

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

**Hyper-Automaton:
Avaliação Interativa de Alunos em
Cursos na WEB Baseado em Autômatos Finitos**

por

CARLOS TADEU QUEIROZ DE MORAIS

Dissertação submetida à avaliação como requisito parcial para
a obtenção do grau de Mestre em Ciência da Computação.

Prof. Dr. Paulo Fernando Blauth Menezes

Orientador

Prof. Dr. Ricardo Reis

Co-Orientador

Porto Alegre, fevereiro de 2002.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Morais, Carlos Tadeu Queiroz de

Hyper-Automaton: Avaliação Interativa de Alunos em Cursos na WEB Baseado em Autômatos Finitos / por Carlos Tadeu Queiroz de Moraes. – Porto Alegre: PPGC da UFRGS, 2002.

111 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2002. Orientador: Menezes, Paulo Fernando Blauth; Co-Orientador: Reis, Ricardo.

1. Teoria dos Autômatos. 2. Exercícios e Avaliações baseados em autômatos. 3. Modelos de hipertexto. 4. WWW. I. Menezes, Paulo F. Blauth. II. Reis, Ricardo. III. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Profa. Wranna Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitor Adjunto de Pós-Graduação: Prof. Jaime Evaldo Fensterseifer

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Carlos Alberto Heuser

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*“Crianças diferentes devem ser ensinadas diferentes,
porque o indivíduo compreende o mundo de diferentes maneiras”.*

Emmanuel Kant

*“Nao é a primeira vez que a aparição de novas tecnologias intelectuais
é acompanhada por uma modificação nas normas do saber.”*

Pierre Lévy

*“O professor bem sucedido já não é uma eminência que bombeia conhecimento a
pressão em receptáculos passivos [...] É um estudante mais velho ansioso por ajudar os
seus colegas mais novos “.*

William Osler

Agradecimentos

A conclusão de um curso de Mestrado, além do término de mais uma etapa de formação acadêmica, significa o amadurecimento do pesquisador através da adoção sistemática e consistente do método científico em seus trabalhos. Essa capacidade de pensar de forma organizada, de embasar opiniões, de conhecer o passado e suas conseqüências, de aplicar o conhecimento de maneira coerente, enfim, de ser capaz, técnica e eticamente, de formular mais do que palavras ao vento de acordo com interesses pessoais devem ser o maior aprendizado a se obter pelo aluno e também cidadão. Infelizmente, essas são qualidades raras nos tempos atuais, onde a aparência vale muito mais do que os conteúdos, expressões da moda carecem de significado e perde-se o compromisso com a verdade.

Neste momento de alegria, presto as minhas homenagens àqueles que tornaram possível a superação dos obstáculos pessoais que sempre surgem em longas caminhadas. Em primeiro lugar, sem dúvida, minha esposa Maria José Martins de Moraes e aos meus filhos Tiago e Lucas Martins de Moraes, que me proporcionam sempre carinho, segurança e amor em qualquer situação. Sem este apoio, seria impossível qualquer tipo de conquista.

O maior agradecimento que eu posso fazer neste momento é ao meu orientador, Prof. Dr. Paulo Fernando Blauth Menezes e ao co-orientador Prof. Dr. Ricardo Reis, que me ajudaram de forma decisiva no aprendizado de como fazer pesquisa e ciência. Como seus grandes conhecimentos acadêmicos e profissionais foram decisivos para a realização deste trabalho, pois, além de possuí-lo, teve competência para ensiná-lo. Minha homenagem muito especial, que há mais de dois anos me dedica seu apoio e amizade Júlio Machado ao GRUPO especial de bolsistas organizado pelo professor Blauth, sem eles seria impossível à realização operacional desta dissertação: Guilherme, Kaqui, Link, Penczek e nosso novo integrante Roges.

Também merecem meu profundo agradecimento os demais professores e funcionários do Instituto de Informática da UFRGS e da Universidade em geral. Acredito que essas pessoas, ou sua grande maioria, estejam entre aquelas que acreditam e lutam pela Universidade pública, gratuita e de qualidade e que, com dedicação e doação, defendem a educação em um país cujos governos possuem todos os motivos para querer vê-la cada vez mais restrita e ineficiente.

Por fim, agradeço ao CNPq pelo auxílio financeiro e ao Instituto de Informática da UFRGS pelo apoio material, imprescindível para que se tenha a dedicação necessária à elaboração de um importante trabalho. Este trabalho de pesquisa foi financiado em parte por FAPERGS (Projeto QaP-For), CNPq (Projetos HoVer-CAM, MEFIA) e CAPES (Projeto TEIA).

Sumário

| | |
|---|-----------|
| Lista de Abreviaturas..... | 7 |
| Lista de Figuras..... | 8 |
| Lista de Tabelas.. .. | 9 |
| Resumo..... | 10 |
| Abstract..... | 11 |
| 1 Introdução | 12 |
| 1.1 Contexto do Trabalho | 12 |
| 1.2 Objetivo | 14 |
| 1.3 Descrição do Trabalho | 16 |
| 1.4 Proposta e Benefícios da Dissertação..... | 19 |
| 1.5 Organização do texto..... | 20 |
| 2 Avaliação Interativa de Alunos em Cursos na WEB Baseado em Autômatos Finitos | 24 |
| 2.1 Principais características de ferramentas existentes para WEB..... | 24 |
| 2.2 Modelos de Hipertextos Trellis..... | 24 |
| 2.2.1 Modelo <i>Object-Oriented Hypertext Design Model</i> | 24 |
| 2.2.2 Modelo <i>Relationship Management Methodology</i> | 25 |
| 2.2.3 Padrões de Projeto Hipermídia | 26 |
| 2.3 Ferramentas para Desenvolvimento para WEB..... | 29 |
| 2.3.1 Linguagem HTML..... | 29 |
| 2.3.2 A Linguagem de Programação CGI | 30 |
| 2.3.3 A Linguagem de Programação Java | 31 |
| 2.4 Mecanismos para Exercícios | 33 |
| 2.5 Ferramentas para Exercícios..... | 34 |
| 2.5.1 Question Mark Designer..... | 34 |
| 2.5.2 Hot Potatoes..... | 39 |
| 2.5.3 Pathware 4 | 41 |
| 2.5.4 Outros Sistemas | 45 |
| 2.6 Hyper-Automaton..... | 46 |
| 2.7 Comparação entre Sistemas | 52 |
| 3 Study of Rooms for the Teaching Mediated by Computer..... | 55 |
| 3.1 Abstract | 56 |
| 3.2 Introduction | 56 |
| 3.3 Researches Accomplished at the Universities and Schools..... | 56 |
| 3.4 Comparison of the Models | 58 |
| 3.5 Suggestions for an ideal Room | 59 |
| 3.6 Proposal | 60 |
| 3.6.1 Proposed I | 61 |
| 3.6.2 Proposed II..... | 61 |
| 3.7 Future Works..... | 62 |
| 3.8 Conclusion | 62 |
| 4 Exercícios e Avaliações Interativo Mediadas por Computador Baseado em Sistema Formal | 64 |
| 4.1 Resumo | 65 |
| 4.2 Introdução | 65 |

| | | |
|----------|---|------------|
| 4.3 | Configuração do Ambiente de uma Sala | 66 |
| 4.4 | Exercícios e Exames Mediados por Computadores..... | 67 |
| 4.5 | Mecanismo para Elaboração de Exercícios e Exames. | 68 |
| 4.6 | Característica do Sistema de Exercícios e Avaliações..... | 69 |
| 4.7 | Sistema Proposto Utilizando Autômato para Elaboração dos Exercícios... | 69 |
| 4.8 | Implementação..... | 70 |
| 4.9 | Caso Real | 70 |
| 4.10 | Conclusão | 73 |
| 4.11 | Trabalhos Futuros | 73 |
| 5 | A WEB Teaching System Based on Formal Methods..... | 74 |
| 5.1 | Introduction | 75 |
| 5.2 | Project TEIA..... | 76 |
| 5.3 | Hyper-Automaton System | 76 |
| 5.4 | Exercises and Evaluations using the Automata..... | 78 |
| 5.5 | Adaptive Evaluations | 80 |
| 5.6 | Conclusion | 81 |
| 6 | Modelagem para Geração Automática de Exercícios e Avaliações..... | 82 |
| 6.1 | Abstract | 83 |
| 6.2 | Introduction | 83 |
| 6.3 | At the Present Time..... | 84 |
| 6.4 | Proposed Situation..... | 84 |
| 6.5 | Noonexistence of Technical Restrictions | 86 |
| 6.6 | Specification of the System | 86 |
| 6.7 | Applying in a Real Case | 89 |
| 6.8 | General Conception of an Automatic Generator | 90 |
| 6.9 | Important Aspects for Modeling / Implementation | 91 |
| 6.10 | Conclusion | 92 |
| 7 | Sistema Hyper-Automaton para Cursos e Avaliações Mediadas por Computador Baseado em Autômatos Finitos..... | 93 |
| 7.2 | Introdução | 94 |
| 7.3 | Cursos no Hyper-Automaton | 95 |
| 7.4 | Exercícios e Provas no Hyper-Automaton | 95 |
| 7.5 | Provas e Exercícios Adaptativos no Hyper-Automaton..... | 96 |
| 7.6 | Adaptação do Sistema com Geração Automática de Exercícios e Provas... | 96 |
| 7.7 | Integração do Curso com Geração Automática de Exercícios e Provas | 97 |
| 7.8 | Conclusão | 97 |
| 8 | Conclusão..... | 99 |
| 8.1 | Principais Propostas da Dissertação | 99 |
| 8.1.1 | Conceituação de Hyper-Automaton | 100 |
| 8.1.2 | Razões para usar o Hyper-Automaton:..... | 101 |
| 8.1.3 | Abordagem Proposta em Hyper-Automaton | 101 |
| 8.1.4 | Ambiente de Desenvolvimento Baseado em Hyper-Automaton..... | 103 |
| 8.2 | Benefícios do Hyper-Automaton | 103 |
| 8.3 | Trabalhos Futuros | 104 |
| | Bibliografia..... | 106 |

Lista de Abreviaturas

| | |
|--------|--|
| ABEND | Associação Brasileira de Educação a Distância |
| ABENGE | Associação Brasileira de Ensino de Engenharia |
| AFS | Autômatos Finitos com Saída |
| BD | Banco de Dados |
| CGI | Common Gateway Interface |
| CNPq | Conselho Nacional de Desenvolvimento Científico e Tecnológico |
| DTD | Document Type Definition |
| EVI | Ensino via Internet |
| EAD | Ensino a Distância |
| ER | Entidade Relacionamento |
| HOT | Hot Potatoes |
| HPA | Hyper-Automaton |
| HTML | Hypertext Mark-up Language |
| HTTP | Hypertext Transfer Protocol |
| ICECE | International Conference on Engineering and Computer Education |
| IEEE | Institute of Electrical and Electronic Engineers |
| ISTEC | Ibero-American Science and Technology Education Consortium |
| JDBC | Java Database Connectivity |
| M/SET | International Conference on Mathematics/Science Education and Technology |
| PPGC | Programa de Pós-Graduação em Computação |
| OO | Orientado a Objetos |
| OOHDM | Object-Oriented Hypertext Design Model |
| QML | Question Mark-up Language |
| RMDM | Relationship Management Data Model |
| RMM | Relationship Management Methodology |
| SBC | Sociedade Brasileira de Computação |
| SBIE | Simpósio Brasileiro de Informática na Educação |
| SENAC | Serviço Nacional de Aprendizagem Comercial |
| SGBD | Sistema de Gerência de Banco de Dados |
| SGML | Standard Generalized Mark-up Language |
| SQL | Structured Query Language |
| TEIA | Técnicas de Ensino Interativas Assistidas por Computador |
| UFRGS | Universidade Federal do Rio Grande do Sul |
| UML | Unified Modeling Language |
| URL | Uniform Resource Locator |
| ULBRA | Universidade Luterana do Brasil |
| VLSI | Very Large Scale Integrated |
| WWW | World-Wide WEB |
| XML | Extended Mark-up Language |
| XSL | Extended Style Language |

Lista de Figuras

| | |
|---|-----|
| FIGURA 1 - Diagrama RMDM (sem <i>slices</i>) | 26 |
| FIGURA 2 - Exemplo de hipertexto | 28 |
| FIGURA 3 - Funcionamento de um programa CGI | 30 |
| FIGURA 4 - QML Construção de um arquivo de perguntas | 35 |
| FIGURA 5 - QML Bloco de Perguntas | 36 |
| FIGURA 6 - QML Tipos de Perguntas | 36 |
| FIGURA 7 - QML Estilo Múltipla-Escolha | 37 |
| FIGURA 8 - QML Editor de Questões..... | 39 |
| FIGURA 9 - HOT Tela Principal | 40 |
| FIGURA 10 - PTW Arquitetura Pathware | 42 |
| FIGURA 11 - PTW Módulo Home | 42 |
| FIGURA 12 - PTW Módulo Tarefa. | 43 |
| FIGURA 13 - PTW Módulo de Resultados | 44 |
| FIGURA 14 - PTW Relatório mede a efetividade do curso | 44 |
| FIGURA 15 - PTW Relatório Avaliação on-line. | 45 |
| FIGURA 16 - Hyper-Automaton – Modo de Criação | 47 |
| FIGURA 17 - Hyper-Automaton Modo de Criação | 48 |
| FIGURA 18 - Hyper-Automaton Montagem do Alfabeto | 48 |
| FIGURA 19 - Hyper-Automaton Escolha do Arquivo..... | 49 |
| FIGURA 20 - Hyper-Automaton Montagem da Função Programa | 49 |
| FIGURA 21 - Hyper-Automaton Ligações Lógicas..... | 50 |
| FIGURA 22 - Hyper-Automaton Resposta Corretas/Incorretas..... | 51 |
| FIGURA 23 - Layout format "U" | 61 |
| FIGURA 24 - Layout type "M" or two rooms in "U" | 61 |
| FIGURA 25 - TEIA Project Layout Room. | 62 |
| FIGURA 26 - Proposed Layout Projects TEIA..... | 62 |
| FIGURA 27 - Layout Sala em Linha..... | 67 |
| FIGURA 28 - Layout Proposto em "U" | 67 |
| FIGURA 29 - MPI007 Projeto de Sistemas VLSI | 71 |
| FIGURA 30 - Linguagens Formais Teste 2. | 72 |
| FIGURA 31 - Student Interface..... | 77 |
| FIGURA 32 - Courses on VLSI Projects | 78 |
| FIGURA 33 - Assessment on Formal Languages | 78 |
| FIGURA 34 - Exercises/Evaluations Automata's Model | 79 |
| FIGURA 35 - Adaptive Test Moore Machine Model | 80 |
| FIGURA 36 - Adaptive Test Mealy Machine Model..... | 80 |
| FIGURA 37 - DER Relationship basic processes of the exercises/exams | 85 |
| FIGURA 38 - Architecture of the System | 86 |
| FIGURA 39 - CGI Event..... | 87 |
| FIGURA 40 - General Architecture of the Automatic Uses Cases | 88 |
| FIGURA 41 - Maintenance of Attributes Exercises..... | 89 |
| FIGURA 42 - Adaptive Question Moore Machine | 90 |
| FIGURA 43 - Adaptive Question Mealy Machine..... | 90 |
| FIGURA 44 - Modelo de AFS nas Transições (Maquina Mealy)..... | 95 |
| FIGURA 45 - Modelo de Prova Maquina Mealy | 96 |
| FIGURA 46 - Integração das Ferramentas. | 105 |

Lista de Tabelas

| | |
|--|----|
| TABELA 1 - QML Tipos de exercício..... | 38 |
| TABELA 2 - PTW Módulos Pathware | 41 |
| TABELA 3 - Comparação de Mecanismos Resolução de Exercícios..... | 52 |

Resumo

O objetivo desta dissertação é a elaboração de uma técnica da aplicação do formalismo de Autômatos Finitos com Saída (Máquina de Mealy e Máquina de Moore) como um modelo estrutural para a organização de hiperdocumentos instrucionais, em destacar especial, Avaliação e Exercício. Esse objetivo é motivado pela organização e agilização do processo de avaliação proporcionado ao professor e ao aluno.

Existem diferentes técnicas de ensino utilizadas na Internet, algumas dessas continuam sendo projetadas com o uso de metodologias tradicionais de desenvolvimento, outras têm a capacidade de modelar de forma integrada e consistente alguns aspectos necessários para uma aplicação WEB. Para alcançar o objetivo proposto, foram realizadas pesquisas nas várias áreas abrangidas pelo tema em evidência, tanto relativo ao processo tradicional (aplicação de prova utilizando metodologia tradicional), como o desenvolvimento de *software* mediado por computador e uso da Internet em si. A modelagem de desenvolvimento para Internet deve integrar características de técnicas de projeto de sistemas de hipermídia devido à natureza hipertextual da Internet. O uso de hiperdocumento como autômatos com saída está na forma básica de representação de hipertexto, em que cada fragmento de informação é associado a um nodo ou a um link (estado/transições) do grafo. Sendo assim, os arcos direcionados representam relacionamentos entre os nodos ou links, ou seja, uma passagem do nodo origem para o nodo destino. As n-uplas dos autômatos apresentam uma correspondência as estruturas de hiperdocumentos na WEB, seu estado/transição inicial corresponde a sua primeira página e suas transições definidas na função programa, funcionam como ligações lógicas, quando selecionadas durante a navegação do hipertexto. Entretanto, faz-se necessário um levantamento dos modelos de hipertextos e das ferramentas de implementação disponíveis para a Internet, a fim de que seja capaz de suportar as peculiaridades do ambiente. Tudo isso deve ser integrado preferencialmente em um paradigma de desenvolvimento amplamente aceito, para que os projetistas não tenham muitas dificuldades em assimilar os conceitos propostos.

A proposta apresentada nesta dissertação, batizada de Hyper-Automaton (hipertexto e autômato), consiste na integração de um Curso na WEB, utilizando formalismo de Autômatos Finitos com Saída para a modelagem dos conceitos necessários e definição das fases adequadas para completar a especificação de Sistema Exercício e Avaliação, bem como a especificação da Geração Automática dos Exercícios e Avaliações baseadas em autômatos para a WEB. Os modelos criados abrangem conceitos de Máquina de Mealy, Máquina de Moore e Aplicações de Hiperdocumentos e Ferramentas de Programação para Internet, os mesmos já testados em caso real. Os parâmetros apurados, nos testes, serviram a uma seqüência de etapas importantes para modelar e complementar a especificação do sistema projetado. Com os parâmetros e etapas de modelagem, a metodologia Hyper-Automaton consegue integrar, de forma consistente, as vantagens de várias técnicas específicas de modelagem de documentos e sistemas de hipermídia. Essas vantagens, aliadas ao suporte às ferramentas de desenvolvimento para Internet, garantem que a metodologia fique adequada para a modelagem de Sistemas com aplicação de métodos de autômatos para exercícios e avaliação na WEB.

Palavras-chaves: Teoria dos Autômatos, Exercícios e Avaliações baseados em autômatos, Modelos de hipertextos, WWW.

TITLE: “INTERACTIVE EVALUATION OF STUDENTS IN COURSES IN THE WEB BASED ON FINITE AUTOMATA”.

Abstract

This dissertation aims to elaborate a technique for applying Output Finite Automata (Mealy and Moore Machines) as a structural model for organizing instructional hyperdocuments, specially for assessment and exercises. We intend to organize and speed up the assessment process helping the teacher and the student. There are many different techniques for teaching through the Internet. Some of them are still projected following traditional developing methodologies, and others have the capacity to model applications for the Web in a more integrated and consistent way. In order to reach the objectives we realized many researches reporting to the assessment process itself and the developing process of software for distance learning (more specifically through Internet) and other applications of computers in education. When we develop Internet directed software we must integrate hypermedia techniques to the project because of its hypertext nature. Output finite automata are basic representations for hyperdocuments where each information piece is associated to a graph node or link (state/transition). In this way, directed arcs represent relationships among nodes and links, i.e., among source and target nodes. When we formally define automata, its n-uples present a natural relationship with WEB hyperdocuments. Its initial state might represent its first page and its program function might work as logical links selected into the hyperdocument. It is necessary to figure out which hypertext models and tools are available for developing systems for the Internet in order to have support for its peculiar environment. Ideally, all of its characteristics have to be integrated into a well-known and accepted paradigm. In this way, designers do not have too many difficulties to assimilate its concepts. The Hyper-Automaton model (hypertext and automaton) proposed here consists of integrated WEB course that makes use of Output Finite Automata formalisms for modelling. It is necessary to define adequate steps and concepts for completing the specification of exercises and assessment systems, as well as to specify their automatic generation for Web. The created models have Mealy and Moore machines concepts applied to Internet and were already successfully tested in real educational programs. The parameters generated from these tests served as important data to complement the specification of this proposed model. Having these parameters and steps modelled, Hyper-Automaton is able to consistently integrate with advantages many modelling techniques for specifying hypermedia systems. These advantages give adequate support for developing educational Internet systems, specially for applying exercises and for assessment on the Web.

Keywords: Automata theory, Automata base exercises and assessment, hypertext models, WWW.

1 Introdução

Este documento descreve a dissertação submetida à avaliação, como requisito parcial, para obtenção do grau de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Computação (PPGC) da Universidade Federal do Rio Grande do Sul. A dissertação aborda a análise e a implementação de um sistema baseado em conceitos da teoria de autômatos para o controle e a organização de hiperdocumentos na *World Wide Web*, com ênfase na avaliação interativa.

A dissertação está organizada como uma coletânea de artigos produzidos durante o curso. Apresenta-se, a seguir, uma introdução e um capítulo introdutório extenso com objetivo de propiciar ao leitor um resumo estendido da dissertação. Os demais capítulos estão organizados em forma de artigos, constituindo uma coletânea dos principais artigos produzidos e publicados durante o período de pesquisa. Esses artigos, portanto, apresentam um detalhamento da dissertação.

1.1 Contexto do Trabalho

A utilização da informática na educação indica que estamos vivendo um período forte de transformação, surgindo então, a necessidade de formar professores e alunos capacitados de interagir com a máquina na busca de resultados positivos e preciosos para a solução dos problemas de aprendizagem.

Nesse aspecto, o primeiro passo para a solução é desafiar professor e aluno a explorar a potencialidade da máquina, buscando a habilidade do raciocínio lógico da criatividade e do senso crítico. Para isso, não existirá um único instrumento de ensino na aprendizagem.

A interface dos modelos de testes e listas de exercícios tornar-se-á interativa entre o aluno e o sistema, com o surgimento de cursos mediados por computadores. Observa-se que os estudantes utilizam os computadores quase que diariamente na execução de tarefas, tanto em classe como extraclasse.

Dentro desse contexto, a máquina é considerada como um mecanismo físico, com a finalidade de substituir o homem na execução de algumas tarefas. O uso do computador, nesse caso, parece ter sido iniciado por estudantes que contestavam em realizar avaliação no papel, enquanto, no curso inteiro usavam computadores. [THO 96]

O termo "interatividade" significa associar uma atividade com recurso informático e não informático nas relações aluno-aluno, aluno-docente e docente-docente. [BRU 94]

Uma situação de aprendizagem compreende diversos atores: os alunos e as atividades que eles supõem efetuar; o professor e o papel que ele deve representar e o sistema informático.

Entretanto, deve-se considerar o fato de que esses instrumentos não atuam sozinhos e que eles estão ligados a uma classe de situações. Assim, é necessário prever possibilidades de extensão e de verificação. Para tanto, se faz necessário avançar para o

conceito de interação. Esse pressupõe a criação, ou seja, a colocação do aluno na função de autor.

A interatividade consiste na manipulação de todos os recursos disponíveis dentro de uma modelagem, ou seja, tudo aquilo que é fornecido pela máquina e colocado à disposição do usuário, enquanto a interação pressupõe a intervenção do aluno como autor, ou vale dizer, permite a autonomia do usuário.

[VAR 91] Afirmam que toda cognição é baseada em ações em tempo real no mundo, o que implica que a própria percepção não é possível sem ação. Mesmo que se possua grande habilidade no uso do computador, as informações que o computador nos apresenta sempre requerem reflexão antes de responder a ele, é sempre objetiva, e geralmente vem de outra pessoa que não nós mesmos evitando a interatividade natural.

