o ensino de um leque variado de disciplinas. Grande parte deles se limita a fornecer conteúdos ou testes ao usuário sem levar em consideração o perfil deste usuário, isto é, sem avaliar a capacidade efetiva do usuário para entender ou lidar com a situação sendo proposta. Por outro lado, as aplicações ou exercícios sugeridos, podem ser apresentados aleatoriamente ou então num esquema gradativo que não leva em conta se o usuário de fato já adquiriu habilidades suficientes para lidar com o nível de abstração sendo proposto.

O modelo apresentado aqui lida com estas situações de uma forma inovadora, propondo a promoção ou não do usuário em função do seu desempenho atual e passado. Da mesma forma a base de aplicações vai sendo atualizada em função do acertos ou desacertos dos diversos usuários do sistema, mantendo assim uma classificação realista e adequada ao perfil de cada curso. Com o modelo proposto o aluno terá exercícios na medida certa da sua capacidade de entendimento, melhorando desta forma o seu rendimento e estimulando-o a prosseguir em busca de níveis de dificuldade cada vez maiores.

Nas situações em que o usuário se encontrar num impasse sem conseguir prosseguir um agente que monitora seus movimentos vai sugerir dicas ou exemplos semelhantes a aplicação sendo trabalhada de forma a estimulá-lo a completar a tarefa proposta. O modelo embora dirigido permite que o usuário tente, se se sentir em condições, a resolução de exercícios com classificação mais alta, permitindo desta forma que ele pule etapas e chegue aos objetivos propostos mais rapidamente.



8. A IMPLEMENTAÇÃO DO MODELO

8.1. Introdução

Esta talvez seja a etapa mais problemática para a viabilização de um modelo de multiagentes. Não pela falta de opções, mas sim pelo excesso delas. O número de alternativas disponíveis é tão grande e diversificada que a tomada de uma decisão nem sempre poderá ser feita sem uma análise prolongada e cuidadosa de todas as alternativas disponíveis.

As possíveis alternativas para a implementação de um modelo multiagentes envolvem considerações sobre três categorias de caminhos:

- ❶ utilização de uma camada intermediária de software;
- ❷ utilização de uma linguagem especificamente projetada e
- ❸ utilização de uma linguagem de propósito geral.

8.2. Camadas de software

Existem diversas formas para implementar uma plataforma multiagentes. Uma opção é o desenvolvimento de agentes sobre camadas de software projetadas para facilitar o desenvolvimento e a implementação e que rodam sobre um sistema operacional, ou seja, são extensões do sistema operacional com o propósito específico de gerenciamento, desenvolvimento, coordenação e comunicação entre agentes. A tabela 8.1 apresenta algumas destas opções. A utilização de um destes ambientes pode permitir alguma flexibilidade na escolha da linguagem de programação ou então forçar o desenvolvimento em uma linguagem específica (a Java, por exemplo).

A grande virtude desta forma de implementação é a facilidade de criação e manutenção dos agentes, pois todo o trabalho miúdo e pesado de programação já foi realizado, sendo necessário apenas uma adaptação às características do ambiente sendo utilizado. Esta opção, por um lado viabiliza alguns projetos e reduz consideravelmente o ciclo de desenvolvimento de outros, mas por outro lado limita drasticamente a flexibilidade do projetista. A limitação decorre do fato de que a maioria destes ambientes exige a programação em uma linguagem específica e o sistema resultante só poderá rodar em plataformas que possuem esta camada adicional de software instalada ou nas quais este ambiente também rode.

O sistema denominado "Aglets" da empresa IBM é uma destas camadas de software que foi desenvolvida em Java e que requer que os agentes também sejam programados nesta linguagem. Situação semelhante ocorre com o Concordia da empresa Mitsubishi Electric. Menos mal já que a Java é uma linguagem que vem ganhando popularidade rapidamente. Já o ambiente Mozart desenvolvido por um consórcio de instituições requer que os agentes sejam programados em uma linguagem própria denominada de Oz. Os exemplos citados acima são bastante restritivos pois fornecem apenas uma linguagem como opção de programação, o mesmo não ocorre com o projeto TACOMA que oferece suporte para seis diferentes linguagens, entre elas o C++ e o

Visual Basic e roda nas plataformas Windows 98 e NT e Unix. De qualquer forma se por um lado estes ambientes agilizam parte da programação, por outro lado será necessário estudá-los e entendê-los e possivelmente aprender a linguagem suportada para se poder fazer uso deles.

Tabela 8.1 - Camadas de software para a implementação de agentes

Módulo	Características
Aglets	Módulo para o desenvolvimento de agentes da empresa IBM. Foi desenvolvido em Java [CHA96].
Concordia	Esforço do laboratório de sistemas Horizonte da empresa Mitsubishi Electric Information Center America. É desenvolvido inteiramente em Java e apresenta versões para rodar na plataforma Windows (95, 98 e NT) e para o sistema Solaris. Possui classes específicas de agentes permitindo o desenvolvimento dos mesmos de forma simples [WON97].
Inferno	Camada de software da Lucent Technologies, o laboratório de pesquisa dos laboratórios Bell. É uma linguagem para aplicações distribuídas que inclui protocolos e APIs, isto é, um sistema completo [PAR98].
Mole	Módulo de desenvolvimento de agentes que utiliza a linguagem Java. Está sendo desenvolvido pelo grupo de Sistemas Distribuídos da universidade de Stuttgart (Alemanha). O módulo permite a escrita e execução de agentes móveis em Java [STR97].
Mozart	Sistema de suporte a computação distribuída aberta e a inferência baseada em restrições. Implementa a linguagem concorrente e objeto orientada Oz, para o desenvolvimento de sistemas multiagentes. O projeto é uma colaboração entre os departamentos de Ciências Computacionais das universidades DFKI (Alemã) e Católica de Louvain (Bélgica) e o Laboratório de Sistemas Inteligentes do Instituto Sueco de Ciências Computacionais [MUL95].
MSAgent	Camada de software desenvolvido pela empresa Microsoft que é integrada ao sistema operacional Windows (95, 98 e NT). Utiliza a tecnologia COM e ActiveX da empresa. Pode ser utilizado diretamente pelos navegadores através de uma marcação <OBJECT> ou então inserido como componente nas linguagens que suportam a tecnologia COM.
TACOMA (Tronso And Cornell Moving Agents)	Camada de software para o suporte de processos (agentes), escrito em C e que utiliza o protocolo TCP. Suporta as linguagens C, C++, Perl, Python, Scheme e Visual Basic e as plataformas Unix e Windows (95 e NT). Está sendo desenvolvido como um esforço conjunto entre os departamentos de Ciências Computacionais das universidades de Cornell (EUA), Tronso (Noruega) e da Califórnia (São Diego) [JOH96].

8.3. Linguagens específicas

Uma segunda alternativa para a implementação de sistemas multiagentes é a utilização de uma linguagem específica para este fim. Diversas opções já estão disponíveis. A tabela 8.2 apresenta algumas destas opções.

Tabela 8.2 - Linguagens específicas para a programação de agentes

Linguagem	Características
AKL (<i>Agents Kernel Language</i>)	É uma linguagem concorrente desenvolvida no Instituto Sueco de Ciências Computacionais. Possui bibliotecas para o suporte a escrita de agentes.
April (<i>Agent PProcess Interaction Language</i>)	Linguagem orientada a construção de sistemas multiagentes. É uma linguagem concorrente baseada em objetos, onde os objetos são processos. Oferece uma interface para a linguagem C [MCA94].
Java	É uma linguagem de programação objeto orientada bastante semelhante ao C++. É interpretada e precisa de um interpretador de código de bytes na máquina que irá rodar a aplicação [BIG98].
Lalo	É uma linguagem orientada a agentes e uma estrutura para o desenvolvimento de sistemas multiagentes. Um programa escrito em Lalo é traduzido para o C++. Os agentes se comunicam através da KQML. A linguagem está disponível para as plataformas Unix e Windows (95 e NT).
Obliq	É uma linguagem não tipada e interpretada que suporta computação objeto orientada distribuída. A linguagem é uma especialização da Modula-3 que deve estar no computador do usuário para que possa rodar a Obliq [CAR95].
Python	É uma linguagem objeto orientada e interpretada, semelhante a Tcl, Perl, Scheme ou Java [CAM97].
Scheme	Linguagem de programação derivada do Lisp que incorpora o cálculo lambda, distinguindo procedimentos de expressões lambda e símbolos, utilizando um único ambiente léxico para todas as variáveis. A implementação do MIT é um ambiente completo que roda em plataformas Unix, Windows e IBM OS/2.
Tcl (<i>Tool Command Language</i>)	É a combinação de uma linguagem de script com uma biblioteca. A linguagem foi projetada com o objetivo de escrever programas interativos tais como editores de textos. A biblioteca pode ser embutida em aplicativos e consiste de um parser para a linguagem e rotinas para a implementação de comandos e procedimentos. A linguagem possui um ambiente gráfico de programação denominado de Tk (Tool Kit).
Telescript	É uma linguagem interpretada e objeto orientada que objetiva o desenvolvimento de aplicações distribuídas. A primeira classe da hierarquia de classes da linguagem é a classe "processo" [NWA96]. A linguagem foi desenvolvida pela empresa General Magic e é acompanhada de um ambiente de desenvolvimento.

8.4. Linguagens gerais

Uma terceira e última opção é o desenvolvimento a partir de linguagens objeto orientadas que, apesar de não possuírem facilidades específicas para implementação de agentes, apresentam a vantagem da proximidade conceitual entre agentes e objetos. Entre as linguagens que podem ser utilizadas estão o C++, o Java e o Delphi. Destas, a que é atualmente mais utilizada nas implementações relatadas na literatura é a linguagem Java pela fato de possuir uma camada de implementação, não de agentes em si, mas através de uma máquina virtual Java que é utilizada pelos navegadores de rede e que permite a implementação de software em múltiplas plataformas, sem a necessidade de adaptações a cada hardware específico. Desta forma um agente escrito para uma plataforma Windows, poderá rodar sem problemas numa estação Unix.

8.4.1. O Paradigma da orientação a objetos

Embora ainda extensivamente usada, principalmente comercialmente, a abordagem estruturada apresenta uma série de pontos fracos, mesmo tendo apresentado uma significativa melhora no tratamento da complexidade de software. Os principais problemas apresentados são a pouca flexibilidade à mudanças, os baixos índices de reaproveitamento do código gerado e a dificuldade de manutenção deste código.

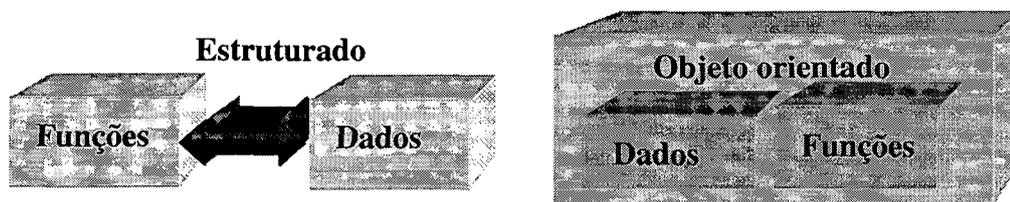


Figura 8.1 - Relação entre os paradigmas estruturado e objeto orientado

Analisando-se as deficiências da decomposição funcional, pode-se constatar que a principal causa é a separação entre dados e procedimentos. Eles são tratados com entidades diferentes em um programa. Esta dualidade torna-se evidente na independência das abordagens que as disciplinas de Banco de Dados e de Análise Estruturada dão ao desenvolvimento de sistemas [COA91]. No paradigma da Orientação ao Objeto ou Objeto Orientado esta dicotomia não existe. Os dados e os procedimentos são integrados em um só elemento que é denominado de objeto. A figura 8.1 faz um paralelo entre os dois paradigmas.

8.4.2. Vantagens da orientação a objetos

A possibilidade de que um objeto pode ser definido em termos de outros previamente construídos torna possível a manipulação de objetos em níveis crescentes de abstração, formando uma hierarquia de especialização. A solução para um problema pode, então, ser definida no nível mais apropriado de abstração. Isto reduz a complexidade das aplicações, apresentando as seguintes vantagens específicas, que surgem do suporte direto à abstração [BRO91]:

Dados ocultos. O estado interno de um objeto fica oculto de seus usuários. Como consequência, os clientes de um objeto precisam somente se preocupar com os serviços que o objeto fornece, não com a forma com que estes serviços são prestados.

Independência dos dados. Pela restrição do uso de um objeto a um conjunto fixado de operadores, pode-se controlar a extensão para o qual o objeto cliente depende dos detalhes internos do objeto servidor.

Modularidade. A definição de objetos apropriados pode agir como um ponto central para modular a implementação de um grande sistema de software, muitas vezes fornecendo um limite para distribuição ou concorrência de objetos.

Reuso. Pela organização das classes dentro de uma hierarquia de especialização, propriedades comuns de uma classe podem ser herdadas pelas subclasses, encorajando, não somente o reuso dentro do sistema mas, também, o reuso entre diferentes sistemas através do compartilhamento de bibliotecas de classes.

8.4.3. A orientação a objetos e os agentes

Com o paradigma objeto orientado, o analista distancia-se do perigo da dicotomia “dados e procedimentos”. A abstração passa a ser fundamentada em uma entidade que combina ambos e que traz uma série de características que o ajudam a enfrentar melhor a complexidade de problemas, produz sistemas menores do que os tradicionalmente formulados, aumenta consideravelmente o reaproveitamento do software e facilita a manutenção do código já gerado [BOO91, FIC92].

A combinação dessas características faz com que a orientação a objetos seja o paradigma mais adequado para se lidar com a programação de objetos complexos, como os tratados na hipermídia, na teoria dos agentes, e portanto a escolha natural para este trabalho. A figura 8.2, mostra uma relação entre agentes e objetos, conforme [LUC97].

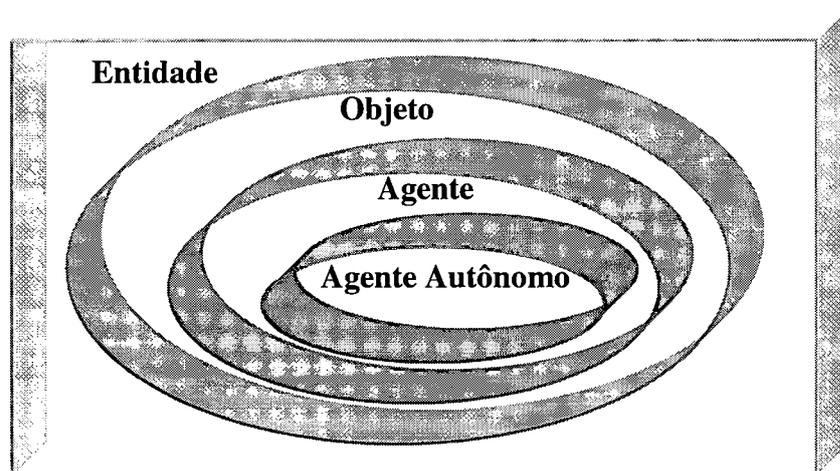


Figura 8.2 - Relação entre agentes e objetos

8.4.4. Os componentes de software

Os componentes de software estão se tornando tradicionais em todas as plataformas. Conforme a plataforma utilizada, eles podem receber diferentes denominações comerciais. Na plataforma Windows e Macintosh, os componentes foram denominados de tecnologia COM (*Component Object Module*), na OS/2 da IBM de SOM (*Some Other Model*) e nas demais de CORBA (*Common Object Request Broker Architecture*). Por definição COM é a implementação da OLE (*Object Linking and Embedding*) [CAN97]. A OLE é uma extensão do modelo DDE (*Dynamic Data*

Exchange). A utilização da área de transferência (*clipboard*) num PC permite a cópia de uma aplicação para outra. A DDE permite a conexão entre dois documentos e pode ser vista como uma extensão da área de transferência. A OLE permite a cópia dos dados de uma aplicação servidora para uma aplicação cliente. Os dados podem ser copiados junto com a conexão (*link*) formando a "inserção do objeto" ou deixados na aplicação original caracterizando a "conexão OLE".

A tecnologia OLE original foi atualizada para a OLE2 e recebeu novas características tais como a "automação" e "controles". Além disso o sistema operacional Windows recebeu uma camada de software para suportar esta nova versão da tecnologia, que foi renomeada de "controles ActiveX", permitindo que controles mais "leves" sejam apropriados para a distribuição através da Internet.

Os termos OLE e COM são utilizados normalmente da mesma forma, uma vez que o COM especifica os detalhes técnicos da OLE e qualquer linguagem COM-compatível poderá ser utilizada para escrever objetos COM/OLE. A tecnologia COM incorporou o conceito de objetos distribuídos através do modelo DCOM. A empresa Microsoft e o OMG (*Object Management Group*), responsável pelo modelo CORBA, concordaram em estabelecer uma abordagem que permita que um objeto DCOM possa se comunicar com um CORBA e vice-versa.

O DCOM é um conjunto de conceitos e objetos de interface em que um programa cliente ou objeto pode solicitar serviços, utilizando RPC (*Remote Procedure Call*), a um programa servidor ou objeto, ou a outros computadores em uma rede. Na Internet são utilizados os protocolos TCP/IP e o HTTP. O COM fornece um conjunto de interfaces para a comunicação entre programas, módulos ou objetos em um mesmo computador. O DCOM faz parte do ambiente NT a partir da versão 4.0 e pode ser gratuitamente adicionada ao Windows 95. Estará em breve disponível para a plataforma Unix e IBM.

8.4.5. A linguagem de implementação

Uma área de software de intensivos investimentos reside na criação de pacotes de desenvolvimento rápido - RAD (*Rapid Application Development*). Um pacote deste tipo consiste de uma linguagem de programação, um modelo de componentes para a conexão de elementos de interação, denominados de "controles", e um ambiente personalizável (os editores de recursos). Um dos mais bem sucedidos ambientes RAD tem sido o Borland Delphi, entretanto o primeiro destes pacotes a ter ampla utilização foi o Microsoft Visual Basic [HUG97]. O produto da Borland utiliza uma versão proprietária do Pascal objeto, enquanto que o da Microsoft utiliza uma versão proprietária do Basic. O Delphi não é apenas um compilador mas um ambiente completo que inclui um editor de texto com amplos recursos, um compilador, um depurador, bibliotecas de procedimentos, bibliotecas de classes e objetos, biblioteca encapsulando a API do Windows e outras ferramentas úteis, como o Winsight, um visualizador de classes.

Como o ambiente Java é tido como robusto e uma linguagem de programação segura ele está sendo visto como um inimigo natural dos ambientes RAD, mas a linguagem Java não suportava uma tecnologia considerada fundamental: a de componentes. Componentes de software são uma convenção para a escrita de controles de interface com o usuário, de forma que as ferramentas de desenvolvimento de aplicações e ambientes de execução possam manipulá-los. O JavaBeans é a resposta dos desenvolvedores do Java para o ingresso na arquitetura dos componentes. Para fazer frente a esta nova ameaça ao ambiente Visual Basic, a Microsoft estendeu a tecnologia COM para a Internet através dos controles ActiveX, oferecendo suporte para que controles

ActiveX sejam invocados a partir da linguagem Java e que a Java possa invocar controles ActiveX.

As primeiras linguagens utilizadas com o propósito de implementar objetos COM foram a C e a C++, mas a Delphi é inteiramente compatível com esta tecnologia. De fato, se for observado o código fonte, o Pascal objeto, utilizado pela Delphi, parece ser mais fácil de usar do que o C++ para escrever objetos COM. Isto deriva do fato de que com a versão 3 da linguagem (Delphi 3) foi introduzido o conceito de interface, que é, a propósito, a mesma idéia de interfaces que foi utilizada para integrar a linguagem Java com a tecnologia COM na plataforma Windows [CAN97].

8.4.6. Controles ActiveX

A linguagem Visual Basic da Microsoft foi o primeiro ambiente de programação a utilizar a idéia de fornecer componentes de software, se bem que a idéia de componentes reutilizáveis seja tão antiga quanto o conceito de orientação a objetos.

Essencialmente, um controle deste tipo fornece condições para que componentes possam se comunicar uns com os outros não importando a linguagem de implementação ou a plataforma suportada. Isto não quer dizer que um controle ActiveX seja portátil, mas sim que a comunicação pode ser realizada entre diferentes plataformas que suportam a tecnologia, da mesma forma que muitos outros ambientes exigem camadas intermediárias de software para serem portáteis. Por exemplo agentes construídos sobre o sistema Concordia e as applets precisam de um interpretador Java rodando na máquina cliente.

Tabela 8.3 - Opções para utilizar controles ActiveX [COF96]

Linguagem	Vantagens	Desvantagens
C++	Melhor acesso a API do controle. Código mais rápido e compacto. Grande número de implementações. Grande número de recursos testados.	Difícil de aprender, escrever e manter. Itens adicionais acrescentadas na linguagem em diferentes épocas e por vendedores de compiladores diferentes.
Delphi	Produtiva, fácil de manter e com recursos altamente integrados.	Linguagem proprietária. Requer recursos de terceiros para gerar controles ActiveX.
Java	Altamente portátil. Segura. Muitas implementações, mas todas em conformidade com as especificações da empresa Sun.	O desempenho varia conforme a plataforma. Tecnologia ainda imatura.
Visual Basic	Melhor conhecida do que qualquer outra ferramenta de desenvolvimento. Altamente portátil para codificar aplicações GUI.	O desempenho em computação intensiva é inferior as demais linguagens. A geração de controles ActiveX só está disponível na versão 5, o que poderá acarretar erros.

Um controle ActiveX incorpora os serviços COM e DCOM. Apesar de ser teoricamente compatível com outras plataformas, é necessário que o controle seja recompilado, antes, na plataforma que deverá ser utilizado. Ele é uma versão mais leve da OLE. Cada controle OLE possui uma janela que ocupa espaço na memória, mesmo que esta janela

nunca apareça. O uso de uma janela também significa que o controle, quando visível, deve ser retangular e não transparente, devendo suportar automação, edição visual e deve ser capaz, ainda, de manejar eventos entre ele e o local onde está inserido. Já um controle ActiveX não precisa ser uma janela e não precisa fornecer estas habilidades. De fato, um controle ActiveX deve satisfazer somente duas propriedades: ele deve poder responder questões sobre o que ele pode fazer e deve poder determinar quando não é mais necessário e liberar a memória por ele próprio. Isto é feito através da implementação do conceito de interface desconhecida (*IUnknown*).

8.5. Outras facilidades

A programação de agentes possui algumas particularidades que a programação convencional não apresenta. Algumas características dos agentes como a necessidade de comunicação entre eles, a mobilidade e a inteligência requerem esforços adicionais de programação. Como consequência algumas linguagens estão sendo criadas para atender a estas características.

Independente da forma e da arquitetura utilizada para implementar um conjunto de agentes, eles precisam trocar mensagens. Para que a comunicação possa ser mais efetivamente realizada, algumas padrões de linguagens de comunicação estão sendo propostos. A tabela 8.4 apresenta os principais projetos na área. A classificação utilizada nem sempre poderá ser isenta de críticas, pois o que para alguns autores é uma linguagem de programação para outros a linguagem serve mais a comunicação. Por isto é possível o encontro de divergências na literatura a respeito das classificações utilizadas neste trabalho.

Tabela 8.4 - Principais linguagens de comunicação de agentes

Linguagem	Características
ACE (<i>Agent Communication Language</i>)	Linguagem de comunicação de agentes desenvolvida através de uma iniciativa da DARPA. A linguagem é composta de um "vocabulário", um método de acesso externo através da KQML e uma linguagem interna de representação de conhecimentos a KIF [FIN94].
Agent Talk	É uma linguagem acrescida de um protocolo de coordenação. O protocolo pode ser definido de forma incremental e facilmente customizado através de um mecanismo de herança. Está sendo desenvolvida como um esforço conjunto do Laboratório de Ciências da Comunicação NTT e o Laboratório Ishida do departamento de Ciências da Informação da universidade de Kyoto.
KQML (<i>Knowledge Query and Manipulation Language</i>)	É uma linguagem mais um conjunto de protocolos que objetiva fornecer facilidades para que programas computacionais (agentes) se conectem e troquem informações entre si. É formada pelas camadas de conteúdo, de mensagem e de comunicação [FIN94].