As ferramentas de avaliação on-line [TIN 96] permitem a verificação de deficiências no aprendizado dos alunos pela aplicação de testes e exercícios objetivos. Muitos dos sistemas incluem a geração ou seleção automática de questões disponíveis em um banco de dados, cálculo automático das notas resultantes e analisadores da performance dos alunos. Alguns sistemas também oferecem a possibilidade do gerenciamento integrado do conteúdo com os testes, permitindo ao professor estabelecer pontos de realimentação de informações de forma a conseguir acompanhar os pontos mais frágeis do currículo estabelecido.

Se as avaliações serão realizadas ou não por computador, ainda é muito questionado, visto que trabalhos de pesquisa são os mais fáceis de administrar em relação ao tempo de preparação e correção de provas, ainda mais que requer uma segurança para que os estudantes não tenham acesso às informações das provas na rede.

É fato que as universidades públicas enfrentam dificuldades, em particular no Brasil, com relação às densidades demográficas, à distância geográfica, à formação de professores e ao aumento do número de vagas de alunos. Para isso, os sistemas que apresentam formas de aproveitamento e controle dos alunos são de suma importância para ensino a distância, principalmente diminuindo a distância entre professor e aluno buscando dar um atendimento individualizado para cada estudante.

Como o trabalho proposto pode ser usado tanto remotamente como localmente, no decorrer deste texto, serão apresentados diferentes layouts de salas de aula, para aplicação de prova local e atividades de ensino em geral, ou seja, dentro da universidade. Tendo como métrica: aplicação de provas via computador para turmas com 20 a 40 alunos, colocando os alunos em posições difíceis de visualização da tela do computador do colega, facilitando a aproximação do professor ao aluno e possibilitando uma melhor movimentação na sala para formação de grupos e apresentação de seminários.

Este trabalho originou-se de uma proposta inicial feita pelo Prof. Blauth ao seu grupo de pesquisas, em desenvolver um software de navegação Avaliação Interativa de Alunos em Cursos na WEB Baseado em Autômatos Finitos denominado Hyper-Automaton. Foi desenvolvida pelo grupo uma base de criação e edição do Hyper-Automaton, implementado pelos seus bolsistas. Após alguns testes, surgiram algumas propostas: modelar e implementar um curso utilizando como estrutura de navegação o

autômato finito com saída, proposta já defendida como dissertação de mestrado e trabalho de diplomação. A outra fica para este trabalho, o qual propõe desenvolver um sistema de exercícios e avaliação de alunos com aplicação em ambiente remoto e local, constituído por vários exercícios referentes ao curso, passando para o professor um retorno de suas aulas e aos alunos uma forma de revisar e estudar o conteúdo, utilizando uma bateria de exercícios e finalizando com provas.

Foram aplicadas listas de exercícios e testes, para aos cursos de graduação e pós-graduação nas disciplinas Linguagens Formais e Autômatos e Projeto de Sistemas VLSI - UFRGS, com a participação dos educadores na elaboração de seus cursos, exercícios e testes, numa forma de medir parâmetros e dificuldades individuais de seus educandos, tornando a correção mais rápida.

O controle do educando nesse processo de ensino à distância, é bastante relevante, uma vez que existe a personalização da aprendizagem em função: dos vários processos de navegação existentes, dos objetivos específicos que cada aula possui e dos interesses que cada um tem com relação ao conteúdo provido.

Os recursos tradicionais como lápis e caderno continuarão a existir, junto com a nova tecnologia, pois o sistema desenvolvido será uma ferramenta importante para o trabalho do professor, uma vez que pode ser utilizado para diversas atividades, tais como: introduzir um sistema de exercícios permitindo ao aluno aprofundar-se num tema, revisar matérias já estudadas, preparar alunos para provas, atender a alunos de diferentes perfis [KAW 96].

1.2 Objetivo

O objetivo traçado para esta dissertação é a elaboração de uma técnica de desenvolvimento de *software* adaptada à especificação de um modelo de autômato finito de forma completa e consistente, aplicado a um sistema de avaliação interativa de alunos em cursos na WEB. Para alcançá-lo, é proposta a integração ao curso que tem extensão das notações das máquinas de Moore e Mealy de modelagem orientada a saídas em estados e transições, respectivamente, que definem a semântica dos novos conceitos necessários à modelagem de sistemas na WEB. No que se refere ao modelo de curso já foi desenvolvido por [MAC 2000a] e [PEC2000]. Esses modelos acrescentam ao autômato a capacidade de especificação de conceitos relacionados à Modelagem de Documentos, Modelagem de Sistemas Hiperídia e utilização de ferramentas de desenvolvimento para Internet.

Além da adição de conceitos às notações das máquinas de Moore e Mealy, propõe um ciclo de etapas de modelagem para especificação do modelo. Essas etapas têm como objetivo:

- Garantir que as necessidades de modelagem de autômato sejam atendidas de forma consistente e completa, através do uso apropriado do formalismo proposto;
- Desenvolver uma ferramenta interativa de apoio (elaboração, acompanhamento, aplicação e dados para estatística), para os professores (exercícios e avaliações) [MOC 99a, MOC 99b];
- Estudar a viabilidade de um gerador automático de exercícios e provas (autômatos) a partir de um banco de exercícios e provas. [MOC 2000a];

- Integrar o módulo de avaliação ao sistema de curso via WEB, também baseado em autômatos finito [MOC 2000b];
- Estudar e propor modelo de salas para aplicação de exercícios e avaliações locais [MOC 99].

A ferramenta proposta possibilita aos alunos o conhecimento de experiências novas e desafiadoras durante o processo de ensino, viabilizando sua avaliação imediata e agilizando o retorno do seu aprendizado perante o curso local ou remotamente.

Entre os objetivos específicos relativos a exercícios e avaliações, destacam-se:

- Analisar as linguagens de programação para WEB e os modelos de hipertextos que fazem parte dos sistemas de cursos na WEB para Ensino a Distância;
- Apresentar as características de diversos produtos disponíveis no mercado, destacando a funcionalidade e os mecanismos como ferramentas para elaboração de exercícios e teste de avaliação que cada um apresenta;
- Efetuar testes de casos reais, com alunos dos cursos de graduação e pós-graduação nas disciplinas Linguagens Formais e Projeto de Sistemas VLSI - UFRGS, usando linguagens formais e autômatos como contexto de trabalho [MOC 99b].

Com os resultados dos testes realizados, referentes aos objetivos específicos, também serão analisados os parâmetros de:

- Desempenho da sistemática;
- Textos e maneabilidade;
- Receptividade dos alunos.

Os resultados apurados e comparados com a metodologia tradicional servirão de métrica para validar e fornecer parâmetros para opinar sobre a metodologia proposta.

Através dessas etapas, são especificados, além de aspectos comuns a outros sistemas e presentes na ferramenta proposta, a estrutura dos documentos, os contextos de navegação existentes, tabelas de atributos, a estrutura navegacional do sistema e os processos de geração automática.

Com esses elementos e etapas de modelagem, realmente é propiciada a modelagem integrada aos cursos na WEB, especificando seus conceitos fundamentais de forma consistente.

A modelagem de sistemas baseados em documentos também é beneficiada, pois o sistema permite a especificação de estrutura e relacionamento entre documentos, além da capacidade de reuso de documentos dinâmicos e virtuais e integração com bases de dados.

Outro aspecto importante é a integração de mecanismos de busca e índice remissivo para processos de ajuda on-line (dicas) organizadas por tópicos. Por fim, o sistema oferece uma abordagem nova e adequada à modelagem de sistemas hipermídia com relevante comportamento dinâmico, através da definição de fornecer maior flexibilidade e facilidades a professores e alunos.

Por isso, o sistema é apropriado para a especificação de sistemas hipermídia na *WEB*, de forma independente do autômato, com potencial de modularização do material instrucional, servindo como ferramenta de apoio para EAD.

1.3 Descrição do Trabalho

No contexto do ambiente de ensino, a estrutura dos documentos disponibilizada na *WWW*, pode ser categorizada da seguinte maneira: nós, ligações e *WEBS*, as classes as quais podem ser utilizadas dinamicamente são: scripts ou applet Java. Para definição e controle da navegação, podem conter elementos específicos a serem explorados em material didático.

Uma das questões importantes na aplicação do sistema de avaliação é o processo de navegação que o sistema permite para a utilização do World Wide *WEB*. A flexibilidade do hiperdocumento pode servir de apoio à aprendizagem, utilizando os recursos de hipertextos, gerenciando as informações em que estas são armazenadas em uma rede de nós, conectadas através de ligações, podendo ser caracterizada como hipermídia, onde podem ser incluídas outras mídias como som, imagem, desenho e animação.

O fato de o material estar disponível não é necessariamente de grande valia se o usuário não consegue achar facilmente o tópico desejado. Portanto é importante que, as necessidades específicas do usuário sejam consideradas. O que caracteriza um hipertexto é um mecanismo de estruturação de informação textual que divide os dados em porções (nodos), relacionadas entre si (através de *links*). Cada nodo contém informações sobre um assunto específico e fornece meios de se atingir (navegar para) outros nodos que contenham informações relacionadas a uma *âncora* no nodo origem. Hipertexto não é uma idéia nova. A primeira proposta de sistema com essas características surgiram em 1945 no projeto Memex [BUS 45]. No fim da década de 1980, produtos comerciais de hipertexto tornaram-se mais difundidos como o *HYPERCARD*, da Apple e *Hyperties* de Shneiderman e o *NOTECARDS* da Cia. Xerox (citados por [STO 89]). O termo hipermídia se refere a sistemas com estrutura hipertextual, mas com informação em formatos diferentes de texto, como imagens, sons, vídeos, etc.

O uso de hiperdocumento na forma básica de representação de hipertexto através de grafos direcionados, em que cada fragmento de informação é associado a um nodo do grafo e os arcos direcionados representam relacionamentos entre os nodos, como uma passagem potencial do nodo origem para o destino. Entretanto, esse formalismo poderá apresentar deficiência para representar todas as potencialidades de sistemas de hipertexto, gerando a necessidade de métodos mais formais de modelagem, a qual pode ser usada para prover interface de programação para o controle de material hipermídia. O modelo que utiliza o contexto e mecanismo de abstrações (citados em [STO 89]), é o uso da Rede de Petri.

O modelo formal de estruturação de hipertextos baseados em Autômatos Finitos com Saída utilizado no Hyper-Automaton tem a vantagem do fato de autômatos não somente capturarem o poder descritivo de grafos diretos, conhecidos por serem uma abstração muito útil em sistemas de hipertexto [CON 87], como também provê um formalismo matemático preciso de uma máquina abstrata para o controle e a análise da

execução ou "navegação" em um hiperdocumento. Como é um autômato, o modelo apresenta ainda propriedades de linguagens formais e pode ser visto como gerador ou reconhecedor de linguagens [HOP 79], de forma que podem ser utilizados algoritmos e técnicas de verificação bem conhecidos na literatura para a análise de algumas propriedades de um curso na WWW, como por exemplo: garantir que um estudante seja capaz de acessar todos os materiais instrucionais através de pelo menos um caminho de "navegação", verificar a existência de nodos inatingíveis na hiperbase, ou ainda analisar a linguagem reconhecida e/ou gerada pelo autômato para certificar-se que os caminhos possíveis a serem percorridos são coerentes com a estratégia de ensino utilizada pelo autor [MAC 2000]. Demais trabalhos relacionados ao uso do formalismo de autômatos para hiperdocumentos estão citados em [FUR 89, MOL 98, STO 89, TOM 89, WAN 98], estão voltados para modelos formais no estudo da semântica de sistema de hipertexto. As demais técnicas e soluções serão descritas no decorrer deste texto.

A partir de suas características, Nielsen sugeriu as **regras de ouro** para uso de hipertexto. Segunda elas, hipertexto é adequado [NIE 90]:

- Para grande quantidade de informação organizada em pequenas partes;
- Quando partes de informação são relacionadas entre si;
- Quando há muita interação com o usuário;
- Quando o usuário só necessita de uma pequena porção de informação em um determinado momento.

Baseando-se em suas regras, Nielsen também sugeriu exemplos de **sistemas que mais se beneficiariam** de uma estrutura hipertextual [NIE 90]:

- Documentação e ajuda on-line;
- Engenharia de *Software*, para elos entre documentos de fases do ciclo de desenvolvimento e funções relacionadas. Bons exemplos desse uso são os trabalhos de Perin [PER 92] e Ortigosa [ORT 95]. Ambos utilizam hipertexto em ambientes de desenvolvimento de *software* para auxiliar o acesso e a compreensão dos diferentes documentos que compõem a especificação de um programa;
- Processo de auditorias (informação de várias fontes, muita procura e organização);
- Catálogos, jornais, revistas, guias para lojas, livrarias, bibliotecas, museus, etc.

Pesquisas recentes na área de hipermídia estão avançando na direção da incorporação de funcionalidade hipertextual em sistemas tradicionais. O objetivo desses trabalhos é oferecer ao usuário uma forma associativa de acessar, analisar e organizar informações. Segundo Oinas-Kukkonen, “uma interessante direção para **pesquisas futuras é o desenvolvimento da funcionalidade hipermídia** como um método complementar a metodologias estruturadas ou orientadas a objetos para desenvolvimento de sistemas de informação” [OIN 97]. Os benefícios dessa abordagem surgem dos seguintes fatores [BAI 91], [OIN 97]:

- A hipermídia oferece acesso contextual e navegacional à informação, onde facilita acesso a dados não-estruturados, não impõe formas rígidas de consultas, a representação hipertextual de conhecimento é próxima à utilizada pelas pessoas, permitindo um melhor entendimento da informação e facilita o uso do sistema por pessoas não-especializadas;

- A hipermídia é um meio natural de comunicação entre diferentes tipos de usuários, sua estrutura livre de armazenamento capta informação que não se enquadra à rede organizacional (principalmente bancos de dados).

Entretanto, à medida que os sistemas hipertextuais ficam mais complexos, o risco de desorientação do usuário na estrutura de navegação aumenta. Para que isso não aconteça, é necessário que o sistema de hipertexto saiba onde o usuário está e de onde ele veio, tudo em relação à informação (contexto) em cada ponto da navegação [HEA 97]. Alguns mecanismos genéricos de ajuda à orientação do usuário são, por exemplo, botões de retorno, *tours* guiados, índices, “história” da navegação e *bookmarks* [NIE 90].

Atualmente, a tecnologia de hipertexto está em grande evidência por causa do sucesso da WWW, que se tornou o maior ambiente hipermídia da história. Suas características, principalmente sua arquitetura aberta e multiusuário adicionaram novas dificuldades ao desenvolvimento hipermídia. Como forma de diminuir essa complexidade, tanto na Internet como em outros ambientes, tem havido um crescente interesse em Sistemas de Gerência de **Hiperbases**, que separam as camadas de armazenamento e de interface gráfica. Esses gerenciadores de hiperbases devem se preocupar com [WII 97]:

- **Modelos e arquiteturas:** aspectos de escalabilidade, extensibilidade, abertura da arquitetura, interoperabilidade, entre outros;
- **Gerência de nodos, links e estruturas:** identificação de objetos, integridade de objetos e da estrutura de ligações;
- **Navegação e busca:** integração das abordagens de acesso através de hipermídia e de consultas tradicionais;
- **Controle de versões:** especificações de quais entidades são versionadas e quando o versionamento ocorre;
- **Controle de concorrência e transações:** suporte a transações de variadas durações, colaboração e compartilhamento de informação.

Por tudo que foi exposto, é possível caracterizar um conjunto de hiperdocumento no conceito de autômato como um curso que difere do projeto de sistemas tradicionais em vários pontos. Como exemplos, podem ser citadas a captura e a organização da estrutura da informação de domínios complexos e as dificuldades acarretadas pelo uso de diferentes tipos de mídia [ISA 95]. Além disso, quando se considera a Internet e suas características próprias, outras dificuldades surgem, como ferramentas disponíveis, integração com outros sistemas, distribuição de dados, acessos remotos, etc.

O sistema Hyper-Automaton é um sistema semi-automatizado com base em uma arquitetura cliente/servidor na internet, para suporte a cursos na WEB, baseado no conceito de autômato, com aplicação on-line e também off-line.

A seqüência deste capítulo abordará algumas técnicas de desenvolvimento de sistemas hipermídia, padrões e linguagens de programações, direcionada para desenvolvimento de aplicações para WEB.

O objetivo desse estudo é identificar outras características e soluções que devem estar presentes em uma metodologia apropriada à modelagem da funcionalidade existente nesse tipo de sistema.

1.4 Proposta e Benefícios da Dissertação

De uma forma geral, esta dissertação contribui para as áreas de Engenharia de Software, Hipermídia, Informática na Educação e Sistemas Formais, onde existe a preocupação em desenvolver soluções práticas, flexíveis e poderosas para o aperfeiçoamento de sistemas de hipertexto voltados para ambientes distribuídos de ensino.

Como principais contribuições deste trabalho de pesquisa é possível obter:

- a) Apresentação de uma técnica de desenvolvimento de Software adaptado à especificação de um modelo de autômatos finitos de forma completa e consistente, aplicada a um sistema de avaliação interativa de alunos em cursos na WEB, baseando em Autômatos Finitos;
- b) Alcançar e propor a extensão das notações das máquinas de Moore e Mealy de modelagem orientada a saídas em estados e transições respectivamente, que definem a semântica dos novos conceitos necessários à modelagem de sistemas na WEB, conceitos esses inexistentes no modelo convencional. Esses modelos acrescentam ao autômato a capacidade de especificação de conceitos relacionados à Modelagem de Documentos, Modelagem de Sistemas Hipermídia e utilização de ferramentas de desenvolvimento para Internet;
- c) A utilização deste modelo de exercícios será desenvolvida para suporte do ensino de Informática Teórica na UFRGS, tendo como base a criação de uma ferramenta para resolução de exercícios e avaliação que incorpore as novas tecnologias de redes;
- d) Desenvolver uma proposta de um sistema de **Avaliação Interativa de Alunos em Cursos na WEB Baseado em Autômatos Finitos**, de modo simples e flexível tanto para alunos como para professores. As características que a proposta apresenta são: o reuso de listas de exercícios; banco de provas; geração automática de exercícios e provas.

Além de aspectos comuns a outros sistemas e presentes no Hyper-Automaton, a estrutura dos documentos, os contextos de navegação existentes, tabelas de atributos, a estrutura navegacional do sistema e os processos de geração automática, são etapas de modelagem que realmente propiciam a integração aos cursos na WEB, especificando seus aspectos fundamentais de forma consistente.

A modelagem de sistemas baseados em documentos também é beneficiada, pois o sistema permite a especificação de estrutura e relacionamento entre documentos, além da capacidade de reuso de documentos dinâmicos e virtuais e integração com bases de dados.

Outro aspecto importante é a integração de mecanismos de busca e índice remissivos para processos de ajuda on-line (dicas) organizadas por tópicos. Por fim, o sistema, oferece uma abordagem nova e adequada à modelagem de sistemas hipermídia com relevante comportamento dinâmico, fornecendo maior flexibilidade de usabilidade e facilidades aos professores e aos alunos.

Por isso, o sistema é apropriado para a especificação de sistemas hipermídia na WEB, com potencial de modularização do material instrucional, servindo como ferramenta de apoio para EAD.

1.5 Organização do texto

A dissertação está organizada como uma coletânea de artigos produzidos durante o curso. Até o momento da redação deste volume foram produzidos 08 artigos, tanto em nível de autoria quanto co-autoria e publicados em anais dos eventos relacionados aos temas da dissertação (Informática na Educação, Métodos Formais, Hipermídia,...), listados abaixo:

a) Artigos Completos, como autor principal

1. **Study of Rooms for the Teaching Mediated by Computer.** International Conference on Engineering and Computer Education, 1999, **ICECE'99 - IEEE**, Rio de Janeiro. [MOC 99], é uma análise de modelos para salas de aula dirigida ao ensino apoiado pelo uso de computadores. É uma peça auxiliar neste trabalho, pois introduz soluções a serem aplicadas nos ambientes descritos pelo projeto TEIA [WEB98].
2. **Exercícios e Avaliações Interativo Mediado por Computador Baseado em Sistema Formal.** Simpósio Brasileiro de Informática na Educação, 10. 1999 **SBIE'99 - SBC**, Curitiba [MOC 99b]. Desenvolve a análise do modelo de autômatos finitos com saída para a definição de avaliações interativas via WWW.
3. **A WEB Teaching System Based on Formal Methods,** World Computer Congress - **ICEUT2000 - IFIP** Educational Uses of Communication and Information Technologies. Beijing – China [MOC 2000] apresenta um estudo de caso da utilização prática dos sistemas em avaliações nos cursos de Linguagens Formais e Autômatos e Microeletrônica (Projeto VLSI) no Instituto de Informática da UFRGS.
4. **Modelagem para Geração Automática de Exercícios e Avaliações.** International Conference on Engineering and Computer Education, 2000, **ICECE2000 - IEEE**, São Paulo, [MOC 2000a], Apresenta pontos importantes para modelagem do processo de avaliação tradicional para geração automática de exercício e de avaliação, centrada na análise e aplicações do formalismo do autômato finito com saída.

b) Resumos e posters como autor principal

5. **Sistema Hyper-Automaton para Cursos e Avaliações Mediado por Computador Baseado em Autômatos Finitos.** Simpósio Brasileiro de Informática na Educação, 11. 2000, **SBIE2000 – SBC**, Maceió [MOC 2000b], é demonstrado os conceitos e as aplicações dos módulos cursos e exercícios/avaliações, principalmente a integração dos módulos.

c) Artigos Completos como co-autor

6. **Structuring WEB Course Pages as Automata: revising concepts.** Recherche d'Informations Assistee par Ordinateur 2000, Conference on Content-based Multimedia Information Access, 6., 2000, Paris [MAC 2000] desenvolve uma análise crítica do modelo de autômatos finitos com saída em face de conceitos pertinentes a área de hiperdocumento e hipermídia.
7. **Autômatos Finitos: um formalismo para cursos na WEB.** Simpósio Brasileiro de Engenharia de Software, 13., 1999 Florianópolis. [MAC 99a] apresenta questões importantes referentes à implementação do sistema em uma arquitetura cliente-servidor na Internet.
8. **Computer in Classroom: Impact on the Environment and on the Relationships.** International Conference on Engineering and Computer Education, 1999, Rio de Janeiro. [OLI 99] discute o impacto de sistemas de ensino que utilizam o computador como ferramenta de intermediação entre os professores e alunos. Tal estudo surgiu da necessidade de fundamentação de conceitos a serem utilizados na validação do sistema proposto nesta dissertação.

d) Outras publicações

- **Avaliação Interativa de Alunos em Cursos na WEB Baseada em Sistemas Formais**, Semana Acadêmica do PPGC, 4., 1999, Porto Alegre [MOC 99a]

Os artigos publicados completos como primeiro autor, bem como o artigo curto Sistema Hyper-Automaton para Cursos e Avaliações Mediado por Computador Baseado em Autômatos Finitos constituem os capítulos subseqüentes. Cada um desses capítulos inicia-se por uma subseção de introdução, na qual estão descritas as contribuições de cada artigo constante no capítulo, juntamente com informações pertinentes ao evento no qual o artigo foi publicado e um resumo dos principais assuntos tratados no seu texto.

Os artigos estão distribuídos em capítulos que descreve uma evolução de cada etapa do trabalho realizado, de acordo com o crescimento das pesquisas realizadas durante a montagem do texto.

O volume desta dissertação está organizado da seguinte forma:

- capítulo 2, “Avaliação Interativa de Alunos em Cursos na WEB Baseado em Autômatos Finitos”, esse capítulo tem como principal objetivo

apresentar um resumo estendido da dissertação, onde se propõe mostrar um estudo de caso com os principais modelos de hipertextos, linguagens e ferramentas. Tendo como contribuição principal validar o Hyper-Automaton como uma aplicação exclusiva para exercícios e provas. Para concluir essa validação realizaram-se testes com outras aplicações, onde constam as suas principais características dos mecanismos para exercícios e provas e por fim uma comparação com as ferramentas e o protótipo do modelo de autômato. Para construção desse capítulo foram realizados trabalhos paralelos para reforçar os objetivos propostos, esses trabalhos geraram a publicação de artigos que estão citados a seguir:

- capítulo 3, “Study of Rooms for the Teaching Mediated by Computer.”, apresenta um estudo inicial para aplicabilidade de “Cursos On-line” em salas de aula e relaciona pesquisa em universidades em escola. A contribuição total nesse capítulo, para a dissertação, foi pesquisar ambientes escolares onde se aplicam aulas em laboratórios de informática, bem como a realização de provas e exercícios buscando focalizar o espaço entre máquinas e alunos e a movimentação entre alunos e professores, tanto visual quanto a segurança das informações via rede local. Dados que foram importantes para a realização do teste piloto que realizou-se para validar o autômato, descrito no capítulo seguinte.
- capítulo 4, “Exercícios e Avaliações Interativo Mediado por Computador Baseado em Sistema Formal.”, esse capítulo contém uma análise de desenvolvimento do modelo de autômatos finitos com saída para definição de exercícios e avaliações interativas na WWW. A principal contribuição foi testar o modelo de autômato em realização de testes reais, dentro de um ambiente descrito no capítulo anterior, com o objetivo de validá-lo para aplicação de provas e exercícios por computador. Foi realizada uma análise comparativa dos modelos de provas tradicional e on-line. Bem como uma análise da aplicabilidade das provas. A conclusão é que os testes realizados validaram o modelo proposto. Os dados gerados pelos testes definiram uma nova necessidade para o protótipo. Esses e outros aspectos, bem como uma apresentação do modelo como ferramenta de ensino são explorados no capítulo seguinte.
- capítulo 5, “A WEB Teaching System Based on Formal Methods”, apresenta depois de testar e validar o autômato dentro de um ambiente on-line, a aplicabilidade da ferramenta Hyper-Automaton em relação ao ensino como possibilidade de fornecer apoio aos professores e aos alunos nos cursos na WEB. Com a realização de novos testes, testou-se um novo modelo de autômato com provas adaptativas e se obteve um bom resultado. Depois de apresentar o Hyper-Automaton como ferramenta de apoio ao ensino, surgiu a necessidade da geração automática de exercícios e avaliações, descrito no capítulo seguinte.
- capítulo 6, “Modelagem para Geração Automática de Exercícios e Avaliações”, apresenta uma metodologia que especifica um sistema para

geração automática de exercícios e avaliações, utilizando conceitos e características da modelagem de autômatos, bem com técnica de Engenharia de Software. Assim, este capítulo transcende o teste e a validação do modelo de autômatos para avaliações interativas, propondo novos módulos a serem adicionados ao hyper-automaton. Resta agora, a integração do curso, exercícios e avaliações, o que é apresentado no capítulo seguinte;

- capítulo 7, “Sistema Hyper-Automaton para Cursos e Avaliações Mediado por Computador Baseado em Autômatos Finitos”, apresenta a integração entre todos os módulos do sistema Hyper-Automaton, ou seja, cursos, exercícios e provas, módulos adaptativos e a geração automática de exercícios/provas;
- capítulo 8, “Conclusão”, resume os principais pontos da abordagem da dissertação, incluindo contribuições, modelagem e implementação do sistema, benefícios, adequação à modelagem de sistemas para a *WEB* e finalmente trabalhos futuros.

Os demais artigos gerados como co-autor serviram como base para elaboração desta dissertação. Entretanto, como parte do seu conteúdo está completamente inserido nesse texto, seria redundante sua anexação ao trabalho.

2 Avaliação Interativa de Alunos em Cursos na WEB Baseado em Autômatos Finitos

Este capítulo propõe apresentar um resumo estendido da dissertação com objetivo de proporcionar ao leitor uma visão geral da dissertação, onde se propõem mostrar um estudo das principais ferramentas existentes para WEB. Tendo como contribuição principal validar o Hyper-Automaton como uma aplicação exclusiva para exercícios e provas.

2.1 Principais características de ferramentas existentes para WEB

2.2 Modelos de Hipertextos Trellis

O **modelo Trellis** [STO89] de hipertexto utiliza Redes de Petri para dar uma estrutura mais formal ao hipertexto e uma **base matemática** para **documentos**. Lugares de uma rede representam nodos dos hipertextos, e arcos representam *links*. Quando uma ficha está em um lugar, a informação relacionada ao lugar é exibida ao usuário. **As transições habilitadas** representam **os links de navegação** disponíveis no momento. As duas **vantagens principais** do modelo Trellis sobre grafos direcionados normais são a especificação de **caminhos concorrentes** e de **semântica de navegação** [STO 89].

O **modelo Trellis** não especifica uma metodologia completa de modelagem, nem se preocupa com desenvolvimento de sistemas para *WEB*. Entretanto, sua capacidade de modelagem formal de navegação é de grande interesse.

Em primeiro lugar, Redes de Petri são boas para a representação de atividades paralelas, logo, são apropriadas para a modelagem de exibição concorrente de múltiplos elementos e caminhos alternativos de navegação.

Em segundo lugar, Redes de Petri são capazes de modelar os modos como o usuário deve navegar sobre o hipertexto (semântica). Pode-se especificar, por exemplo, a seqüência na qual os nodos devem ser percorridos, pois nem todos os *links* podem estar disponíveis em um determinado momento. Em Trellis, esse controle é especificado de forma integrada à estrutura do hipertexto, não sendo necessária à aplicação de mecanismos externos de navegação. Lugares em uma rede sem informação associada são usados para a modelagem dessas estruturas de controle [STO 89].

2.2.1 Modelo *Object-Oriented Hypertext Design Model*

Object-Oriented Hypertext Design Model cuja sigla é **OOHDM** [SCH 95b] é uma abordagem baseada em modelos para construção de aplicações hipermídia. Durante o ciclo de desenvolvimento, um conjunto de modelos orientados a objetos é construído. **OOHDM possui quatro etapas** de desenvolvimento: Projeto Conceitual, Projeto de Navegação, Projeto de Interface Abstrata e Implementação [SCH 95b].

A **primeira etapa** consiste de um esquema OO (com novas primitivas específicas, como tipos de dados alternativos para um atributo) do domínio da

aplicação. O Diagrama de Classes resultante captura de forma neutra a semântica do domínio (classes) [SCH 95b].

A **segunda etapa** traduz o esquema conceitual para um modelo navegacional, já se preocupando com o tipo de usuário alvo da aplicação e as tarefas realizadas. Existem dois esquemas no modelo navegacional:

- Classes de Navegação especifica os nodos reais (cada nodo é uma classe de navegação), *links* e estruturas de acesso do sistema; cada nodo é uma visão específica do esquema conceitual, combinando atributos de diferentes classes;
- Contextos de Navegação agrupa informação sobre a forma como os nodos podem ser visualizados pelo usuário. O esquema usa Contextos de Navegação para especificar quais nodos e *links* estão disponíveis e qual tipo de navegação é permitida (*tours* guiados, índices, etc.) [SCH 95b].

Na **terceira etapa**, o Projeto de Interface Abstrata, são definidos o comportamento e a aparência da interface com o usuário do sistema. Essa fase está fora do escopo deste trabalho.

Na **última etapa**, Implementação, os modelos OOHDM devem ser traduzidos manualmente para objetos concretos disponíveis em cada ambiente de implementação (HyperCard, Toolbook, Director, HTML, etc.) [SCH 95b]. Essa tradução manual revela a falta de auxílio da metodologia para o aproveitamento da modelagem no processo de implementação [ISA 95].

2.2.2 Modelo *Relationship Management Methodology*

Relationship Management Methodology cuja sigla é **RMM** [ISA 95] propõe o desenvolvimento hipermídia como um veículo de gerência de relacionamentos entre objetos de informação. A metodologia RMM é mais apropriada para aplicações que apresentam um domínio com estrutura regular, como catálogos de produtos e *front-ends* hipermídia para bancos de dados tradicionais. Para alcançar esse objetivo, RMM usa um modelo de dados, o RMDM (*Relationship Management Data Model*), que oferece uma linguagem para descrição de objetos de informação e mecanismos de navegação. O modelo é uma mistura de primitivas E-R (entidades, atributos e associações), estruturas de navegação (*links*, índices, *tours* guiados e mecanismos de agrupamento) e primitivas do *Relationship Management Diagram (slices)* [ISA 95].

As etapas de modelagem mais importantes para a fase de análise são:

- **Projeto E-R**, é representar o domínio de informação da aplicação através de um diagrama E-R. Esse domínio é implementado através de um banco de dados relacional e é o único meio de armazenamento no sistema [ISA 95];
- **Projeto de Slices** (Entidades) é a modelagem interna das entidades. *Slices* são estruturas especiais de hipertexto usadas para agrupar informação sobre entidades, definindo a apresentação ao usuário. Isso é feito pelo desmembramento da entidade em partes relacionadas (por

exemplo, dados pessoais, publicações, etc.) e organização das partes em uma rede de hipertexto. Apenas *links* podem ser usados entre *slices*, não estruturas complexas (índices ou *tours* guiados). Para cada entidade, é definida uma *slice* principal, para onde é direcionada a navegação para a entidade. O resultado dessa fase é um diagrama “E-R+”, onde cada entidade do E-R é substituída pela sua estrutura de *slices* [ISA 95];

- **Projeto de Navegação** são definidos os caminhos de navegação, como ilustrado na figura. 1. Cada associação E-R no diagrama E-R+ é analisada. Se a associação deve ser acessível por navegação, ela é substituída por uma ou mais estruturas de acesso RMDM (agrupamento, índices e *tours* guiados). Como essas estruturas são responsáveis por navegação entre entidades frequentemente atualizadas, não existem *links* fixos entre instâncias dessas entidades [ISA 95].

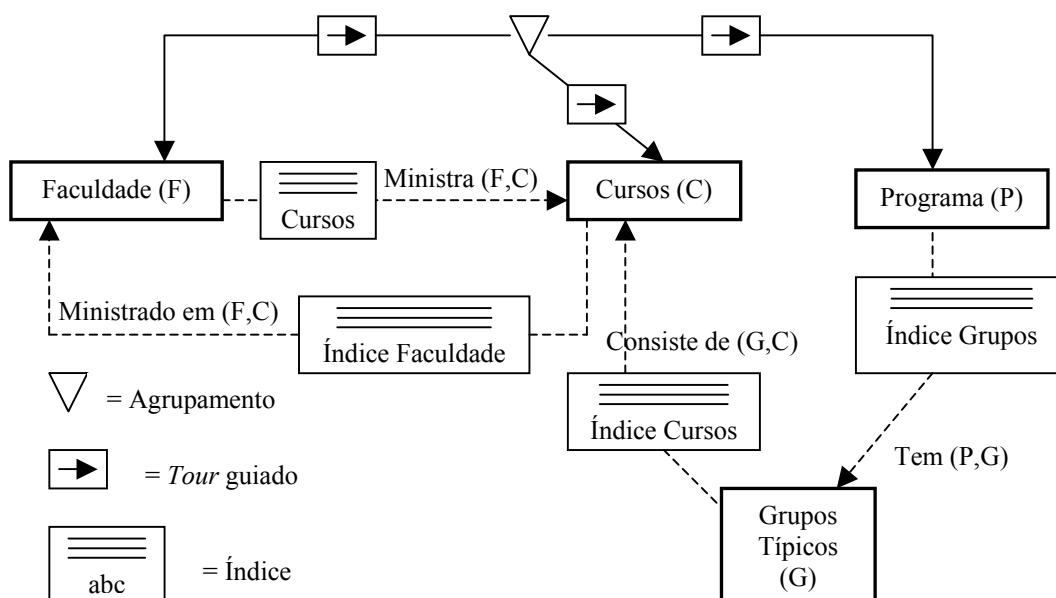


FIGURA 1 - Diagrama RMDM (sem *slices*)

Agrupamentos têm o papel de *menus* do sistema, disponibilizando uma série de *links*. A integração entre nodos e base de dados é feita através das estruturas de acesso: índices representam uma lista de instâncias de uma entidade e *tours* guiados implementam um caminho linear sobre uma coleção de itens. Cada instância é, em tempo de execução, buscada da base de dados e apresentada. As condições qualificando índices e *tours* guiados determinam quais instâncias de uma entidade serão acessadas.

2.2.3 Padrões de Projeto Hipermídia

Com a evolução das técnicas de modelagem e desenvolvimento de sistemas hipermídia, pesquisas sobre padrões de projeto específicos para essa área começam a ganhar importância. As pesquisas nessa área desenvolveram-se principalmente a partir dos trabalhos de Rossi et al. [ROS 97], baseados na metodologia OOHDM. Ambos os esforços seguem na direção de conferir à modelagem hipermídia características de orientação a objetos.

Da mesma forma que padrões tradicionais hipermídia (ver [GAM 95] para uma referência completa) têm, como à documentação e a reutilização de conhecimento e de soluções já consagradas, auxiliando no desenvolvimento do projeto. Como capturam aspectos essenciais de problemas e soluções, padrões de projeto transcendem a escolha de uma ou outra técnica de modelagem [GAM 95], [ROS 97].

Rossi et al. [ROS 97] definem duas categorias de padrões para hipermídia:

Padrões para sistemas hipermídia: usados na construção de ambientes de desenvolvimento de aplicações hipermídia;

Padrões para aplicações hipermídia:

- Padrões de navegação: ajudam na organização da estrutura de navegação da aplicação;
- Padrões de interface: ajudam na organização da informação exibida através da interface de usuário.

Entre vários padrões propostos na área de hipermídia, é merecedor de atenção o padrão *Navigational Context* [ROS 97], [GAR 97c]. Seu objetivo é oferecer ao usuário subespaços de navegação com informação e *links* próprios, melhorando a apresentação e a compreensão dos mesmos.

Isso é alcançado através do conceito de visões dos documentos, que são apresentados de formas diferentes dependendo do contexto em que o usuário se encontra.

Um exemplo é a navegação sobre obras de arte, onde um usuário pode pesquisar obras por assunto, por autor ou por data. Em cada um desses contextos, informações diferentes podem ser exibidas, bem como *links* apropriados para obras relacionadas.

Resumo

Hipertextos são sistemas gerenciadores de informações nos quais estas são armazenadas em uma rede de nós conectados através de ligações. Quando as informações incluem, além de texto, outras mídias como som, imagem, desenho e animação, recebem a denominação de hipermídia.

A principal característica de um documento hipertexto ocorre no momento em que existe uma ligação para um outro documento ou uma outra posição do documento como um objeto remoto (por exemplo: exibir uma determinada página), que são disponibilizados o acesso e a apresentação desse objeto remoto. Existe uma série de aspectos e conceitos usualmente presentes em sistemas de hipertexto. Independente da aplicação à qual o sistema se destina, a informação nele contida estará estruturada em dois elementos básicos: nós e ligações.

Assim como o pensamento é por natureza associativo, o mecanismo de leitura num hipertexto procura representar este relacionamento associativo por meio de nós e de ligações como ilustrado na figura 2. Esse processo de leitura é proporcionado pelo texto convencional, em que o início e fim estão claramente definidos.

A leitura, através de uma série de ligações interconectando partes de um documento, não pressupõe a existência de um fim definido pelo autor, mas sim de um ponto de chegada que o autor definiu como suficiente. Outro conceito importante é o acesso aos dados.

A idéia de navegação, ligada ao movimento que o usuário executa ao se deslocar dentro da estrutura do hipertexto, diferencia-se da atividade convencional de leitura linear.

A leitura em um hipertexto é entendida como um processo descontínuo e não linear em que, através das ligações, o leitor navega de uma informação a outra, não necessariamente numa ordem seqüencial. Logicamente possuem um nome ou título que os identifica, cujo conteúdo poderá ser exibido na tela.

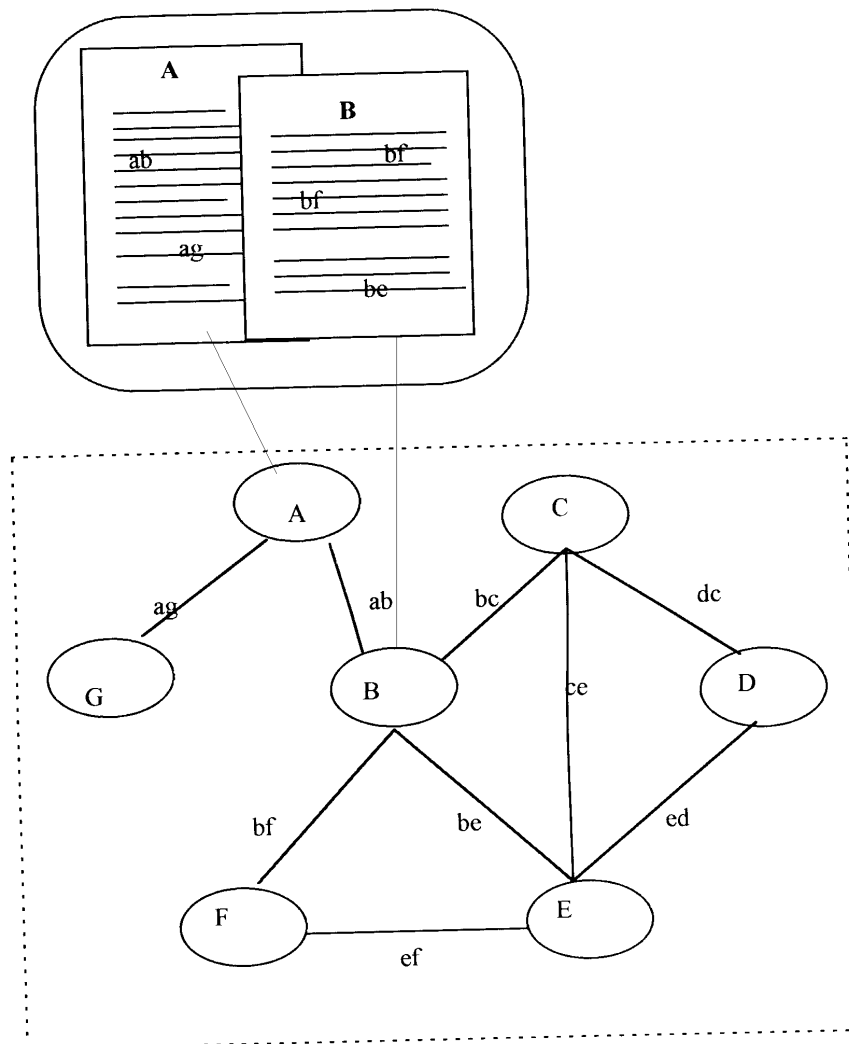


FIGURA 2 - Exemplo de hipertexto

2.3 Ferramentas para Desenvolvimento para WEB

2.3.1 Linguagem HTML

A linguagem HTML define uma estrutura bastante simples para os documentos disponibilizados na WEB. Muitas vantagens podem ser obtidas quando uma estrutura formal mais elaborada é utilizada. É possível, por exemplo:

- **Validar** automaticamente uma estrutura lógica pré-definida;
- **Realizar** buscas restritas a um conjunto de elementos estruturais;
- **Concentrar** no conteúdo dos documentos, preocupando-se inclusive com a formatação e a apresentação visual dos mesmos;
- Prontamente **filtrar**, permutar e reutilizar a informação.

Estas vantagens são diretamente associadas ao fato de o HTML ser um exemplo de linguagem descrita por um padrão bem mais amplo e genérico, o SGML (Standard Generalized Markup Language-ISO 1996), o qual foi proposto para que as propriedades acima fossem obtidas.

A diferença entre um documento em SGML e um em HTML é que o primeiro contém a descrição da sua própria gramática (definida no próprio documento), enquanto o segundo deve obedecer a uma gramática reduzida pré-definida [BOS 97]. Essa simplificação na linguagem facilita a criação de aplicativos, mas impõe importantes restrições [BOS 97], [HOL 98]:

- **Extensibilidade:** não é possível para o usuário especificar novas *tags* em HTML;
- **Estrutura:** não é possível a especificação de estruturas complexas, como as necessárias para a descrição de tabelas em bancos de dados relacionais ou hierarquias orientadas a objetos;
- **Validação:** aplicativos não podem assegurar a validade de um documento em HTML.

Apesar dessas limitações, HTML tornou-se o padrão de linguagem de descrição de documentos suportada por todos os navegadores. Sua simplicidade traz duas **importantes vantagens** que contribuíram para sua grande aceitação: documentos em HTML são relativamente pequenos (**menor tempo de transferência na rede**) e portáteis.

As possibilidades da linguagem HTML que devem ser refletidas em elementos de modelagem incluem:

- **Estruturas comuns em documentos:** títulos, parágrafos, divisões de texto, listas e formatação de texto (tamanho e cor de letra, por exemplo);
- **Inclusão de componentes externos:** figuras, áudio, vídeo, *applets* Java;
- **Formulários gráficos:** campos de texto, listas de opções, *check boxes* e botões;
- **Elementos de ligação entre páginas:** *links*.

Ao longo de sucessivas versões, a linguagem HTML tem evoluído com os objetivos de oferecer maior controle de formatação sobre o texto (novas tags) e maior capacidade de referência aos elementos do texto e suas propriedades (*Document Object*

Model). Essas novas características visam a uma melhor adequação às necessidades de sistemas mais complexos.

O fato de o HTML suportar apenas ligações com um único destino (uma ligação 1:1) pode restringir o processo de estruturação do hiperdocumento. Um autor pode querer associar a um mesmo ponto de partida de uma ligação várias opções de destino para o usuário leitor.

Usando HTML o autor pode utilizar um arquivo intermediário para apresentar a lista de ligações ou inserir a lista de destinos diretamente no arquivo fonte.

No primeiro caso, uma carga extra é exigida da rede e do usuário, que tem que acessar um arquivo apenas para escolher um entre vários itens de uma lista.

No segundo caso, o autor é obrigado a organizar o conteúdo do arquivo fonte de modo a comportar a lista de itens desejada.

2.3.2 A Linguagem de Programação CGI

A primeira forma de extensão a servidores *WEB* foi à programação CGI (*Common Gateway Interface*) [GAR 97b]. Basicamente, a especificação CGI permite que um servidor *WEB* execute programas externos e incorpore os dados de saída desses programas ao arquivo enviado de volta ao navegador. Argumentos são fornecidos ao programa através de variáveis de ambiente e da sua entrada padrão (stdin) e os resultados da execução são retornados através da saída (stdout). Com esse mecanismo, o servidor *WEB* tem sua funcionalidade estendida [DWI 96], [GAR 97b]. A figura 3 abaixo ilustra o funcionamento de um programa CGI [TIT 96].

O relacionamento entre uma página HTML e um programa CGI ocorre através de *links* ou, mais normalmente, através de formulários HTML, que contêm elementos gráficos de interface. Um desses elementos é o botão especial “SUBMIT”, que envia o conteúdo dos campos do formulário para o programa CGI especificado na definição do formulário. Essa passagem de argumentos pode ser realizada através de dois métodos: GET ou POST. O método GET concatena os argumentos na própria *string* URL (Uniform Resource Locator) requisitada ao servidor. Já o método POST cria um objeto (um arquivo, por exemplo) que é passado ao programa [DWI 96], [TIT 96].

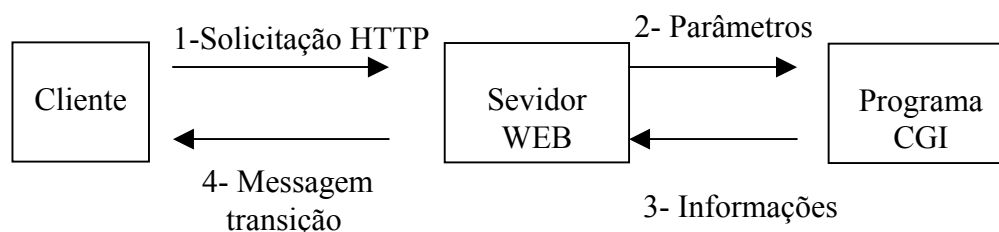


FIGURA 3 - Funcionamento de um programa CGI

Atualmente, a arquitetura CGI é insuficiente para a implementação de complexos sistemas cliente-servidor. Até mesmo novas extensões, como a “Microsoft

Internet Services API” (ISAPI) e a “Netscape Server API” (NSAPI), não atendem aos requisitos dos sistemas atuais. O principal problema dessas tecnologias é o uso do protocolo HTTP para todas as comunicações cliente/servidor: formulários HTML continuam a unidade básica de interação. Essa abordagem não é apropriada para aplicações que requeiram intensa comunicação entre componentes, além de não ser escalonável [ORF 97].

O que é importante na arquitetura CGI em termos de metodologias de modelagem pode ser resumido em:

- É um componente executável independente;
- Está localizado e executa no servidor;
- Pode ser referenciado por um *link* ou um formulário em uma página HTML;
- Recebe parâmetros;
- Pode acessar recursos no servidor ou em outros computadores ligados ao servidor;
- Retorna resultados na forma de um arquivo HTML gerado dinamicamente.