Agentes inteligentes devem possuir uma base de conhecimentos. Atualmente não existe um padrão universal para a troca de conhecimentos. Assim, cada aplicação constrói sua própria base de conhecimentos e utiliza linguagens e metodologias particulares, acarretando a inviabilidade do aproveitamento do esforço de desenvolvimento em outras aplicações.

Para fornecer suporte ao reaproveitamento e o compartilhamento de conhecimentos formalmente representados entre sistemas multiagentes é útil a definição de um vocabulário comum em que o conhecimento compartilhado possa ser representado. Uma especificação de um vocabulário para um domínio de discurso compartilhado, incluindo definições de classes, relações, funções e termos, é denominado de **ontologia**. Alguns projetos estão realizados no sentido de formalizar um padrão para a troca de conhecimentos entre agentes de diferentes aplicações. A tabela 8.5 apresenta alguns resultados.

Tabela 8.5 - Linguagens de troca de conhecimentos

Linguagem	Características
KIF (<i>Knowledge Interchange Format</i>)	É uma versão do cálculo de predicados de primeira ordem com extensões para raciocínio não-monotônico e definições [FIN94]. A linguagem está sendo desenvolvida pelo <i>Interlingua Group</i> .
Ontolingua	O KSL (<i>Knowledge Systems Laboratory</i>) da universidade de Stanford está desenvolvendo ferramentas e serviços para suporte ao processo de obtenção de ontologias compartilhadas comuns por grupos geograficamente distribuídos [GRU93]
SRKB (<i>Shared, Reusable Knowledge Bases</i>)	O grupo SRKB está trabalhando no sentido de formalizar o conteúdo de bases de conhecimento compartilhadas. Um dos objetivos é compartilhar o conhecimento de áreas específicas mas com metodologias e ferramentas independentes de conteúdo [FIN94].

8.6. Conclusão

Nos últimos anos, os agentes estão sendo utilizados nas mais diversas áreas para solucionar problemas complexos. O ensino por computador é um problema deste tipo. Geralmente são sistemas grandes, pesados e pouco flexíveis. A utilização da hipermídia, flexibilizando o fornecimento dos conteúdos aliada a uma estrutura de multiagentes para coordenar e testar o aluno poderá ser uma alternativa promissora sobre o sistema tutorial tradicional.

A implementação de sistemas multiagentes, assim como de qualquer outro sistema de software, depende hoje de linguagens, módulos, bibliotecas, para que possam ser efetivamente implementados. O paradigma objeto orientado permite o reaproveitamento de código e hoje é impraticável se construir sistemas a partir do zero como ocorria no início da década quando o sistema operacional para micros era o DOS. Com as novas interfaces gráficas cada vez mais sofisticadas, é inviável a construção de qualquer sistema sem o apoio de uma base sólida de componentes pré-construídos. A escolha do melhor caminho é um problema bastante difícil devido à diversidade de opções oferecidas. No capítulo foram listadas várias destas opções e a escolha será feita evidentemente mais por critérios de oportunidade e preferências pessoais do que propriamente técnicas, uma vez que praticamente inexistem mecanismos que possam avaliar ou recomendar a melhor opção entre a diversidade disponível.



9. UMA APLICAÇÃO NA ESTATÍSTICA

9.1. Introdução

A estatística moderna surgiu no início deste século [VEL91] e tem sido computo-rizada essencialmente através de ferramentas que pretendem facilitar a análise de dados. Certas técnicas só podem ser aplicadas a grandes conjuntos de dados com a ajuda do computador. No entanto, a grande maioria destes sistemas, apesar da migração para Windows e da melhora na interface, não são fáceis de usar. Uma outra opção disponível hoje são documentos html que podem ser visualizados ou recuperados através de qual-quer software de navegação na Internet. Alguns destes locais apresentam *applets* Java, que proporcionam alguma interatividade e permitem a visualização dinâmica de distri-buições e animações de experimentos probabilísticos.

As planilhas (notadamente o Excel) também vêm fornecendo, com as sucessivas atualizações, mais recursos estatísticos e, normalmente, com uma maior facilidade de uso do que os pacotes específicos. No entanto, ela é uma ferramenta que não foi proje-tada para lecionar Estatística e muito menos como um recurso didático. É um software projetado com outro propósito e que apresenta alguns recursos para a análise de dados. Notadamente os conceitos estatísticos na planilha são de tal forma confusos que é ne-cessário que o professor dedique parte do tempo a explicá-los. Além disto, a organiza-ção não segue nenhum plano pedagógico. É uma mistura de recursos que junta todas as áreas em um único local, prestando-se a uma confusão maior entre os usuários do que a que eles normalmente fazem. Para observar que o Excel, por exemplo, deixa muito a desejar, para ser um software estatístico razoável, basta observar o grande número de extensões (*add-ins*) que estão disponíveis no mercado.

No caso de softwares específicos, como o SPSS, por exemplo, apesar de apresenta-rem recursos para quase qualquer tipo de análise estatística, não se deve esquecer que eles foram e são projetados para serem eficientes na análise de dados, e não são, de for-ma alguma, transparentes no como e porquê estes cálculos estão sendo realizados. Ou seja, são softwares destinados a especialistas e que só podem ser usados no ambiente didático com o auxílio de um especialista. Além disto, estes softwares são excelentes para eliminar o trabalho braçal, digamos assim, envolvido na aplicação das técnicas es-tatísticas, mas eles não são apropriados para a exploração de conceitos e relações. Ou seja, eles são ferramentas e não recursos didáticos.

9.2. Os recursos existentes

Os recursos computacionais existentes para se lidar com dados e que poderão ser utilizados como recursos instrucionais para se lecionar Estatística podem ser divididos em dois tipos:

- Recursos *online*;
- Recursos *off-line*.

9.3. Os recursos *online*

Os recursos *online* são os que exigem que o computador esteja conectado a rede. Isto é, cursos ou bases de dados ou programas (*applets*) que só podem ser alcançados via uma conexão remota, através da WWW, e que são apresentados através de páginas html. Os recursos *online* são via de regra locais disponibilizados por professores ou grupos de pesquisas ligados a universidades. Alguns são versões html de livros que servem para divulgação da versão impressa ou, ainda, algum software de seus autores.

9.3.1. Os cursos (tutoriais) na rede

A grande gama de cursos, recursos, bases de dados, casos, histórias e tudo o mais que a imaginação dos autores permita pode ser encontrado hoje na Internet com o objetivo de lecionar Probabilidade e Estatística. Grande parte deles são versões html de livros que funcionam mais como um comercial do produto do que propriamente como material didático. A tabela 9.1 apresenta um resumo de alguns dos locais mais facilmente encontrados. A lista é obviamente parcial porque diariamente novos recursos estão sendo acrescentados.

Tabela 9.1 - Alguns cursos (tutoriais) *online*

Pacote	Identificação	Artigos
GASP (<i>Globally Accessible Statistical Procedures</i>)	Responsável Webster West do Departamento de Estatística da Universidade da Carolina do Sul.	
	Local http://www.stat.sc.edu/rsrch/gasp/ Propósito O sistema apresenta duas abordagens: uma interface de formulários, onde o usuário interage com o computador controlador do sistema; e uma segunda abordagem realizada através de <i>applets</i> Java.	
Rice <i>Virtual Lab in Probability and Statistics</i>	Responsável David Lane do Departamento de Psicologia e Estatística da Universidade Rice.	
	Local http://www.ruf.rice.edu/~lane/ Propósito É um livro eletrônico com conteúdos introdutórios de Estatística e conexões para outros recursos estatísticos na rede.	
<i>Psychological Statistics at SMSU</i>	Responsável David W. Stockburger do Departamento de Psicologia da Universidade Estadual do Sudoeste Missouri (SMSU).	
	Local http://www.psychstat.smsu.edu/ Propósito Versões html de livros do autor, sendo um sobre Estatística Básica e outro sobre Estatística Multivariada.	Estatística
SMART (<i>Statistics and Mathematics as Advanced Research Tools</i>)	Responsável Grupo BioSS (<i>Biomathematics & Statistics Scotland</i>)	TAL98
	Local http://www.bioess.sari.ac.uk/smart/ Propósito Treinamento em técnicas estatísticas especializadas, dirigido a pessoas que já possuam familiaridade com as técnicas básicas e que desejam se familiarizar rapidamente com novas técnicas.	

<i>SurfStat. australia</i>	Responsável	Keith Dear do Departamento de Estatística da Faculdade de Economia e Comércio da Universidade de Newcastle, Austrália.	DEA97
	Local	http://surfstat.newcastle.edu.au/surfstat/	
<i>Virtual Laboratories in Probability and Statistics</i>	Responsável	Kyle Siegrist do Departamento de Ciências Matemáticas da Universidade do Alabama em Huntsville.	
	Local	http://www.math.uah.edu/stat/intro/index.html	
<i>WISE (Web Interface for Statistics Education)</i>	Responsável	Dale Berger do Departamento de Psicologia da Universidade Claremont.	
	Local	http://www.projects.cgu.edu/wise/	
<i>XploRe</i>	Responsável	W. Haerdle e MD*Tech Method and Data Technologies.	MÚL98
	Local	http://141.20.100.244/www.xploRe-stat.de/	
	Propósito	Pacote com uma linguagem de programação completa que fornece recursos interativos e gráficos dinâmicos, que pode ser utilizado na rede através de uma interface CGI e com auxílio de applets Java.	

9.4. Os recursos off-line

Estes recursos envolvem principalmente softwares comerciais como os pacotes genéricos, os pacotes específicos, as versões computacionais de livros, cursos em CD-ROM, pacotes para análises gráficas de dados, planilhas e suas as extensões.

Nos últimos anos surgiram uma série de pacotes estatísticos para microcomputadores. Este crescimento ocorreu principalmente pela migração de antigos pacotes baseados em *mainframes*. O crescimento do poder computacional dos micros com discos rígidos cada vez maiores e mais rápidos tornaram esta migração possível e também possibilitaram que estes softwares se tornassem cada vez mais sofisticados e que manejassem conjuntos de dados cada vez maiores. Muitos destes novos pacotes oferecem, agora, um leque abrangente de procedimentos estatísticos e uma grande flexibilidade para explorar e analisar dados. No entanto, alguns deles ainda conservam as hábitos anteriores, como o SPSS, por exemplo, que fornece facilidades, no caso um ambiente específico, para se trabalhar com a linguagem de comandos. É possível com o SPSS para Windows continuar a trabalhar da mesma forma que se fazia no ambiente DOS.

Os pacotes anteriores baseados em *mainframes* eram projetados para utilização através de uma linguagem de comandos, fazendo com que o usuário tivesse que aprender linguagens específicas nem sempre claras antes de tentar executar qualquer análise estatística. As novas versões baseadas no Windows oferecem novas facilidades, como interface gráfica, a possibilidade de recortar e colar dados através de aplicações diferentes, troca dinâmica de dados e Inserção e Ligação de Objetos (OLE), que as tornam muito mais úteis e poderosas do que as antigas versões comando orientadas dos *mainframes*. No entanto, como o observado, por Morgan [MOR98], "*existe uma grande necessidade, na pesquisa científica e na comunidade de negócios, por pacotes estatísticos que permitam a não profissionais avaliar dados de forma acurada e eficiente*" (p. 1). Eu acrescentaria ainda que esta necessidade é mais premente na área educacional, isto é, existe ainda uma lacuna não-preenchida por um software que permita o ensino de estatística de forma eficiente para alunos de vários níveis e com variados conhecimentos de informática. A idéia é construir uma ferramenta que se adapte aos alunos e ao professor, e não, como acontece, quando se usa ou se tenta utilizar diferentes pacotes que foram projetados com outros objetivos, filosofias ou ambientes.

9.4.1. Os pacotes de propósito geral

A diversidade de pacotes estatísticos varia de um extremo, que inclui os denominados genéricos ou de propósito geral, a outro extremo, que engloba pacotes especializados em uma única tarefa ou técnica estatística. Por pacote genérico entende-se todo e qualquer software que utilize grande parte das principais técnicas incluindo as multivariadas. Estes pacotes invariavelmente adotam o paradigma da planilha como entrada de dados e normalmente fornecem saídas na forma de objetos, possibilitando a troca ou intercâmbio de resultados com outros softwares projetados na mesma plataforma. Como são ferramentas que envolvem muitos recursos, geralmente custam caro, requerem conhecimento especializado para o seu uso eficiente e consomem muito espaço no disco rígido, necessitando de processadores rápidos ou coprocessadores matemáticos (para os computadores mais antigos) e grande quantidade de memória volátil.

Apesar de não declarado, muitos pacotes são projetados explicita ou implicitamente para um tipo específico de audiência. Muitos dos principais pacotes foram desenvolvidos por ou sob a influência de especialistas em certas áreas. Da área Biomédica, o BMDP; da Estatística ou por estatísticos, o GLIM; da área de Agricultura, o Genstat; por ou para professores de Estatística, o Minitab; por ou para psicólogos, o Systat e por ou para as Ciências Sociais, o SPSS [CUR94].

Muitos desses pacotes são originários do início da era computacional e vários ainda apresentam interface de linha de comando, mas a tendência é em direção a uma interface JIMM (janelas, ícones, *mouse* e menus deslizantes). Esta metáfora de interface torna a utilização acessível a novatos ou usuários ocasionais e eles podem, desta forma, serem utilizados, eventualmente, como recursos instrucionais.

Os pacotes específicos são projetados para aplicar uma ou algumas poucas técnicas estatísticas, invariavelmente técnicas multivariadas.

Tabela 9.2 - Resumo dos principais pacotes genéricos

Pacote	Identificação		Artigos
Axum	Responsável	Mathsoft	ASH98
	Local	http://www.mathsoft.com/axum/	
ConStatS	Responsável	Estúdio de projeto de software curricular da universidade Tufts.	KIN97
	Local	http://www.tufts.edu/tccs/services/css/	
Data Desk	Responsável	Data Description Inc	CUR98
	Local	http://www.datadesk.com/data/	
GAUSS	Responsável	Aptech Systems, Inc.	
	Local	http://www.aptech.com/	
Genstat (<i>General Statistical System</i>)	Responsável	Departamento de Estatística do IACR	GRA97, TAY97
	Local	http://www.res.bbsrc.ac.uk/stats/genstat/	
JMP	Responsável	Instituto SAS	CUR96
	Local	http://www.jmpdiscovery.com/	
Minitab	Responsável	Minitab, Inc.	CUR97, DAV97
	Local	http://www.minitab.com/products/minitab/	
NCSS	Responsável	NCSS Statistical Software	
	Local	http://www.ncss.com/html/ncsswin.html	
Origin	Responsável	Microcal Software	
	Local	http://www.microcal.com/www/	
Prism	Responsável	GraphPad Software	MOR98
	Local	http://www.graphpad.com/prism/	
SAS/ Insight	Responsável	Instituto SAS	CUR96
	Local	http://www.sas.com/software/components/	
SAS/ STAT	Responsável	Instituto SAS	COE91
	Local	http://www.sas.com/software/components/	
SigmaPlot	Responsável	SPSS Inc.	NEW97
	Local	http://www.spss.com/software/science/	
SigmaStat	Responsável	SPSS Inc.	
	Local	http://www.spss.com/software/science/	
Simstat	Responsável	KCS (<i>Kovach Computing Services</i>)	
	Local	http://www.kovcomp.com/simstat.html	
S-PLUS	Responsável	Mathsoft	RIP97
	Local	http://www.mathsoft.com/splus/splsprod/	

SPSS	Responsável SPSS Inc.	CUR98, LUR95
	Local http://www.spss.com/software/spss/	
SPSS Diamond	Responsável SPSS Inc.	NEW98
	Local http://www.spss.com/software/	
Stata	Responsável Stata Co.	FER95
	Local http://www.stata.com/info/	
Statgraphics	Responsável Manugistics, Inc.	DAV92
	Local http://www.statgraphics.com/frame/html/	
Statistica	Responsável StatSoft	
	Local http://www.statsoft.com/addition.html	
Statistix	Responsável Analytical Software Co.	MOR98
	Local http://www.sigma-research.com/bookshelf/	
Statmost	Responsável Datamost	MOR98
	Local http://www.datamost.com/products.htm	
StatView	Responsável Instituto SAS	KAY97, BIE93
	Local http://www.statview.com/productinfo/	
Systat	Responsável SPSS, Inc.	CUR97
	Local http://www.spss.com/software/science/systat	
Vista	Responsável Forrest Young - Departamento de Psicologia - Universidade da Carolina do Norte.	TAG97, YOU95
	Local http://forest.psych.unc.edu/research/vista-frames/	
WINKS	Responsável TexaSoft	FOR98
	Local http://www.texasoft.com/	
Xlisp-Stat	Responsável Luke Tierney	
	Local http://stat.umn.edu/~luke/xls/xlsinfo/	

9.4.2. Pacotes de propósitos específicos

Por pacotes ou sistema específico, entende-se aquela categoria de software projetado para aplicar uma técnica específica (como por exemplo: regressão, análise de variância, análise de conglomerados, etc.) ou atender determinada categoria de consumidores (economistas, biólogos, etc.) ou área de aplicação (saúde pública, indústria, etc.). O sistema Epi Info, por exemplo, foi projetado com o objetivo de atender a profissionais de saúde pública. A quantidade de recursos que podem ser colocados nesta categoria é elevada e cresce rapidamente. Portanto não existe a pretensão de um levantamento abrangente. Na tabela seguinte estão colocados os recursos que na minha visão são os mais significativos.

Tabela 9.3 - Resumo dos principais pacotes específicos

Pacote	Identificação	Artigo
Amos	Responsável SPSS. Inc.	
	Local http://www.spss.com/software/spss/base/	
	Propósito Equações estruturais	
Epi Info	Responsável CDC (<i>Center for Disease Control</i>)	
	Local http://www.cdc.gov/epo/epi/epiinfo.htm	
	Propósito Análise de estatísticas de saúde.	
EViews	Responsável Quantitative Micro Software	
	Local http://www.wviews.com/index.html	
	Propósito Análise econométricas e previsões.	
First Bayes	Responsável Anthony O'Hagan do Departamento de Matemática e Estatística da Universidade de Nottingham, Inglaterra.	YOU95
	Local http://www.maths.nott.ac.uk/personal/aoh/	
	Propósito Análise Bayesiana	
GLIM	Responsável Grupo de trabalho GLIM da <i>Royal Statistical Society</i> Britânica.	HIN94
	Local http://www.nag.co.uk/xtats/gdgc/html	
	Propósito Principalmente ajuste de modelos lineares.	
MacAnova	Responsável Gary Oehlert e Christopher Bingham do Departamento de Estatística da Universidade de Minnesota.	HIN97
	Local http://www.stat.umn.edu/~gary/macanova/	
	Propósito Análise de variância e de de séries temporais.	
MLab	Responsável Civilized Software	
	Local http://www.civilezed.com/	
	Propósito Modelagem matemática e estatística.	
Oriana	Responsável Waren Kovach	UPT96
	Local http://www.kovcomp.co.uk/oriana/index.html	
	Propósito Análise de dados circulares.	
R-Code	Responsável Luke Tierney	ROB95
	Local http://www.stat.umn.edu/~rcode/	
	Propósito Análise de Regressão	
Secos	Responsável Statistics for Education	REI98
	Local http://www.statsed.co.uk/secos/index.htm	
	Propósito Bases de dados para realizar cálculos estatísticos. Programa para interpretação de dados e realização de gráficos.	

Soritec	Responsável	Full information Software	
	Local	http://fisiosoft.com/	
	Propósito	Análises econométricas.	
Xtremes	Responsável	Rolf Reiss e Michael Thomas	COL96
	Local	http://www.xtremes/math.uni-siegen.de/	
	Propósito	Analisar dados extremos e comparar estimadores.	

9.4.3. Cursos

Aqui, o objetivo é fornecer condições para que o interessado adquira conhecimentos básicos sobre certas áreas ou conteúdos Estatísticos e Probabilísticos. Na maioria são produtos comerciais e, quando não, geralmente servem de publicidade para algum outro produto como um software ou um livro escrito pelo autor do curso.

Tabela 9.4 - Resumo dos principais cursos off-line

Curso	Identificação		Artigo
ActivStats	Responsável	Paul Velleman do Departamento de Estatística da Universidade de Cornell.	CUR97
	Local	http://www.datadesk.com/activstat/index.htm	
	Propósito	Apresentar conteúdos introdutórios num formato multimídia. Os conteúdos incluem vídeos, simulação, animação, narração, texto e experimentos interativos.	
Against All Odds: Inside Statistics	Responsável	David Moore do Departamento de Estatística da Universidade de Purdue e COMAP (<i>Consortium for Mathematics and its Applications</i>).	
	Local	http://www.pbs.org/adultlearning/als/publication/	
	Propósito	É um curso introdutório de Estatística para alunos que não estudaram Cálculo, na forma de um telecurso englobando 26 programas de meia hora cada um.	
An Eletronic Companion to Statistics	Responsável	Cogito Learning	BOW98
	Local	http://www2.viaweb.com/cogitodirect/	
	Propósito	Material de suporte que apresenta os conteúdos na forma de notas que são ilustradas por vídeo, gráficos e animações. Apresenta testes em cada seção.	
CrashCourse in Statistics	Responsável	SPSS, Inc.	ROB97
	Local	http://www.spss.com/software/science/crash/	
	Propósito	Pacote auxiliar que apresenta conteúdos introdutórios e avançados de Estatística. Inclui, som, vídeo e exemplos reais.	

Quercus	Responsável	TLTP (<i>Teaching and Learning Technology Programme</i>) da Universidade de Strathclyde, Escócia.	McC96
	Local	http://www.stamns.strath.ac.uk/external/CAL/	
	Propósito	Tutorial interativo sobre a análise básica de dados e a escrita de relatórios.	
Statistics for the Terrified	Responsável	ITTI (<i>Information Technology Training Initiative</i>) da Universidade de Ulster, Irlanda.	MOR99
	Local	http://www.icbl.hw.ac.uk/itti/ittiprod.html	
	Propósito	Ensinar Estatística para aqueles que não possuem base matemática sólida como alunos dos cursos das áreas Humanas e Sociais.	
StatTutor	Responsável	Peter Richardson do Departamento de Psicologia da Universidade britânica Nene.	CUR96, SNE95
	Local	http://www.york.ac.uk/inst/etipsych/web/CTI/	
	Propósito	Ensino de estatística básica para estudantes universitários de Psicologia.	
STEPS (<i>Statistical Education through Problem Solving</i>)	Responsável	TLTP (<i>Teaching and Learning Technology Programme</i>) da Universidade de Strathclyde, Escócia.	McC95, MAC95, SNE95
	Local	http://www.stats.gla.ac.uk/steps/	
	Propósito	Módulos baseados em problemas para dar suporte ao ensino de Estatística em vários cursos.	
Statwise	Responsável	Projeto SUMSMAN (<i>Scottish Universities Mathematics and Statistics across the Metropolitan Area Networks</i>) da Universidade Napier, Escócia	DAV97
	Local	http://www.maths.ed.ac.uk/~ama/data/	
	Propósito	Ensinar conceitos introdutórios de Estatística alunos universitários de Administração e Ciências Sociais.	

9.4.4. Planilhas

As planilhas vão se firmando cada vez mais como um recurso instrucional em laboratórios de Estatística. Além dos recursos típicos, elas oferecem um grande número de funções estatísticas e probabilísticas, se bem que bastante limitados. As principais vantagens da planilha são sua grande base instalada, a possibilidade de programação de novas funções e seu custo relativamente baixo. Além disso, o paradigma da planilha é conhecido por boa parte dos alunos, diminuindo, desta forma, o tempo gasto na aprendizagem da mecânica de uma nova ferramenta de software. Felizmente, ou infelizmente, nesta área, parece que existe uma tendência, quase que universal, em favor do software da Microsoft, a planilha Excel. Isto é confirmado pelo grande número de empresas e mesmo pessoas isoladamente que escrevem extensões para este programa, pelos artigos na literatura técnica, pelos exemplos e exercícios disponíveis na Internet e até pelas listas de discussões, exclusivamente dedicadas a este software.