2.3.3 A Linguagem de Programação Java

A história da linguagem Java começou em 1990 nos laboratórios da Sun Microsystems, onde se realizavam pesquisas sobre os componentes eletrônicos de aparelhos domésticos. O objetivo era uniformizar a interface de programação dos eletrodomésticos, permitindo a comunicação entre eles e com computadores. O primeiro produto em mente seria um controle remoto universal, capaz de controlar todos os aparelhos de uma casa. Apesar do fracasso do projeto, um de seus produtos vingou: a linguagem de programação desenvolvida, em 1991, batizada de Oak [NEW 96].

Hoje, Java é uma linguagem especializada para aplicações em Internet e suportada pelos navegadores mais populares. Seu sucesso também se deve à evolução constante de suas bibliotecas de programação, que, a cada nova versão, incluem importantes recursos de desenvolvimento. Novos produtos, como servidores *WEB* especializados e ferramentas de programação Java no lado do servidor, também contribuem para a ampla aceitação da linguagem [MOM 97], [NEW 96]. As três sessões seguintes explicam um pouco das características das ferramentas de desenvolvimento relacionadas à linguagem Java.

Java surgiu da necessidade de criação de uma nova linguagem de programação adequada à utilização em pequenos aparelhos eletrônicos. Baseada em C++, Java foi projetada com o objetivo de ser, antes de tudo, segura, pequena e portátil. As conhecidas características de Java podem ser resumidas em [MOM 97], [NEW 96]:

Simplicidade: Java é uma adaptação de C++ sem características como ponteiros e manipulação de memória; tipagem forte também colabora para a simplicidade de compiladores e interpretadores;

Orientação a objetos: Java é puramente orientada a objetos e oferece uma grande variedade de bibliotecas; algumas características de linguagens OO, como herança múltipla, foram abolidas;

Segurança: existem quatro níveis de segurança em Java: segurança na sintaxe da linguagem, segurança na verificação do *bytecode* pré-compilado; restrições aplicadas durante a execução de *applets* e assinatura digital de *applets*;

Portabilidade: como Java é interpretada, ela pode ser executada em qualquer máquina com um interpretador Java; atualmente, os navegadores mais populares suportam a linguagem, e ambientes de execução estão disponíveis para uma grande variedade de plataformas.

O maior interesse despertado por Java foi a possibilidade de utilização de uma verdadeira e completa linguagem de programação, projetada especialmente para a *WEB*, com a capacidade de transferência pela rede e execução na máquina cliente.

Novas possibilidades foram abertas em termos de interfaces gráficas mais interativas, acesso a recursos remotos e transferência de processamento de servidores para clientes, além de uma variedade de aplicações distribuídas portáteis com suporte para programas já existentes [MOR 97].

A linguagem Java foi o primeiro passo em direção da “*Object WEB*”, a *WEB* com suporte a aplicações baseadas em objetos distribuídos [ORF 97].

A linguagem de programação Java torna possível aplicação de hiperdocumentos. Java é também uma plataforma de linguagem independente que assegura todas as aplicações incluídas que possam carregar em qualquer sistema na Internet. Como uma linguagem de programação orientada a objeto, Java permite o reuso de código fonte. Essa característica é muito importante para o curso, simulação, prova e ferramentas de avaliação a um documento, controlado por uma aplicação Java. A construção de um sistema **modelado em autômatos escrito em Java**, possibilita aos estudantes testar e aprender a funcionalidade do sistema, criando seu próprio curso e exercícios para simular um teste de avaliação.

Para efeitos de modelagem, uma *applet* é um componente de código que executa no cliente. A sua transferência do servidor para o cliente se dá através de uma referência à classe da *applet* no documento HTML no qual ela será incluída [LIM 99].

Portanto, a *applet* pode ser relacionada a um documento como um componente, via agregação. A *applet* em si é uma classe da linguagem Java, portanto, é facilmente representada por um estereótipo baseado em Classe.

Na maioria das situações, a *applet* se relaciona com outras classes, tanto as incluídas nas bibliotecas da linguagem quanto às definidas pelo usuário. Em termos de modelagem, esse relacionamento pode ser encarado como semelhante à estrutura do padrão *Facade* [GAM 95], onde uma classe serve de intermediário (fachada) para os métodos fornecidos por outras classes (invisíveis ao cliente do intermediário).

Baseando-se nesse padrão, a *applet* representa em sua classe os métodos implementados por ela e também por suas colaboradoras, e esses métodos podem ser acessados pelo usuário da aplicação ou por outros componentes de software. A capacidade da *applet* de se comunicar com outros componentes também deve ser refletida na modelagem, como, por exemplo, através da possibilidade da definição de

relacionamentos entre *applets* e outras classes e, também, de acessos a recursos remotos [LIM 99].

2.4 Mecanismos para Exercícios

Mecanismo de resolução de exercícios para EAD são processos que incluem a manipulação de textos, gráficos, processos administrativos, acompanhamento do desenvolvimento do aluno, exercícios, testes de avaliações, trabalhos extraclasse, enfim tudo que é necessário em um ambiente de ensino. Esses processos são as funcionalidades de um sistema para a construção de material instrucional.

Existem vários mecanismos com componentes que podem facilitar ao professor e ao aluno a construção e resolução dos exercícios:

- Tipo de exercícios;
- Paleta de cores;
- Tipos de fontes;
- Ferramenta de desenho;
- Monitoramento;
- Bloco de rascunho;
- Importar figuras;
- Importar símbolos;
- Importar arquivos;
- Baralhar alternativas;
- Serviço de dicas;
- Ligações com Cursos;
- Geração automática ou semi-automática;
- Outros.

Durante o processo de pesquisa, realizamos testes em alguns aplicativos em busca de bons mecanismos importantes para o processo de aprendizagem. Aplicamos teste real do protótipo do sistema Hyper-Automaton em desenvolvimento, em que o professor pode gerar seus exercícios e aplicar provas on-line [MOC 99].

Nesse teste, o professor e os alunos acrescentaram algumas melhorias no protótipo do sistema Hyper-Automaton, em que identificamos itens importantes para a nossa pesquisa de um bom mecanismo de resolução de exercícios e exames.

Para que possamos atender as necessidades do professor e dos alunos, foram destacadas as seguintes sugestões:

1) Exercícios:

- a) Questões objetivas
 - Tipo de questões objetivas: múltipla escolha, escolha simples, verdadeiro ou falso, associativos, pintar, sublinhar e outros.
- b) Questões descritivas
 - Tipo de questões descritivas: respostas curtas, palavras cruzadas e redação.
- c) Montagem das Questões
 - Exercício (questão)

Em caso de uma questão, inserção de conteúdo, figuras, nível de raciocínio, dificuldades, atribuir peso, dicas com referência ao material do curso.

- Lista (questões)

Em caso Lista, forma de exibição (uma por página, ou todos por página), embaralhar questões ou itens e exibir exercícios aleatórios.

2) Tempo

É um dos fatores importante para resolução e elaboração de uma lista de exercícios, onde se mede o tempo de execução, exercícios, acesso a dicas e tempo de encerramento.

3) Resposta Rápida (Automática)

Preparação de listas de exercícios com retorno das respostas automáticas para o aluno acompanhar seu rendimento.

4) Material de apoio ao aluno

Bloco de rascunho, ferramenta de desenho para colorir fontes, deixar em destaque (circular e sublinhar).

5) Modelo de exercícios

- a) Tradicional e exercícios seqüenciais;
- b) Exercícios adaptáveis conforme seu grau de entendimento geram uma seqüência diferente.

6) Tipo de Geração de Exercício ou Prova

- a) Semi-Automático;
- b) Automático.

7) Criação de um banco de itens

- a) Possibilita a indexar um grande volume de questões, do semestre atual e dos semestres anteriores;
- b) Rotulados para facilitar no mecanismo de busca.

8) Monitoramento

Durante a aplicação de uma Lista/Exercícios e provas, torna-se necessário o acompanhamento para analisar o perfil do aluno.

9) Segurança dos dados criptografados.

Com os dados acima relatados, busca-se na pesquisa de campo comparar com as necessidades do professor e dos alunos em busca de uma ferramenta que o atenda totalmente ou parcialmente. Tais ferramentas serão descritas a seguir.

2.5 Ferramentas para Exercícios

Com o recurso do mecanismo de busca na internet, participação de seminários e eventos para ensino a distância, encontraram-se vários software com finalidade exclusiva para a construção de material instrucional tendo como base principal os mecanismos para construção e resolução de exercícios e provas mediada por computadores, mecanismos administrativos para monitoramento do perfil do aluno e outros. Nas ferramentas analisadas, verificou-se o componente para construção e resolução de exercícios.

2.5.1 Question Mark Designer

Question Mark Designer é um software desenvolvido pela Question Mark Computing Ltd., estando disponível, atualmente, para as plataformas Windows 3.11, 95,

98 e NT e System 7 da Macintosh. Para esse fim, foi desenvolvida a Question Markup Language [QUE99], uma linguagem de marcação de hipertexto proprietária, que permite exportar para a Internet e Intranet, os exercícios aplicados aos cursos de treinamento interativos.

Através de um mecanismo inteligente de percepção o Question Mark Designer consegue administrar automaticamente o nível de questionamento do aluno, utilizando como modelo Kirkpatrick [KIR99].

Estrutura do Software

O modelo conceitual do QuestionMark está estruturado em arquivos de bibliotecas, bloco de questão e arquivo de perguntas. Os arquivos de biblioteca são constituídos pelos blocos de questões. Os arquivos de testes contêm perguntas criadas por um autor, sendo que o arquivo de tese às vezes, é conhecido como bancos de artigo, e o arquivo de perguntas como parte de um teste.

O arquivo de perguntas não só cria perguntas, como questiona a informação relacionada que será apresentada a um usuário, mas também quais as bibliotecas de perguntas que são incluídas em um teste.

Testes construídos no Question Mark são freqüentemente integrados de um ou mais arquivos de perguntas ou arquivo de biblioteca referenciada (bibliotecas de perguntas).

Demonstração da construção de um arquivo de perguntas vistos na figura 4, em que as questões do arquivo de biblioteca são como um “bloco de edifício” integrante na construção do arquivo de perguntas.

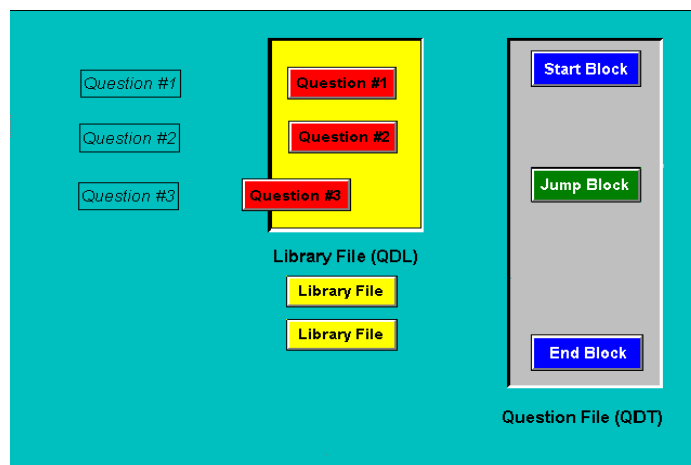


FIGURA 4 - QML Construção de um arquivo de perguntas

Na seqüência da construção, as perguntas selecionadas do arquivo de biblioteca são inseridas em um arquivo de perguntas figura 5.

Existem quatro blocos importantes na construção de um arquivo de perguntas:

- StartBlock e EndBlock apresentam informações que controla um teste no começo e no fim entre os blocos de um arquivo de perguntas;
- Jump Block, determina quais perguntas que serão apresentados aos usuários;
- Library Block, os blocos de bibliotecas, são perguntas do arquivo de biblioteca.

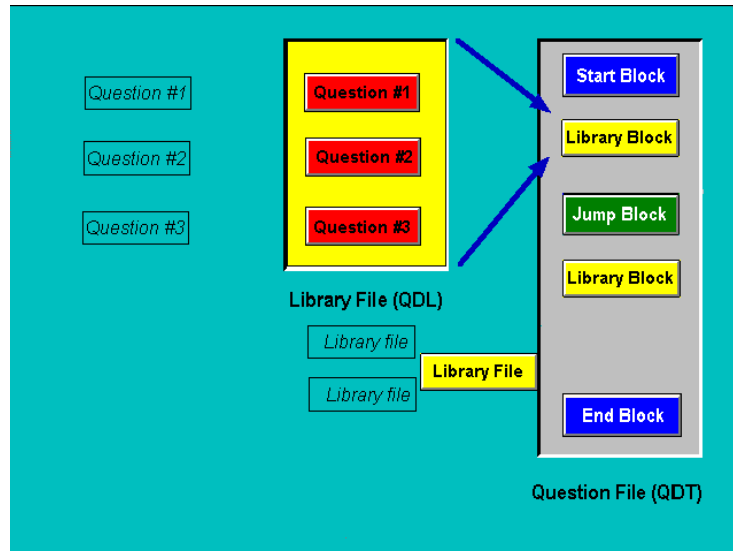


FIGURA 5 - QML Bloco de Perguntas.

Tipos de Perguntas (Questões)

Question Mark possui oito tipos de diferentes perguntas, visto na figura 6, como característica de elaborar provas dinâmicas, interativas e adaptáveis, motivacional que aprende com o ambiente, avaliação do conhecimento e opiniões. Usa técnica de multimídia em seus testes ou pesquisas e técnica de segurança em teste que tenha randomização e muitos outros tipos de perguntas.

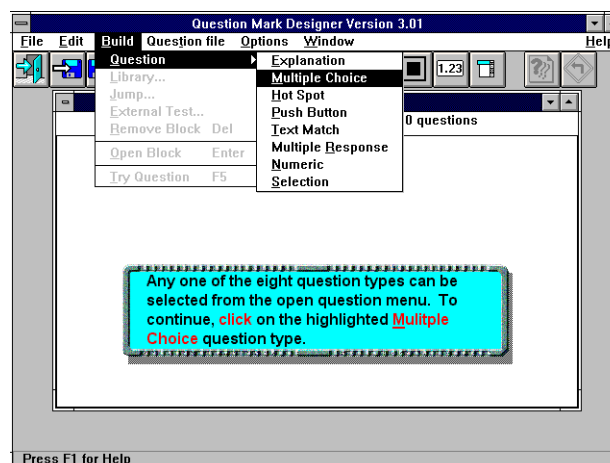


FIGURA 6 - QML Tipos de Perguntas

- 1) Explicação - tipo de perguntas descritivas.
- 2) Múltipla Escolha - tipo de pergunta de múltipla escolha, que permite escolher de duas ou mais possíveis respostas. É possível utilizar gráficos,

fontes misturadas, sobrescrito e subscições em perguntas de múltipla escolha (figura 7).

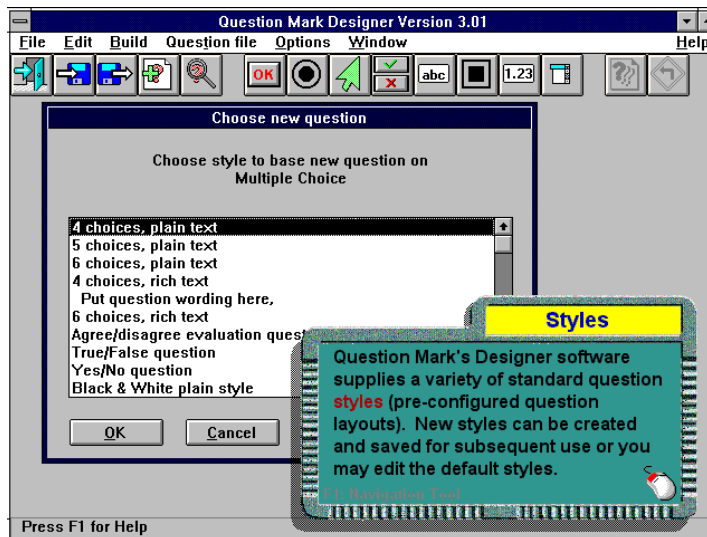


FIGURA 7 - QML Estilo Múltipla-Escolha

- 3) Múltipla Resposta: o tipo de pergunta de múltipla resposta, deixa escolher mais de uma resposta correta para a pergunta.
- 4) Botão de Arrastar: oferecem uma escolha entre duas ou mais respostas especialmente boas para menu.
- 5) Mancha quente: pode usar qualquer gráfico e qualquer símbolo ou marcador para mover perguntas típicas que poderiam mostrar um mapa, fotografia ou diagrama e uma peça para o usuário que localize algo nisto. Também é possível utilizar neste tipo de pergunta um botão de OK.
- 6) Texto partido é o tipo de pergunta que permite ao usuário digitar uma palavra ou texto mais longo como uma resposta. Define também quando se deve digitar, e qual a resposta certa é. Também pode usar este tipo de pergunta para fazer o usuário preencher espaços em branco em texto que provê.
- 7) Numérica: no tipo de perguntas numéricas, as respostas poderão ser digitadas em números. Define a resposta correta, ou um grupo de respostas corretas.
- 8) Seleção, tipo de seleção de pergunta, por exemplo, deixa nomear uma série de escolhas ou declarações e pergunta para o usuário se elas foram verdadeiras ou falsas.

Na tabela 1 apresenta um resumo das características usuais dos tipos de exercícios do QML.

TABELA 1 - QML Tipos de exercício.

| Tipo de Exercício | Característica |
|--------------------------|--|
| Múltipla Escolha | Uma alternativa entre várias, incluindo: <ul style="list-style-type: none"> • Múltipla escolha clássica; • Verdadeiro e falso; • Sim e não; • Escala de concordância (concordo plenamente, discordo plenamente). |
| Múltipla Resposta | Mais de uma resposta entre várias. |
| Numérica | Resposta é um número. |
| Texto | Resposta através de um texto com: <ul style="list-style-type: none"> • Apenas uma palavra; • Várias palavras; • Ensaio; • Parágrafo; • Texto livre. |
| Seleção | Seleções em uma lista tipo: <ul style="list-style-type: none"> • Hot Spot: A alternativa é selecionada considerando sua posição, utilizando um dispositivo apontador (<i>mouse</i>) ou uma tela sensível ao toque (<i>touch screen</i>); • Matrix: Permite seleção em uma lista matricial de alternativas em uma página; • Explanation: Explicar situações. |

Biblioteca de Questões

Na estrutura do módulo de bibliotecas existem três opções:

- Editor básico de perguntas: Fundos e Gráficos;
- Propriedades de componente: Escore, Avaliação, e Multimídia;
- Prova e Simulações.

Editor de QuestãoO editor de questão provê bordas opcionais para realçar os componentes individuais que, junto, forma uma única pergunta, figura 8.

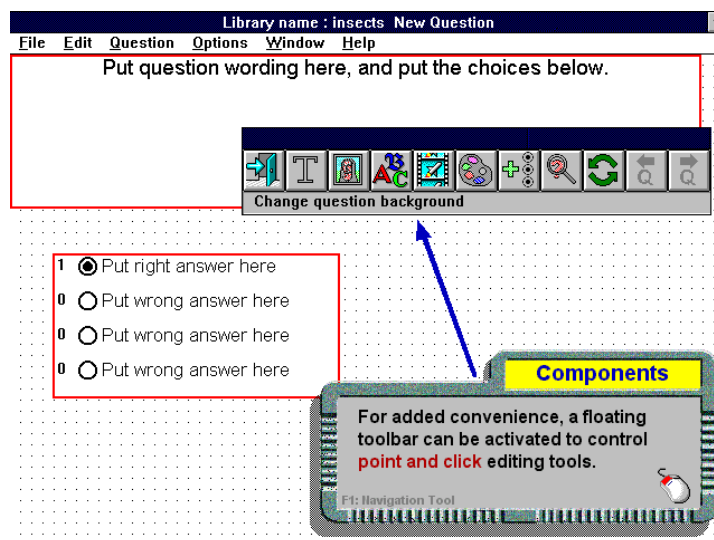


FIGURA 8 - QML Editor de Questões

O limite de componentes por pergunta pode conter até 50 componentes separados. Há 4 componentes gerais que podem ser usados em todo tipo de pergunta.

- texto padrão;
- componente gráfico;
- componente de texto;
- componente de multimídia;

Usa-se, nas perguntas, botões interativos, que permitem ao participante ver e/ou ouvir som, vídeo, animação, além de executar outro programa DOS/Windows, todos componentes têm várias características ou propriedades que podem ser modificadas.

2.5.2 Hot Potatoes

O Hot Potatoes é um programa interativo desenvolvido pela Universidade de Victoria (Martin Holmes, Stewart Arneil) para criação de exercícios para a WEB, disponível em plataformas IBM-PC Windows 95 e Macintosh System 7. É um programa freeware Win95 que é interativo com WWW.

O Hot Potatoes possui seis tipos de ferramentas, figura 9: [HOT99].

- JBC: exercícios de múltipla-escolha;
- JCloze: encha os espaços das aberturas de exercícios;
- JCross: palavras cruzadas;
- JMatch: emparelhando / ordenando exercícios;
- JQuiz: perguntas e resposta curtas;
- JMix: orações confundidas.



FIGURA 9 - HOT Tela Principal

- 1) O Hot Potatoes possui as seguintes características:
 - interfaces simples, consistentes em Windows e Macintosh;
 - permitem múltiplas respostas corretas;
 - possuem todos os tipos de fontes de caracteres;
 - suporte para hierarquia de usuários (administrador, usuário comum).
- 2) Ambiente:
 - múltiplas plataformas;
 - browsers Netscape 3 & 4 e MS IE 3 & 4;
 - requer nenhuma configuração de servidor especial ou software;
 - são fáceis de integrar em rede local e WEB.
- 3) Possibilidades de integração:
 - anexar quadros (frames) e ligações (links);
 - modificar o código fonte existente das páginas (HTML);
 - anexar seu próprio código HTML;
 - ajuda para usuarios.
- 4) No ambiente local de WEB:
 - Download dos programas;
 - tutoriais para Macintosh e Windows;
 - perguntas e comentários para o quadro de anúncios;
 - suplementares e atualizações;
 - FAQs e relatórios de bug.

A nova versão beta Hot Potatoes 4.0 inclui um formato de produção nova, e também opção para exportar DHTML. A versão 4.0 também unificará dados dos arquivos com formatos em XML, de forma que o Macintosh e diferentes versões do Windows possam, finalmente, ler dados de outros arquivos, e outras aplicações também possam ler dados do Hot Potatoes. Enquanto usando XML e XSL, serão habilitados a interatividade com sofisticadas ferramentas para browsers.

2.5.3 Pathware 4

Pathware é o processo de aprendizagem on-line com módulos distintos que enviam o processo de aprendizagem on-line inteiro. Isso habilita acesso feito sob encomenda ao sistema baseado em permissão de usuário. Se os usuários não tiverem permissão para usar um módulo cliente, não estará disponível o acesso.

O Pathware é um administrador de sistema que controla acesso aos módulos e operações de sistema. Os módulos são projetados com quatro usuários distintos em mente: estudantes, instrutores, administradores, e os autores dos conteúdos.

TABELA 2 - PTW Módulos Pathware

| | | |
|------------|---|-------------------------------------|
| Home | Colocações de sistema, contra-senhas de mudança. | Pathware o administrador de sistema |
| Tarefas | Cursos leves de treinamento On-lines. | Estudantes |
| Planejador | Construção e sucessão currículos on-lines. | Autores |
| Perfis | Administra o perfil do curso, o perfil do usuário, e permissões de sistema. | Pathware o administrador de sistema |
| Usuários | Criar e modificar os usuários. | Pathware o administrador de sistema |
| Matrículas | Administra matrículas de estudante. | Instrutores |
| Resultados | Manutenção dos registros dos resultados dos estudantes. | Instrutores |
| Repórter | Cria e produz relatórios de sistemas. | Todos |
| Mensagem | Envia mensagens de e-mail. | Os instrutores e administradores |

A arquitetura de desenvolvimento de Scalable de Pathware

Arquitetura Pathware figura 10 que obtém os conteúdos de acesso desenvolvido com o Authorware 5, Dreamweaver e outras ferramentas de autoria, retornam para os usuários como eles progridem em treinar os conteúdos, onde são armazenados os resultados do treinando, inclusive estados de conclusão do curso, contagens, habilidades, e respostas do estudante no banco de dados.