A planilha Excel

Ela foi lançada em 1987 numa versão desenvolvida originalmente para os computadores Macintosh. A primeira versão para Windows foi rotulada como 2, para corresponder a versão já existente para o Macintosh. Em 1990 foi lançada a versão 3, que incluía barras de ferramentas, capacidade para desenhar, suportes para programas adicionais e gráficos em 3 dimensões. A versão 4, a primeira realmente popular, foi lançada em 1992. Em 1993 foi lançada a versão 5, com melhorias expressivas, como planilhas múltiplas e suporte para a linguagem Visual Basic. Em 1995 foi lançada a versão 7 da planilha, conhecida como Excel 95 e a primeira versão de 32 bits. A versão 6 da planilha não existe, pois a Microsoft resolveu remunerar seus produtos para escritório, de modo que todos eles tivessem a mesma versão. A versão 8 foi lançada em 1997 e conhecida como Excel 97. A versão 9, que está sendo anunciada como Excel 2000, está prevista para 1999/2000.

O Excel pode ser utilizado no ensino de Estatística. No entanto, este software apresenta uma série de deficiências e vários problemas para ser utilizado como recurso instrucional. Os problemas variam desde os de tradução, passando pela falta de homogeneidade de nomenclatura e representação até a estrutura "espaguete" dos conceitos estatísticos, que mistura todos em uma única categoria, fazendo com que o aluno já confuso o fique ainda mais.

Além do Excel existem várias outras alternativas. O problema é que a maioria delas é praticamente desconhecida. Com exceção da Lotus 1-2-3, que já foi extremamente popular, e da Quatro Pro, as demais só serão descobertas com uma garimpagem minuciosa na Internet, mas não existe, praticamente nada, de literatura sobre ensino de Estatística com estas planilhas.

Extensões para a planilha Excel

Uma área de software que vem crescendo substancialmente são os pacotes (ou extensões) para a planilha que viabilizam utilizá-las como recurso instrucional de Estatística. A literatura apresenta apenas este tipo de software para a planilha Excel. A tabela 9.5, um resumo dos encontrados atualmente.

Tabela 9.5 - Resumo das principais extensões para a planilha Excel

Pacote	Identificação	Artigos
DISCUS (<i>Discovering Important Statistical Concepts Using Spreadsheets</i>)	Responsável Neville Hunt e Sidney Tyrrel da Escola de Matemática e Ciências da Informação, da Universidade Coventry, Inglaterra. Local http://www.mis.coventry.ac.uk/research/ Propósito Coleção de 8 livros de exercícios, onde cada um cobre um assunto e consistem de arquivos contendo conjuntos de dados, diagramas, tabelas e gráficos de vários tipos.	GRA98
Utilizando dados	Responsável Statistics for Education Local http://www.statsed.co.uk/secos/secos1.htm Propósito Fornecer aos calouros ferramentas que os habilitem a interpretar e utilizar Estatística Descritiva de uma forma significativa.	RID98

Unistat	Responsável Unistat Limited	CUR98
	Local http://www.unistat.com/index.html	
	Propósito Programa complementar que fornece uma grande variedade de procedimentos estatísticos dentro da planilha Excel e permite direcionar as saídas para o MSWord.	
xlSTAT	Responsável Thierry Fahmy	
	Local http://www.xlstat.com/	
	Propósito É um conjunto de procedimentos estatísticos e de manipulação de dados, principalmente multivariados.	
XLStatistics	Responsável Rodney Car do Departamento de Sistemas de Gerenciamento de Informações da Universidade Deakin, Austrália.	
	Local http://www.man.deakin.edu.au/rodneyc/	
	Propósito Traçado de distribuições de probabilidades acumuladas, regressão linear e não-linear, análise de regressão e correlação múltiplas.	

9.5. A aplicação do modelo

O modelo multiagentes proposto será aplicado ao ensino de Probabilidade e Estatística através de um sistema modular. Este sistema é acrescido, devido as particularidades do ensino de Estatística de um módulo Planilha, com uma interface inovadora que incorpora conhecimento e diminui as possibilidades de análises errôneas que podem ser cometidas com os pacotes tradicionais.

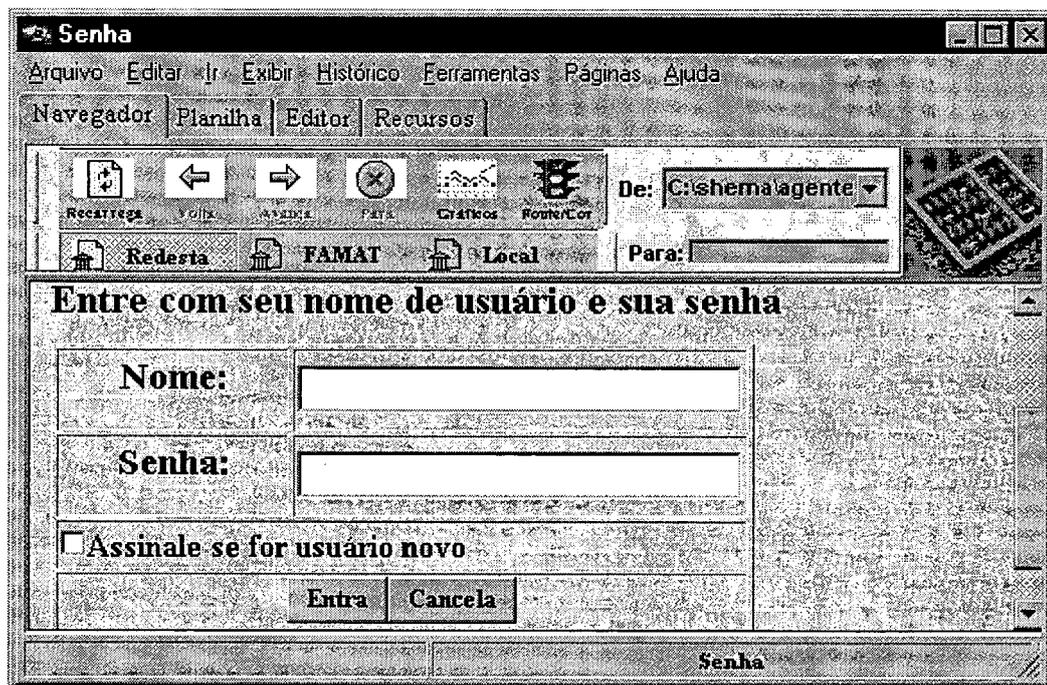


Figura 9.1 - Uma visão do sistema, destacando o módulo de navegação

Ao executar a aplicação, o usuário terá disponível quatro ambientes distintos:

- ☐ um ambiente de navegação, através de um interpretador html;
- ☐ um ambiente gráfico, composto de um painel 2D e outro 3D;
- ☐ uma planilha com interface inteligente e
- ☐ um ambiente de comunicação.

A primeira página acessada por default é o ambiente do navegador. É neste ambiente hipermídia que o usuário pode ler o conteúdo da disciplina na forma de um hipertexto. Este módulo funciona de forma integrada aos demais através de recurso inovador de extensão do conceito de URL da WWW.

O módulo planilha oferece recursos de um software estatístico tradicional (como o SPSS por exemplo), apenas com a diferença de possuir os recursos enfocados nas diversas disciplinas e não técnicas avançadas de análise, só que de uma maneira inovadora, enfocando e reforçando os conceitos como forma de ser um recurso didático e não meramente um recurso de manipulação de dados.

O ambiente gráfico possui três painéis distintos que funcionam integrados à planilha e ao navegador através de conexões por URLs.

Os conceitos de Estatística, bem como os de Probabilidade, podem ser apresentados através de um sistema de hipermídia com inúmeras vantagens. Primeiro, o material pode ser enriquecido com conexões que levem a arquivos sobre a história e o desenvolvimento tanto da Estatística quanto da Probabilidade. Segundo, referências a figuras ilustres no desenvolvimento das disciplinas, que num texto normal ou numa aula serão apenas citados, podem ser acrescentadas ao conteúdo na forma de uma conexão, contendo a biografia ilustrada e, desta forma, sem interferir com o texto normal da disciplina, pois esta conexão poderá ou não ser seguida, dependendo apenas da curiosidade e interesse individual de cada um.

No entanto, a principal vantagem do hipertexto na Estatística e na Probabilidade são as conexões a nodos contendo outros programas que podem ser simulações e ilustrações de conceitos, como as distribuições de probabilidade em Probabilidade e as distribuições amostrais na Estatística. No caso do sistema modelo proposto, as conexões podem remeter à planilha auxiliar, onde o aluno pode praticar na hora o conceito sendo apresentado, sem mudanças de ambiente e sem a necessidade de recursos adicionais, como rodar um novo pacote estatístico ou mesmo utilizar uma calculadora.

O modelo agrega um recurso para a realização de análises gráficas dinâmicas e o traçado de distribuições tanto de funções contínuas quanto discretas. O sistema, além de apresentar gráficos de distribuições pré-embutidas no sistema, pode traçar o gráfico de qualquer outra função através de um analisador sintático. O aluno pode optar por fazer um único traçado num sistema cartesiano ou plotar várias funções num mesmo diagrama para comparações, por exemplo, uma mesma distribuição com diferentes parâmetros. Este recurso é talvez o mais importante do sistema e o que o diferencia de outros existentes que apresentam lições ou conteúdos de Estatística através de páginas html, onde somente é possível utilizar recursos gráficos de forma limitada com a utilização de *applets Java*. O uso de *applets* apresenta duas grandes desvantagens. A primeira delas é que o navegador deve possuir embutido um interpretador *Java* e, a segunda, é que cada função precisa ser especificamente programada através de uma nova *applet*. Já com o sistema proposto isto não é necessário, pois o analisador sintático pode interpretar qualquer nova função que não esteja pré-programada no sistema. O analisador sintático será útil também para traçar diagramas de distribuições de probabilidade contínuas que nenhum outro pacote computacional contempla. Por exemplo a função $f(x) = 1,5.(x - 1)^2$

para $0 \leq x \leq 2$ é uma função densidade de probabilidade. Basta que esta equação seja digitada e o intervalo fornecido para que o usuário tenha a idéia do gráfico desta distribuição. Através do gráfico, o usuário pode ter uma melhor idéia das medidas da variável, como por exemplo a média, e ficar mais seguro sobre a correção de valores eventualmente calculados.

9.6. O Módulo de navegação (browser)

Em qualquer caso, o hipertexto é a técnica natural para suportar interfaces multimídias, uma vez que é baseado na interconexão de nodos contendo diferentes mídias [NIE93]. Os nodos típicos em um ambiente hipermídia são: texto, gráficos, vídeo e som. Para este sistema, o objetivo é utilizar esses nodos típicos, com destaque a texto, gráficos animados e programas. Em menor grau, poderá ser integrado som. Neste caso, o som deve ser combinado a um dos nodos existentes, uma vez que pela característica do sistema proposto, isto ressaltaria certos conteúdos, reforçando desta forma, a atratividade do sistema. Os nodos constituídos de programas vão estabelecer um diferencial em relação aos sistemas de hipertexto tradicionais. Apesar de mencionados frequentemente como nodos típicos em um ambiente hipermídia, nodos constituídos de programas raramente aparecem em sistemas existentes. No sistema proposto os nodos constituídos de programas vão desempenhar um papel importante para dinamizar o ensino. Ao invés de obter uma única ilustração estática, como nos textos tradicionais, o usuário, neste caso, poderá, através do nodo programa e com o auxílio da simulação, rodar quantos exemplos forem necessários para o perfeito entendimento dos conceitos sendo apresentados.

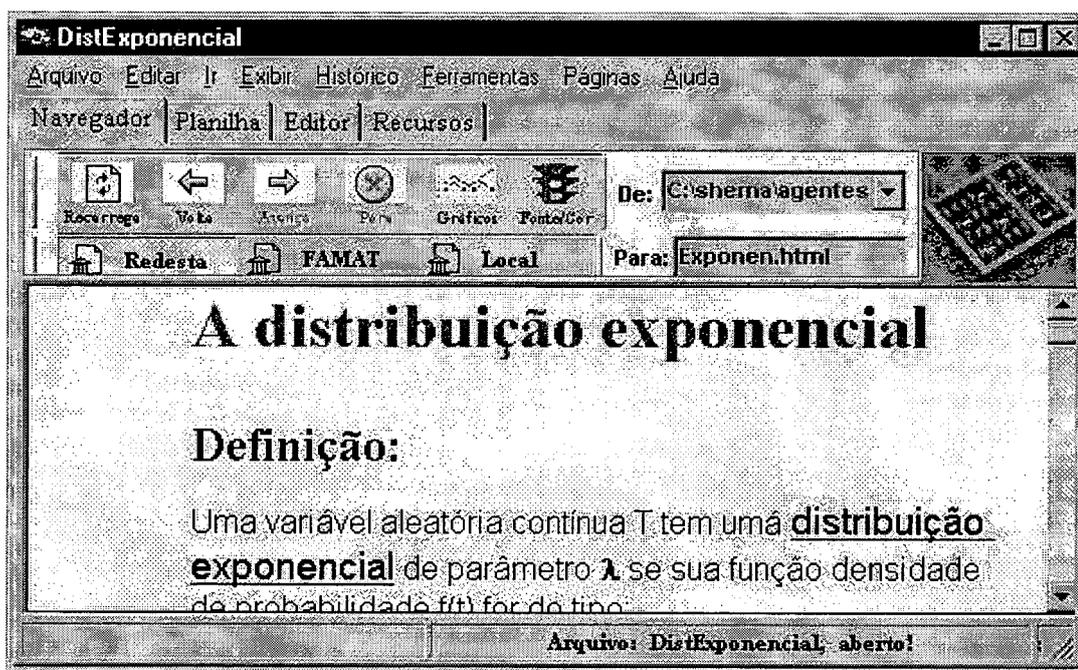


Figura 9.2 - Uma visão do módulo de navegação

Características do sistema

Quanto à navegação, o sistema possui os principais recursos disponíveis nos navegadores tradicionais. Possui capacidade de recarregar um documento, caso ocorra algum problema durante a abertura inicial do documento. Esta característica também é útil para

interagir com o ambiente do usuário (o editor de texto) e recarregar um documentos após algumas modificações, para ver como se parecem as modificações no navegador. Esta recarga é feita diretamente através de um botão de atalho na barra de ferramentas. Não há necessidade, portanto, de abrir uma caixa de diálogos e selecionar um nome de arquivo.

Estão presentes, também na barra de ferramentas, os botões de atalho de voltar e avançar, para que o usuário possa retroceder um passo, e avançar também um passo, na lista dos arquivos percorridos. Existe também um botão intitulado de "parar" que muda de cor quando um arquivo está sendo aberto, mostrando ao usuário que está havendo atividade no sistema e que pode estar levando algum tempo. Isto é útil para não deixar o usuário com a sensação de que alguma coisa possa estar errada porque não está acontecendo nada na tela.

Um recurso adicional é a possibilidade de abrir uma caixa de diálogo para alterar as características básicas de apresentação de texto, como cor da fonte, tamanho da fonte e cor de fundo. Isto pode ser feito e as modificações são refletidas automaticamente sem a necessidade de recarregar o documento ou o próprio sistema.

9.7. O módulo estatístico (planilha)

A Estatística é lecionada nos cursos universitários essencialmente a não-especialistas, tipicamente em disciplinas de 4 créditos (4 horas semanais) e por apenas um semestre. Nas Engenharias pode envolver seis horas e alguns cursos duas disciplinas de 4 créditos, mas quase nunca além disso. Cobrir todos os conteúdos ou grande parte deles dentro de tais restrições de tempo e espaço é sem dúvida uma façanha. Além do pouco tempo disponível, a disciplina se torna difícil não somente para o aluno, mas também para o professor, devido aos seguintes os motivos:

Primeiro, a Estatística, como um conhecimento aplicado, envolve conceitos abstratos e complexos, pelo fato de ser uma aplicação que envolve basicamente conhecimento matemático. Além disso, idéias específicas tais como aleatoriedade, distribuições amostrais e conclusões probabilísticas não são familiares à maioria dos alunos por não serem vistas em praticamente nenhum curso de nível médio ou disciplina anterior.

Segundo, a Estatística requer habilidades analíticas na formulação de problemas, na seleção e identificação de variáveis e na construção de modelos que normalmente não são fáceis de abordar ou lecionar, pois a maneira mais efetiva de ensinar estas habilidades, como de resto as demais, é praticar mais, mas isto é difícil devidos às restrições de tempo.

Terceiro, aprender a utilizar as ferramentas técnicas da Estatística requer algumas habilidades matemáticas específicas, mas conforme já mencionado, mesmo estudantes mais preparados nesta área, como os de Engenharia, apresentam grandes deficiências em sua formação matemática.

Finalmente, o uso efetivo da Estatística requer a habilidade para sintetizar e transferir o conhecimento adquirido a outras áreas e isto é praticamente impossível com o ensino voltado ao micromundo enfatizando técnicas e procedimentos limitados em escopo e domínio.

Em virtude destas dificuldades, o desenvolvimento de habilidades técnicas que sejam de uso efetivo para a maioria dos estudantes não-especializados parece um objetivo inacessível em cursos introdutórios de Probabilidade e Estatística. Por isto é que o modelo hipermédia multiagentes é adequado ao ensino da disciplina, mas não é

ainda suficiente. Neste trabalho, propõe-se ainda a extensão do modelo através de uma planilha com características inovadoras.

A planilha pretende ser uma interface inteligente para a elaboração de cálculos estatísticos da forma como são lecionados e não da forma que *outros* softwares o fazem. Neste caso, o que se está fazendo é adaptar o sistema ao ensino e não tentar adaptar os alunos ou o ensino ao sistema.

A planilha incorpora inteligência na identificação das variáveis, através de sua identificação automática. Isto é, não será o usuário que informará ao sistema o tipo de variável que está sendo utilizada, como acontece nos sistemas atualmente em uso, mas sim o sistema que as reconhecerá de forma automática. Este procedimento é básico para que se possa evitar que o usuário inexperiente use o sistema para realizar cálculos e análises sem sentido, que ocorre com os sistemas tradicionais.

Uma vez que o sistema identifique o tipo de variável sendo considerada, ele fornecerá ao usuário (através de menus flutuantes, por exemplo) somente aquelas opções que forem pertinentes àquele tipo de dado, evitando desta forma a realização de cálculos estatísticos que violem hipóteses básicas sobre a escala de medida envolvida. (Como, por exemplo, que o usuário calcule média e desvio padrão de uma variável categórica). Nos sistemas atuais não existe um mecanismo que evite que tais tipos de cálculos sejam executados e sejam apresentados pelo sistema como se fossem análises perfeitamente legais.

9.7.1. Os Sistemas Especialistas e a Estatística

Os esforços para colocar inteligência em softwares de análise de dados começaram no início da década de 80. E no início da década de 90 os ambiciosos objetivos articulados na época ainda não haviam sido alcançados [PRE91].

Gérard Hatabian e Hervé Augendre [HAT91] colocam que a principal função de um ESS (*Expert Statistical System*), que seria o resultante de um Sistema Especialista (*Expert System*) e um Sistema Estatístico (*Statistical System*), não seria representar o conhecimento de um campo em particular, mas guiar o usuário na utilização do software estatístico. Já um ISS (*Intelligent Statistical Software*) seria um software na qual a representação do conhecimento não está explícita.

O sistema proposto seria enquadrado nesta segunda categoria, uma vez que o conhecimento está implícito no software, principalmente na interface. O sistema não pretende ser um conjunto de regras que auxiliam o usuário a escolher entre este ou aquele teste estatístico. Isto de nada adianta se o usuário não souber por que este teste é o adequado ou recomendado e principalmente por que está aplicando um teste ou fazendo uma determinada análise.

O que se faz necessário, e isto foi percebido em muitos anos lecionando, é que o aluno entenda o que está sendo feito e, só então, saiba quais recursos (técnicas) existem para melhor explorar ou resolver a situação. Se o software apenas disser qual a técnica que deve ser usada, sem despertar a compreensão do conjunto ou do que está sendo realizada e com que propósito, a situação apenas vai mudar de um conselheiro humano para um eletrônico. Técnicas estatísticas para lidar com determinadas situações só devem ser vistas quando o estudante perceber a necessidade delas.

É preciso, antes de mais nada, enfatizar e mostrar a necessidade de um tratamento estatístico em situações reais. Algumas destas situações consistem de aplicações específicas ao campo de estudo particular do estudante, mas outras envolvem situações de interesse geral ou experiências da vida diária [YIL96].

A análise não sendo centrada em procedimentos e sim nas variáveis e no que pode ser feito com uma (duas ou mais) delas diminui sensivelmente a curva de aprendizagem e manejo do software. Rolf Biehler [BIE93] coloca que "*cada software estatístico introduz uma linguagem idiossincrática própria, conceitos e objetos que precisam ser aprendidos e dominados antes que a ferramenta possa ser razoavelmente utilizada. As pessoas estão certas em considerar esta característica como um obstáculo e não como uma novo potencial para aprendizagem e prática*" (p 75). . E acrescenta: "*pode-se observar uma reorganização qualitativa não intencional da atividade estatística quando se utilizam recursos computacionais: os estudantes não estão mais realizando qualquer trabalho ou pensamento estatístico, mas estão inteiramente ocupados com as tecnicidades de uma interface ou com os detalhes de uma linguagem de comandos*" (p.76).

9.7.2. As formas tradicionais

O uso de pacotes computacionais dispensou professores e alunos dos cursos de Estatística e Probabilidade do tédio provocado pela excessiva atividade braçal em executar um grande número de cálculos irrelevantes que não acrescentavam nada em termos de aprendizagem. Softwares convencionais como: Minitab, SAS, SPSS, Statgraphics, Statistica e Systat entretanto, não têm auxiliado muito no entendimento dos conceitos subjacentes. Os estudantes não refletem sobre o que estão fazendo e nem por que o estão fazendo. Simplesmente lançam os dados e aplicam o teste [STE91].

Sistemas especialistas na área de Estatística em geral têm se preocupado em auxiliar na escolha da técnica estatística a ser aplicada. Isto é feito através de uma série de questões que são colocadas ao usuário e à medida que ele as responde o software vai descendo em um diagrama de árvore e se aproximando da técnica correta. O único problema que esta abordagem acarreta é que ela parte do pressuposto que o usuário saiba e vai responder corretamente a todas as questões propostas. Vejamos um exemplo de um diálogo típico "computadorxusuário" de um destes "sistemas especialistas". O exemplo abaixo é do sistema SPRINGEX, que é caracterizado como um módulo adicional especialista para o sistema TURBO SPRING-STAT. O propósito do "sistema especialista" é auxiliar o usuário no manejo dos dados e nas necessidades analíticas com os procedimentos disponíveis no sistema TURBO SPRING-STAT. O sistema foi escrito em TURBO PROLOG e foi baseado num exemplo de um sistema especialista do guia de referência do turbo Prolog denominado GENI [RAE92].

A seguir, a tela de abertura do sistema e algumas questões colocadas ao usuário:

Bemvindo ao SPRINGEX, um recurso para ajudá-lo a selecionar procedimentos Spring-Stat. Use o menu à direita para responder cada questão.	Sim Não Porquê
É verdade que você quer rodar um procedimento ESTATÍSTICO:	Sim
É verdade que você quer entrar/revisar/editar dados:	Não
É verdade que você tem UMA variável DEPENDENTE:	Não
É verdade que você quer ASSOCIAR para ver tanto os objetos quanto os RÓTULOS das variáveis:	Não
É verdade que seus dados estão mensurados em escala de INTERVALO, pelo menos:	Não
É verdade que você quer PLOTAR até 10 objetos em 3 variáveis:	Não

É verdade que sua variável DEPENDENTE é categórica (membro de um GRUPO):	Não
É verdade que você salvou um arquivo MANOVA [.DAN]:	Não
É verdade que você salvou um arquivo CORRELAÇÃO [.CRM] no formato matricial:	Não
É verdade que você quer analisar EXATAMENTE 2 variáveis:	Não
É verdade que você quer analisar UMA variável de cada vez:	Não
É verdade que você salvou um arquivo no formato BASE DE DADOS [.HDG]:	Sim
É verdade que a distribuição é NÃO-NORMAL e dos dados não estão em escala de INTERVALO:	Sim
É verdade que seus dados tem o MESMO N para as variáveis/grupos:	Sim

Supondo que o pretendo usuário tenha conseguido responder a todas as questões acima sem a necessidade de terminar o curso de Estatística, o "sistema especialista" daria seu "conselho":

Eu acho que você quer rodar: [Não-par] FRIEDMAN ou KENDALL

Ora, o suposto especialista computacional é construído para auxiliar no uso do sistema e na seleção de uma técnica estatística, mas em contrapartida exige que o usuário tenha conhecimentos mais do que suficientes para chegar a mesma conclusão que ele (sistema especialista) por si próprio! Se um hipotético usuário do sistema acima tivesse habilidades suficientes tanto em estatística quanto no uso do software para responder a todas as questões ele certamente não precisaria de um especialista, pois ele seria o especialista!

Bem, pode-se argumentar que este é um exemplo antigo e que hoje as coisas são diferentes. Veja-se então um outro exemplo descrito em [GRA96] sobre o diálogo com o usuário de um software com objetivos semelhantes, isto é, selecionar o tipo de teste estatístico, onde o conteúdo focado é o "teste de hipóteses", que é parte integrante de praticamente todos os cursos de estatística básica:

Que tipo de variável resposta você tem?

Quantas amostras?

Qual é o tamanho da amostra (n)?

Você possui uma alternativa bilateral?

A variância é conhecida?

Os dados são emparelhados?

Pode-se ver que o diálogo é bem mais simplificado e bem mais objetivo que o anterior. No entanto, ele ainda não eliminou a necessidade de o usuário tomar decisões ou possuir um conhecimento que ele supostamente ainda não tem e que teoricamente o sistema deveria estar ensinando. Este, entretanto, não é principal problema que vejo aqui. Para mim o mais grave é o enfoque restrito, que reforça a aplicação mecânica dos

conhecimentos. A única diferença é que agora o aluno fará isto com o auxílio da máquina e não às suas próprias custas.

9.7.3. A alternativa proposta

Como forma de encaminhar alguns dos problemas dos sistemas especialistas tradicionais na área de Estatística, o sistema proposta está sendo construído, não para demandar questões específicas por parte do aluno e sim para que ele tome decisões mais gerais, decisões globais. As decisões específicas como as sobre tipos de variáveis, tamanho de amostra, etc., serão tomadas automaticamente pelo sistema ou solicitadas ao usuário de forma indireta, reforçando o conhecimento global e não o "micro" conhecimento que reforça o caráter do ensino tradicional voltado a aplicação mecânica de fórmulas.

Desta forma, a proposta não é solicitar que o usuário informe que tipo de dados ou variáveis ele possui e sim que inicialmente ele clique com o mouse sobre uma ou mais colunas da planilha, selecionando uma ou mais variáveis. A partir desta seleção, a um clique do botão direito do mouse, um menu flutuante, figura 9.1, é apresentado, supondo que apenas uma coluna tenha sido selecionada.

As alternativas oferecidas, já que uma variável numérica, no caso quantitativa discreta, foi selecionada são:

Executar uma análise:

Gráfica;
Descritiva ou
Inferencial.

Agrupar (isto é, construir uma distribuição de frequências).

Um clique na alternativa "Descritiva", por exemplo, remeteria o usuário para um submenu com as seguintes opções:

De tendência ou posição central;
De dispersão ou variabilidade;
De assimetria e
De achatamento ou curtose.

As alternativas para a opção "De tendência ou posição central", estão logo abaixo no próprio menu, para evitar o excesso de submenus e consistem de:

Média;
Mediana;
Moda ou
Todas (permitindo o cálculo das três anteriores).

Já as opções para a alternativa "De dispersão ou variabilidade", consistem em:

Variância;
Desvio padrão;
Desvio médio;
Variância relativa;
Coeficiente de variação;
Todas (as consideradas nesta categoria).

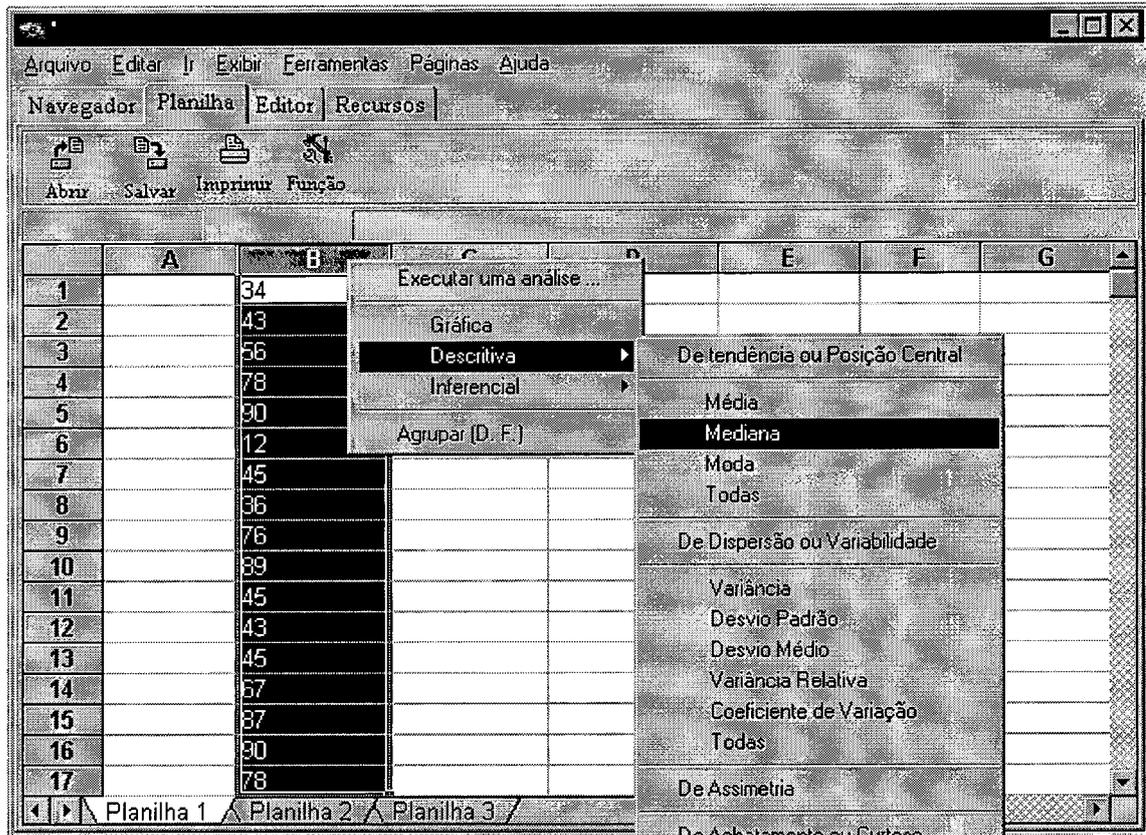


Figura 9.3 - Planilha mostrando o menu flutuante com as opções ativas

Já se uma variável qualitativa for selecionada, será necessário a interferência do usuário para informar se esta variável é "nominal" ou "ordinal", uma vez que, neste caso, o sistema não tem condições de fazer a identificação de forma automática. Com a variável identificada as opções das caixas de diálogo ficariam limitadas às pertinentes. Por exemplo, uma análise descritiva, não forneceria as opções de cálculo da "média" e "mediana", mas poderia apontar a "classe" ou "categoria" modal.

Acredito que se possa ter uma boa visão das intenções do sistema com os exemplos apresentados acima. Neste caso o que se está procurando enfatizar e salientar são os conceitos gerais, e cada opção oferecida reforça esta idéia.

Obviamente o sistema poderá eventualmente oferecer opções de atalho para o usuário mais experiente ou que já tenha incorporado as idéias básicas. Só que isto seria uma opção a ser considerada posteriormente, pois o objetivo é que ele seja utilizado por alunos que enfrentam pela primeira vez um curso de Estatística Introdutória e neste caso todo e qualquer reforço conceitual nunca será excessivo. Além disso o sistema oferecerá uma descrição (resumo), de forma opcional, através de um botão (ou menu) intitulado "você quer ajuda" ou "e agora ...".

A inteligência do sistema estará embutida nas opções fornecidas ao usuário, que serão somente aquelas pertinentes àquele tipo de dado e nada mais. Hoje, o usuário acessa através dos menus todas as ferramentas e recursos disponíveis no sistema para aplicar inteiramente ao seu critério, não importa se quem o esteja manipulando seja um especialista ou algum aluno sem experiência tentando descobrir o que ele pode fazer.

9.8. O módulo gráfico

A parte gráfica do modelo proposto é formada por um recurso que nenhum curso *online* através dos navegadores tradicionais pode oferecer. Neste caso, o máximo que pode obter são *applets* Java que normalmente demoram para rodar e que muitas vezes apresentam erros, frustrando o usuário.

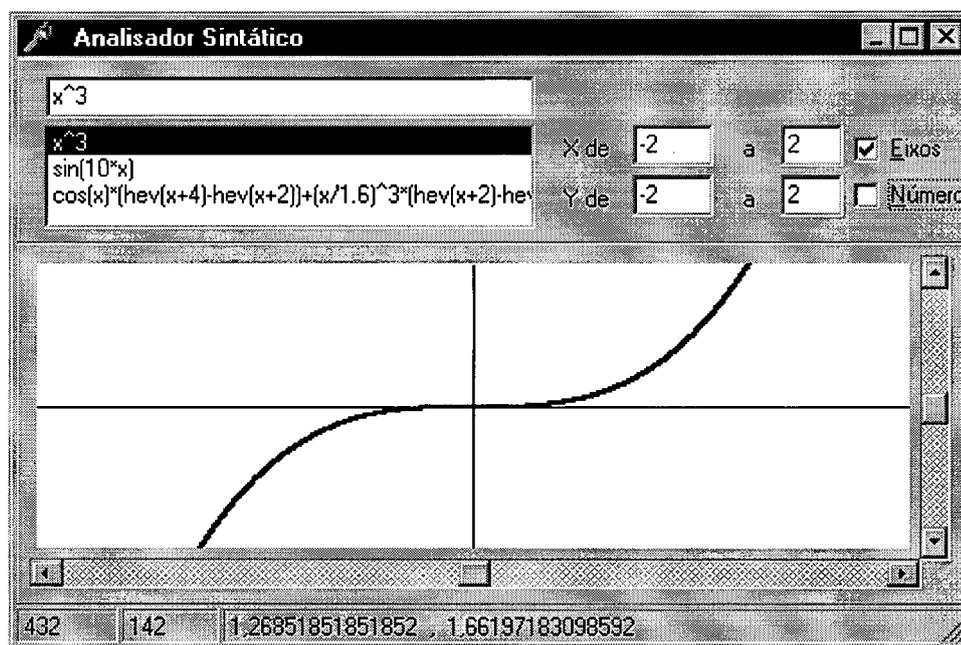


Figura 9.4 - Uma visão do analisador sintático do sistema

O editor gráfico é composto de três painéis distintos. Um analisador sintático, um módulo para diagramas em 2D e um módulo para diagramas em 3D. O analisador sintático pode mostrar imediatamente qualquer equação digitada pelo usuário, além das pré-incluídas no sistema. É possível interagir com o editor, fornecendo os intervalos de variação desejados e representando um único diagrama ou vários simultaneamente num mesmo sistema cartesiano. As opções de diagramas em 2D e 3D são oferecidas através de um componente ActiveX denominado Olectra.

9.8.1. Gráficos dinâmicos interativos

Os estudantes precisam saber não somente onde as técnicas funcionam mas também onde elas falham e o porquê. Eles podem adquirir mais experiência tentando novas abordagens em muitos exemplos. Obter esta experiência, sem o aborrecimento de cálculos exaustivos ou extensiva programação, requer uma ferramenta computacional rápida e relativamente fácil de usar. Tal necessidade pode ser suprida por um sistema computacional interativo. Se mecanismos de gráficos dinâmicos forem adicionados, os resultados de tais análises podem ser mais rápido e facilmente absorvidos [AND88]. Tradicionalmente, os gráficos não suportam interação ou quando suportam ela é muito limitada; eles são meramente imagens para serem olhadas. No entanto, pode-se fazer um pouco melhor do que isto. O módulo gráfico do sistema incorpora um mecanismo gráfico, onde qualquer gráfico pode ser manipulado de inúmeras maneiras. Pode-se passar de

uma visão de duas dimensões para uma de três, girar e rotar a visão tridimensional, com o auxílio do mouse, para qualquer ângulo ou perspectiva desejado.

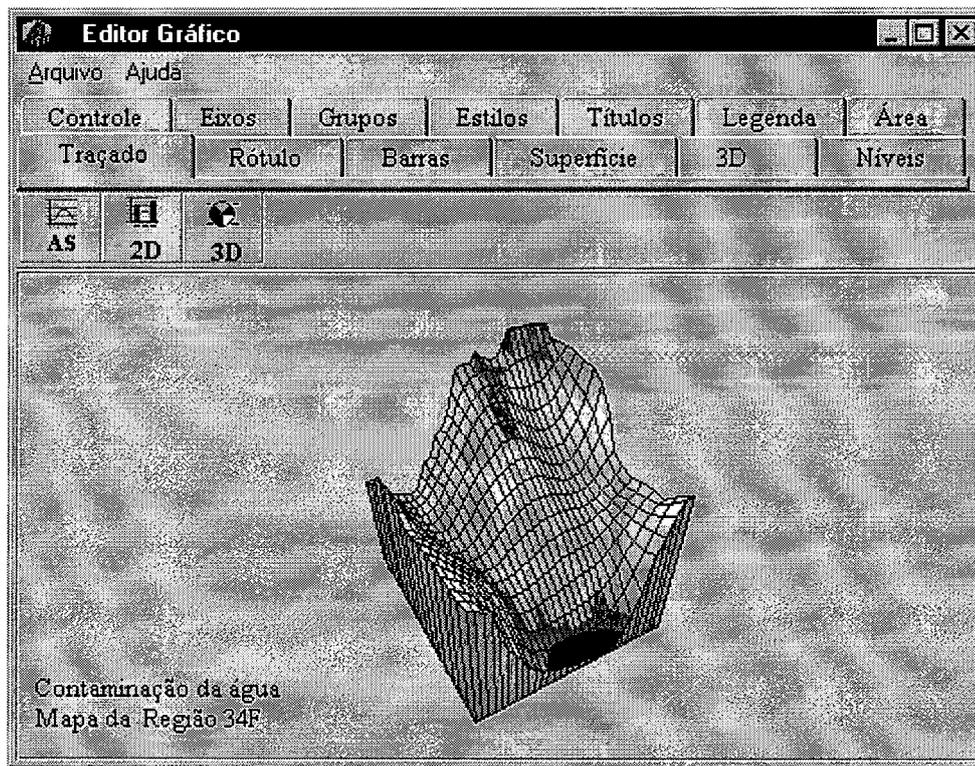


Figura 9.5 - Uma visão do módulo gráfico do sistema

O reconhecimento do potencial da manipulação direta, com resultados em tempo real, de gráficos para análise de dados remonta ao início dos anos 60, quando apareceram os primeiros sistemas computacionais gráficos. No final dos anos 60 foi desenvolvido o primeiro sistema dinâmico para desenhar curvas probabilísticas. Pela rotação de um mouse esférico, o usuário podia continuamente mudar o parâmetro forma para a distribuição de referência exibida na tela. O mouse podia ser roteado para tornar o padrão de pontos tão retos quanto possível. O sistema demonstrou que a técnica dos gráficos dinâmicos tinha um bom potencial para a análise de dados [BEC88]. Os gráficos dinâmicos são muito semelhantes aos estáticos, exceto que certos aspectos, podem ser alterados rapidamente sem a sua completa reconstrução. Os gráficos podem ser construídos seqüencialmente, com o processo de construção sendo parte do exercício de aprendizagem [CUR96].

9.8.2. Conceito e métodos

A técnica dos gráficos dinâmicos possui duas propriedades importantes que a tornam apropriada ao processo de ensino de disciplinas como Estatística que lidam com dados: uma é a manipulação direta dos elementos gráficos na tela do computador e a outra é a mudança virtualmente instantânea dos elementos. O usuário executa uma ação através da manipulação manual de um mecanismo de entrada (mouse, teclado) e observa o resultado em tempo real na tela.

A figura 9.6, mostra um exemplo na qual um método dinâmico é utilizado para mostrar e esconder legendas. Quando o usuário move um retângulo sobre um ponto do gráfico cobrindo-o, sua legenda aparece e assim que o retângulo é removido a legenda

desaparece. Acredita-se que esta técnica é útil para aumentar o aproveitamento do usuário do sistema, pois pode-se perceber mais e melhor quando se interage com o gráfico, principalmente se ele reage instantaneamente, do que meramente olhar. Isto não significa que os métodos estáticos tradicionais não serão utilizados, mas que este método virá somar no esforço de proporcionar um ambiente mais produtivo para o entendimento dos conceitos sendo apresentados ao usuário do sistema.

Gráficos dinâmicos podem ser implementados através de diversos métodos. Entre eles podem-se destacar: identificação, exclusão, conexão, escalonamento e rotação. Informações adicionais sobre o assunto podem ser vistas em [BEC88, STU88].

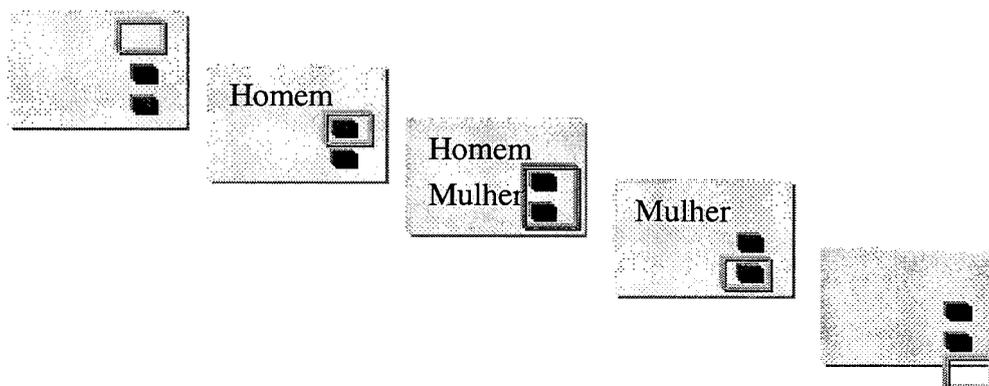


Figura 9.6 - Um exemplo de gráfico dinâmico [BEC88]

9.9. O módulo de comunicação

O módulo de comunicação é formado por três componentes. Um dos componentes é o de correspondência eletrônica (*e-mail*) que objetiva a comunicação entre os usuários do sistema. Um segundo componente é o de conferência online (*chat*) entre os usuários do sistema e o terceiro componente (a ser implementado posteriormente) objetiva fornecer condições para a vídeo conferência.

9.9.1. A correspondência eletrônica

Este recurso permite que dois usuários do sistema troquem correspondência entre si. Também poderá ser utilizado para acessar um painel, no local (*site*), com dicas de outros usuários sobre aplicações. Poderá ser utilizada para a distribuição de informações sobre a disciplina, bem como de alterações de datas de provas, entrega de trabalhos, etc.

9.9.2. A conferência online

O propósito da conferência *online* é que dois os mais alunos da disciplina possam discutir conteúdos e/ou exercícios sem a necessidade de deslocarem para um ponto comum. Poderá ser utilizado para acessar o professor, caso este esteja disponível, ou então monitores da disciplina.

9.9.3. A vídeo conferência

O sistema apresenta um módulo, não implementado ainda, para o acréscimo futuro de vídeo conferência. A vídeo conferência poderá ser utilizada para que o aluno possa assistir as aulas sem a necessidade de se deslocar a universidade.

9.10. O editor

O editor na primeira versão apresenta a possibilidade de visualizar e editar o código fonte do arquivo que está sendo apresentado no navegador. É possível imprimir, salvar e salvar com um novo nome o arquivo sendo editado. A grande vantagem desta abordagem é a possibilidade do autor do material didático realizar correções no material sendo apresentado no navegador sem a necessidade de recursos ou editores de texto adicionais, num procedimento extremamente prático, que apresenta vantagens sobre os sistemas existentes, que exigem o uso de no mínimo dois pacotes diferentes. Observado um erro qualquer no material, basta clicar o botão do editor e imediatamente será apresentado o código fonte do arquivo e a correção pode então ser realizada. Para visualizar o material corrigido, basta salvar o arquivo no editor e voltar, então, para o navegador, fazendo a recarga (*reload*) do arquivo.

Numa segunda versão, o editor deverá agregar novas características, como a possibilidade de visualização idêntica a do navegador, mas com a diferença de possibilitar a edição do arquivo, coisa que no navegador não é possível fazer.

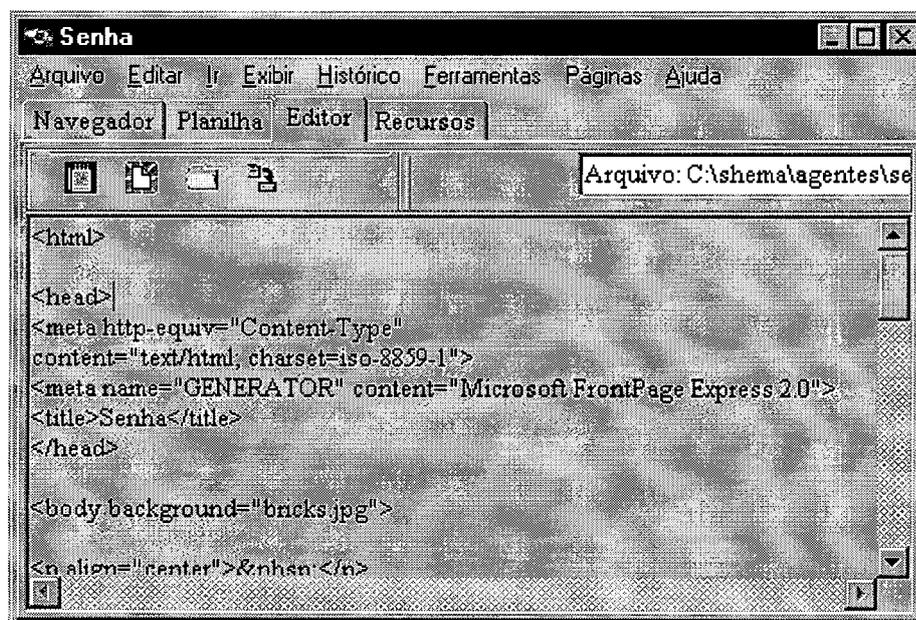


Figura 9.7 - O módulo editor do sistema

9.11. Características adicionais

O uso da simulação já se tornou uma ferramenta tradicional no ensino de Probabilidade e Estatística [DER95, CHA92, STE91]. No modelo proposto ela também será utilizada. Por simulação, neste caso, entende-se basicamente a aplicação do método Monte Carlo, isto é, a geração de números pseudo-aleatórios e distribuições de probabilidade. Uma grande parte da Estatística consiste fundamentalmente na descrição, resumo e interpretação de dados. Coletar dados não é uma tarefa de todo fácil e, além disso, é um trabalho que consome tempo. Num sistema de ensino eficaz os dados devem estar prontamente disponíveis e isto só pode ser feito através da simulação destes dados.