Com suas três bancadas, a arquitetura Pathware pode ser configurada em um único servidor ou em múltiplos servidores para qualquer nível de desempenho desejado.

Aplicação Pathware possui um banco de dados de empreendimento que instalam no mesmo servidor para uma aplicação departamental ou em servidores diferentes para grandes desenvolvimentos de empreendimentos.

NETWORK ARCHITECTURE

Content developer and subject matter experts

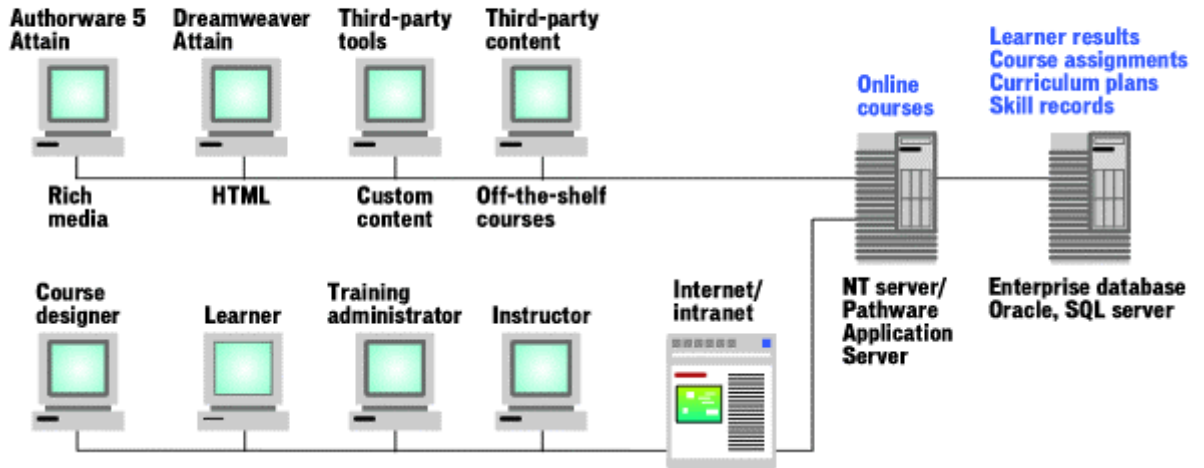


FIGURA 10 - PTW Arquitetura Pathware

Módulo Home

Os administradores de sistemas usam o módulo Home para personalizar o sistema de Pathware, mostrado na figura 11. Para criar uma comunidade de campus online, usa gráficos, logotipos e páginas de HTML. Também podem ser configurados outros parâmetros como terminologia de sistema e permissões de acesso com o módulo Home.

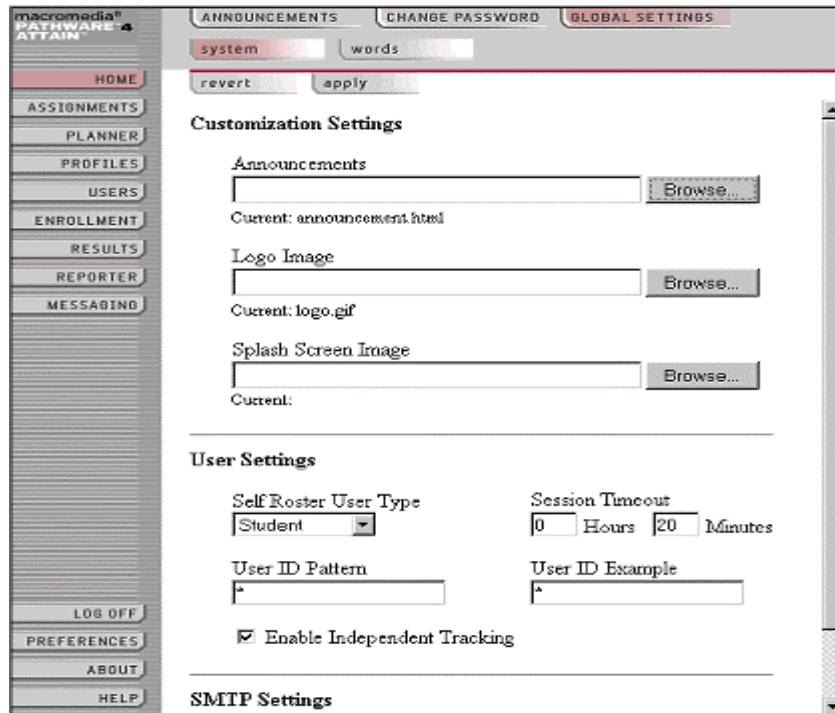


FIGURA 11 - PTW Módulo Home

Módulo de tarefa usuário designado: Estudantes

O módulo de tarefa provê um único ponto de acesso para estudantes assinarem Pathware, tarefas de treinamento de cursos, onde terão que se matricular nos mesmos. Resultados do treinando, inclusive com pontuação, estado de conclusão, cronometragem, respostas dos estudantes são capturadas, automaticamente, pelo servidor de Pathware para recuperação posterior.

Pathware é compatível com uma gama extensa de ferramentas de desenvolvimento de conteúdos e idiomas, inclusive Dreamweaver Attain, Authorware Attain, como também editores de HTML standards e Java.

Pathware pode lançar materiais de treinamento desenvolvidos com outras ferramentas de mesa, inclusive PowerPoint e MS Word. Pathware também pode administrar conteúdo comercialmente disponível de principais vendedores, inclusive NETg, Teach.com, Catapulta (uma divisão de IBM), e Sistemas de Tarragon (uma divisão de Sistemas de CBT). Estudantes usam o módulo de tarefas para acessar os cursos on-line, como mostra a Figura 12.

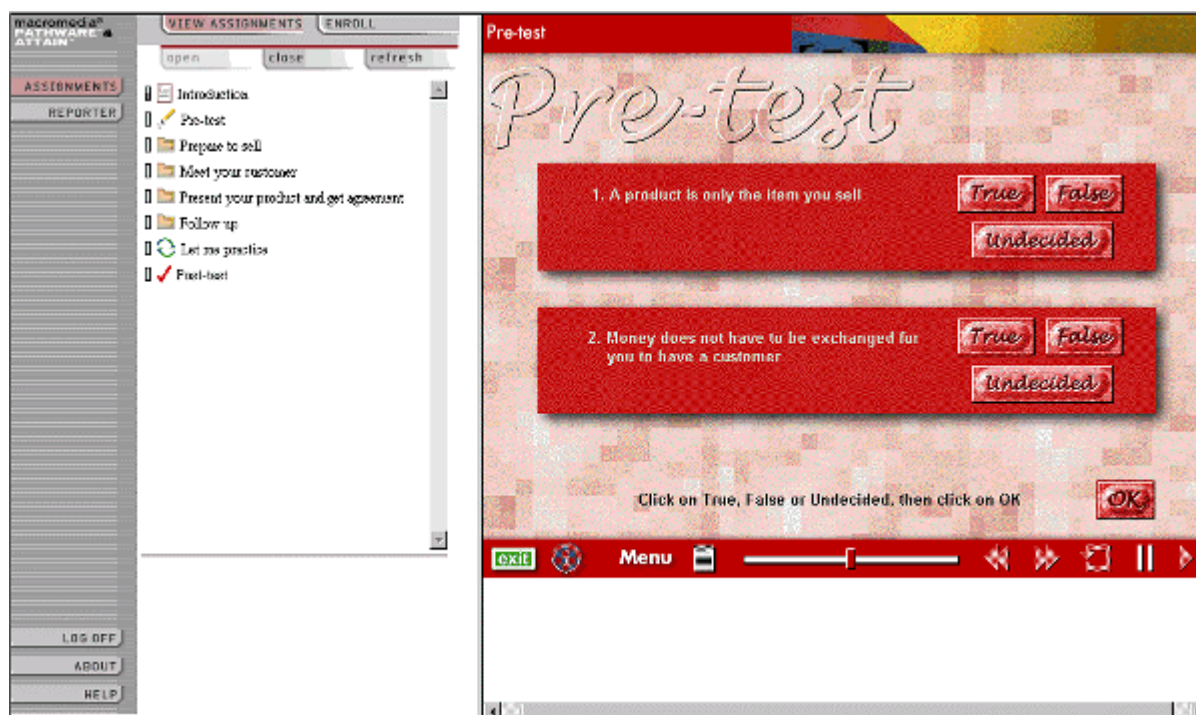


FIGURA 12 - PTW Módulo Tarefa.

Módulo de resultados usuário primário: Instrutor

Pathware trabalha com conteúdo de modo off-line, possibilitando aos estudantes e administradores a capacidade de atualizar o estado de conclusão do curso e a pontuação (contagens) manualmente para vídeo, instrução da sala de aula, leituras e exercícios de laboratórios.

Os instrutores e os gerentes usam o módulo de resultados para verificar os mesmos nos cursos de modo off-line bem como, editá-los de cursos on-line. Os

instrutores utilizam este módulo visto na figura 13 para editar os registros e entrar na informação da conclusão do curso em off-line.

| Last Name | First Name | Logon Name | Score | Completion Date | Duration | Progress |
|-----------|------------|------------|-------|-----------------|----------|------------|
| Adrian | Adrian | Adrian | 70 | March 29, 1999 | 27 | Incomplete |
| Howard | Chris | choward | 0 | | 119 | Incomplete |
| Howard | Chris | choward | 0 | March 29, 1999 | 119 | Incomplete |
| Andrews | Fai | FAndrews | 80 | March 29, 1999 | 27 | Incomplete |
| Eabb | Buffy | EEabb | 67 | March 29, 1999 | 28 | Incomplete |
| Eabcock | Will | WEabcock | 89 | March 29, 1999 | 34 | Incomplete |

FIGURA 13 - PTW Módulo de Resultados

Módulo de relatório usuário primário: Todos

Com o módulo relatório, o usuário pode ver e criar relatórios on-line a partir do seu browser. Pathware 4 apresenta quinze relatórios standards que provêm dados completos, matrículas, cursos, e progresso de estudante. Os administradores podem projetar e personalizar disponibilizando relatórios. A ferramenta de relatório é a mais popular para bancos de dados de empreendimento. Utilizando outras modificações de segurança, você pode administrar acesso de relatórios individuais no módulo de Perfis. O relatório mostrado, na figura 14, mede a efetividade do curso mostrando latência, cronometragem e porcentagem correta para cada interação.

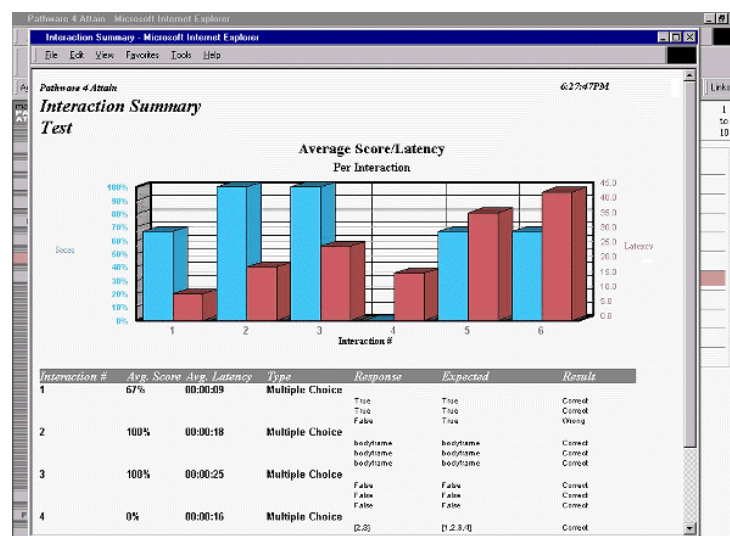


FIGURA 14 - PTW Relatório mede a efetividade do curso

Este relatório demonstrado na Figura 15 mostra o progresso do estudante em uma avaliação on-line.

Pathware 4 Attain Wednesday, March 24, 1999, 6:35:41PM

User Interaction Detail for Chris Howard
Activity: Test

| Interaction # | Last Accessed | Type | Response | Expected | Result | Weight | Latency |
|---------------|-------------------|-----------------|----------------|----------------|---------|--------|----------|
| 1 | 11/3/00 8:04:17PM | Multiple Choice | True | True | Correct | 10 | 00:00:11 |
| 1 | 3/24/00 5:18:30PM | Multiple Choice | True | True | Correct | 10 | 00:00:11 |
| 2 | 11/3/00 8:04:31PM | Multiple Choice | bodyframe | bodyframe | Correct | 10 | 00:00:28 |
| 2 | 3/24/00 5:18:44PM | Multiple Choice | bodyframe | bodyframe | Correct | 10 | 00:00:16 |
| 3 | 11/3/00 8:04:41PM | Multiple Choice | False | False | Correct | 10 | 00:00:30 |
| 3 | 3/24/00 5:18:51PM | Multiple Choice | False | False | Correct | 10 | 00:00:27 |
| 5 | 11/3/00 8:04:55PM | Multiple Choice | 111in | 111in | Correct | 10 | 00:00:50 |
| 5 | 3/24/00 5:19:05PM | Multiple Choice | 111in | 111in | Correct | 10 | 00:00:40 |
| 6 | 11/3/00 8:05:03PM | Matching | 1.1a.2.2a.3.3a | 1.1a.2.2a.3.3a | Correct | 10 | 00:00:58 |
| 6 | 3/24/00 5:19:15PM | Matching | 1.1a.2.2a.3.3a | 1.1a.2.2a.3.3a | Correct | 10 | 00:00:48 |

PREFERENCES

- Profile Reader Summary: Shows all users who belong to a user profile.
- Single Profile Progress: Shows progress for a single course and user profile.
- Single User Progress: Shows progress for a single user in multiple courses.
- User Interaction Detail: Shows a user's results for all interactions in an activity.

FIGURA 15 - PTW Relatório Avaliação on-line.

2.5.4 Outros Sistemas

Entre outros sistemas disponíveis como ferramentas para elaboração de exercícios e provas, destacam-se os seguintes:

- **TextToys:** desenvolve exercícios no formato de texto *on-line*. Utiliza o formato de exercícios Rhubarb e Sequitur desenvolvido originalmente por John e Muriel Higgins. Implementa páginas para a WEB, isto significa que o estudante pode ter acesso aos exercícios utilizando qualquer browser capaz de executar Javascript, e também pode ter acesso aos exercícios de CD-ROM, disco, ou através de uma intranet ou da Internet;
- **Markin32:** é um programa que o estudante marca os textos submetidos por e-mail ou um documento em MSWord. O programa foi desenvolvido como um suprimento de sistema de cursos pela internet, e pode produzir trabalho no formato de RTF ou documento texto ou como arquivos de HTML que pode ser visto usando um browser de WEB;
- **Clozemaker:** programa que permite o professor projetar na tela, exercícios de palavras cruzadas para os estudantes;
- **Choices:** programa com questões de múltipla escolha que são criadas em modo professor e completadas pelo aluno, em modo estudante;
- **Multi-Replace:** é um programa de freeware que procura-e-substitui operações em muitos arquivos de texto ao mesmo tempo. É uma ferramenta extremamente útil para qualquer administrador de WEBSites de grande ou médio tamanho. O programa auxilia, automaticamente, em todos seus arquivos para um diretório que é especificado antes de fazer qualquer mudança, e restabelece os originais qualquer hora;
- **JBC:** é um programa freeware que compila questões de múltipla-escolha em formato HTML, usando Javascript. Os estudantes podem completar estas questões sem ter o programa de JBC instalado no PC deles. O JBC possui uma série de páginas HTML auto-explicativas para orientar como compilar as

- questões, que pode ser usado dentro de qualquer browser da WEB que tenha suporte Javascript;
- **Jquiz:** é um programa de freeware que permite os professores criarem questões interativas para uso no World Wide WEB ou intranet. O Jquiz usa Javascript, portanto o usuário não precisa ter qualquer conhecimento de HTML ou Javascript, o gerador de questões faz toda a codificação automaticamente;
 - **Minitools:** é um site freeware de três programas que permitem aos professores a criação de uma variedade de idiomas interativos para serem exercitados no World Wide WEB, ou em uma escola local ou intranet.
 - **WEBble:** é um utilitário de conversão para texto de página da WEB. É especialmente útil para assegurar que os caracteres em suas páginas de WEB sempre apareçam como foi digitado, respectivos sistemas operacionais ou browser são usados para ver as páginas pretendidas;
 - **Annotation Templates for MSWord and WordPerfect:** este modelo é freeware e contém barra de botão feito sob encomenda que podem ser usados para anotar texto dentro de um documento do processador de texto MSWord, ideal para professores que marcam composições submetidos como documentos de Word Perfect, ou para esses que precisam ler prova ou anotar documento;
 - **The Evil Landlady Action Maze** é uma ilustração de páginas interativas, baseado em páginas da World Wide WEB que o estudante pode seguir usando um browser.

2.6 Hyper-Automaton

Hyper-Automaton é um sistema interativo de aprendizagem local ou remoto com módulos distintos e integrados, desenvolvido pelo grupo de pesquisa da Universidade Federal do Rio Grande do Sul - UFRGS para criação de cursos, exercícios e avaliações, disponível para todas as plataformas WEB.

O sistema apresenta uma forma de modelagem de cursos, exercícios e avaliações na WEB, utilizando Autômatos Finitos com Saída. [MEN 98]. Esta técnica esta direcionada para suporte de cursos, exercícios e avaliações em rede local e remota. O Hyper-Automaton é um sistema semi-automatizado, onde foi aplicado e testado para o ensino de Informática Teórica no Instituto de Informática da UFRGS.

Nesse sistema, tanto o curso, os exercícios e as avaliações, podem ser apresentados sob dois enfoques:

- Máquina de Moore (saídas associadas aos estados);
- Máquina de Mealy (autômato modificado de forma gerar uma palavra de saída para cada transição).

Portanto, cada página de exercício/conteúdo do curso corresponde a um estado(Moore) ou a uma transição(Mealy). A vantagem é que a estrutura dos autômatos com saída permitem a criação do material hipermídia de forma independente do autômato em si, possibilitando a programação de seqüências de estudo com objetivos e enfoques específicos, reuso de exercícios ou conteúdo do curso de forma completa ou parcial das páginas WEB em diversos cursos/exercícios (autômatos), eliminado a redundância de informações e criações de páginas, ou seja, modularizando o sistema que facilite a expansibilidade, onde a base do hiperdocumento é projetada de forma independente da estrutura de controle da aplicação hipermídia.

O sistema utiliza uma arquitetura cliente-servidor para executar na WWW, como pode ser visto na figura 16, o servidor funciona como um intermediário entre o cliente (usuário) e o servidor (HTTP) que provê o material instrucional.

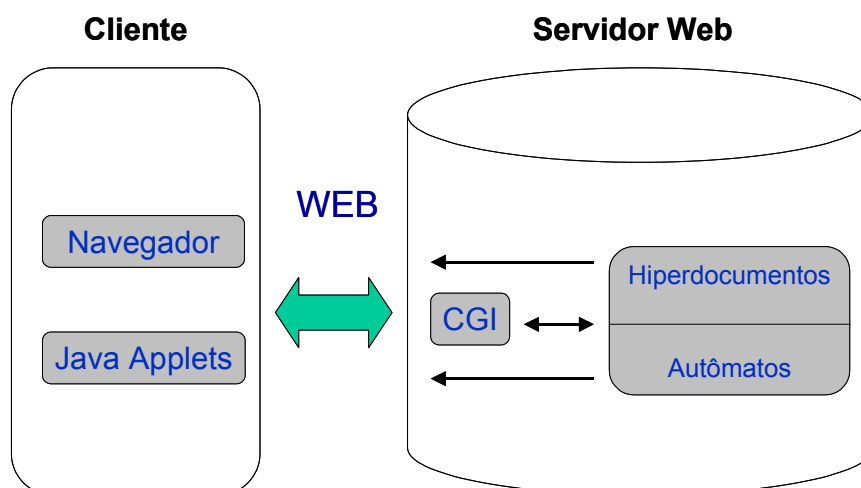


FIGURA 16 - Hyper-Automaton – Modo de Criação

O servidor é formado por um conjunto de programas *CGI* (desenvolvidos na linguagem *Perl*) alocados em um servidor *HTTP* e oferece serviços de controle sobre as estruturas dos autômatos e acesso à hiperbase. Um cliente pode utilizar qualquer navegador *WWW* com capacidade de execução de *applets Java*. As páginas *HTML* e material hipermídia em geral estão localizadas no servidor *HTTP* em questão. O sistema não utiliza nenhum editor de *HTML* integrado, podendo o professor utilizar o programa com o qual está mais familiarizado.

Embora os arquivos estejam armazenados em um servidor *HTTP*, como na maioria dos sistemas destinados ao ensino pela Internet, há uma diferença fundamental: os arquivos não contêm *links* armazenados diretamente em seu código *HTML*. Dessa forma, a seqüência das informações a serem "navegadas" pelos usuários não está contida no próprio documento, mas separada do conteúdo instrucional.

O sistema consiste de três componentes principais:

- uma ferramenta de autoria de autômatos;
- uma interface de navegação;
- um controle de armazenamento.

A ferramenta de autoria permite ao professor criar, modificar e reutilizar definições de autômatos. Quando o professor grava uma definição de curso, um processo específico garante o armazenamento dos dados no servidor. Por fim, um usuário acessa os hiperdocumentos pelo módulo de navegação sobre autômatos. Como consequência, o reuso do material instrucional é muito simples. Pode-se alterar o autômato sem alteração nas páginas *HTML* e vice-versa, bem como utilizar uma mesma página para diversos exercícios. O professor pode armazenar no servidor um grande número de autômatos diferentes, cada um com um objetivo específico como: preparar uma bateria de exercícios para uma avaliação final, ou até mesmo disponibilizar testes e exercícios dos anos anteriores. Com isso o professor poderá acompanhar a evolução do aluno no curso e fazer cruzamento com exercícios/provas e correlação e estatísticas dos diversos módulos. O Hyper-Automaton é um sistema interativo em desenvolvimento

para criação de módulos de cursos e exercícios/provas, disponível em plataformas WEB, como demonstrado na figura 17.



FIGURA 17 - Hyper-Automaton Modo de Criação

O Hyper-Automaton possui um conjunto de programas no servidor que operam em dois modos: edição e execução.

No modo de edição existem as seguintes ferramentas:

Alfabeto - cria ou altera o alfabeto de entrada do autômato, armazena os dados em um arquivo chamado alfabeto.fl. O alfabeto de entrada do autômato refere-se ao número de estados definido pelo usuário/professor conforme número de questões de sua lista de exercícios ou teste, representado pela figura 18.

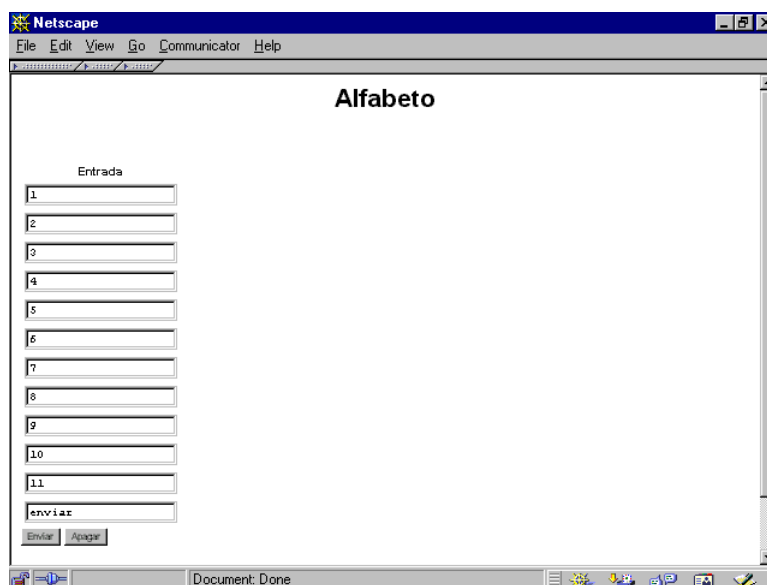


FIGURA 18 - Hyper-Automaton Montagem do Alfabeto

Arquivo – cria ou altera o conjunto de páginas a serem utilizadas pelo autômato, atribuindo um código para cada página. Neste caso, o professor possui, já preparado um conjunto de páginas/questões, onde serão relacionadas pelo número de estados ou transições definida no alfabeto de entrada, visto na figura 19. Os dados são armazenados em arquivo.fl.

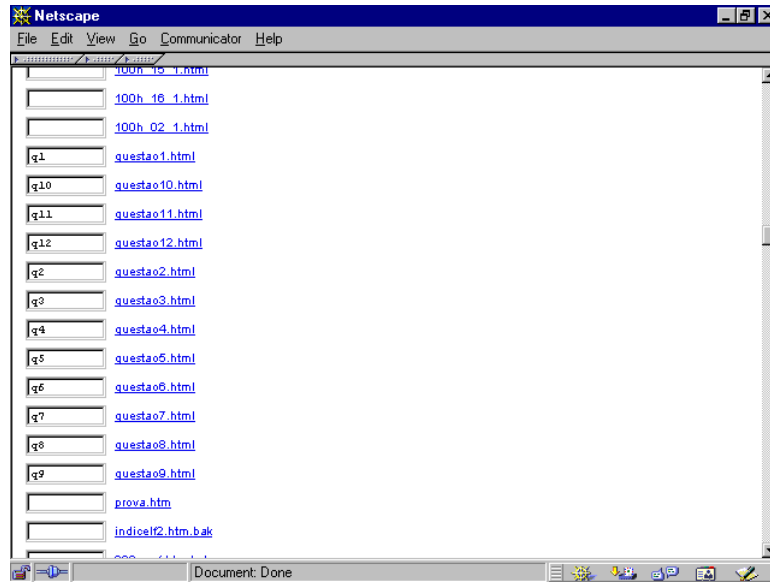


FIGURA 19 - Hyper-Automaton Escolha do Arquivo

Programa – cria ou altera a função programa do autômato, utilizando os **códigos das páginas** como **saída** para cada **estado** (no caso da máquina de Moore) ou **transição** (no caso da máquina de Mealy). Armazena as informações no arquivo programa.fl. Neste momento visto na figura 20, podemos verificar na prática a estrutura Hyper- Automaton, onde visualizamos o autômato com seus links hipertexto lógicos.

| Estado | Saida | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | enviar |
|--------|-------|---|----|----|----|----|----|---|---|---|----|----|--------|
| 0 | q1 | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 | q2 | 0 | | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 2 | q3 | 0 | 1 | | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 3 | q4 | 0 | 1 | 2 | | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 4 | q5 | 0 | 1 | 2 | 3 | | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 5 | q6 | 0 | 1 | 2 | 3 | 4 | | 6 | 7 | 8 | 9 | 10 | 11 |
| 6 | q7 | 0 | 1 | 2 | 3 | 4 | 5 | | 7 | 8 | 9 | 10 | 11 |
| 7 | q8 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | | 8 | 9 | 10 | 11 |
| 8 | q9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | 9 | 10 | 11 |
| 9 | q10 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | | 10 | 11 |
| 10 | q11 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | | 11 |
| 11 | q12 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 12 | p1q1 | | 13 | 14 | 15 | 16 | 17 | | | | | | 18 |

FIGURA 20 - Hyper-Automaton Montagem da Função Programa

Curso – gerência a criação e a remoção de cursos no servidor. Armazena as informações no arquivo curso.fl.

As n-uplas dos autômatos de exercícios/provas apresentam uma correspondência as estruturas de hiperdocumentos na WEB, seu estado inicial corresponde uma página HTML, onde o conjunto do símbolo de entrada do alfabeto é as alternativas das questões elaboradas pelos docentes.

As transições do autômato, definidas na função programa (questões) funcionam como ligações lógicas (links HTML figura 21), quando selecionadas durante a navegação do hipertexto.

No último estado, o autômato gera uma transição de saída (Mealy) com todas as questões selecionadas, alternativas, dados do aluno, gabarito como mostrado na figura 22.

Nesta estrutura Hyper-Automaton, a manutenção e o reuso dos estados em outra montagem, torna-se transparente para o usuário e muito prático para o professor, visto que o sistema Hyper-Automaton permite o reuso das páginas para montagem em outros exercícios.

estado inicial q_0

O alfabeto Σ de símbolos de entrada é uma seleção de alternativas (resposta)

A função Programa (questões), corresponde as ligações links lógicas ou HTML, durante a navegação do hipertexto

INF05514 Questões de Prova

Linguagens Formais - Capítulo 2 - Teste #2

Questão 3:

Considere o seguinte autômato sobre o alfabeto $\{a, b\}$:

Então, está CORRETO:

a) Aceita a linguagem $(aa + \epsilon)(a + b)^*$

b) Aceita a linguagem $(aa a^* + b)^*$

c) Aceita a linguagem $(a + b)^* aa(a + b)^*$

d) Não é uma linguagem regular

e) Nenhuma das afirmações acima

Last update Sept 15, 1999.

[1] [2] [3] [4] [5] [6] [enviar]

FIGURA 21 - Hyper-Automaton Ligações Lógicas

No modo de execução existem dois módulos:

Autômato – módulo que recebe como entrada o estado atual e, com base nas informações armazenadas nos três arquivos mencionados anteriormente, devolve a(s) página(s) indicada(s) pelo código (uma palavra formada pela concatenação de páginas) na saída do estado/transição atual. Este módulo também retorna os links para os

próximos estados disponíveis no autômato de acordo com a respectiva função programa.

Resposta – módulo que recebe como saída do estado atual uma transição de saída, com base nas informações armazenadas nos arquivos de dados mencionados anteriormente, devolve a comparação do arquivo com gabarito na saída do estado ou transição atual.

Uma das vantagens do sistema tanto para o aluno como professor é o retorno das questões corretas e incorretas de uma lista de exercício ou provas, definido no último estado ou transição como mostrado na figura 22.

Através da utilização do protótipo em desenvolvimento do sistema Hyper-Automaton para criação de cursos e exercícios/provas na WEB, surgiram novas propostas de ferramentas, como por exemplo:

- Provas Adaptativas;
- Geração automática ou semi-automática para criação de exercícios e provas;
- Barras de Ferramentas (atributos de cores, lápis, bloco de rascunho etc.);
- Interface;
- Símbolos/Figuras;
- Integração Curso, Exercícios/provas/Dicas;
- Ferramenta de busca;
- Editor de Questões.

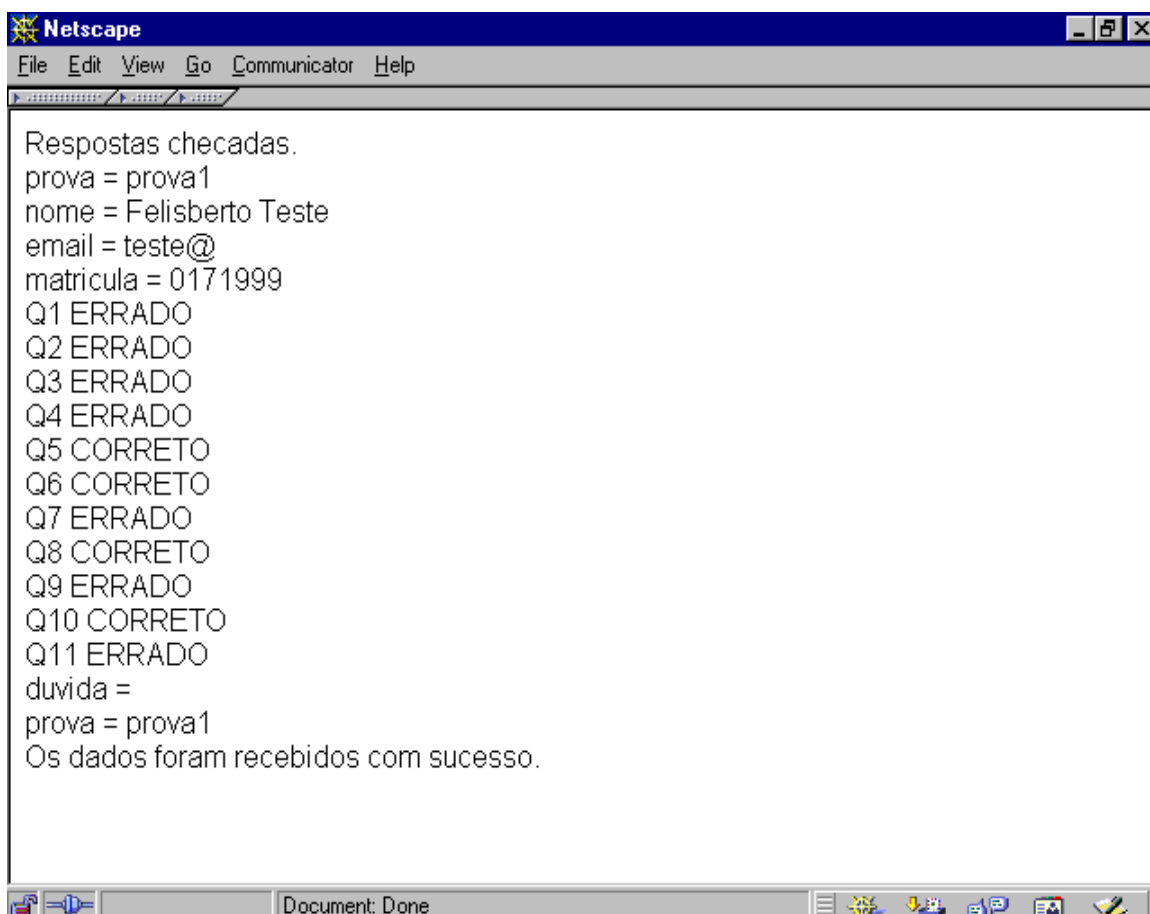


FIGURA 22 - Hyper-Automaton Respostas Corretas/Incorretas

2.7 Comparação entre Sistemas

Nesta seção, realizamos comparação entre os sistemas que possuem mecanismos de resolução de exercícios, com o objetivo de apresentar um quadro comparativo, ressaltando a facilidade de uso, confiabilidade e a qualidade da informação exibida aos usuários.

O processo consiste em comparar as seguintes funcionalidades:

- a) Plataformas suportadas;
- b) Possibilidade de Integração;
- c) Suporte Browser;
- d) Características do Produto;
- e) Gerenciamento do Usuário;
- f) Interface / Exercícios;
- g) Distribuição de Programas;
- h) Monitoria da Rede;
- i) Manipulação dos Dados;
- j) Segurança dos dados;
- k) Opção de Notificação;
- l) Opções de Relatórios.

TABELA 3 - Comparação de Mecanismos Resolução de Exercícios

| Produto | Hyper-Automaton | Question Mark | Hot Potatoes | Pathware 4 |
|---|-----------------|---------------|--------------|------------|
| Plataformas Suportadas | | | | |
| Windows / 95/98 | S | S | S | S |
| Windows NT | S | S | S | S |
| Macintosh | S | S | S | S |
| Unix | S | N | N | N |
| Linux | S | N | N | N |
| Possibilidade de Integração | | | | |
| Sistema Aberto para Manutenção | S | N | S | N |
| Importa dados | S | S | S | S |
| Exporta HTML | NI | S | S | S |
| Requer um Servidor Dedicado | S | S | N | S |
| Linguagem | Pearl | QML | Java | Java |
| Suporte Browser | | | | |
| Netscape | S | S | S | S |
| MS Iexplore | S | S | S | S |
| Característica do Produto | | | | |
| Interface simples | S | S | S | S |
| Banco itens biblioteca de questões | S | S | S | S |
| Gerencia segurança e contas de usuários | NI | S | N | S |
| Backup de dados de configuração de servidores | NI | N | N | S |
| Fornecer informações sobre as conexões | NI | S | N | S |

| Produto | Hyper- Automaton | Question Mark | Hot Potatoes | Pathware 4 |
|--|---------------------|------------------|-----------------|---------------|
| Monitora os servidores para compor estatísticas | NI | S | N | S |
| Gerenciamento | | | | |
| Gerencia a configuração Usuário | NI | S | N | S |
| Monitora os usuários para compor estatísticas | NI | S | N | S |
| Identifica e corrige problemas de aplicativos | NI | S | N | S |
| Interface/Exercícios | | | | |
| Quantidade de tipos de exercícios | NI | 8 | 6 | NI |
| Editor de Questões | S | S | S | NI |
| Randomização | S | S | S | NI |
| Opções aleatórias | S | N | N | NI |
| Multimídia (som, vídeo e animação) | NI | S | S | S |
| Escore/Pontuação | S | S | S | S |
| Provas/Simulados | S | S | S | N |
| Prova Interativa e Adaptável | S | N | N | N |
| Gabarito Automatizado | S | S | S | N |
| Distribuição de Programas (1) Freeware ou (2) shareware | 1 | 2 | 1 | 2 |
| Monitoria da Rede | | | | |
| Programa inclui monitor de acesso | NI | S | N | S |
| Oferece um histórico geral | S | S | N | S |
| Apresenta análise do Log. da Rede | NI | S | NI | S |
| Oferece estatísticas globais | NI | S | S | S |
| Manipulação de Dados | | | | |
| Base de dados é baseada em SQL | S | S | N | S |
| Índice Remissivo e Mec. De Busca | S | N | N | N |
| Segurança de Dados | | | | |
| Site protegido por login e senha | N | N | N | S |
| Arquivos de base de dados protegidos por senha | N | N | N | S |
| Aceita diferentes níveis de privilégios administrativos | N | N | N | S |
| Opções de Notificação | | | | |
| Mensagem por email | N | N | N | S |
| Relatório | | | | |
| Oferece scripts (templates) de relatório | N | N | N | N |
| Habilita ordenação / Filtros | N | N | N | S |
| Exporta relatório no formato de plan. Eletrônica / BD | N | N | N | S |

(NI) Não implementado

(N) Não

(S) Sim

A tabela acima exposta apresenta a comparação entre quatro sistemas de aplicação para ensino na WEB, onde se destacam alguns itens importantes:

- Quanto à plataforma de sistema suportada, até o presente momento o Hyper-Automaton utiliza todas as plataformas e os demais não apresentaram uma versão para os sistemas operacionais Unix e Linux;
- Em relação à integração de importar/exportar dados para o formato HTML, todos os sistemas permitem, mas o único sistema que permite a alteração de seu código fonte é o HOT POTATOES, codificado em linguagem HTML;
- Quanto à linguagem para WEB, o diferenciado é o sistema Question Mark, que utiliza uma linguagem proprietária QML, utilizando as características da linguagem HTML;
- Quanto aos browsers todos utilizam o Netscape e MS IExplore, bem como utilizam módulos em off-line, sem precisar estar conectado ao servidor.
- Quanto às características do produto, todos apresentam uma interface simples, banco de questões, monitoramento da rede e informam o perfil do usuário.

Na análise acima descrita, a diferença do Hyper-Automaton em relação aos outros sistemas, está na navegação dos hipertextos e na capacidade de definição dos controles (links ou botões) navegacionais. Enquanto os demais sistemas contêm somente botões predefinidos do tipo próximo anterior, o alfabeto de entrada do Autômato Finito com Saída fornece ao autor um mecanismo de definição dos links de navegação com flexibilidade total do autor (professor-aluno).

A metodologia de criação de material instrucional para o Hyper-Automaton utiliza a premissa de não utilizar links internos às páginas a fim de promover a reutilização do material desenvolvido. A funcionalidade de navegação externa ao hipertexto implica na utilização de mecanismos de frames (quadros) HTML ou de programas no servidor capazes de interceptar o acesso a URLs externas, de forma a garantir a possibilidade de retorno ao hiperdocumento a qualquer momento que o usuário necessitar.

Os outros sistemas estudados aplicam um pré-processamento sobre as páginas na WEB a fim de incluir um mecanismo de navegação guiada sobre os documentos originais. Eles agregam código HTML para inclusão de botões e links na especificação original dos documentos. Tal solução tem o inconveniente de alterar o layout original das páginas acessadas, ao contrário do Hyper-Automaton o qual utiliza links lógicos, facilitando o processo de reuso dos documentos instrucional em diversos cursos.

O Hyper-Automaton é o único que apresenta a facilidade e funcionalidade de um índice remissivo, construído através de palavras-chave ligadas a cada hiperdocumento do alfabeto de saída do autômato, permitindo mecanismos de busca automatizada, bem como mecanismos de dicas para exercícios e avaliações.

3 Study of Rooms for the Teaching Mediated by Computer

A mais nova proposta de salas de aulas representa uma solução para a idéia que esta dissertação descreve, sobre a necessidade de prover aplicação de exercícios e teste mediado por computador, ou grupos que estejam trabalhando com aprendizagem, seja aplicado na universidade, seja na escola, ou qualquer organização pública ou privada com ambientes que facilitem a produtividade, tanto local como remoto. A proposta é exploração de espaço entre alunos-alunos e alunos-professor, um espaço totalmente flexível e de fácil reconfiguração, permitindo uma aproximação entre o professor e aluno, possibilitando trabalhos em grupos, seminários, videoconferência e aplicação de provas.

Esta proposta originou-se de um artigo aceito e publicado no evento descrito a seguir.

Evento – Internacional: ICECE'99 – International Conference on Engineering and Computer Education

Principais Patrocinadores

- IEEE – Institute of Electrical and Electronic Engineers;
- ASEE – American Society for Engineering Education;
- ISTECC – Ibero-American Science and Technology Education Consortium;
- ABENGE – Associação Brasileira de Ensino de Engenharia.

Local: Rio de Janeiro - Brasil **Período:** de 9 a 12 de agosto de 1999

Categoria: Artigo Completo.

Tema do Congresso: O congresso está voltado para a divulgação de experiências instrucionais inovadoras e novas tecnologias no ensino de engenharia e computação em todo o mundo.

A principal contribuição como autor desta dissertação, foi abordar concepções, teórica e prática sobre a aplicação de provas mediadas por computadores. Assim sendo, são tecidas algumas considerações a respeito da utilização de computadores na sala de aula, através de um estudo que investiga como os computadores estão dispostos nas universidades e escolas, enfatizando a abordagem dada por alguns professores. Com essas perspectivas, busca-se responder os problemas localizados, em que pretende-se com a abordagem desta pesquisa, tecer métricas proporcionado aos professores modelos de salas, para adaptação a sua prática de ensino, adequando-a as novas necessidades como aplicação de provas mediada por computadores em um ambiente local.

Neste capítulo, são apresentados os levantamentos realizados nas universidades e escolas, comparação entre os modelos, sugestões para uma sala ideal, proposta, trabalhos futuros e conclusão.

Study of Rooms for the Teaching Mediated by Computer

Carlos Tadeu Q. de Moraes, Júlio P. Machado, Paulo B. Menezes
ICECE99

3.1 Abstract

This article has for objective to propose models of classrooms for the teaching mediated by computer. Based on the results observed in the visits these rooms, evidenced by the responsible of the schools and universities, they are pointed some models of classrooms, of the courses of Computer Science, Medicine and Engineering, in order to provide a general vision of the same ones. The principal problem is students' super capacity and workspace permeated by obstacles between teacher and student. In this text we will present well in way summarized the models researched, well as comparisons between them and proposal of models for classrooms.

3.2 Introduction

The visits made possible to develop proposed for study of a model of room of ideal class. For the use of teaching the distance, defined for the examples of traditional rooms and laboratories as: the knowledge acquisition and ability through the information and mediated instruction including every technology and other learning forms at the distance. Comparing with the model of the future room of class, for high school of USP [ESC 99]. We verified the needs that are the universities, in matter in Brazil: with relation the teacher formation and in increasing the number of student vacancies, so that these can give an attendance individualized for each student as we propose. Besides, the class full the students of the classrooms are a difficulty for a teaching more individualized, centered in the student. With those models we can change this scenery.

The schools and universities look for alternatives, changing the format of disposition of the tables and computers in the rooms, leaving the in line traditional for other formats as in column format, type " U " or " M ", trying to approximate the teacher's student, offering them best learning conditions and teaching.

The paper is organized in the following way: section 3.3 the researches accomplished in the schools and universities, of the models of classroom. Section 3.4 we will make a comparison among the models. Section 3.5 we will present suggestions for a room of ideal class. Section 3.6 proposed of models of classrooms. Section 3.7 future works. Section 3.8 it contains the conclusions.

3.3 Researches Accomplished at the Universities and Schools

In this section we will present the important points of the research that orientated the search of an ideal room, facts as:

- Space between teacher and students;
- Ergonomic points between student and computer;
- Communication between student and teacher mediated by circulate dislocate of the chair or of some approach of the teacher until the student.

Adopted criterion:

- layout of the room, format type (line, columns or format " U ");
- space among students;
- space between teacher and student, facilitating the approach for support;
- space among the computers;
- type of chairs fixed or circulate;
- illumination in relation to the harmful reflexes to the videos (clear walls, windows);
- ergonomic points: posture when sitting down, support of the hands for fingering and angle of the students' vision in relation to the teacher;
- Support tools for the teacher.

Next we classified in four models the results research in the visits:

MODEL I

In this model the classrooms have laboratory with total of 14 rooms, with forecast for 18 rooms.

The layout of the room has traditional format, In format line. All the tables with the computers come back forward from the picture, how much of difficulty for the teacher to approach to the student.

Each room have 08 tables, with a length of 4ms, with 2 to 4 points of nets and capacity from 4 to 6 students for table, fixed chairs, an average of 18 computers. These are classrooms with point the way nets, without computers on the tables, so that the students connect their personal computers.

The machines of the laboratories are computers Pentium 100 MHz, linked in nets with Internet access. Support tools for the teacher, are the picture and a computer with image projector.

The attendance for their students, they work in three groups: morning, late and night and more intermediate courses. Day with occupation total: Tuesday, Wednesday and Thursday. Total 136 groups, forecast until the end of the semester 190 groups. Project future: 23 computers for room.

The current difficulties in the rooms of classes are:

- updating of Software;
- growth from 30 to 40% of the number of groups per semester;
- Schedule conflicts.

MODEL II

The capacity of the Center of Technology is of approximately 5.000 students, being 3.000 destined to the course of Computer Science. The Center of Technology has a resource total of nine laboratories.

The layout of the room, is described this way: each lateral of the room is 05 computers with the backs of the videos gone back to the wall, in the center they are 10

computers, where there are two arrays with 05 turned computers of backs one for the other. The main idea is to facilitate the teacher's circulation join to the student.

The machines of the laboratories are computers Pentium 100 MHz, linked in nets with Internet access and matricial printer for reports of programs. The support tools for the teachers are blackboard and retro project. The attendance for the students is in three groups: morning, afternoon and evening and more 02 intermediate courses. Their days with larger occupation are Tuesdays, Wednesdays and Thursdays

MODEL III

This laboratory model possesses an average of 18 computers, with two arrays of tables in each lateral one in form of having supported capacity to support video, CPU, keyboard and mouse.

The layout of the classroom has the format in " U ", and in each lateral of the room have 08 computers, with the backs of the videos gone back to the wall. In each the extremity they are 04 computers, the main idea is to facilitate the teacher's circulation close to the student. Capacity for 40 students (two for computer) and fixed chairs. The equipment's of the laboratories are computers 486 100 MHz, linked in nets with access Internet, printer ink jet for reports of programs and the students' works. Support tools for the teacher: blackboard and retro projector.

The attendance in two groups, morning and afternoon. Day with total occupation: Tuesday, Wednesday and Friday. Capacity of approximately 700 students, destined to the Technical course in Computer science and Initial Series.

MODEL IV

This laboratory model has an average of 30 computers, 30 tables with capacity of supporting video, CPU, keyboard and mouse. Layout of the room, line formats. The equipments of the rooms are computers 486 and Pentium of 100 MHz, linked in nets with access Internet, printer ink jet for reports of programs and the students' works. Support tools for the teacher: blackboard and retro projector.

The attendance for two groups afternoon and evening with capacity of approximately 500 students, destined to the Technical course in Computer Science and Technical in Electronics capacity for 30 to 40 students, fixed chairs.

3.4 Comparison of the Models

To be analyzed by us the examples above, we can verify that all the classrooms face the same difficulties:

- Super capacity in the beginning of every school semester, could have from two to three students for computer in the universities;
- schedule competition, besides the computer science courses, engineering and architecture, also compete for the use of the room the other courses;
- software not updated or not installed in all the computers, hindering like this the classes;
- computers in maintenance, increasing the student number for computer;

- administrative problems: it misses of available monitors for support the teachers during the classes, growth of the universities disproportional the classrooms and laboratories;
- before these facts some layouts, as the models II and III, are already adapting if and standardizing their classrooms;
- Important points: space among students, appropriate tables for computers, limiting a number of 20 students and 20 computers for classroom, with the objective of facilitating the learning with smaller groups.