9.11.1. Interatividade

Os recursos mencionados anteriormente estarão disponíveis para a análise estatística. No entanto, sempre que se lecionam disciplinas de Estatística também se leciona Probabilidade. Tópicos de Probabilidade são pré-requisitos para se abordar inferência estatística. A probabilidade pode ser abordada com o auxílio dos sistemas através da geração dinâmica das principais distribuições de probabilidade. O sistema pode gerar graficamente as principais distribuições contínuas (Normal, t, Uniforme, Exponencial, etc.) e as principais distribuições discretas (Binomial, Poisson, etc.). Esta geração é interativa, isto é, o usuário dispõe de controles deslizantes que poderá manipular para verificar o comportamento da distribuição com a variação dos parâmetros. Optou-se por esta abordagem por julgá-la superior à tradicional de deixar o usuário fornecer valores em entradas editáveis, pois, neste caso, o usuário inexperiente pode ter que tentar inúmeras vezes até fornecer os valores corretos dos parâmetros. Com algumas repetições da experiência o aluno terá uma visão clara e completa de todas as formas possíveis da distribuição, evitando desta forma a visualização distorcida fornecida pelos livros textos que apresentam quase que invariavelmente apenas o formato típico da curva negligenciando os casos extremos ou variações atípicas.

A distribuição normal, por exemplo, é invariavelmente apresentada como uma curva em forma de sino, que é verdadeiro apenas em parte pois dependendo dos valores do desvio padrão, isto é, se ele for muito grande ou muito pequeno, isto poderá não ser verdadeiro. Com um experimento simples o usuário poderá em alguns segundos verificar que alterando o parâmetro σ (desvio padrão) a distribuição normal poderá passar da forma de sino tradicional (a apresentação default) para uma linha reta horizontal, num extremo, quando $\sigma \rightarrow \infty$ a uma linha reta vertical no outro extremo quando $\sigma \rightarrow 0$.

9.11.2. A simulação

A simulação é um modelo munido de uma interface. Esta interface permite que o usuário interaja com o modelo e veja o comportamento resultante do modelo. Uma das vantagens da simulação (sobre o sistema real) é que o comportamento (resposta) pode ser representado através de diferentes modalidades. Tipicamente a simulação fornece dados numéricos, gráficos e animação de variáveis. Estas variáveis podem ser as reais, do modelo, ou valores que podem ser mais significativos para o usuário, derivados de tais variáveis através de algum tipo de transformação.

A representação mais comum do estado de um modelo é a representação gráfica de uma variável contra o tempo. Isto pode ser feito de muitas maneiras diferentes, colocando mais de uma variável num mesmo gráfico e contrastando duas variáveis (uma contra a outra). No entanto, a característica essencial da representação gráfica é que ela mostra num único diagrama as mudanças que ocorrem quando o modelo simulado é executado.

Para que o leitor possa tirar partido da simulação, ele deve poder controlá-la. O controle possui três aspectos: estabelecer ou mudar as condições iniciais, atribuir valores aos parâmetros do modelo e especificar a forma dos estados sucessivos do modelo e seus comportamentos pela duração de uma replicação. Seria desejável ainda interagir com o sistema enquanto ele está sendo executado, fornecendo, desta forma, um controle dinâmico da situação. Tradicionalmente, a simulação seria utilizada como um último recurso, pelos menos teoricamente, mas o que se tem observado [VIA91] é que a simulação é largamente empregada em todas as áreas do conhecimento como uma das principais técnicas de análise de sistemas discretos, como corroboradora de soluções analíticas e fundamentalmente como fornecedora de dados na realização de experimentos, so-

luções de problemas e demonstração de propriedades e mesmo na verificação de teoremas. Neste caso, a técnica empregada é a Monte Carlo, ou seja, a geração de valores e distribuições aleatórias.

9.11.3. Utilização educacional da simulação

A Tabela 9.2 [BRE90] fornece um panorama da simulação em aplicações instrucionais, considerada em referência às condições de instrução.

Tabela 9.2 - Usos educacionais da simulação

Condições da Instrução	Demonstração	Treinamento	Modelagem	Representação ambiental
Atividades de Aprendizagem	Observação das mudanças no sistema.	Aplicação de procedimentos.	Alteração de variáveis e inter-relações.	Tomada de decisões orientadas pelo problema.
Objetivos Educacionais	Conhecimento de estruturas e da dinâmica de sistemas complexos.	Internalização de procedimentos definidos.	Estratégias de descobertas científicas.	Conhecimento e refinamento de habilidades e desenvolvimento de heurísticas.

9.11.4. Razões para utilização da simulação no ensino

A simulação é uma ferramenta tutorial valiosa pelas seguintes razões [MER96]:

- ☐ Envolve menos riscos que a realidade. Erros, que em sistemas reais seriam fatais, numa simulação podem no máximo causar uma pequena frustração. O erro se transforma em experiência e tenderá a ser repetido cada vez menos.
- ☐ Os custos de treinamento são reduzidos. Embora alguns sistemas de simulação tenham um custo inicial alto, a longo prazo eles certamente sairão mais barato do que o treinamento no objeto real.
- ☐ Ela é normalmente mais conveniente do que o treinamento real, pois possibilita geralmente o treinamento de mais estudantes ao mesmo tempo.
- ☐ As experiências em simulação podem ser repetidas. Os estudantes podem repetir uma experiências tantas vezes quanto o necessário para entendê-las e enfrentá-las com habilidade.

Muitas dificuldades de aprendizagem ocorrem em virtude de se possuir apenas uma visão estática da representação de sistemas naturais ou artificiais. Com um livro esta é a única maneira possível. No entanto com o computador os modelos podem e devem ser dinâmicos.

9.12. Conclusão

Como forma de se colocar inteligência nos softwares estatísticos, o que se está fazendo é colocando sistemas especialistas, ou os denominados conselheiros (*wizards*), que através de interrogação do usuário pretendem conduzi-lo à decisão correta sobre o tipo de técnica que deve ser utilizada. Talvez para um sistema que pretenda analisar da-

dos, este tipo de enfoque seja conveniente. No entanto, este tipo de inteligência não é apropriado para um sistema que pretende lecionar estatística, pois primeiro solicita do usuário conhecimentos que ele normalmente não possui e segundo leva a uma visão segmentada do conteúdo, enfatizando a idéia de que a estatística é apenas um conjunto de regras que levam à seleção de uma determinada técnica ou fórmula.

Para fugir a estes preceitos tradicionais, o modelo proposto apresenta uma interface em que a maioria das decisões é feita automaticamente pelo sistema e enfatiza a visão de conjunto do conteúdo. O sistema também fornece, através de opções de ajuda sensível ao contexto, explicações do tipo "por que determinada opção não está ativada em determinado tempo".

Toda nova tecnologia, para ser aplicada ao ensino, envolve uma fase de adaptação. Com a WWW não foi diferente. Apesar de existir um conhecimento acumulado bastante grande da utilização do computador, no ensino este conhecimento se referia basicamente a tutorais, isto é, esquemas mais ou menos rígidos onde o computador era usado ou tentava ser usado como um substituto do professor. Eram sistemas de ensino pouco flexíveis que tentavam adaptar ao novo meio as técnicas de ensino tradicionais.

Com a tecnologia do hipertexto e o surgimento da WWW permitindo acesso imediato a base de conhecimentos situadas em qualquer parte do mundo, esta postura precisa e está sendo revista. O ensino através destes meios deve mudar inteiramente seu foco do professor para o aluno. Com o conhecimento descentralizado e acessível diretamente na casa do aluno, um ensino individualizado que sempre foi um dos objetivos do ensino tradicional pode ser agora implementado. Não na concepção original de um professor para cada aluno, o que seria proibitivo em termos de custos, mas com o aluno interagindo muito mais com o conhecimento em si e o professor agindo apenas nas horas em que esta interação seja interrompida de uma forma ou de outra. Nestas situações, a comunicação com o professor pode ser feita via e-mail, via chat ou via uma teleconferência.

Um novo paradigma de ensino está surgindo, que é o de **aprender a aprender**. Ao invés de se tentar ensinar o aluno, o papel do professor é o de facilitador do acesso ao conhecimento, isto é, o professor deve incentivar e fornecer condições para que o aluno aprenda por si próprio. Ele deve fornecer os meios e as facilidades para que o aluno construa seu próprio conhecimento. O papel tradicional do professor mudará de agente transmissor do conhecimento para o de agente organizador do conhecimento, cabendo ao computador o papel de transmissor e interagente do conhecimento.

Ao invés de preparar as tradicionais aulas para serem passadas para o quadro e do quadro para o caderno de anotações dos alunos, o professor agora passará as aulas diretamente para um local na Internet, onde o aluno poderá acessá-las sem a etapa intermediária da passagem pelo tradicional "quadro-negro" que consome e desperdiça tempo tanto do professor quanto do aluno.

O argumento contra este novo sistema é que o contato direto com o professor é fundamental para que o aluno possa interagir e perguntar. No entanto, a experiência tem mostrado que poucos alunos universitários tiram proveito deste contato direto com o professor. A grande maioria, silenciosa, se limita a copiar os conteúdos do quadro, sem nenhum pensamento crítico e deixando para avaliar o material em casa quando for estudar para a prova. Ora, nesta hora o professor não está acessível, mas com o novo paradigma o aluno estabelece uma sessão *online* ou manda suas dúvidas por mail ou então consulta outro colega, e não precisa deixar a dúvida para a próxima aula quando já é dia de prova ou, então, quando ele já esqueceu qual era o problema.



10. CONCLUSÕES E RECOMENDAÇÕES

10.1. Conclusões

Com a informação crescendo substancialmente a todo instante não é possível ignorar um recurso como o computador para transmitir mais conhecimentos, de uma forma mais dinâmica, interativa e prazerosa. A tecnologia da hipermídia já está suficientemente fundamentada para que se possa utilizá-la com proveito para ensinar praticamente qualquer conteúdo. Cabe ao professor descobrir o que funciona melhor com seus alunos e com sua disciplina, de forma a motivá-los e tirar proveito tanto da atratividade exercida pelo computador quanto pela motivação despertada pela Internet e pela WWW.

Lecionar Estatística, como de resto disciplinas da área de Matemática, é uma tarefa difícil por vários motivos. Geralmente estas disciplinas envolvem abstrações que não encontram suporte em recursos visuais. Os alunos apresentam, quando não uma aversão completa pela disciplina, um desinteresse que dificulta tanto o relacionamento professor aluno quanto a aprendizagem. Melhorar este quadro, tornando o ensino mais visual, mais atrativo, mais analítico e mais abrangente, envolvendo perspectivas globais, é o propósito deste trabalho. Um das maneiras que acredito que isto possa ser feito é mudando substancialmente uma das principais ferramentas de suporte dos pacotes computacionais usados para analisar dados, que via de regra não são adequados para servirem como recurso didático. Estes pacotes reforçam a impressão transmitida pelo ensino tradicional, de que a Estatística é apenas um conjunto de técnicas analíticas aplicadas a certas situações ou conjuntos que satisfazem certas restrições que nunca ficam exatamente claras. O que se quer é deixar clara uma perspectiva que tudo o que está sendo apresentado tem um propósito específico e está interrelacionado. A grande maioria dos softwares estatísticos foram construídos com o objetivo de realizar cálculos, isto é, permitir que determinadas técnicas que demandariam computação exaustiva, e desta forma impraticáveis manualmente, pudessem ser aplicadas. Eles normalmente não foram projetados para lecionar, isto é, com objetivos pedagógicos claramente definidos. O leque de recursos é expressivo na maioria dos pacotes, no entanto, o que torna um pacote adequado ao ensino não é exatamente o que ele pode fazer e sim como ele faz.

Além dos softwares estatísticos direcionados à computação intensiva, podem-se utilizar as planilhas. A grande vantagem da planilha é que ela é programável e dinâmica. Só que como ela não foi projetada para servir como recurso pedagógico e muito menos para lecionar estatística. Para tal torna-se necessário fazer adaptações e estas adaptações não são sempre simples. Tanto é verdade que isto é atestado pela série de softwares construídos com a finalidade de torná-las adequadas ao ensino, e com alguns tornando-se de meros adendos a pacotes computacionais completos por si só.

Existem ainda os livros eletrônicos ou cursos multimídias, embelezados com sons e fotos que muitas vezes mais distraem do que atraem, disponíveis em grande profusão no mercado comercial e da Internet em muitas áreas além da Estatística, como Cálculo, Álgebra, Física, Química, etc. Nestes livros eletrônicos multimídia, a informação é apresentada em uma seqüência mais ou menos fixa e o estudante deve aprender absorvendo

toda e qualquer informação apresentada na tela do computador. Um teste típico deste tipo de software consiste de questões para o qual a resposta é correta ou não e que invariavelmente produzem uma mensagem do tipo "tente novamente".

Ao meu ver, num software multimídia os eventos e a seqüência do material sendo apresentado devem ser controlados pelo aluno. O material multimídia adicional, como o som e o vídeo e a animação, deve sempre ter o propósito de aumentar ou explicar ou ainda apresentar o material num formato diferente, e deve sempre estar sob o comando do usuário que em última instância decide se vê, houve, participa ou não. Ele não deve ser apenas um atrativo para embelezar ou atrair mas sim servir a um propósito pedagógico específico. O estudante deve sempre ter a última palavra sobre o quanto vai estudar, que material vai estudar em um determinado tópico e, é claro, advertido que para entender este ou aquele assunto é necessário que primeiramente fique bem assimilado tais e tais conceitos. Que ele possa, sempre que achar necessário, voltar atrás e revisar pontos que não ficaram claros e por isto é necessário que ele possa passar de um tópico para outro a qualquer tempo ou ponto.

As questões ou testes devem sempre apresentar além do "correto" e "errado", as razões ou do porquê aquele é o resultado correto e não aquele outro. Além disso, oferecer sempre uma alternativa de "dica" ou "pista" antes de fornecer a resposta e, como etapa final, a "solução" completa do exercício, para que ele possa ver o caminho percorrido para chegar lá, de forma que possa aplicá-lo numa situação semelhante.

O propósito do sistema, ao menos na sua primeira fase, é somente demonstrar funcionalidade, ou seja, que um sistema hipertexto pode ser aplicado com vantagens ao ensino de disciplinas da área de Ciências Exatas e que pode ser construído como um híbrido entre um sistema de hipermídia e um programa computacional clássico de Estatística.

Neste trabalho apresenta-se um modelo de agentes para auxiliar no ensino de conteúdos que são transmitidos praticamente da mesma forma desde que a escola foi instituída. Acredito que sua utilização possa aumentar a motivação dos alunos pelas disciplinas de Estatística/Probabilidade, melhorando, desta forma, o rendimento e colaborando para um aumento da produtividade no ensino.

Entre as contribuições que se acredita sejam relevantes neste trabalho, pode-se destacar uma classificação e justificativa da utilização do computador no ensino; um levantamento e classificação dos sistemas de hipertexto/hipermídia na educação, e particularmente no ensino de estatística, onde foram tentadas novas abordagens; uma análise dos recursos computacionais atuais para lecionar Estatística, na qual se observa que os pacotes existentes ou planilhas não satisfazem totalmente as necessidades para um ensino de graduação para grupos grandes e heterogêneos de alunos, normalmente despreparados. É feita uma justificativa do uso de gráficos dinâmicos e da interatividade para melhorar a aprendizagem, pois, como o colocado por Gordon [GOR95], *"o maior desafio atualmente enfrentado por pesquisadores da Estatística educacional é melhorar nosso entendimento da aprendizagem (online)"*.

O principal resultado do trabalho é apresentação de um modelo multiagente para dosar e calibrar os conteúdos sendo apresentados ao aluno e para principalmente detectar e classificar o usuário do sistema de forma que ele receba exercícios de acordo com a sua capacidade, mantendo-o, desta forma, estimulado a prosseguir para patamares mais altos de dificuldades. Os agentes podem ainda classificar as aplicações de modo a adequá-los ao perfil do usuário do sistema e resolver eventuais dúvidas através do fornecimento de dicas e exemplos semelhantes.

Já o modelo proposto integra um navegador convencional, acrescido de facilidades para estudar, com uma planilha de interface inteligente num mesmo sistema, que é integrado, ainda, por um módulo gráfico e um de comunicação, mais um editor de hipertexto onde alterações e correções podem ser feitas e visualizadas imediatamente. Ao invés de se tentar novas abordagens com resultados duvidosos, foi feito o uso de paradigmas de interface e programação que comprovadamente já mostraram, através de sua larga aceitação e popularidade, que funcionam. Casou-se o sistema de hipertexto mais popular da atualidade e talvez o software com maior utilização no mundo atualmente, que é o sistema de hipertexto da rede mundial, com o da planilha, também já sedimentado e de largo uso, combinados com uma interface inteligente para a planilha e do sistema multiagentes. O sistema vai dirigindo o aluno em função do tipo de análise pretendida para a obtenção dos resultados através de menus inteligentes que são auto-aplicáveis em função da identificação automática do tipo de variável sendo utilizada.

A Estatística rapidamente tirou vantagem da computação ágil e barata. Métodos antigos, como a regressão, estão agora equipados com uma grande variedade de ferramentas de diagnóstico. Classes de modelos mais gerais, tais como os lineares generalizados e os aditivos generalizados, descrevem uma grande variedade de fenômenos. As técnicas de *bootstrapping* e subamostragem produzem estimativas de erros e intervalos de confiança em situações que antes eram intratáveis. A cada ano aparecem novas formas de analisar dados pela aderência de classes gerais de funções. A natureza, tanto da prática quanto da pesquisa estatística, mudaram dramaticamente sob o impacto da tecnologia.

A forma de lecionar, entretanto, mudou muito pouco. Como observado por MOORE [MOO95], "*nosso modo fundamental de interagir com os estudantes continua como sempre foi*" e acrescenta "*a revolução computacional não alterou nem a natureza do ensino e nem tampouco a produtividade dos professores*" (p. 298).

O propósito deste trabalho é dar uma contribuição para que de alguma forma o ensino de Estatística possa iniciar também a sua revolução, a exemplo do que ocorreu com a técnica e a pesquisa.

10.2. Recomendações

O trabalho apresenta um modelo multiagentes que pode ser utilizado para lecionar a distância ou então como um recurso de auto-instrução. Desta forma existem muitas maneiras de aperfeiçoá-lo, expandi-lo e aplicá-lo. O leque de disciplinas em que ele poderá ser utilizado abrange principalmente as da área de Matemática e de Matemática Aplicada. No entanto, nada impede que ele possa ser utilizado em disciplinas de outras áreas, bastando que estas disciplinas envolvam ou requeiram abstração e que necessitem, desta forma, que o aluno pratique os conceitos, isto é, realize muitos exercícios.

Entre as disciplinas sugeridas em que o modelo proposto poderá ser utilizado, além da aplicação proposta neste trabalho que é o ensino de Estatística e de Probabilidade. Estão os diversos Cálculos, a Álgebra Linear e a Geometria Analítica, bem como, em uma variedade de disciplinas dos cursos de graduação das áreas humanas e sociais que envolvam conteúdos básicos de Matemática como as disciplinas de Complementos de Matemática, por exemplo.

A inteligência dos agentes poderá utilizar outros modelos além do proposto aqui, pois o leque de possibilidades é extenso já que as pesquisas em modelos neuro-difusos

são recentes. Muitas outras propostas de modelos neuro-difusos poderão ser tentadas. O acréscimo de algoritmos genéticos, por exemplo, já foi mencionado e poderá ser utilizado. Da mesma forma, o modelo poderá ser ampliado com o acréscimo de novos agentes. Pode-se propor agentes móveis, na forma de mensagens, que se encarreguem de avisar o usuário quando alguém conseguir resolver um exercício mais difícil ou que passe entre os membros da turma alguma dica que possa ajudar em alguma tarefa mais complicada. Também, um agente que se encarregue de manter um cronograma das atividades da disciplina e sempre que necessário envie uma mensagem ao aluno para lembrá-lo da data de uma prova ou da entrega de um trabalho ou ainda de alguma atividade como palestra ou seminário relacionado ao conteúdo.

Na versão inicial muitas das características desejáveis e recomendáveis de um sistema de software educacional são sacrificadas em função da funcionalidade. Então é recomendado o aumento da capacidade do sistema pela agregação de novas habilidades e a atualização constante na interface para o acompanhamento da tendência mundial na construção de software. Isto não será difícil, pois além da arquitetura modular o sistema é construído através de componentes de software (componentes Delphi e componentes ActiveX). Estes componentes poderão ser atualizados, ou até mesmo trocados, sem a necessidade de alterar todo o sistema.

O sistema foi projetado para lecionar determinadas disciplinas para alunos das áreas técnicas (Engenharias principalmente), mas nada impede que ele seja utilizada para o ensino nas demais áreas como a humanas e sociais. Neste caso não é propriamente o sistema que deve ser adaptado e sim as diferentes bases de conhecimentos como os conteúdos do sistema de hipertexto, a base de aplicações (listas de exercícios), a base de casos (exemplos).

A validação do sistema em qualquer uma destas áreas poderá ser efetuada através de um ensino tradicional comparada com o sistema em turmas piloto e pela comparação estatística dos resultados.

Recomenda-se, finalmente, a implementação do módulo de comunicação que foi parcialmente realizada nesta etapa. Este módulo é composto por um subsistema de correspondência eletrônica, um de conversação online (*chat*) e um de vídeo conferência.



11. GLOSSÁRIO

- ACM** (*Association of Computer Machinery*). Fundada em 1947, é a maior e mais antiga sociedade internacional computacional e industrial. Através dos seus SIGs (*Special Interest Groups*) a sociedade promove a pesquisa e a comunicação em um amplo leque de áreas computacionais.
- Ajuda do Windows**. É um leitor de hipertexto que está disponível, em qualquer computador que use o MSWindows, através da tecla "F1", ou através do menu "Ajuda", disponível em qualquer software que rode sob o sistema operacional Windows. É um sistema de hipertexto simples que oferece navegação através de conexões com uma função de procura através de palavras-chaves. O sistema também registra o caminho percorrido, oportunizando, desta forma, o retorno às conexões estabelecidas através de uma tecla de atalho "retorno".
- Alan Kay**. É um dos inventores da linguagem de programação Smalltalk e um dos pais da idéia da programação objeto orientada. Concebeu a idéia do computador laptop e da interface por janelas (Windows).
- Andrew Lippman**. Trabalha a 27 anos no MIT, onde é atualmente Diretor associado do Laboratório de mídia e professor. É um dos principais investigadores do programa "Televisão do Amanhã". Responsável, juntamente com outros colegas pelo desenvolvimento do primeiro sistema de hipermídia o Aspen Movie Map.
- Andries Van Dam**. Em 1968 junto com Ted Nelson desenvolveu o primeiro sistema de hipertexto que foi chamado de HES (*Hypertext Editing System*). Um ano mais tarde desenvolveu o sistema FRESS que segundo o autor foi o primeiro software a apresentar a função "undo".
- API** (*Application Program Interface*). É o conjunto de funções de um sistema operacional através do qual um programa aplicativo ou programador pode se comunicar com o sistema ou com outros aplicativos.
- Applet**. Pequeno programa de aplicação que foi popularizado como um programa compacto, escrito em Java, embutido nas páginas escritas em html, que é executado no computador cliente.
- Archie**. É um sistema para localizar arquivos disponíveis em diretórios públicos na Internet, que podem ser acessados através de ftp.
- ASCII** (*American Standard Code for Information Interchange*). É o formato de arquivos de texto mais comum. Num arquivo ASCII, cada letra, número ou caracter especial é representado através de um número binário de 7 bits.
- Aspen Movie Map** (1978). Este talvez tenha sido o primeiro sistema hipermídia (utilizava videodisco). Por ser o pioneiro e por suas características avançadas ele foi incluído entre os de segunda geração, apesar de ter sido desenvolvido em 1978, pelo grupo de arquitetura de máquinas do MIT. O sistema é uma aplicação em turismo que permite ao usuário simular uma viagem pela cidade de Aspen.
- Augment/NLS** (1962-1976). Projeto desenvolvido por Douglas Engelbart e equipe no SRI (*Stanford Research Institute*). Um módulo do projeto era o NLS (*oN-Line System*), que se diferenciava da outra parte, o FLS (*ofF-Line System*), que

apresentava várias características de hipertexto, mesmo que não tenha sido desenvolvido como tal.