Working in a pleasant atmosphere, the student can learn more spontaneously, using the appropriate resources for the theory in practice, become an active part in the student learning.

"... In the era of the information the diversified educational experience will be the fundamental base for the success. The think that the students need is not to dominate the content and yes to dominate the learning " process [CAN 96].

3.5 Suggestions for an ideal Room

To crescent it looks for the computer science in the education, it is done necessary the search of a new layout of classrooms and computer science laboratories for application of new techniques.

Explain some examples of classrooms, in the educational entities, of then, which we verified that the room with format of " U " left the students the will to work. A good free space for the teacher's displacement to the student, circulate chairs for the students to turn in the direction of the picture and if necessary to form a debate group it will be enough to rotate the chairs for the center. Tables and ergonomic chairs and support for short students, all the tied computers the net and Internet.

Some requirements of modification of layout of the classrooms defined by Fred Bennett [BEN 99]

- a computer for each student;
- next machines one of the other with the back part gone back to the wall or for the back part of another machine;
- larger classrooms than the current ones that were drawn to assist 30 students;
- with the computerized education the students won't have fixed room, because of any machine they will access their configurations;
- the demand of the silence will be smaller;
- an architecture server-customer to assist the whole demand;
- can change the presence demand once the student can advance their study in relation to the remaining of the class;
- a support service in the school so that the student can access their machine of the own house; and
- Won't have somebody teaching in the classroom, there will be a facilitator of the learning.

It's true that such configuration it will demand heavy investment on the part of the school and their leaders. But it's also true, that if the school doesn't accompany this movement more and more it will be moved way of a reality and provoking the inhibition of a learning that can cost more still expensive to the society.

As an example of use of the so many computers in classroom or out, nowadays, it is already a reality, transparent for some universities.

In the course of Medicine of ULBRA where the computation is present, in all ambits, so much in classroom, as well as, serving as resource so that the student can accomplish works and studies extra-class.

The important is that the proposals focus the efforts of the schools and universities in improving the quality of the teaching, in search of a pleasant atmosphere, cooperative and productive for the progress of the education. I mention some educational points that validate the to change:

- of liberty that the student starts with have, with the entrance of the computer in the classroom is a new variable that needs to be worked with the teacher. [BEN 99]
- Our education system prepares for systematic evaluations in order to approximate how the student learned to be what the teacher has brought or stimulated was to look for in suitable sources.
- With the computer to the disposition, the student has a volume of information that can generate the contradictory what the teacher brought. [BEN 99]

"It is necessary that the educational technologists and the educators get to break the difficulty of working inter disciplines, protecting her the paper of each one in the process educational " [RAP 90].

Another possible dream of taking place with the computer science in the education is it of ending with the distance among the disciplines. Taking advantage of the technological resources close to for course assembly the computer, approximating the teachers for the new resources.

The Program of Teaching the Distance in Science of the Theory Computers [MEN 98a,MEN 99,WEB 98] it includes several modules: Electronic book (constituted of tutorials), Evaluation and Management, Animations, Simulators, System of Help, Bank of Exercises, References and it Consults Bibliographical, and Applications [MOE 98].

3.6 Proposal

It is important to observe that, before beginning to think in the solution of a problem, it should be understood the problem fully to be solved, in way to look for a better use of the students in the classrooms.

The schools and universities look for alternatives, changing the format of disposition of the tables and computers in the rooms, leaving the traditional in line for

other formats as in column, type " U " or " M ", trying to approximate the teacher's student, offering them best learning conditions and teaching.

3.6.1 Proposed I

The Layout that we propose is based on the model III of the classroom of format type " U " visa in the fig. 23, for 18 to 20 students for room. Where it will be a student for computer, facilitating the teacher's circulation in the room, approximating the same of the student.

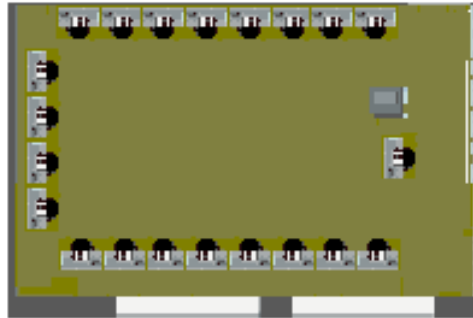


FIGURA 23 - Layout format "U"

3.6.2 Proposed II

The previous model, proposed in relation to above numbers of 20 students by room reinforces the proposed Layout II; we presented the Layout format " M " for groups from 40 to 50 students for classroom. In this case the format " M " can become two rooms in ' U ", as illustration in the fig. 24 to proceed:

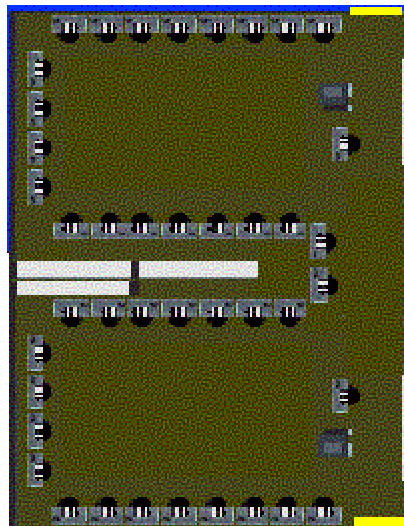


FIGURA 24 - Layout type "M" or two rooms in "U"

We presented the model accordingly above, two germinated rooms, with separate and with movement possibility if moving the side doors for the left song of the classroom, turning her larger, making possible the educator's movement among the rooms.

The tendency, even in Institute of Computer science of UFRGS, it provides that the student uses the same equipment, same software, and in most of the American schools the equipment's Macintosh are used. Another tendency is the use of rooms with wait of net point, for the students they swallow their individual equipment's to use in the classes. Fact that already happens, as for instance, Medicine of courses in ULBRA.

We concluded that we should make possible the students to have a global vision, as much as hardware, as of software, with the use of more than a platform, for us not to induce to use an only platform.

3.7 Future Works

With the challenge of placing in practices the proposed model, a model of viable room will be developed, economically, in acceptable time, assisting the project TEIA - Techniques of Teaching Interactive, of UFRGS, that today this in line format as example of the Fig. 25.

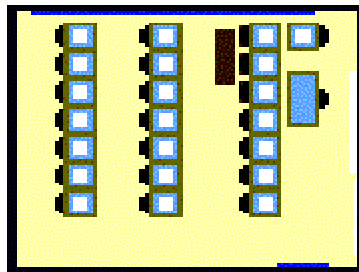


FIGURA 25 - TEIA Project Layout Room.

Assisting to the criteria of the proposed model, it will only be able to be reached with the conclusion of the application of this model in line, shown in the Fig. 25. Transforming for the format " U " as sees in the projection of the Fig. 26, without larger costs to not to be in the adaptation, tends as base the same number of equipment foreseen in the proposed model I..

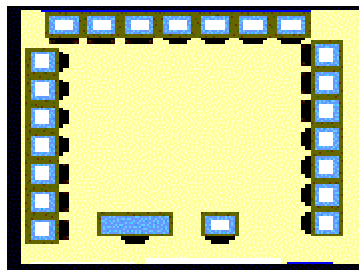


FIGURA 26 - Proposed Layout Projects TEIA

3.8 Conclusion

This text presents a rising of models of classrooms for courses of Computer science, Medicine and Engineering, presenting in a summarized way their examples of classrooms, allowing a comparison among them.

We placed some suggestions, where we propose two models of rooms of formats " U " and other " M ".

We showed that they exist the possibility to change the format in line for format " U " without possible additional costs, tends the same characteristic of the proposed format.

As it was evidenced along the text, the objective of this work, is a proposed of model that approximates the students' teachers. Where we will have a considerable improvement in the teaching, because, the students feeling more comfortable in rooms or laboratories, they will produce more and the teacher tends a larger access close to the students, he will notice the difficulties better presented by the same ones.

Breaking like this, the barrier a lot of times created by the students in relation to the teachers.

4 Exercícios e Avaliações Interativo Mediadas por Computador Baseado em Sistema Formal

Hoje muitos testes de proficiência, certificação, seleção e pesquisa de mercado já utilizam o computador para sua aplicação, com o objetivo de apurar e agilizar as informações de forma rápida e flexível para o usuário final. Com esta facilidade, o professor e o aluno buscam agilizar o processo de aplicação, correção e respostas de forma automática. Uma nova aplicação utilizando a construção formal, conhecida como autômatos finitos com saída, foi testada em paralelo a um teste tradicional (em papel) com um teste real de um protótipo para aplicação de exercícios e provas mediada por computadores, utilizando como estrutura de navegação um autômato finito com saída. Após, buscou-se verificar alguns pontos de avaliação entre os dois testes para aprimorar o protótipo. Esta proposta originou-se de um artigo aceito e publicado no evento descrito a seguir.

Evento Nacional:

SBIE99 – Simpósio Brasileiro de Informática na Educação.

Principais Organizadores

- SBC - Sociedade Brasileira de Computação,
- Departamento de informática da Universidade Federal do Paraná;

Local - Curitiba; Paraná. Brasil **Período:** 23 a 25 de novembro de 1999

Categoria – Artigo Completo.

Página do Evento - <http://www.inf.ufpr.br/sbie99/index.htm>.

Tema do Congresso - Tem como objetivo divulgar a produção científica nacional nessa área e proporcionar um ambiente para a troca de experiências e idéias entre profissionais, estudantes e pesquisadores nacionais e estrangeiros.

Neste capítulo, são apresentados os conceitos básicos da configuração de modelos de salas (rever no item 3.3 um resumo do artigo [MOC 99]), pontos relevantes à adaptação do aluno nos cursos mediados por computador, os mecanismos de suporte aos alunos para realização de provas e exercícios, características do sistema de exercícios e avaliação, implementação de um protótipo e, por fim, uma análise da aplicação de exercícios e provas como Teste Real.

Exercícios e Avaliação Interativa mediada por Computador Baseada em Sistemas Formais

*Carlos Tadeu Q. de Moraes, Júlio P. Machado, Paulo B. Menezes, Mauro Nogueira, Ricardo A.L. Reis.
SBIE99*

4.1 Resumo

Este artigo tem por objetivo propor uma ferramenta para elaboração de exercícios e de uma avaliação interativa de alunos para o ensino mediado por computador, onde trabalha em conjunto com o curso disponibilizado via WEB para o ambiente remoto e local, utilizando-se construções formais, conhecidas como Autômatos Finitos Determinísticos, dando continuidade a uma análise mais profunda no programa de Ensino a Distância de Informática Teórica. Os principais problemas são a superlotação de alunos e espaço de trabalho permeado por obstáculos entre professor e aluno. Neste texto apresentaremos de forma bem resumida a configuração do ambiente de uma sala de aula, bem como exercícios e exames mediados por computador, mecanismo, característica e proposta do sistema de exercícios.

Palavras-chave: ensino a distância, exercícios e avaliação, WWW, autômatos.

4.2 Introdução

O avanço da informática na educação indica que estamos vivendo um período forte de transformação, surgindo então, a necessidade de formar alunos capacitados a interagir com a máquina na busca de resultados positivos e preciosos para a solução dos problemas de aprendizagem.

Nesse aspecto, o primeiro passo para a solução é desafiar a potencialidade da máquina e do educando, buscando a habilidade do raciocínio lógico, da criatividade e do senso crítico. Para isso, não existirá uma forma única de avaliação desse processo.

Este trabalho visa desenvolver um sistema de exercícios e de avaliação interativa de alunos em cursos na WEB, baseado em Sistemas Formais a dar continuidade a uma análise mais profunda no programa de Ensino a Distância de Informática Teórica [MEN 98a, MEN 99, WEB 98] em relação aos módulos: Livro Eletrônico (constituído de tutoriais), Avaliação de Alunos e Gerência, Animações, Simuladores, Sistema de Ajuda, Banco de Exercícios, Aplicações, Referências e Consultas Bibliográficas [MOE 98].

A utilização deste modelo de exercícios será desenvolvido para um suporte do ensino de Informática Teórica na UFRGS, tendo como base a criação de uma ferramenta simples, de fácil implementação, manutenção e reuso, independente de outros sistemas (como banco de dados, sistema operacional, etc.).

Desenvolver um sistema de exercícios e de avaliação de alunos com aplicação em ambiente remoto e local, que consistirá em vários exercícios referentes ao curso, passando para o professor um retorno de suas aulas e aos alunos uma forma de revisar e estudar o conteúdo, utilizando uma bateria de exercícios e finalizando com provas.

No ambiente local, foi preparada uma lista de exercícios e testes, junto aos cursos de graduação e pós-graduação nas disciplinas INF05514 Linguagens Formais e MPI007 Projeto de Sistemas VLSI - UFRGS, com a participação dos educadores em elaborar junto aos seus cursos exercícios e testes, numa forma de medir parâmetros e dificuldades individuais de seus educandos, tornando a correção de problemas relevantes a aplicação mais rápida.

O sistema desenvolvido será uma ferramenta importante para o trabalho do professor, uma vez que pode ser utilizado para diversas atividades, tais como: introduzir um sistema de exercícios permitindo ao aluno aprofundar-se num tema, revisar matérias já estudadas, preparar alunos para provas, atender a alunos de diferentes perfis [KAW 96].

4.3 Configuração do Ambiente de uma Sala

Uma das questões mais importante em qualquer aplicação de exame é o local em que serão realizadas as provas nos aspectos individualizados. A constante busca pela informática na educação, faz-se necessária a busca de novo layout de salas de aula e de laboratórios de informática para aplicação de novas técnicas [MOC 99].

Alguns quesitos de modificação de layout das salas de aula definidos por Fred Bennett [BEN 99]

- Um computador para cada estudante;
- salas de aula maiores que as atuais que foram desenhadas para atender 30 estudantes;
- com a educação computadorizada os estudantes não terão lugar fixo, pois de qualquer máquina acessarão suas configurações;
- a exigência do silêncio será menor;
- arquitetura servidor-cliente para atender toda a demanda;
- períodos normais de duração das aulas continuarão porque permite que os estudantes interajam;
- um serviço de suporte na escola para que o estudante possa acessar sua máquina da própria casa.

É importante observar que, antes de se começar a pensar na solução de um problema, deve-se compreender plenamente o problema a ser solucionado, de forma a buscar um melhor aproveitamento dos alunos nas salas de aula.

As escolas e universidades buscam alternativas, mudando o formato de disposição das mesas e computadores nas salas, saindo do tradicional em linha para outros formatos como em coluna, tipo “U” ou “M”, procurando aproximar o aluno do professor, oferecendo-lhes melhores condições de aprendizagem e de ensino [MOC 99].

Inclusive como verificado em casos reais, nessas configurações não há necessidade de salas de aulas maiores. Como por exemplo, o projeto TEIA – Técnicas de Ensino Interativas, da UFRGS, que hoje esta em formato de linha como exemplo da figura 27, sendo transformado para o formato “U” como mostrado na figura 28.

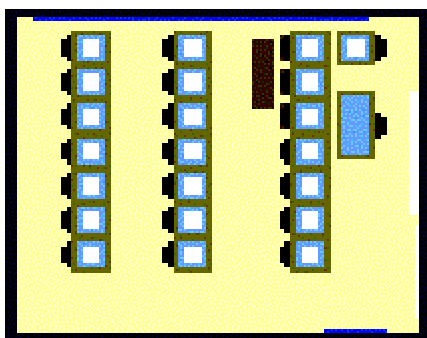


FIGURA 27 - Layout Sala em Linha

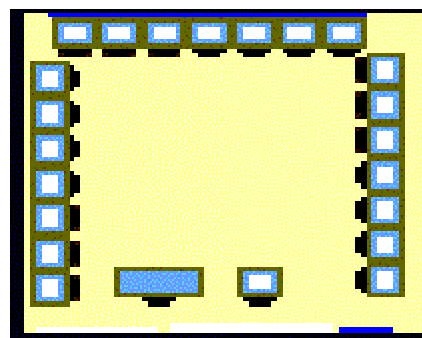


FIGURA 28 - Layout Proposto em “U”

4.4 Exercícios e Exames Mediados por Computadores

Com o surgimento de cursos mediados por computadores, o estudo e a experiência com modelos de interação, interfaces é altamente necessária a integração de testes e exercícios. Observa-se que os estudantes utilizam os computadores quase que diariamente na execução de tarefas, tanto em classe como extraclasse.

Se os exames serão usados ou não por computador, ainda é muito questionado. Sabemos que questões de pesquisa são as mais fáceis de administrar, mas questões de informações individualizadas requerem segurança para que os estudantes não tenham acesso as informações da rede.

O uso do computador nesse caso, parece ter sido iniciado por estudantes que contestavam em realizar exames no papel, quando o curso inteiro usava computadores [THO 96].

Questiona-se muito na adaptação dos alunos em relação à mudança, pontos que parecem relevantes, mas convém analisar, como por exemplo:

- Conhecimento do equipamento apresentará dificuldades para alunos que conhecem pouco o equipamento, e mensagem que o sistema possa dar uma parada da máquina;
- uso das informações, saber utilizar os recursos que o sistema permite acessar, desde a navegação até a resolução dos problemas;
- dúvidas e apoio do colega, a troca de informações entre estações na rede é o principal problema que o administrador deverá solucionar, sendo que não presente uma solução que garanta a redução do risco, tendo um aluno que tenha uma boa memorização ele pode visualizar a tela do colega;
- bloco de rascunho, os alunos estão acostumados a rabiscas nas provas ou um papel de rascunho, em que o sistema deverá apresentar algo similar para apoio ao aluno;
- confiança no equipamento, esta é da responsabilidade do administrador em deixar todos os equipamentos em condições.

É importante ressaltar que faz-se necessário um treinamento aos estudantes para adaptar-se o novo ambiente. Em que deverão ter vários testes de adaptação, principalmente na administração do tempo e na utilização de todos os recursos que ferramentas disponibilizam para efetuar um exame.

Convém observar que cada aluno tem o seu próprio ritmo e seu conhecimento tanto de hardware como software podem ser limitados, contudo a administração do tempo desses alunos pode ser crucial nesse novo ambiente.

4.5 Mecanismo para Elaboração de Exercícios e Exames.

Este artigo descreve uma forma de desenvolver uma ferramenta que efetue um cruzamento dos exercícios e da avaliação com o curso, de maneira de apresentar suas informações organizadas.

O objetivo desta ferramenta é para que o professor consiga acompanhar a evolução do aluno no curso, através das premissas do cruzamento com exercícios, provas e correlação.

A elaboração dos exercícios será de forma simples e flexível, disponibilizando uma bateria de exercícios e provas dos anos anteriores para que os alunos possam utilizá-los.

A estrutura das questões dos exercícios será mostrada em uma página principal, em que constará um índice de grupos de exercícios e diversas opções por tópicos.

Os mecanismos de apoio para os alunos, seguindo seus costumes utilizados em provas escritas, buscarão adaptar alguns recursos para sua utilização como, por exemplo, papel rascunho e anotações na prova.

Notou-se a necessidade de viabilizar uma caneta de destaque para anotações textuais com a qual o aluno poderá sublinhar seu texto, através da qual também poderemos exibir as questões marcadas com palavras chaves para resolução do exercício.

Disponibilizaremos como apoio ao aluno um mecanismo de busca e navegação que será baseado em autômato, o qual descreveremos a seguir.

Cada módulo de testes constitui de um curso definido sobre um conjunto de páginas independentes, conseqüentemente elas poderão pertencer a diversos módulos.

A função desse sistema é buscar do aluno o que realmente ele aprendeu, tendo assim uma bateria de testes para o mesmo executar a qualquer momento. Além de verificar os seus escores de acertos, o aluno poderá ter um retorno do grau de entendimento, passando para o professor um retorno de suas aulas e aos alunos uma forma de revisar e estudar o conteúdo, utilizando uma bateria de exercícios e finalizando com uma prova.

A fuga do aluno dos objetivos propostos pelo curso é diminuída através desse controle eficiente. A avaliação de alunos contínua desse processo será através de

exercícios e de pesquisas que possibilitam através da WEB a troca de experiência entre os alunos, com o objetivo de proporcionar o senso de equipe e cooperação.

Um dos problemas dos atuais sistemas de curso na WEB é que a maior parte dos cursos não tem tido a preocupação de acrescentar em sua página um acompanhamento do aluno em relação ao seu conteúdo.

O sistema aqui apresentado procura contornar esse problema, sendo um sistema interativo entre conteúdo e aluno.

4.6 Característica do Sistema de Exercícios e Avaliações

O sistema consiste em um ambiente local e remoto para apoio interativo ao professor, com recursos que facilitam elaborar seus exercícios e avaliação, bem como uma interface amigável para os serviços disponibilizados do curso. A primeira etapa do desenvolvimento está sendo feita no ambiente local, onde foi realizado um estudo sobre o melhor layout para ambiente local [MOC 99].

A função principal do sistema consiste em: elaborar bancos de exercícios; desenvolver um sistema de avaliação interativa de alunos em cursos na WEB baseado em Sistemas Formais; gerar bateria de Exercícios; desenvolver uma ferramenta interativa de apoio para os professores (exercícios e avaliação); integrar ao sistema de curso via WEB; estudar a viabilidade de um gerador automático de exercícios e provas (autômatos) a partir de um banco de exercícios e provas.

4.7 Sistema Proposto Utilizando Autômato para Elaboração dos Exercícios

A utilização deste modelo de exercícios está sendo desenvolvida para um suporte do ensino de Informática Teórica na UFRGS, tendo como base à criação de uma ferramenta simples, de fácil implementação, manutenção e reuso, independente de outros sistemas (como banco de dados, sistema operacional, etc.).

O site provê material instrucional para cursos de Teoria da Computação, Linguagens Formais e Teoria das Categorias, baseado em notas de aula e em livros já publicados [DIV 99, MEN 98] por professores do Instituto de Informática da UFRGS. O site também inclui ferramentas automatizadas que facilitam a comunicação entre alunos e professores, como quadro de avisos e listas de discussão.

O módulo de avaliação está sendo integrado a um amplo sistema de ensino na WEB. Com esta integração é possível definir exercícios "inteligentes" com vários níveis de "dicas" para a solução de problemas e referências cruzadas para o material instrucional relativo à questão apresentada.

Além disso, a criação de exercícios modelados como autômatos finitos serve à categoria de sistemas capazes de oferecer estudo individualizado, com caminhos construídos especificamente para cada nível de habilidade dos alunos.

O sistema que permite a construção de avaliações interativas de alunos faz parte de um modelo para definição de cursos a serem disponibilizados na WEB utilizando o

formalismo de Autômatos Finitos com Saída (Máquina de Mealy, Máquina de Moore). Tal modelo é descrito em maiores detalhes em [MAC 99a, MAC 99, MEN 99b].

Desta forma, como a criação do material hipermídia é independente do autômato, é possível criar diversos autômatos sobre um mesmo material, cada um com objetivos específicos ou enfoques diferentes.

4.8 Implementação

O sistema consiste em um ambiente de apoio interativo ao professor, baseado em uma arquitetura cliente-servidor na *WWW* [VAS 97], com recursos que facilitam uma representação gráfica dos exercícios, assim como uma interface amigável para os serviços disponibilizados (criação e manutenção de exercícios).

O servidor é formado por um conjunto de programas *CGI* (desenvolvidos na linguagem *Perl*) alocados em um servidor *HTTP* e oferece serviços de controle sobre as estruturas dos autômatos e acesso à hiperbase.

Um cliente pode ser de qualquer navegador *WWW* com capacidade de execução de *applets Java*. As páginas *HTML* e o material hipermídia em geral estão localizados no servidor *HTTP* em questão.

O sistema não utiliza nenhum editor de *HTML* integrado, podendo o professor utilizar o programa com o qual está mais familiarizado.

Embora os arquivos estejam armazenados em um servidor *HTTP*, como na maioria dos sistemas destinados ao ensino pela Internet, há uma diferença fundamental: os arquivos não contêm *links* armazenados diretamente em seu código *HTML*. Dessa forma, a seqüência das informações a serem "navegadas" pelos usuários não está contida no próprio documento, mas separada do conteúdo instrucional.

Como consequência, o reuso do material instrucional é muito simples. Pode-se alterar o autômato sem alteração nas páginas *HTML* e vice-versa, bem como utilizar uma mesma página para diversos exercícios.