Authorware. Particularmente é um software da empresa Macromedia que auxilia o desenvolvimento de aplicações multimídias para CD-ROM e para a WWW. Genericamente designa uma ferramenta de autoria (*authorware*) que permite a criação de aplicações.

AVI (Audio Video Interleave). Formato de vídeo desenvolvido pela Microsoft. O formato é intercalado (*interleaved*), isto é, os dados de áudio e vídeo são armazenados consecutivamente com um segmento de vídeo seguido imediatamente por um de áudio.

Ben Shneiderman. É diretor do projeto HyperTies, professor do departamento de Ciências da Computação e chefe do Laboratório de Interação homem-máquina da universidade de Maryland.

Bitmap. Processo de representação gráfica que utiliza mapas de bits. O outro método de representação de imagens é conhecido como vetorial e utiliza equações matemáticas. Os formatos gráficos que se valem deste processo são o bmp, o gif e o tiff.

BMP (BitMaP). Formato de armazenagem de imagens que define o espaço de localização e a cor de cada pixel ou bit. Os formatos GIF e JPEG são exemplos de arquivos imagens gráficas que contém bitmaps.

Brian Atkinson. Foi o chefe da equipe responsável pelo desenvolvimento da GUI e o projetista chefe do desenvolvimento do HyperCard.

CCITT (Consultative Committee on International Telephone and Telegraphy). É agora denominado de ITU-T e é o principal comitê internacional para promover a padronização cooperativa de equipamentos de telecomunicações.

CERN (Centre Européen pour la Recherche Nucleaire). Trata-se do *European Laboratory for Particle Physics*, localizado em Genebra na fronteira entre a Suíça e a França. O acrônimo CERN relaciona-se ao nome anterior. A WWW teve sua origem aqui, devido aos esforços Tim Berners-Lee.

Class (Computer-based Laboratory for Automation of School Systems). Foi desenvolvido pela empresa Systems Development Corporation.

CNIDR (Center for Networked Information Discovery and Retrieval). Criado como um projeto de pesquisa pela NSF é um grupo especializado na pesquisa e desenvolvimento de sistemas de informações distribuídas.

David Ausubel (1918 -). Um dos expoentes da teoria cognitiva da aprendizagem e na explicação da aprendizagem verbal significativa, que ele vê como o método predominante de aprendizagem em sala de aula.

DBMS (DataBase Management System). É um pacote computacional que executa as tarefas de gerenciamento de um base de dados.

DNS (Domain Name System). É uma base de dados distribuída que traduz nomes computacionais como (*music.pucrs.br*) em endereços numéricos da Internet (*como 195.54.47.2*) e vice-versa.

Douglas Engelbart (1923-). Responsável pelo desenvolvimento, em 1968, do hipertexto NLS (*oN Line System*). No desenvolvimento deste sistema foi responsável por várias inovações técnicas tais como o mouse (possui a patente), a interface de janelas, suporte sensível ao contexto, etc.

DTD (Document Type Definition). Um arquivo que descreve formalmente uma linguagem de marcação (*markup*) específica tal como o html, por exemplo.

- DVD** (*Digital Video Disc*). Um novo tipo de CD-ROM que pode armazenar um mínimo de 4,7 gigabytes até um máximo de 17 gigabytes e com uma taxa de acesso de 600 KBps até 1,3 MBps.
- Dynabook** (1970). Sistema desenvolvido no início dos anos 70 por Alan Kay, para dar suporte a escrita e a leitura de texto.
- Edward Lee Thorndike** (1874-1949). Psicólogo americano que trabalhou com animais e estabeleceu que todo o comportamento de aprendizagem é regulado por punição e recompensa. Faz parte da escola conexionista através de sua estrutura de aprendizagem denominada de S-R (*Stimuli and Response*).
- EMYCIN**. É a shell do sistema MYCIN.
- EPS ou PS** (*Encapsulated PostScript*). Formato de arquivo gráfico criado pela empresa Adobe e utilizado em artes gráficas.
- Estrutura Geral de Hipertexto**. Rao e Turoff [RAO90] propuseram uma estrutura baseados no modelo de Guilford para o intelecto, argumentando que os sistemas de hipertexto tendem a sofrer de falta de coerência devido a ambigüidade nos significados associados a nodos e conexões. Esta estrutura classifica os nodos em seis tipos semânticos diferentes e as conexões em dois tipos principais. Eles acreditam que uma estrutura deste tipo seria de ajuda aos projetistas no desenvolvimento de melhores metáforas de interface.
- Expertext**. É uma combinação de hipertexto com sistema especialista. Utiliza predicados nos nodos e implicações nas conexões de forma que o grafo do hipertexto pode suportar inferência. É um ITS no formato de hipertexto, construído com a intenção de testar a utilidade, praticidade e generalização do conceito de sistema inteligente de ensino. O modelo do estudante e o especialista foram codificados em centenas de regras que são avaliadas por uma shell de um sistema especialista comercial [JON94].
- Felix Klein** (1849-1925). Matemático nascido na Prússia (Alemanha agora) e conhecido por seu trabalho nas geometrias não-euclidianas.
- F(rederic) B(urrhus) Skinner** (1904-1990). Psicólogo comportamental americano. Suas "leis de aprendizagem", derivadas de experimentos com animais foram largamente aplicadas na educação e na terapia comportamental. Sua teoria da aprendizagem é conhecida como Condicionamento Operante.
- FRESS** (*File Retrieval and Editing System*). Desenvolvido em 1968, pela equipe de Andries van Dam foi implementado em um mainframe IBM. Em virtude de sua plataforma extremamente estável foi possível rodar uma demonstração do seu código mais de vinte anos depois, na conferência de 1989 em hipertexto da ACM.
- FTP** (*File Transfer Protocol*). Protocolo de transferência de arquivos sem autenticação (anônimo). O ftp e o http são semelhantes só que no ftp o arquivo não precisa estar em um formato específico (texto html) e ele é bidirecional, isto é, pode buscar e enviar arquivos entre computadores, enquanto que o http pode apenas receber.
- gIBIS** (*Graphical Issue Based Information Systems*). É um hipertexto projetado para facilitar e capturar discussões de projetos de sistemas grandes e complexos. Foi projetado para suportar a construção colaborativa de equipes de qualquer tamanho dispersas através de uma rede.
- GIF** (*Graphics Interchange Format*). Formato de arquivo utilizado para a apresentação de imagens estáticas. Utiliza baixa compressão sem perda de informações.

- Glasgow online.** Sistema de hipertexto construído sobre o HyperCard por Patricia Baird e aproximadamente outras 30 pessoas na universidade de Strathclyde. É um guia completo da cidade de Glasgow incluindo mapas e muitos gráficos.
- Gopher.** Sistema de procura de informações que utiliza um protocolo semelhante ao http e, em virtude disto, ele está incluído agora em cada cliente WWW, de modo que este serviço pode ser acessado como parte da rede.
- GUI (Graphical User Interface).** A interface gráfica do usuário foi a responsável pelo sucesso e desenvolvimento acelerado do microcomputador. É o software que foi desenvolvido inicialmente na Apple Computer e mais tarde popularizado através do sistema Windows da Microsoft.
- Guide (1986).** Primeiro sistema comercial de hipertexto lançado para a plataforma Macintosh em 1986, logo após sendo lançada a versão para o PC e é ainda um dos únicos sistemas de hipertexto disponíveis para as duas plataformas. Mais recentemente o sistema tem sido lançado apenas para o Windows.
- GUIDON.** É um ITS para diagnóstico médico. É um exemplo de sistema tutorial que foi construído para servir de interface a um sistema especialista existente. É baseado no MYCIN.
- HAM (Hypertext Abstract Machine).** Uma das primeiras abordagens para a implementação de um modelo genérico de hipertexto foi o HAM (*Hypertext Abstract Machine*). A ênfase do projeto foi posta no desenvolvimento de um modelo apropriado de armazenagem, que seria flexível e geral de modo a ser utilizado em diferentes aplicações. A arquitetura do sistema está baseado nas camadas: **interface**, um ambiente interativo baseado em janelas para a comunicação com o usuário; **aplicação**, a aplicação real que pode ou não rodar na mesma máquina que a HAM; **HAM**, um mecanismo que gerencia todas as informações sobre o hipertexto e se comunica com a aplicação através de um protocolo de fluxo de bytes; **sistema de armazenagem**, um repositório que armazena todos os elementos do hipertexto.
- HDM (Hypermedia Design Model).** As características deste modelo incluem, a representação de aplicações através das primitivas: entidades (compostas de hierarquias de componentes); perspectivas diferentes para cada componente; unidades correspondentes a pares componente-perspectiva; corpos (representando o verdadeiro conteúdo das unidades); ligações estruturais (relacionando componentes pertencendo a mesma entidade); ligações de aplicação (relacionando componentes pertencendo a entidades diferentes); semântica de navegação (determinando a visualização e as propriedades dinâmicas da aplicação). Estas primitivas são semelhantes a objetos definidos no HAM.
- HES (Hypertext Editing System).** Foi desenvolvido em 1967, por uma equipe liderada por Andries van Dam, sendo o primeiro hipertexto funcional a ser construído no mundo. Ele rodava em 128K de memória em um pequeno *mainframe* IBM/360 e foi desenvolvido através de um contrato de pesquisa com a empresa IBM. Foi vendido para o centro aeroespacial de Houston onde foi utilizado para produzir a documentação das missões Apolo [NIE90].
- HyperCard (1987).** Sistema de hipertexto criado por Brian Atkinson, que foi originalmente construído como um ambiente de programação gráfico e muitas aplicações obtidas com ele nada tem a ver com hipertexto. É, talvez, um dos mais conhecidos produtos de hipertexto, em virtude de ter sido distribuído, sem custos adicionais, com cada computador Macintosh vendido a partir de 1987.
- HyperTalk.** Linguagem de script objeto orientada do HyperCard e seus clones.

- HyperTies** (1983). Foi iniciado como um projeto de pesquisa por Ben Shneiderman na universidade de Maryland em torno de 1983. Foi denominado inicialmente de TIES (*The Electronic Encyclopedia System*), mas como este nome já era marca comercial ele foi alterado para HyperTies para indicar a utilização de conceitos de hipertexto no sistema.
- IGES** (*Initial Graphics Exchange Specification*). É um conjunto de protocolos para a transferência e a apresentação de informação gráfica em mecanismos remotos através de uma linha telefônica ou rede computacional.
- IMAP** (*Internet Message Access Protocol*). É um protocolo para o acesso de mensagens de correio eletrônico em servidores locais. É um protocolo cliente/servidor na qual a mensagem é recebida e mantida pelo servidor de Internet. O software permite que o cliente acesse e manipule *folders* remotos de correio eletrônico no servidor, de maneira semelhante a caixas de correio locais.
- Intermedia** (1985). O sistema é produto de um dos mais antigos grupos de pesquisa em hipertexto, o IRIS da universidade Brown. Vários sistemas inovadores de hipertexto foram desenvolvidos na Brown sob a liderança de Andries van Dam, incluindo o HES de 1968 e o FRESS de 1969.
- IP** (*Internet Protocol*). O mais importante dos protocolos no qual a Internet está baseada. Permite que um pacote de dados seja transferido através de muitas redes de diferentes especificações.
- IRC** (*Internet Relay Chat*). Foi escrito por Jarkko Oikarinen em 1988.
- IRIS** (*Institute for Research in Information and Scholarship*). A universidade Brown criou o Instituto para explorar novas tecnologias na pesquisa, ensino e aprendizagem. Durou de agosto de 1983 a junho de 1994. O legado mais importante do Instituto foi a criação do sistema Intermedia.
- ISO** (*International Standard Organization*). É uma organização internacional e não governamental, fundada em 1947, de padrões nacionais de aproximadamente 100 países. Tem como objetivo promover a padronização, em atividades relacionadas, através do mundo, de forma a facilitar o comércio de bens e serviços.
- Ivan Petrovich Pavlov** (1849-1936). Psicólogo soviético que com seus estudos do reflexo (resposta) condicionado em cães influenciou a teoria comportamental. Ganhou o prêmio Nobel de Medicina ou Psicologia de 1904.
- Jerome Bruner** (1915 -). Professor e pesquisador do departamento de Psicologia de Harvard. Suas idéias exerceram grande influência sobre a teoria da aprendizagem cognitiva. Ele declarava que as pessoas interpretam o mundo em termos de semelhanças e diferenças e que formavam um arranjo hierárquico de categorias relacionadas iniciando do mais alto nível e tornando-se cada vez mais específica.
- Jean Piaget** (1896-1980). Psicólogo Suíço pioneiro no estudo da inteligência da criança. Nasceu em Neuchâtel. Foi professor (1929-54) de psicologia na universidade de Genebra. Autor da teoria da epistemologia genética da aprendizagem.
- J. P. Guilford**. Professor da USC (Universidade do Sul da Califórnia), que desenvolveu junto com colegas uma teoria sobre inteligência denominada de a "estrutura do intelecto".
- JPG** ou **JPEG** (*Joint Photographic Experts Group*). Formato de arquivo definido por um grupo semelhante ao MPEG só que formado por pessoas diferentes e com o objetivo de definir padrões para a compressão de imagens estáticas. Utiliza compressão com perda de informação.

KMS (*Knowledge Management System*). Desenvolvido em 1983 como descendente direto do sistema ZOG. Tornou-se um produto comercial a partir de 1983 e tem sido usado em várias aplicações, incluindo gerenciamento de projetos, manuais técnicos, procedimentos políticos e editoração eletrônica.

Memex (*Memory extender*). Considerado o primeiro sistema de hipertexto, apesar de nunca implementado. Idealizado por Vannevar Bush em 1945.

MIME (*Multipurpose Internet Mail Extensions*). É um protocolo que estende o formato de correio eletrônico de forma a permitir caracteres diferentes do ASCII. É compatível com a RFC 822 que define a SMTP e incorpora acréscimos como: novos campos de cabeçalho que fornecem informações sobre o corpo da mensagem; vários formatos para suportar mensagens eletrônicas multimídias e códigos de transferência para a inserção de outros tipos de arquivos.

Modelo Canto de hipertexto. Este sistema foi projetado para servir como base para um hipertexto onde os dados possam ser facilmente colocados, localizados e removidos sem que o usuário fique desorientado. A idéia básica é o de um modelo integrado de hipertexto que descreva os dados e especifique os atributos e relacionamentos entre os vários tipos de dados.

Modelo de Hipermídia Harmonia. É baseado num modelo de hipertexto objeto orientado denominado de modelo hiper-objeto. Neste modelo um nodo é um objeto e as relações entre os nodos são representadas como conexões de um hipertexto convencional. O modelo estende a idéia da ligação do modo que possa lidar com conexões de mídia ativa e temporal, como vídeo, som e animação.

Modelo Dexter. O objetivo é fornecer uma base sistemática para a comparação de sistemas e desenvolver padrões de intercâmbio e operacionalidade. O modelo é composto de três camadas: a **de armazenagem** (principal componente) modela a base de dados, que é composta de uma hierarquia de elementos de dados interconectados por ligações; a **de execução** trata da apresentação do hipertexto e da dinâmica da interação com o usuário e a **interna** lida com a estrutura interna e com o conteúdo de cada componente (nodo) da rede do hipertexto. A hipótese é que um padrão estrutural do documento tal como ODA, SGML, IGES, etc., será usada em conjunto para capturar o conteúdo/estrutura.

Modelo Trellis (ou de navegação semântica). É um modelo baseado em redes de Petri [STO89] que permite que sejam expressos tanto os possíveis estados em uma seção de leitura do hipertexto quanto as condições necessárias para se transferir de um estado para outro. Esta capacidade torna o modelo apropriado para certos usos educacionais, pois assegura que o leitor tenha acesso aos componentes críticos do hipertexto somente numa seqüência apropriada e por facilitar a adaptação ao comportamento do usuário.

Modelo Formal de Hipertexto. A motivação principal para o desenvolvimento deste sistema foi a falta de mecanismos de trocas de dados entre outros sistemas existentes. Da mesma forma que o Dexter, este modelo enfatiza a estrutura de dados dos hiperdocumentos, só que vai além na definição dos nodos, definindo também botões, campos e *slots*. Ele foi proposto por Danny B. Lange em 1990 como um modelo abstrato de referência [LAN90], que foi formalmente definido na linguagem de especificação VDM (*Viena Development Method*) [JON86].

MPEG (*Moving Picture Expert Group*) ou **MPG**. É um formato de arquivo para vídeo definido sob a orientação da ISO para gerar padrões para vídeo digital (seqüência de imagens no tempo) e compressão de áudio.

- MYCIN.** Sistema especialista para diagnóstico de doenças infecciosas do sangue.
- NCSA** (*National Center for Supercomputing Applications*). Centro nacional de super-computação da universidade de Illinois.
- NESTOR.** É um projeto que envolve as universidades alemães e o centro de pesquisas da empresa Digital Equipamentos. O projeto pretende implementar um ambiente de autoria e aprendizagem multimídia sobre estações de trabalho locais e distribuídas remotamente [MÜH92].
- Niklaus Wirth.** É professor no Instituto de Ciência da Computação – **ETH** (*Eidgenossische Technische Hochschule*) em Zurique. Além do Pascal desenvolveu a linguagem Modula2.
- NNTP** (*Network News Transfer Protocol*). É um protocolo que permite que informações transitórias (notícias) sejam transferidas através da Internet.
- Norman Crowder.** Instrutor da força aérea americana que desenvolveu a "programação intrínseca", uma forma de apresentar conteúdos que é utilizado pela maioria dos programas **CAI**.
- NoteCards** (1985). O NoteCards é um sistema de hipertexto criado no Xerox **PARC**. Foi desenvolvido para auxiliar na tarefa de análise de informações, incluindo leitura, categorização, interpretação e escrita técnica.
- NSF** (*National Science Foundation*). É uma agência independente do governo americano responsável pela promoção da ciência e da engenharia através de programas de investimento em projetos de pesquisa e educação.
- ODBC** (*Open DataBase Connectivity*). É um método padrão desenvolvido pela Microsoft que possibilita acessar dados de qualquer aplicação, não importando o tipo de **DBMS** que está gerenciando estes dados. Isto é feito através da inserção de uma camada intermediária de software, denominada *driver*.
- PARC** (*Palo Alto Research Center*). Centro de pesquisas da empresa Xerox localizado na cidade de Palo Alto – Califórnia.
- Patricia Baird.** Pesquisadora da universidade de Strathclyde (Inglaterra). Editora da revista "Hypermedia" e responsável pelo projeto Glasgow Online.
- PDF** (*Portable Document Format*). Formato de arquivo misto que pode conter imagens e texto e é utilizado pelo software Adobe Acrobat e Acrobat Reader.
- Peter Brown.** Professor da universidade de Kent (Inglaterra). Em 1982 iniciou os trabalhos do que viria a ser o primeiro sistema de autoria de hipertexto comercial para PC que foi denominado **Guide**.
- Petri (rede de).** É uma linguagem formal e gráfica que é apropriada para modelar sistemas concorrentes.
- PLATO** (*Programmed Logic for Automatic Teaching Operation*). Um antigo e influente sistema **CAI**, desenvolvido na universidade de Illinois e comercializado até recentemente pela empresa *Control Data Corporation*. Este talvez seja o projeto **CAI** mais conhecido no mundo [MER90]. Atualmente o sistema é comercializado pela empresa TRO, inc. sediada em Minneapolis e foi rebatizado com o nome de **CYBIS**.
- POP** (*Post Office Protocol*). É um padrão para o recebimento de mensagens de correio eletrônico semelhante ao **IMAP**, só que com este protocolo as mensagens são vistas no servidor como se este fosse o computador cliente.
- Porta.** (a) Um número que identifica um serviço de rede. (b) Um dos canais físicos de entrada/saída do computador. Os dois conceitos são distintos. O primeiro se refere a

um software que executa um serviço (telnet, http, etc.) enquanto o segundo está relacionado diretamente ao hardware.

PostScript. É uma linguagem de descrição de página, desenvolvida pela empresa Adobe Systems em 1985 e que se tornou um padrão da indústria para impressão e transmissão de imagens. Um arquivo PostScript é identificado pelo sufixo ".ps".

Protocolo. Um protocolo é uma convenção de como computadores diferentes devem agir quando se comunicarem entre si. Os computadores podem utilizar softwares completamente diferentes desde que os programas rodando em ambos concordem sobre o significado dos dados sendo trocados.

Randy Trigg. Pesquisador na área de Tecnologia e Prática do Trabalho do laboratório de Sistemas e Práticas do Xerox PARC. Doutor em Ciências da Computação pela universidade de Maryland.

RFC (*Request For Comments*) É um procedimento informal de introduzir novas tecnologias na Internet. RFC são livremente distribuídas e contém descrições da tecnologia pretendida pelos implementadores.

Rlogin. É um programa utilizado para a conexão a um terminal remoto. O telnet tem substituído o rlogin, mas ele ainda é usado em alguns ambientes.

Robert M. Gagné. Ph.D pela universidade Brown. Foi diretor técnico dos laboratórios da Força Aérea Americana. Atualmente é professor da universidade estadual da Flórida (*Florida State University*). Sua obra mais conhecida é o livro "*The Conditions of Learning*" de 1985, que se encontra hoje na quarta edição.

SAID (*Speech Auto-Instructional Device*). Foi desenvolvido na Universidade de Michigan.

SCHOLAR. Sistema inteligente de ensino que utiliza uma rede semântica para representar informações geográficas. É um dos mais antigos sistemas inteligentes de ensino. O sistema utiliza um estilo socrático de tutorial, primeiro tentando diagnosticar os erros implícitos no conhecimento do estudante e então colocando um problema que forçará o estudante a descobrir os erros.

Script (linguagem de). É um programa ou seqüência de instruções que é interpretada ou utilizada por outro programa ao invés do processador do computador como ocorre com um programa compilado.

SmarText. Ferramenta computacional para a criação de hipertexto, ou melhor, livros eletrônicos, produzido pela empresa Lotus. É projetado para ser um organizador de arquivos heterogêneos, podendo ler a maioria dos formatos de arquivos do tipo texto ou gráficos.

SMTP (*Simple Mail Transfer Protocol*) é um protocolo de envio de correspondência entre computadores que usam o protocolo TCP/IP.

SOCRATIC. Sistema inteligente para ensinar diagnóstico médico.

SOPHIE (*Sophisticated Instructional Environment*). É um sistema inteligente de ensino ou ITS, para lecionar como resolver problemas de eletrônica e que utiliza como sistema especialista um programa de simulação eletrônica de propósito geral.

SRI (*Stanford Research Institute*). Fundado em 1946 em conjunto com a Universidade de Stanford é uma das maiores empresas de pesquisa, empregando atualmente mais de 2600 pessoas, possuindo escritórios em vários países. Atualmente é denominado de SRI International.

StratchTutor. É um sistema de hipertexto projetado com o propósito de servir de tutor. Seu principal objetivo é fornecer ferramentas inteligentes para que autores e usuários

não tenham problemas com desorientação na navegação e com a criação de conexões manuais. Oferece um sistema de autoria sem a necessidade de programação para a organização do material de ensino.

Symbolics Document Examiner (1985). Projeto iniciado em 1982 e finalizado em 1985, sendo o primeiro sistema deste tipo a desempenhar um trabalho real. Ele foi construído como uma interface para a documentação das estações de trabalho Symbolics. Foi utilizado não por ser um hipertexto mas sim por ser a melhor maneira de se obter informações sobre as estações.

TCP/IP (*Transmission Control Protocol/Internet Protocol*). É o programa que clientes e servidores da Internet devem rodar. É um software de duas camadas. O **TCP** é o conjunto formal de regras para comunicação que gerencia o empacotamento dos dados, enquanto que o **IP** controla o endereço de cada pacote de dados para que ele seja enviado para o destino correto.

Telnet. (a) Um protocolo de emulação de terminal que permite a conexão em outro sistema computacional na Internet. (b) Um programa aplicativo que permite a conexão em outro sistema computacional utilizando o mesmo protocolo.

Textnet (1986). Sistema de hipertexto desenvolvido com o objetivo de fornecer suporte para a comunidade científica na criação de textos. O Textnet foi projetado e implementado por Randy Trigg na universidade de Maryland.

Theodor Holme Nelson (1937-). Professor de Harvard que cunhou o termo hipertexto. Foi também o responsável pelo desenvolvimento do mais ambicioso dos sistemas de hipertexto conhecido como projeto Xanadu.

TICCIT (*Time-shared Interactive Computer Controlled Information Television*). Este sistema começou a ser desenvolvido em 1971 com o esforço combinado dos engenheiros da MITRE Corporation, em McLean, Virgínia, e educadores do laboratório CAI da universidade do Texas. O sistema é atualmente comercializado pela empresa Ford Aeroespacial e Telecomunicações [MER90].

TIF ou TIFF (*Tagged-Image File Format*). Formato de arquivo do tipo bitmap utilizado para a troca de imagens entre diferentes plataformas. Possui um algoritmo de compactação que não deteriora a imagem.

Tim Berners-Lee. É o criador da WWW. Ele escreveu o primeiro programa cliente-servidor da rede e definiu as especificações do URL, HTTP e HTML sobre os quais a rede funciona enquanto trabalhava no CERN, em 1990.

TUTOR. Uma linguagem de autor para CAI desenvolvida em 1967 na universidade de Illinois para o sistema PLATO III.

URN (*Uniform Resource Name*). É um recurso da Internet com um nome persistente. Ele é semelhante ao endereço de uma página ou um URL. Um URN hipotético seria: "urn:def://objeto_verde", onde "def://" pode indicar uma agência ou um diretório de acesso para todos os dicionários, glossários e enciclopédias na Internet e "objeto_verde" seria o nome do termo. Um URL semelhante seria: <http://www.whatis.com/objetoverde.html>. Neste caso o usuário tem que saber onde o recurso está localizado, bem como escrever o nome correto e o sufixo correto do arquivo. Com um URN a única coisa que o usuário precisa saber é o nome do recurso.

Vannevar Bush (1890-1974). Foi Diretor do departamento de pesquisa científica e desenvolvimento (*Office of Scientific Research and Development*) na gestão do presidente Franklin D. Roosevelt. Projetou o primeiro sistema de hipertexto, embora não tenha usado o conceito.



12. REFERÊNCIAS

- [AUE95] Auer, Karl. Agents. Nov. 1995 [online]. <http://www.tip.net.au/~kauer/project/>
- [AAM94] AAMODT, A., PLAZA, E. Case-Based Reasoning: Foundational Issues, Methodological Variations, and Systems Approaches. *Artificial Intelligence Communications*. USA, v. 7, n. 1, 1994.
- [AND88] ANDREWS, D. F., FOWLKES, E. B., TUKEY, P. A. Some Approaches to interactive Statistical Graphics. In: *Dynamic Graphics for Statistics*. USA:Wadsworth, 1988, p. 73-90.
- [ASH98] ASHCROFT, Mike, NEWELL, John. Axum 5.0. *CTI Statistics. Software reviews*. http://www.stats.gla.ac.uk/cti/activities/reviews/97_02/axum5.html.
- [AUS78] AUSUBEL, D., NOVAK, J., HANESIAN, H. *Educational Psychology: A Cognitive View*. USA: Holt, Reinhart & Wiston, 1978.
- [BAR86] BARR, A, FEIGENBAUM, E. *The handbook of Artificial Intelligence*. USA: Addison-Wesley, 1986, 472 p. v. 2.
- [BAR91] BARRON L., et al. Technology and the design of generative learning environments. *Educational Technology*. Englewood, USA, v. 31, n. 5, p. 34-40, May. 1991.
- [BEC88] BECKER, Richard A., CLEVELAND, Williams S., WILKS, Allan R. Dynamic Graphics of Data Analysis. In: *Dynamic Graphics for Statistics*. USA: Wadsworth, 1988, p. 1-50.
- [BEG93] BEGLEY, S. Teaching minds to fly with disks and mice. *Newsweek*, New York. May 31, 1993, p. 45.
- [BEL95] BELGRAVE, M. The Unified Agent Architecture: A White Paper [online]. http://www.ee.mcgill.ca/~belmarc/agent_root.html
- [BIE91] BIELAWSKI, Larry, LEWAND, Robert. *Intelligent Systems Design: Integrating Expert Systems, Hypermedia, and Database Technologies*. New York: John Wiley & Sons, 1991.
- [BIE93] BIEHLER, Rolf. Software Tools and Mathematics Education: The Case of Statistics. In: *Learning from Computers: Mathematics Education and Technology. Proceedings of the NATO Advanced Research Workshop on Mathematics Education Technology*. Berlin: Spring-Verlag, 1993, p. 68-100.
- [BIG98] BIGUS, Joseph P., BIGUS, Jennifer. *Constructing Intelligent Agents with Java: a programmer's guide to smarter applications*. New York: John Wiley & Sons, 1998, 378 p.
- [BIT96] BITTENCOURT, Guilherme. *Inteligência Artificial: Ferramentas e Teoria*. Campinas: Instituto de Computação, UNICAMP, 1996, 240 p.
- [BOO91] BOOCH, Grady *Object Oriented Design With Applications*. California: Benjamin/Cummings, 1991.

VERONICA (*Very Easy Rodent-Oriented Netwide Index to Computerized Archives*).

É um serviço embutido no Gopher muito semelhante ao Archie. Assim como o Archie permite a procura de arquivos em qualquer local ftp, o Veronica permite que se pesquise itens de menu (arquivos, diretórios e outros recursos) em todos os Gophers.

Vilfredo Pareto (1848-). Nascido em Paris como Marquês Vilfredo Frederico Damaso Pareto, mas educado na Itália onde se formou Engenheiro pela escola politécnica de Turim. O curso de sólida base matemática teve uma influência intelectual profunda que se refletiu em seu trabalho posterior. Em 1870 tornou-se doutor com a tese "Os princípios fundamentais do equilíbrio em corpos sólidos", que despertaram seu futuro interesse na análise de equilíbrio na Economia e Sociologia.

WAIS (*Wide Area Information System*). É um sistema para procurar informações em base de dados (ou bibliotecas) através da Internet. Com o uso do software "freeWAIS" do CNIDR, a WWW pode acessar servidores WAIS diretamente.

Wav. Um dos formatos de arquivo suportados pelo sistema Windows. Grava o som através de amostragem o que se traduz em arquivos que ocupam grande espaços em disco.

Writing Environment (1986). Sistema de hipertexto desenvolvido na universidade da Carolina do Norte, com o objetivo de fornecer suporte à escrita. Ele fornece quatro modos de auxílio à escrita. Além das duas tradicionais (modo texto e modo de edição), ele fornece a estrutura de redes e a de árvore, para auxiliar o escritor a identificar, organizar e seqüenciar as idéias.

WMF (*Windows Metafile Format*). Arquivo de formato gráfico utilizado para a troca de informação gráfica entre aplicações do MS Windows. Os arquivos podem suportar imagens vetoriais ou bitmaps.

Xanadu (1965). A idéia básica é que ele servisse como base de dados de tudo que qualquer um tivesse escrito e seria desta forma um verdadeiro hipertexto universal. A intenção do Xanadu nunca foi realizada e provavelmente nunca será, pelo menos num futuro imediato.

ZOG (1972). O sistema ZOG foi desenvolvido na universidade de Carnegie Mellon, com parte do desenvolvimento iniciando em 1972 e como um todo a partir de 1975. Ele consistia de uma grande base de dados de pequenos segmentos de informação. Sua versão comercial e sucessora foi denominada de KMS.

- [BOW98] BOWMAN, Adrian. An Electronic Companion to Statistics. *Maths&Stats Newsletter*. UK. Feb. 1998. http://www.stats.gla.ac.uk/cti/activities/reviews/98_02/electronic_companion.html.
- [BRA96] BRADSTREET, Thomas E. Teaching Introductory Statistics Courses So That Nonstatisticians Experience Statistical Reasoning. *The American Statistician*. USA, v. 50, n. 1, p. 69-78, Feb. 1996.
- [BRE90] BREUER, Klaus, HAJOVY, Halyna. Adaptive Instrucional Simulations to Improve Learning of Cognitive Strategies. *Expert Systems and Intelligent CAI*. USA, v. 2, p. 154-157, 1990.
- [BRO97] BROOKS, David W. *Web-teaching: A Guide to Designing Interactive Teaching for the World Wide Web*. New York: Plenun Press, 1997, 214 p. (Innovations in Science Education and Technology).
- [BUS45] BUSCH, Vannevar. As We May Think. *Atlantic Monthly*. USA, v. 176, n. 1, p. 101-108, July 1945.
- [CAG88] CAMPBELL, Brad, GOODMAN, Joseph M. HAM: A General Purpose Hypertext Abstract Machine. *Communications of the ACM*, v.32, n.7, July 1988.
- [CAM97] CAMERON, Laird, SORAIZ, Kathryn. Choosing a scripting language. *SunWorld*. USA, Oct. 1997.
- [CAN97] CANTÙ, Marco. *Mastering Delphi 3*. San Francisco: Sybex, 1997, 1476 p. Second Ed.
- [CAR70] CARBONELL, J. AI in CAI: An Artificial Intelligence Approach to Computer-Assisted Instruction. *IEEE Transactions on Man-Machine Systems*. USA, v. 11, p. 190-202, 1970.
- [CAR92] CARRAHER, D. O papel do computador na aprendizagem. *ACESSO: Revista de Educação e Informática*. São Paulo: FDE, v. 3, n. 5, p. 21-30, Jan. 1992.
- [CAR95] CARDELLI, Luca. A Language with Distributed Scope. Palo Alto, CA: Digital Equipment Corporation, Systems Research Center, May 1995.
- [CHA89] CHAN, C. Computer Use in Classroom - I: An Assessment of CAI Software. *Computer and Education*. USA, v. 13, n. 2, p. 109-115, 1989.
- [CHA92] CHANG, Ted C., LOHR, Sharon L., McLAREN, C. Graham. Teaching Survey Sampling Using Simulation. *The American Statistician*. USA, v. 46, n. 3, p. 232-237, Aug 1992.
- [CHA96] CHANG, D. T., LANGE, D. B. Mobile Agents: A New Paradigm for Distributed Object Computing on the WWW. In: *Proceedings of the OOPSLA96 Workshop: Toward the Integration of WWW and Distributed Object Technology*. Oct. 1996.
- [CHU91] CHUNG, J. Collaborative learning strategies. *Educational Technology*: USA, v. 31, n. 12, p. 15-21, Dec. 1991.
- [COA91] COAD, P., YOURDAN, E. *Object-Oriented Analysis*. Englewood Cliffs, NJ: Prentice Hall, 991.
- [COE91] COE, Richard. SAS for Teaching Statistics. *Maths&Stats Newsletter*. Nov. 1991. http://www.stats.gla.ac.uk/cti/activities/reviews/91_11/sas.html.
- [COF96] COFFEE, Peter. Feature. *Windows Source*. 1996, p. 1-8 [online] <http://www4.zdnet.com/wsources/content/0197/sub1.htm>

- [COH96] COHEN, Steve et al. Identifying Impediments to Learning Probability and Statistics From an Assessment of Instructional Software. *Journal of Educational and Behavioral Statistics*. USA, v. 21, n. 1, p. 35-54, Spring 1996.
- [COL96] COLES, Stuart. Laws of Small Numbers: Extremes and Rare Events by Michael Falk, Jurg Husler and Rolf-Dieter Reiss. *Maths&Stats newsletter*. May 1996. http://www.stats.gla.ac.uk/cti/activities/reviews/96_05/small_numbers.html
- [CON96] CONSTANTIN, Von Altrock. *Fuzzy Logic and Neurofuzzy Applications in Business and Finance*. Englewood Cliffs, New Jersey: Prentice Hall, 1996.
- [COS96] COSTA, Evandro de Barros, PERKUSICH, Angelo. Modeling the Cooperative Interactions in a Teaching/Learning Situation. In: *Proceedings of the third international conference on Intelligent Tutoring systems - ITS'96*. Berlin: Springer-Verlag, 1996, p. 168-176.
- [CUR94] CURRAL, James. Statistics Packages: A General Overview. *Maths&Stats Newsletter*. UK, Nov. 1994. http://www.stats.gla.ac.uk/cti/activities/reviews/94_11/overview_packages.html.
- [CUR96] CURRAL, James. JMP 3.1 para Windows. *Maths&Stats Newsletter*. UK, Feb. 1996. http://www.stats.gla.ac.uk/cti/activities/reviews/96_02/jmp.html.
- [CUR96] CURRAL, James. Dynamic or Interactive Graphics. *Maths&Stats Newsletter*. UK, May. 1996. http://www.stats.gla.ac.uk/cti/activities/reviews/96_05/dynamic_graphics.html.
- [CUR96] CURRAL, James. StatTutor. *Maths&Stats Newsletter*. UK, Aug. 1995. http://www.stats.gla.ac.uk/cti/activities/reviews/96_02/jmp.html.
- [CUR96] CURRAL, James. SAS/Insight. *Maths&Stats Newsletter*. UK, Aug. 1996. http://www.stats.gla.ac.uk/cti/activities/reviews/95_08/statutor.html.
- [CUR97] CURRAL, James, YOUNG, Stuart, BOWMAN, Adrian. ActivStats. *Maths&Stats Newsletter*. UK, Nov. 1997. http://www.stats.gla.ac.uk/cti/activities/reviews/97_11/activastats.html.
- [CUR97] CURRAL, James. Systat Release 6 *Maths&Stats Newsletter*. UK. Feb. 1997. http://www.stats.gla.ac.uk/cti/activities/reviews/97_02/systat6.html.
- [CUR97] CURRAL, James. Minitab Release 11. *Maths&Stats Newsletter*. UK, Feb. 1997. http://www.stats.gla.ac.uk/cti/activities/reviews/97_02/minitab.html.
- [CUR98] CURRAL, James. Unistat Version 4. *CTI Statistics Software Reviews*. http://www.stats.gla.ac.uk/cti/activities/reviews/96_11/excel/unistat.html.
- [CUR98] CURRAL, James. SPSS 7.5 for Windows. *Maths&Stats Newsletter*. UK, May 1998. http://www.stats.gla.ac.uk/cti/activities/reviews/98_05/spss.html.
- [CUR98] CURRAL, James. Data Desk Version 6. *Maths&Stats Newsletter*. UK, Feb. 98. http://www.stats.gla.ac.uk/cti/activities/reviews/98_02/data_desk.html.
- [DAV92] DAVIES, Neville. StatGraphics. *Maths&Stats Newsletter*. UK, Feb. 1992. http://www.stats.gla.ac.uk/cti/activities/reviews/92_02/statgraphics.html.
- [DAV97] DAVID, Glyn. Teaching with Minitab and SPSS in a Business School environment. *Maths&Stats Newsletter*. UK, Aug. 1997. http://www.stats.gla.ac.uk/cti/activities/reviews/96_08/minitab_spss.html.
- [DAV97] DAVIDSON, Kirsty. Statwise: multimedia Statistics Courseware. *Maths&Stats Newsletter*. UK, Aug. 1997. http://www.stats.gla.ac.uk/cti/activities/reviews/97_08/statwise.html.

- [DEA97] DEAR, Keith. SurfStat.australia: A Statistics Textbook for the Web. *Maths&Stats Newsletter*. UK, Nov. 1997. http://www.stats.gla.ac.uk/cti/activities/reviews/96_08/minitab_spss.html
- [DER95] DERRY, Sharon, LEVIN, Joel R., SCHAUBE, Leona. Stimulating Statistical Thinking Through Situated Simulations. *Teaching Psychology*. USA, v. 22, n. 1, p. 51-57, Feb. 1995.
- [EGA89] Egan, D. E. et al. Acquiring information in books and SuperBooks. *Machine-Mediated Learning*. USA, n. 3, p. 259-277.
- [EGR89] Egan, D. E. et al. Behavioral evaluation and analysis of a hipertext browser. *Proceedings of ACM CHI'89 - Conference on Human Factors in Computing Systems*, 1989, p. 205-210.
- [FER95] FERRALL, Christopher. Interactive Statistics Tutorials in Stata. *Journal of Statistics Education (Online)*. USA, v. 3, n. 3, 1995. <http://www.stat.ncsu.edu/info/jse/v3n3/ferrall.html>.
- [FIC92] FICHMAN, Robert, KEMERER, Chris F. Object-Oriented and Conventional Analysis and Design Methodologies - Comparison and Critique. *Computer*. USA, v. 25, n. 10, p. 22-39, Oct. 1992.
- [FIN94] FININ, Tim, LABROU, Yannis, MAYFIELD, James. KQML as an agent communication language. Baltimore: University of Maryland, Technical Report, Computer Science Department. 1994.
- [FON97] FONER, Leonard N. Entertaining Agents: A Sociological Case Study. In: *The First International Conference on Autonomous Agents, Marina del Rey, CA*. 1997.
- [FOR96] FORCIER, Richard C. *The Computer as a Productivity Tool in Education*. Englewood Cliffs, New Jersey: Prentice-Hall, 1996, 365 p.
- [FOR98] FOSTER, Peter. WINKS 4.1 Professional Edition. *CTI Statistics Software Reviews*. http://www.stats.gla.ac.uk/cti/activities/reviews/96_08/winks.html.
- [FRA90] FRANKLIN, Carl, KINNELL, Susan K. *Hypertext/Hypermedia in Schools - A Resource Book*. USA: ABC-CLIO, 1990, 274 p.
- [FRA90] FRANKLIN, Carl. A Bibliography on Hypertext and Hypermedia With Selected Annotations. *Database*. USA, v. 13, n. 1, p. 24-32, Feb. 1990.
- [FRA96] FRANKLIN, S, GRAESSER, A. Is It anAgent, or just a Program?: A Taxonomy for Autonomous Agents. In: *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*. Berlin: Springer-Verlag, 1996.
- [GIE98] GIESE, Luiz Fernando. *Estrutura de Agentes para os Processos de Compra e Venda Utilizando Tomada de Decisão Difusa*. Florianópolis: UFSC, 1998, 114 p. Dissertação (Mestrado em Ciências da Computação) - Universidade Federal de Santa Catarina, Fev. 1998.
- [GIL96] GILBERT, Don, JANCA, Peter. IBM Intelligent Agents. Research Triangle Park, Raleigh, USA: IBM Corporation, 1996.
- [GOO93] GOODWIN, Richard. Formalizing Properties of Agents. Pittsburgh: CMU, School of Computer Science. Technical Report, May 1993.
- [GOR95] GORDON, Sue. A Theoretical Approach To Undertanding Learners Of Statistics. *Journal of Statistics Education (online)*. USA. v. 3, n. 3, 1995. <http://www.stat.ncsu.edu/info/jse/v3n3/gordon.html>

- [GRA94] GRANDBASTIEN, Monique, GAVIGNET, Elisabeth. ECSA: An Environment to Design and Instantiate Learning Material. In: *Design and Production of Multimedia and Simulation-based Learning Material*. Netherlands: Kluwer Academic, 1994, p. 31-44.
- [GRA96] GRABOWSKI, Barbara L., HARKNESS, William L. Enhancing Statistics Education with Expert Systems: More than an Advisory System. *Journal of Statistics Education*. USA, v. 4, n. 3, 1996. <http://www.stat.ncsu/info/jse/v4n3/grabowski.html>.
- [GRA97] GRANT, Felix. Genstat and the quantitative revolution. *Scientific Computing World*. USA, Oct. 1997.
- [GRA98] GRAY, Roger. DISCUS. *CTI Statistics Software Reviews*. http://www.stats.gla.ac.uk/cti/activities/reviews/96_11/excel/discus.html.
- [GRU93] GRUBER, T. R. Towards principles for the design of ontologies used for knowledge sharing. In: *International Workshop on Formal Ontology I*. Mar. 1993.
- [HAL90] HALAZ, Frank, SCHWARTZ, Mayer. The Dexter Hypertext Reference Model. *Proceedings of the Hypertext Standardization Workshop by National Institute of Science and Technology (NIST)*, 1990. p. 95-133.
- [HAL97] HALL, Brandon. *Web-Based Training Cookbook*. New York John Wiley & Sons, 1997. 482 p.
- [HAR95] HARRISON, Colin G., CHESS, David M., KERSHENBAUM, Aaron. Mobile Agents: Are they a good idea? *IBM Research Report*. New York: IBM Research Division, Mar. 1995.
- [HAR96] HARTLEY, Stephen et al. Enhancing teaching using the Internet: Report of the Working Group on the World Wide Web as an Interactive Teaching Resource. *SIGCSE Bulletin*. USA, v. 28, Special Issue, p. 218-228, 1996.
- [HAT91] HATABIAN, Gerard, AUGENDRE, Herve. Expert Systems as a Tool for Statistics: A Review. *Applied Stochastic Models and Data Analysis*. USA, v. 7, p. 183-194, 1991.
- [HAY95] HAYES-ROTH, B. An Architecture for Adaptive Intelligent Systems. *Artificial Intelligence: Special Issue on Agents and Interactivity*. n. 72, p. 329-365, 1995.
- [HIN94] HINDE, John. GLIM 4. *Maths&Stats Newsletter*. UK, Nov. 1994. http://www.stats.gla.ac.uk/cti/activities/reviews/94_11/glim4.html.
- [HIN97] HINDE, John. MacAnova 4.04. *Maths&Stats Newsletter*. UK, May 1997. http://www.stats.gla.ac.uk/cti/activities/reviews/97_05/macanova.htm.
- [HOG94] HOGG, Robert V. A Core in Statistics for Engineering Students. *The American Statistician*. USA, v. 48, n. 4, p. 285-287, Nov. 1994.
- [HUB98] HUHNS, Michael N., SINGH, Munindar P. Agents and Multiagent Systems: Themes, Approaches, and Challenges (Chapter 1). In: *Reading in AGENTS*. San Francisco: Morgan Kaufmann Publishers, 1998, p. 1-23.
- [HUG97M] HUGHES, Merlin. JavaBeans and ActiveX go head to head. *JavaWorld*, March 97 [online] <http://www.javaworld.com/javaworld/jw-03-1997/>.
- [IBR95] IBRAHIM, Bertrand, FRANKLIN, Stephen D. Advanced educational uses of the World-Wide Web. *Computer Networks and ISDN Systems*. USA, v. 27, n.6, p. 871-877, Apr. 1995.
- [JAS91] JASPERS, F. Interactivity or instruction? *Educational Technology*, USA, v. 31, n. 3, p. 21-24, Mar. 1991.

- [JEN95] JENNINGS, Nicholas R., WOOLDRIDGE, Michael. Applying Agent Technology. *Journal of Applied Artificial Intelligence (Special Issue on Intelligent Agents and Multi-Agent Systems)*. USA, v. 9, n. 6, p. 357-370, Jun. 1995.
- [JEN96] JENNINGS, Nicholas R., WOOLDRIDGE, Michael. Software Agent. *IEE Review*. USA, v. 42, n. 1, p. 17-21, Jan. 1996.
- [JOH96] JOHANSEN, Dag, SCHNEIDER, Fred B., RENESSE, Robbert van. What TACOMA Taught Us. [online] <http://www.cs.uit.no/DOS/Tacoma/index.html>.
- [JON86] JONASSEN, David H. Hypertext Principles for Text and Courseware Design. *Educational Psychologist*. USA, v. 21, n. 4, p. 269-292, Fall 1986.
- [JON88] JONASSEN, David H. Designing Structured Hypertext and Structuring Access to Hypertext. *Educational Technology*. USA, v. 27, n. 11, p. 13-16, Nov. 1988.
- [JON90] JONASSEN, David H., GRABINGER, R. Scott. Problems and Issues in Designing Hypertext/Hypermedia for Learning. In: *Proceedings of the NATO Advanced Research Workshop on designing Hypertext/Hypermedia for Learning. Held in Rottenburg/Neckar, FRG, July 3-8, 1989*. Berlin: Springer-Verlag, 1990, p. 188-197.
- [JON93] JONASSEN, David H. Applications of Hypertext: Technologies for Higher Education. *Journal of Computing in Higher Education*. USA, v. 4, n. 2, Spring 1993, p. 12-42.
- [JON94] JONASSEN, David H., WANG, Sherwood. The Physics Tutor: Integrating Hypertext and Expert Systems. *Journal of Educational Technology Systems*. USA, v. 22, n. 1, p. 19-28, 1993-94.
- [JON89] JONES, T. Incidental learning during information retrieval a hypertext experiment. *Proceedings of the International Conference on Computer Assisted Learning (ICCAL)*. Berlin: Springer-Verlag, 1989, p. 235-251.
- [KAY97] KAY, Jim. StatView 4.5 for Windows. *Maths&Stats Newsletter*. UK, Feb. 1997. http://www.stats.gla.ac.uk/cti/activities/reviews/97_02/statview.html.
- [KIN97] KINNEAR, Paul. Constats. *Software News*. UK, v. 7, n. 3, Jun. 1997. <http://www.york.ac.uk/ctipsych/web/CTI/DirTxt/reviews/constats.html>
- [KNA96] KNABE, Frederick. An Overview of Mobile Agent Programming. Chile: PUC, 1996.
- [KER98] KERN, Eduardo. *Uma Estrutura de Agentes para o Processo de Licitação*. Florianópolis: UFSC, 1998, 107 p. Dissertação (Mestrado em Ciências da Computação) - Universidade Federal de Santa Catarina, Fev. 1998.
- [KNI96] KNIERRIEM-JASNOCH, A., TRITSCH, B., SCHROEDER, U. Reflection on WWW Functionalities for Educational Purposes. *Computers & Graphics*. USA, v. 20, n. 3, p. 435-443, May/Jun. 1996.
- [KOZ87] KOZMA, R. The implications of cognitive psychology for computer-based learning tools. *Educational Technology*, USA, v. 27, n. 11, p. 20-25, Nov. 1987.
- [LAN90] LANDOW, George. Popular Fallacies About Hypertext. In: *Designing Hypermedia for Learning*. Berlin: Springer-Verlag, 1990, p. 39-59.
- [LAN90] LANGE, D. B. A Formal Model of Hypertext. *NIST Hypertext Standardization Workshop*. Feb. 1990, p. 145-166.
- [LAN92] LANDOW, George P. The Definition of Hypertext and Its History as a Concept. <http://www.stg.brown.edu/projects/hypertext/landow/ht/history.html>.

- [LUC97] LUCK, M., GRIFFITHS, N., D'INVERNO, M. From Agent Theory to Agent Construction: A Case Study. In: *Intelligent Agents III: Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages*. Berlin: Springer-Verlag, 1997, p. 49-63.
- [LUR95] LURIE, Philip M. A Review of Five Statistical Packages for Windows. *The American Statistician*. USA, v. 49, n. 1, p. 99-107, Feb. 1995.
- [MAC95] MACGILLIVRAY, Helen. The Steps Material: Experiences form an Australian Evaluator. *Maths&Stats Newsletter*. UK, Aug. 1996. http://www.stats.gla.ac.uk/cti/activities/reviews/95_08/steps.html.
- [MAD92] MADDUX, Cleborne D. User-Developed Computer-Assisted Instruction: Alternatives in Authoring Software. *Educational Technology*. USA, v. xxxii, n. 4, p. 7-14, Apr. 1992.
- [MAE95] MAES, Pattie. Artificial Life Meets Entertainment: Life like Autonomous Agentes. USA, *Communications of the ACM*, v. 38, n. 11, p. 108-114, Jun. 95.
- [MAG97] MAGUIRE, Liam P., MCGINNITY, T. Martin, McDAID, Liam J. A Fuzzy Neural Network for Approximate Fuzzy Reasoning. In: *Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks, and Genetic Algorithms*. Norwell, MA: Kluwer Academic, 1997, p. 35-58.
- [MAR88] MARCHIONINI, G. Hypermedia and learning: freedom and chaos. *Educational Technology*. USA, v. 28, n. 11, p.8-12, Nov 1988.
- [MAR88] MARCHIONINI, G., SCHENEIDERMAN, B. Finding Facts vs. browsing knowledge in hypertext systems. *Computer*. USA, v. 21, n. 1, p. 70-79, Jan. 1988.
- [MAR90] MARCOULIDES, George A. Improving Learner Performance with Computer Based Programs. *Journal of Educational Computing Research*. USA, v. 6, n. 2, p. 147-155, 1990.
- [MAS94] MARSH, Paul L. A review and Comparison of Five Graphics Programs for PC-DOS and MS-DOS Computers. *The American Statistician*. USA, v. 48, n. 1, p. 44-51, Feb. 1994.
- [MAY90] MAYES, J. T., KIBBY, M. R., ANDERSON, A. Learning about learning from hypertext. In *Designing hypertext/hypermedia for learning*. Heidelberg: Springer-Verlag, 1990 (D. H. Jonassen & H. Mandl Ed.).
- [McA94] McCABE, F. G., CLARK, K. L. April - Agent PROcess Interaction Language. In: *Intelligents Agents*. LNCS, Berlin: Springer-Verlag, v. 890, 1995.
- [McA93] McARTHUR, D. LEWIS, M. e BISHAY, M. The roles of artificial intelligence in education: current progress and future prospects. Santa Monica, CA: RAND, Nov. 1993.
- [McC95] McCOLL, John H. STEPS Forward. *Maths&Stats Newsletter*. UK, Nov. 1995. http://www.stats.gla.ac.uk/cti/activities/reviews/95_11/steps.html.
- [McC96] McCLOSKEY, Moya, BLYTE, Steve, ROBERTSON, Chris. The Quercus Project: Statistics for Biologists. *Maths&Stats Newsletter*. UK, Feb. 1996. http://www.stats.gla.ac.uk/cti/activities/reviews/96_02/quercus.html.
- [MEL89] MELLICHAMP, Joseph M., PARK, Yong H. A statistical expert system for simulation analysis. *Simulation*. USA, v. 52, n. 4, p. 134-139, Apr. 1989.
- [MER90] MERILL, M. et al. Limitations of first generation instructional design. *Educational Technology*. USA, v. 30, n. 1, p. 7-11, Jan. 1990.
- [MER91] MERILL, M. Construtivism and instructional design. *Educational Technology*. Englewood, v. 31, n. 5, p. 43-53, May 1991.

- [MER96] MERRIL, Paul F. et al. *Computers in Education*. Needham Heights, Massachusetts: Simon & Schuster, 1996, 386 p.
- [MIN75] MINSKY, M. A framework to represent knowledge. In: *The Psychology of Computer Vision*. USA: McGraw-Hill, 1975, p. 211-277.
- [MIZ97] MIZUMOTO, Masaharu, SHI, Yan. A New Approach of Neurofuzzy Learning Algorithm. In: *Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks, and Genetic Algorithms*. Norwell, MA: Kluwer Academic, 1997, p. 109-129.
- [MOO95] MOORE, David S. Active Use of Hypertext to Aid Learning and Classroom Instruction. *SIGCE Bulletin - Twnty-Sixth SIGCE Tecnical Symposium on Computer Science Education*. USA, v. 27, n. 1, p. 297-301, Mar. 1995.
- [MOR98] MORGAN, Walter T. A Review of Eight Statistics Software Packages for General Use. *The American Statistician*. USA, v. 52, n. 2, Feb. 1998, p. 70-82.
- [MOR99] MORRIS, Steven, SZUSCIWICZ. The Data Game: Learning Statistics. <http://ctiweb.cf.ac.uk/HABITAT/HABITAT1/Data.html>
- [MOU96] MOULIN, Bernard, CHAIB-DRAA, Brahim. An Overview of Distributed Artificial Intelligence. In: *Foundantions of Distributed Artificial Intelligence*. New York: John Wiley & Sons, 1996, p. 3-55.
- [MÜL98] MÜLLER, Martin, MÜLLER, Tobias, ROY, Peter Van. Multiparadigm Programming in OZ. In: *Visions for the Future of Logic Programming: Laying the Foundations for a Modern Successor of Prolog*. Portland, Oregon, Dec. 7.
- [MÜL98] MÜLLER, Marlene. Teaching Statistics with XploRe. *Maths&Stats Newsletter*. UK, Feb. 1998. http://www.stats.gla.ac.uk/cti/activities/reviews/98_02/xplore.html.
- [NAU94] NAUCK, Detlef. Building neural fuzzy controllers with NEFCON-I. In: *Fuzzy Systems in Computer Science*. Wieweg, Braunschweig, 1994, p. 141-151.
- [NAU95] NAUCK, Detlef, KRUSE, Rudolf. NEFCCLASS - a neuro-fuzzy approach for the classification of data. In: *Proceedings of the 1995 Symposium on Applied Computing, Nashville, Fev. 26-28, 1995*. New York: ACM Press, 1995, p. 461-465.
- [NAU97] NAUCK, Detlef, KRUSE, Rudolf. Neuro-fuzzy systems for function approximation. In: *Proceedings of the 4th International Workshop on Fuzzy-Neuro-Systems '97 (FNS'97), Soest, Germany, 1997*. p. 316-323.
- [NAU97] NAUCK, Detlef, KRUSE, Rudolf. What are Neuro-Fuzzy Classifiers? In: *Proceedings of the Seventh International Fuzzy Systems Association World Congress IFSA '97, vol. IV*. Prague: Academia Prague, p. 228-233, 1997.
- [NEL87] NELSON, Ted. *Computer Lib: Dream Machines*. Redmond, Washington: Tempus Books, 1987, p. 29-33, 64-65, 119-126, 129-147 (Revised Edition).
- [NEL95] NELSON, Theodor Holm. The Heart of Connection: Hypermedia Unified by Transclusion. *Communications of the ACM*. USA, v. 38, n. 6, p. 31-33, Aug. 1995.
- [NEW97] NEWELL, John. SigmaPlot 4.0 for Windows 95. *Maths&Stats Newsletter*. Nov. 1997. http://www.stats.gla.ac.uk/cti/activities/reviews/97_11/sigmaplot.html.
- [NEW98] NEWELL, J. Diamond - A picture says a 1000 numbers. *CTI Statistics. Software Reviews*. http://www.stats.gla.ac.uk/cti/activities/reviews/96_05/
- [NIC93] NICHOLAS, C. K., ROSEMBERG, L. H. Canto: A Hypertext Data Model. *Electronic Publishing*. USA, v. 6, n. 1, Mar 1993, p. 01-23.

- [NIE89] NIELSEN, Jacob. Evaluating Hypertext Usability. In: *Proceedings of the NATO Advanced Research Workshop on designing Hypertext/Hypermedia for Learning*. Berlin: Springer-Verlag, 1990. p. 147-168.
- [NIE90] NIELSEN, J. Navigation through hypertext. *Communications of the ACM*. USA, v. 33, n. 3, p. 297-309, Mar 1990.
- [NIE93] NIELSEN, Jakob. *Hypertext & Hypermedia*. Cambridge, MA: AP Professional, 1993, 296 p.
- [NIE95] NIELSEN, Jakob. *Multimedia and Hypertext: The Internet and Beyond*. Morristown, New Jersey: AP Professional, 1995, 480 p.
- [NWA96] NWANA, Hyacinth. Software Agents: An Overview. *Knowledge Engineering Review*. UK, v. 11, n. 3, p. 205-244, Oct./Nov. 1996.
- [O'HA96] O'HARE, Gregory J. M., JENNINGS, Nick R. Preface. In: *Foundations of Distributed Artificial Intelligence*. New York: John Wiley & Sons, 1996, p. xii-ix (edited by G. M. P. O'Hare and N. R. Jennings).
- [O'NE91] O'NEIL Jr., Harold F., SLAWSON, Dean A., BAKER, Eva L. Design of a Domain-Independent Problem-Solving Instructional Strategy for Intelligent Computer-Assisted Instruction. In: *Intelligent Tutoring Systems: evolutions in design*. USA: Lawrence Erlbaum Associates, 1991, p. 69-103.
- [PAP80] PAPERT, Simon. Teaching children thinking. *The computer in the school: Tutor, tool, tutee*. New York: Teachers College Press, 1980. R. P. Taylor Ed.
- [PAR98] PARK, Joonho. Implementation of SNMP Agent by Inferno. Murray Hill, NJ: Bell Labs Innovations, Lucent Technologies, 1998.
- [PAS90] PARK, Ok-choon, SEIDEL, Robert J. Conventional CBI Versus Intelligent CAI: Suggestions for the Development of Future Systems. *Expert Systems and intelligent CAI*. USA, v. 2, p. 139-145, 1990.
- [PEP89] PEPER, G. L.; MACINTYRE, C.; KEENAN, J. Hypertext: a new approach for implementing an expert system. *IBM Expert Systems Interdivisional Technical Liason*. USA, p 305-309. Nov. 1989.
- [POR96] PORTER, Tod S., RILEY, Teresa M. The Effectiveness of Computer Exercises in Introductory Statistics. *The Journal of Economic Education*. USA, v. 27, n. 4, p. 291-299, Fall 1996.
- [PRE91] PREGIBON, Daryl. Incorporating Statistical Expertise into Data Analysis Software. *Proceedings of a Forum in Probability and Algorithms*. USA: National Academy Press, 1991, p. 51-62. National Research Council.
- [RAD91] RADA, Roy. Small, Medium, And Large Hypertext. *Information Processing & Management*. Great Britain, v. 27, n. 6, p. 659-677, 1991.
- [RAD92] RADA, Roy. Converting a Textbook to Hypertext. *ACM Transactions on Information Systems*. USA, v. 10, n. 3, p. 294-315, Jul. 1992.
- [RAE92] RAES, Jan F. M. Inside two commercially available statistical expert systems. *Statistics and Computing*. USA, v. 2, n. 2, p. 55-62, Jun. 1992.
- [RAO90] Rao, U., TURROF, M. Hypertext functionality: A theoretical framework. *International Journal of Human-Computer Interaction*. USA, n. 2, v. 4, p. 333-357, 1990.
- [REI98] REID, Chris. Software Review: Secos 4 Windows, Data Handling, Graph and Mapping Program. *CHEER (Computers in Higher Education Economics Review)*. *CTI Economics*. UK, v. 12, n. 1, 1988. http://www.ilrt.bris.ac.uk/ctiecon/cheer/ch12_1/ch12_1p40.htm.

- [RID98] RIDDIOUGH, Ralph. Using Data in Business, Economics and Social Sciences. *CTI Statistics*. http://www.stats.gla.ac.uk/cti/activities/reviews/96_11/excel/using_data.html.
- [RIP97] RIPLEY, B. D. S-PLUS for Windows version 4.0. *Maths&Stats Newsletter*. Nov. 97. http://www.stats.gla.ac.uk/cti/activities/reviews/97_11/splus4.html.
- [ROB90] ROBERTS, Franklin C., PARK, Ok-choon. Intelligent Computer-Assisted Instruction: An Explanation and Overview. *Expert Systems and Intelligent CAI*. USA, v. 2, p. 131-136, 1990.
- [ROB95] ROBINSON, Derek. An Introduction to Regression Graphics by Dennis Cook and Sanford Weisberg, University of Minnesota. *Maths&Stats Newsletter*. UK, Aug. 1995. http://www.stats.gla.ac.uk/cti/activities/reviews/95_08/
- [ROB97] ROBINSON, Derek. CrashCourse in Statistics. *Maths&Stats Newsletter*. UK. Aug. 1997. http://www.stats.gla.ac.uk/cti/activities/reviews/97_08/
- [ROM90] ROMISZOWSKI, Alexander J. Expert Systems in Education and Training: Automated Job Aids or Sophisticated Instructional Media? *Expert Systems and Intelligent CAI*. USA, v. 2, p. 17-25, 1990.
- [ROW96] ROWE, Jeff. *Webmaster's: Building Internet Database Servers with CGI*. USA, New Riders Publishing, 1996.
- [RUS95] RUSSEL, Stuart J., NORVIG, Peter. *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, New Jersey: Prentice Hall, 1995, 932 p.
- [SAN95] SANTOS, Neide. Desenvolvimento de Aplicações Hiperídia (Mini-Curso). In: *Multimídia: da Teoria à Prática (Workshop), 1995, Salvador*. Anais... Set. 1995. p. 1-24.
- [SHN89] SHNEIDERMAN, Ben. Reflections on Authoring, Editing, and Managing Hypertext. In *The Society of Text, Hypertext, Hypermedia, and the Social Construction of Information*. USA: The MIT Press, 1989, p. 115-131.
- [SHU94] SHUTE, Valerie J., GAWLICK-GRENDELL, Lisa A. What Does the Computer Contribute to Learning? *Computers and Education*. Great Britain, v. 23, n. 3, p. 177-186, 1994.
- [SME91] SMEATON, A. Using hypertext for computer based learning. *Computers & Education*. UK, v. 16, n. 3, p. 173-179, May 1991.
- [SMI94] SMITH, D. C., CYPHER, A., SPOHRER, J. KidSim: Programming Agentes Without a Programming Language. USA, *Communications of the ACM*, v. 37, n. 7, p. 55-67, Jul. 1994.
- [SNE95] SNELL, J. Laurie. The Journal of Statistics Education Information Service and Other Internet Resources for Statistics Teachers. *The American Statistician*. USA, v. 49, n. 4, p. 372-375, Nov. 1995.
- [SNE95] SNEDDON, Pete. StatTutor. *Software News*. UK, v. 6, n. 1, Sep. 1995. <http://www.york.ac.uk/ctipsych/web/CTI/DirTxt/reviews/stattutor.html>
- [STA90] STANTON, N., STAMMER, R. A comparison of structured and unstructured navigation through a CBT package. *Computers & Education*. UK, v. 15, n. 1-3, p.159-163, Jan. 1990.
- [STA96] STAN, Franklin, GRAESSER, Art. Is It an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In: *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*. Berlin: Springer-Verlag, 1996.

- [STE91] STERLING, Joan, GRAY, Mary W. The Effect of Simulation Software on Students' Attitudes and Understanding in Introductory Statistics. *Journal of Computers in Mathematics and Science Teaching*. USA, v. 10, n. 4, p. 51-56, Summer 1991.
- [STO89] STOTTS, P. D., FURUTA, R. Petri-net-based hypertext: Document structure with browsing semantics. *ACM Transactions on Information Systems*. USA, v. 7, n. 1, , p. 3-29, Jan. 1989.
- [STR88] STROUSTRUP, Bjarne. What is Object-Oriented Programming? *IEEE Software*. USA, v. 5, n.2, p. 10-20, May 1988.
- [STR91] STERLING, Joan, GRAY, Mary W. The Effect of Simulation Software on Students' Attitudes and Understanding in Introductory Statistics. *Journal of Computers in Mathematics and Science Teaching*. USA, v. 10, n. 4, Summer 1991, p. 51-56.
- [STR97] STRABER, Markus, BAUMANN, Joachim, HOHL, Fritz. Mole - A Java Based Mobile Agent System. Stuttgart: IPVR (Institute for Parallel and Distributed Computer Systems), University of Stuttgart, 1997.
- [STU88] STUETZLE, Werner. Plot Windows. *JASA - Journal of the American Statistical Association*. USA, v. 82, n. 398, p. 466-475, Jun. 1987.
- [SYC96] SYCARA, Katia et al. Distributed Intelligent Agents. *IEEE Expert on Intelligent Systems*. USA, p. 36-46, Dec. 1996.
- [TAG97] TAGG, Stephen. ViSta Version 4.27. *Maths&Stats Newsletter*. UK, Aug. 1997. http://www.stats.gla.ac.uk/cti/activities/reviews/97_08/vista.html.
- [TAL98] TALBOT, Mike. SMART: Introducing Specialist Statistical Techniques via de Web. *Maths&Stats Newsletter*. UK. Feb. 1998. http://www.stats.gla.ac.uk/cti/activities/reviews/98_02/smart.html.
- [TAY80] TAYLOR, R. P. (Ed.) *The computer in the school: Tutor, tool, tutee*. New York: Teachers College Press, 1980.
- [TAY97] TAYLOR, Paul. Genstat 5 Release 3.2 for Windows. *Maths&Stats Newsletter*. May 1997. http://www.stats.gla.ac.uk/cti/activities/reviews/97_05/genstat5.html.
- [TEO98] TEODORESCU, Horia-Nicolai, KANDEL; Abraham, JAIN, Lakhmi C. *Fuzzy and Neuro-Fuzzy Systems in Medicine*. Washington, DC: CRC Press LLC, 1998.
- [UPT96] UPTON, Graham. Oriana Version 1.0 for Windows. *Maths&Stats Newsletter*. UK, Aug. 1996. http://www.stats.gla.ac.uk/cti/activities/reviews/96_08/oriana.html.
- [VEL92] VELLEMAN, Paul F. Serendipitous Data and Future Statistical Software. *Probability and Algorithms*. USA: National Academy Press., 1992, p. 15-22. Panel on Probability and Algorithms. National Research Council.
- [VEL96] VELLEMAN, Paul F., MOORE, David S. Multimedia for Teaching Statistics: Promises and Pitfalls. *The American Statistician*. USA, v. 50, n. 3, p. 217-225, Aug. 1996.
- [VEN91] VENEZKY, Richard, OSIN, Luis. *The Intelligent Design of Computer-Assisted Instruction*. White Plains, New York: Y Longman, 1991.
- [VIA91] VIALI, Lorí. *Simulação de Sistemas de Manufatura*. Florianópolis: UFSC, 1991, 186 p. Dissertação (Mestrado em Engenharia de Produção) - Universidade Federal de Santa Catarina, Out. 1991.

- [WIN91] WIN, W. The assumptions of constructivism and instructional design. *Educational Technology*. USA, v. 31, n. 9, p. 38-40, Sep. 1991.
- [WON97] WONG, David, PACIOREK, Noemi, WALSH, Tom. Concordia: An Infrastructure for Collaborating Mobile Agents. Waltham, MA: Horizon Systems Laboratory, Mitsubishi Electric ITA, 1997.
- [WOO95] WOOLDRIDGE, Michael, JENNINGS, Nicholas R. Agents Theories, Arcitectures, and Languages: a Survey. In": *Intelligent Agents (Wooldridge and Jennings Eds)*. Berlin: Springer-Verlag, 1995, p. 01-22.
- [WOO95] WOOLDRIDGE, Michael, JENNINGS, Nicholas R. Intelligents Agents: Theory and Practice. *Knowledge Enginerig Review*. UK, v. 10, n. 2, p. 115-152, 1995.
- [YAN88] YANKELOVICH, Nicole et al. Intermedia: The Concept and the Construction of an Seamless Information Environment. *Computer*. USA, v. 21, n. 1, p. 81-96, Jan. 1988
- [YAZ87] YAZDANI, Masoud. Intelligent Tutoring Systems: An Overview. *Artificial Intelligence and Education*. USA: Ablex Publising, 1987, p. 183-201. Volume One: Learning Environments and Tutoring Systems.
- [YIL96] YILMAZ, Mustafa R. The Challenge of Teaching Statistics to Non-Specialists. *Journal of Statistics Education (Online)*. USA, v. 4, n. 1, 1996. <http://www.stat.ncsu/info/jse/v4n1/yilmaz.html>.
- [YOU95] YOUNG, F. W. LUBINSKY, D. J. Guiding Data Analysis with Visual Statistical Strategies. *Journal of Computational and Graphical Statistics*. USA, v. 4, p. 229-250.
- [YOU95] YOUNG, Karen. First Bayes. *Maths&Stats Newsletter*. UK, Nov. 1995. http://www.stats.gla.ac.uk/cti/activities/reviews/95_08/firstbayes.html.