O professor pode armazenar no servidor um grande número de autômatos diferentes, cada um com um objetivo específico como: preparar uma bateria de exercícios para uma avaliação final, ou até mesmo disponibilizar testes e exercícios dos anos anteriores. Com isso o professor poderá acompanhar a evolução do aluno no curso fazer cruzamento com exercícios/provas e correlação e estatísticas dos diversos módulos.

4.9 Caso Real

Hoje o nosso trabalho parcialmente descrito neste artigo é centrado no desenvolvimento de um sistema semi-automatizado para apoiar o ensino.

É importante destacar que em menos de um ano esta aplicação já está em fase de teste, em conjunto com o curso de Linguagens Formais e Autômato do Instituto de Informática da UFRGS.

Do ponto vista da usabilidade, o sistema foi elaborado de forma de apresentação de autômato de maneira amigável.

Disponibilizamos também uma lista de exercícios para disciplina MPI007 Projeto de Sistemas VLSI – UFRGS elaborada pelo professor para seus alunos testarem o sistema, visto na figura 29.



FIGURA 29 - MPI007 Projeto de Sistemas VLSI

Na análise da usabilidade do sistema de exercícios, tais critérios foram levados em conta: a necessidade do aluno em precisar de um bloco de rascunho, circular e rabiscar sobre os enunciados das questões relevantes às provas e listas.

Portanto os principais objetivos específicos são validar o sistema de autômato para exercícios e avaliação interativa, entretanto constatamos que realmente atende ao objetivo proposto e poucas outras facilidades necessitam ser adicionadas não necessariamente no autômato, mas no mecanismo de apoio a resolução das questões.

Aplicamos duas provas uma on-line e outra tradicional no papel, para uma turma de 25 alunos, em que 12 fizeram a prova on-line e 13 à prova em papel, conforme figura 30.

Acompanhamos durante todo o tempo da prova às reações e a necessidade de cada aluno, a partir disso, constatamos que:

- Todos poderiam acessar o livro para consultar, menos os on-line, onde os mesmos utilizaram o livro eletrônico para consultar, tendo como mecanismo de busca por índice.
- Todos iniciaram juntos, com tempo de duas horas para concluir as respostas.
- Nos primeiros 10 minutos alguns alunos levantaram dúvidas sobre o enunciado de certas questões, ocorreram as mesmas dúvidas tanto para provas em papel como em on-line.
- Na prova on-line houve muita necessidade por partes dos alunos de riscar e anotar qualquer coisa que definiram importante para seu raciocínio.
- Na prova tradicional, alguns alunos riscaram ou circularam os enunciados das questões da prova.

- Tempo de resolução igual, após 30 minutos os alunos já haviam respondido 2 a 4 questões da prova de 11 questões.
- Tempo de conclusão período igual para ambas às provas, duas horas para resolução, o primeiro a entregar foi da prova tradicional e o último a entregar o da prova on-line.
- Ao final da prova on-line os alunos depois de submetidos às respostas surpreenderam com o resultado automático das questões certas ou erradas, fazendo os mesmos revisarem as suas questões erradas.

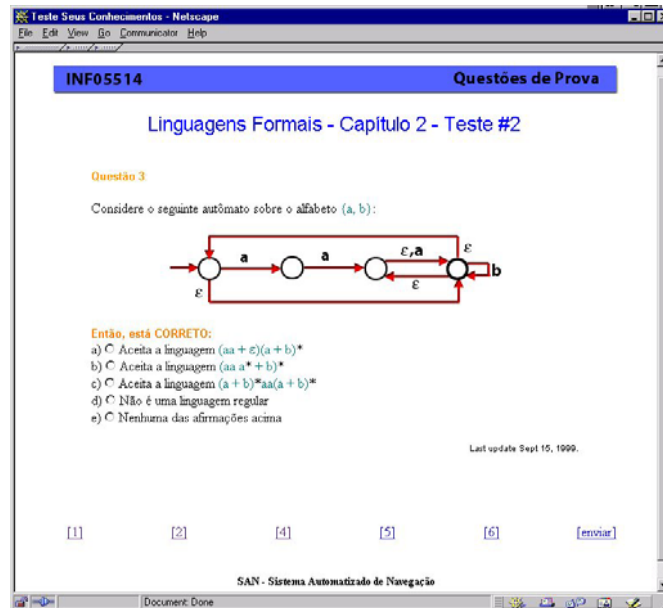


FIGURA 30 - Linguagens Formais Teste 2.

Os resultados imediatos mostraram para os alunos uma idéia de como é importante revisar os conteúdos antes da prova, principalmente a resolução de exercícios disponíveis na rede como material de estudo. Assim, incentivam a estudar o material disponível na rede. Fato que ocorreu após a conclusão da prova muitos retornaram para consultar os exercícios disponíveis para simulação de uma prova, pois quinze minutos depois, já estava disponível como exercícios e com gabarito. Observou-se também que a navegação foi maior do que a de alguns dias antes da prova.

Esse sistema devolve para o professor e aluno os resultados automáticos em relação ao tradicional, possibilitando acompanhar se os alunos estão ou não resolvendo seus exercícios. Durante a resolução disponibilizamos um caixa de dialogo para os alunos acrescentarem suas dúvidas e sugestões.

As médias das notas obtidas tanto tradicional e on-line não apresentaram diferenças significativas, conforme podemos analisar na tabela abaixo:

| CONCEITO | REDE | PAPEL | MEDIA REDE/PAPEL |
|----------|------|-------|------------------|
| C OU > C | 69% | 58% | 64% |
| D OU < D | 31% | 42% | 36% |
| TOTAL | 100% | 100% | 100% |

Notou-se que houve um percentual de 11% a favor da prova on-line, dados que serão analisados junto com outros testes futuros para poder ter um comparativo mais significativo.

4.10 Conclusão

Ao propor uma ferramenta para resolução de exercício e avaliação, buscou-se incorporar as novas tecnologias das redes, utilizando uma proposta simples e flexível, tanto para alunos como para professores, suas vantagens facilitam o reuso de listas de exercícios, banco de provas e geração automática de exercícios e provas.

A ferramenta tem por objetivo que os alunos vivenciem experiências novas e desafiadoras durante o processo de ensino, a possibilidade das respostas imediatas, aproximou-se os alunos a navegar mais sobre a ferramenta.

4.11 Trabalhos Futuros

A partir dos resultados dos primeiros testes on-line, o próximo passo será a implementação das modificações necessárias para otimizar o funcionamento das facilidades, baseadas nos parâmetros analisados durante a aplicação das provas e exercícios. Com esses ajustes, espera-se obter uma melhor adequação do sistema à realidade dos alunos e professores, na elaboração de mais testes de casos reais com alunos, medir parâmetros, validar e opinar metodologia proposta.

Diante dos resultados obtidos na metodologia implantada em comparação com a metodologia tradicional, inicia-se a modelagem para a geração automática de exercícios e avaliação.

5 A WEB Teaching System Based on Formal Methods

Ao início do novo milênio, caminha-se para a construção de um novo perfil de professores, em que a tecnologia entra como um sistema de apoio no dia-a-dia do professor, tanto na criação de um curso, tópicos, ajuda, índice remissivo como os módulos que medem o potencial do curso que são exercícios e avaliações comuns e adaptativas. Toda a sua estrutura foi modelada para ser utilizada em ambiente local (off-line) ou para ambiente remoto (WEB). Sua base de modelagem foi construída utilizando o conceito de autômato finito com saída, seja no curso, seja nos exercícios e nos testes de avaliações. O sistema é uma proposta amigável e de fácil configuração a partir do qual o professor tem a possibilidade de estruturar seu curso reutilizando seus exemplos como exercícios ou dicas. Esse sistema originou um artigo que foi inicialmente submetido na categoria de pôster no congresso MSET/2000 e aprovado. Seu conjunto de informações foi complementado com novas pesquisas, testes realizados e integrado com o curso. Foi o primeiro artigo aceito com o nome do sistema Hyper-Automaton. Submetido e aceito como artigo completo no evento descrito a seguir.

Evento Internacional - ICEUT2000 - International Conference on Educational Uses of Communication and Information Technologies.

Principais Organizadores

- 16th IFIP - International Federation for Information Processing
WCC2000 World Computer Congress
- Ministry of Information Industry - China
- Dept of Computer Science & Technology Peking University

Local: Beijing, China

Período: 21 a 25 de agosto 2000

Categoria: Artigo Completo.

Página do Evento: <http://www.wcc2000.org>

Tema do Congresso A conferência tem como principal objetivo explorar novas tecnologias que influenciam na educação como ferramentas educacionais, tutorias e outras que facilitam a aprendizagem no ensino. Dentro de seu tema destacamos a sua motivação que consta na página do evento: *“The conference is motivated by the need to understand the implications of novel technologies on teachers in the post-industrial information age. Especially in developing countries like China, traditional styles of education inherited from its culture and its long history needs to be re-investigated before reformed and adapted to novel styles of education. This conference will address these issues and questions, investigate traditional assumptions and cognitive styles in education, and evaluate a variety of novel learning styles, such as self-paced learning and learning by playing with information.”*

Neste capítulo são apresentados o projeto TEIA, sistema Hyper-Automaton, exercícios e avaliações utilizando o autômato e avaliações adaptativas.

A WEB Teaching System Based on Formal Methods

*Carlos Tadeu Q. de Moraes, Júlio P. Machado, Paulo B. Menezes, Ricardo A.L. Reis
ICEUT2000*

Abstract

This paper presents a modeling of courses available in the WWW making use of formal constructions known as Deterministic Finite Automata. Our work, partially described in this paper, is centered in developing a semi-automated system to support the teaching (both undergraduate and graduate levels) of Theoretical Computing Science and VLSI Design at the Computer Science Institute of the Federal University of Rio Grande do Sul - Brazil. It was done through the definition of a set of courseware tools based on the manipulation of constructions from the Automata Theory [HOP 79], Category Theory [BAR 90, WIN 97] and Hyperdocuments Technology [GAF 93].

Keywords

WWW, distance education, remote courses, finite automata, course modeling.

5.1 Introduction

With the emergence of the WEB it became possible to provide students with access to a broadly range of information and teaching materials independently of time and geographic constraints.

Integration of computers into classroom can increase the learning potential of a student. Computers can provide immediate feedback on students' progress helping the teachers to better monitor this progress.

The interactive learning environment is a step in the right direction. Computers become tools for cognitive development and supporting participants in a rich learning environment.

The WEB is also a powerful tool that allows us to set hybrid learning environments, combining collaborative aspects from classroom-based teaching with the flexibility provided by WEB-based teaching, thus allowing the students to learn at their own pace [BOR 98].

One of the main problems with the current courseware systems is the lack of integration between course contents pages and the assessment tools.

The proposed system, called Hyper-Automata [MAC 99a, MEN 99b], avoids this drawback through the use of a common formalism for the control and organization of exercises/tests and instructional hyperdocuments. It provides tools for the creation of what we called "intelligent questions", with several levels of cues and cross-references to the WEB course pages [MOC 99b].

In this paper, we intend to introduce the mechanisms behind the relationship between WEB course pages and computer managed exercises/tests. We also provide a

discussion on the use of such strategy in Formal Languages, Category Theory, and microelectronic classes.

5.2 Project TEIA

The efforts are based on the Project TEIA - "Técnicas de Ensino Interativas Assistidas por Computador" (Computer Aided Interactive Teaching Techniques) [WEB 98] whose main objectives are:

- Stimulate student's independence from the teacher and better engagement in their own knowledge building process, allowing students to fulfill their needs at their own pace.
- Provide instructional materials with low cost. Hyperdocuments can be stored in WEB servers, reused and copied by students for use in their own computers.
- Increase the teachers' productivity. Once the instructional materials are in digital media, the teachers don't need to spend so much time reproducing their notes every semester. They can use the time to supervise students' activity individually, to stimulate the students' creativity, to develop experiments and group work, and to use feedback to improve the on-line materials.

5.3 Hyper-Automaton System

The system was implemented [MAC 99a] in Perl language using client/server architecture and it can run inside any WEB server that supports CGI scripts.

In the system, WEB courses are automata with output (Mealy/Moore Machines) [HOP 79,STO 98] and links between pages are automata transitions (not HTML source).

Each automaton defines a course and consists of a set of independent pages (Information Units), which may belong to other courses as well. The automata next-state function acts as a logical link between the Information Units. The result is the basic structure of pages and hypermedia links in a WEB site. It is important to highlight that the links between pages are not static and hard-coded into the HTML source, but they are defined in the automata transitions.

As a consequence, reusing the instructional materials is straightforward. The teacher can alter the automata without modifying the HTML pages and vice versa, as well as use the same pages in several courses. Thanks to the automata, the resulted system was easy to implement, simple but powerful.

The Distance Education Program includes a broad range of tools: electronic book (tutorials), system management, evaluation facilities, animated graphs, and simulators, help system, questions database, references and bibliography and other specific software programs. The system, which is being used and developed, is aimed at:

- Providing access to hypermedia materials from classroom, laboratories or home. The hypertexts include several techniques to enhance learning like animated figures depicting algorithms, the presentation of automata, and so forth. It also contains Java applets that implements abstract machines, from within instructors and students can select from a number of automata to animate. They can modify existing models as well as create entirely new ones. The on-line site (<http://teia.inf.ufrgs.br>) provides instructional materials on Theory of Computing, Formal Languages, Automata Theory and Category Theory, based on class notes and published books [MEN 98].
- Tracking students' paths and progress through logs of their on-line activities. The students' participation can be taken into consideration in the evaluation process. During the browsing, the students can navigate through the hypermedia course trying to attain the objectives of the course. The student can follow the course in compliance with what was implemented by the teacher (see fig. 31). The hyperbase allows, if needed, a free exploratory navigation, while the automata take to a guided navigation. Free navigation is also supported by the addition of an automated index facility. Each output-symbol (HTML page) contains a list of keywords describing its contents. The system then constructs a glossary for accessing all the instructional materials directly. A search facility is not yet implemented. We can also aggregate other useful data into each page like typing, date of creation, author info. In addition to guided navigation, a hypermedia system should keep track of the user's path through the network of information, allowing easy return from exploratory side paths. A very simple User Model allows the inclusion of this information in a database.

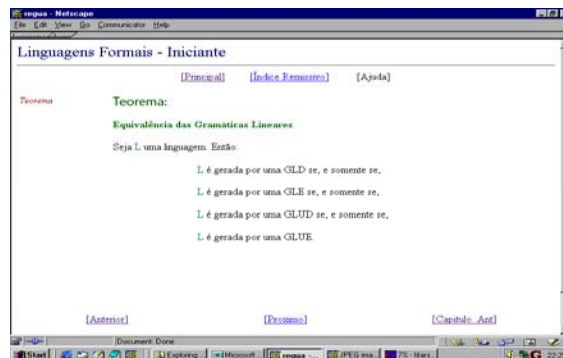


FIGURA 31 - Student Interface

- Organizing instructional contents as an automata with output model, aggregating diffuse information into coherent units; thus helping students to find the information they need without getting lost in the vast amount of files found in the WEB servers.
- Providing communication tools, which stimulate cooperative and/or collaborative work. E-mail, discussion forum and bulletin boards are very important because they explicitly address the process of knowledge construction.

- Organizing a database of computer managed exercises and tests. The assessment system is being developed for both formative and summative purposes. The main idea of formative assessment is to help students to learn.

The main aim of summative assessment is to classify the students in some way. It is currently being used to create a database of multiple-choice exercises with automatic checking of the answers for the courses on VLSI Design (shown in fig. 32) and Formal Languages (shown in fig. 33).



FIGURA 32 - Courses on VLSI Projects

The assessment system was tested in a local classroom with all the students. In the near future we plan to add security features in order to allow the students to take the summative tests at distance.

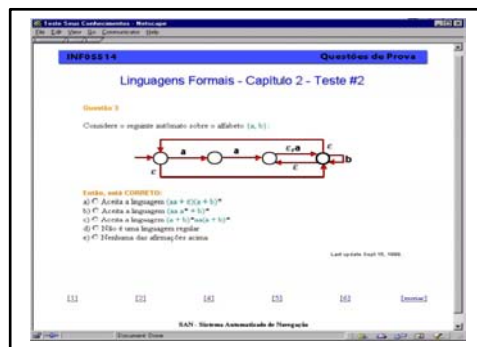


FIGURA 33 - Assessment on Formal Languages

5.4 Exercises and Evaluations using the Automata

The n-tuples of the exercise and evaluation automata (shown in fig. 34) present a correspondence to the hyperdocument structure in the WEB [MOC 99b]. An automaton, with an exit in the last transition, generates an exit tape for a new status (new page html) or it sends the information of the subjects (number, answer) and the student's data for a database recording in the server.

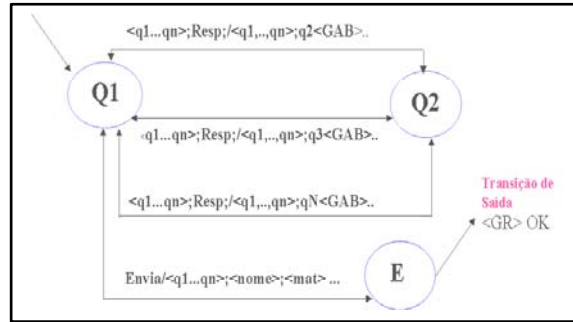


FIGURA 34 - Exercises/Evaluations Automata's Model

- The initial state q_0 corresponds to a common homepage call, the point of entrance to exercises and initial evaluations.
- The alphabet Σ of entrance symbols is a selection of alternatives (answer) that identify the navigation links in the hypertext of the evaluations / exercises.
- The exit words Δ^* , presents in the function program and function of exit δ_s , they can be pages connected as a unique WEB page in the user's navigator.

Notice that the word notion stays. Therefore the exits (of an alphabet Δ) are related to units of information (content of the questions) constituted by WEB pages.

The automata's transitions work as logical connections between the questions of the exam/test. These transitions define possible HTML links that will be selected by the students during the navigation of the hypertext.

Our focus is on using automata for organizing information resources to engage students in understanding, interpreting, and evaluating materials, providing knowledge building and critical thinking. At the same time avoiding problems like disorientation and cognitive overhead.

The visual interface of the browser environment provides the user with a tangible interpretation of the Mealy/Moore machines. The output alphabet is annotated with units of information (hypermedia HTML pages) and, in that case, the result of the next-output function (Moore) or the next-state function (Mealy) is the display of the HTML pages concatenated in one browser window. The input alphabet that labels the transitions in the automata is displayed as HTML links that a user can select. The link itself is the projection of the next-state function in the hypertext environment. If a link is followed, then the current displayed contents is deactivated and the contents mapped to the output nodes are activated, in accordance to the transition executed.

Although the essence of the model is its machine-supported linking, the way the output-alphabet (hypertext contents) is constructed has an important role in the system. The sizing of the HTML files are small and are the result of the modularization of a document into information fragments within a syntactic unit, such as a definition, an example, an explanation paragraph, etc. Thus, keeping the notion of output-word, a hypertext node is composed by several other small fragments (some kind of composite node).

In practice we can analyze the exercises / evaluations automata's n-tuples presenting a model corresponding to the hyperdocuments structures in the WEB.

5.5 Adaptive Evaluations

The Hyper-Automaton system also supplies functionality for the construction of adaptive evaluations via Internet. In this case, adaptive is understood as an evaluation of whose sequence of questions presentation is dependent on the student's performance, when answering previous questions. Basically, such behavior can be modeled (patterned) by an automata according to figures fig 35 and fig. 36. In this example, the questions were elaborated according to three levels of difficulties. First, the student answers an average level question. If he finds the right answers, the system provides a higher-level question, otherwise he will get a chance to recuperate the wrong concept by a question formulated specially for the recovery. It is important to emphasize the difference between the modeling, as a Moore machine (see fig. 35), and the modeling, as a Mealy machine (fig.36). Notices that a Mealy machine has a more adequate formalism to give support for evaluations and exercises. One of the reasons is the explosion of states in the Moore machine, which is necessary to hide the answer set (which could be deduced by the analysis of the state transition destination). It does not happen when using the Mealy machine since the outputs are associates to the transitions and not to the states.

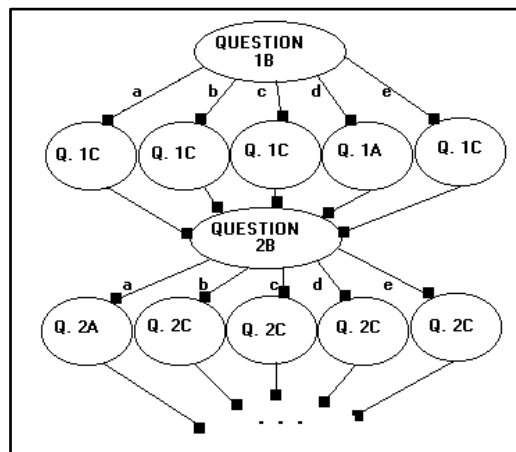


FIGURA 35 - Adaptive Test Moore Machine Model

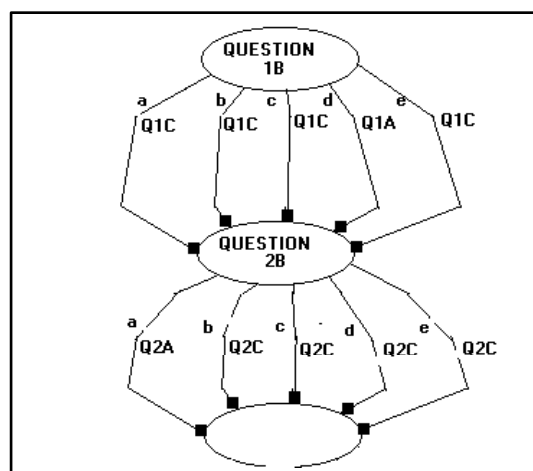


FIGURA 36 - Adaptive Test Mealy Machine Model

5.6 Conclusion

We do not intend to make a "breakthrough" in the teaching paradigms, but we want to show that it is important to develop tools that can be easily and usefully integrated into the daily classes at universities as lecture support or for distance and continuous education.

6 Modelagem para Geração Automática de Exercícios e Avaliações

O Hyper-Automaton é um ambiente livre para construção de sistema de ensino. Espera-se que o Hyper-Automaton abra espaço diferencial e inovador no sistema educacional, proporcionando uma mudança de paradigma quanto à aprendizagem em relação ao que é aprender e ensinar na internet. O computador é uma ferramenta que oferece a possibilidade de um ensino individualizado, permitindo ao aluno conduzir seu próprio ritmo. Essa experiência demonstrou que a implementação de um sistema de avaliação é sempre um desafio. Entretanto, após os testes experimentais, e de acordo com sugestões recebidas dos usuários (aluno/professor), através de questionários aplicados, obteve resultados positivos com aplicação das provas adaptativas, que motivaram e serviram como base para modelagem e desenvolvimento do módulo de geração automática para exercícios e avaliações.

Submetido e aceito como artigo completo no seguinte evento:

Evento Internacional: ICECE2000 – International Conference on Engineering and Computer Education

Principais Organizadores

- SENAC - escola de Engenharia e Tecnologia;
- IGIP - Internationale Gesellschaft für Ingenieurpädagogik;
- ASEE - American Society for Engineering Education;
- ASIBEI - Ibero-American Association of Engineering Education Institutions;
- RBE - Brazilian Network of Engineering;
- ISTECC - Ibero-American Science and Technology Education Consortium;
- SBC - Brazilian Computer Society
- SBPC - Brazilian Association for the Advancement of Science.
- IEEE – Institute of Electrical and Electronic Engineers Inc Brazilian Society for Engineering Education (ABENGE).

Local: São Paulo – Brasil **Período:** 26 a 28 de agosto 2000

Categoria: Artigo Completo

Página do Evento: <http://www.senac-sp.gov.br>

Modeling for Automatic Generation of Exercises and Evaluations

Carlos Tadeu Q. de Moraes, Paulo B. Menezes, Ricardo A.L. Reis
ICECE2000

6.1 Abstract

The proposed article describes a methodology for the development of an automatic generation system of exercises and evaluations, using as base a finite automaton model for creating exercises, where any user can build his own exercises in an automatic or semi-automatic way. The model makes possible that the user previously configures his main parameters as number of subjects, level of difficulty, percentage of subjects and subjects weigh. This article is centered in the analysis and application of Finite Automaton with Output formalism (Mealy/Moore Machine), with the objective of supporting the educators and students.

Index Terms — Finite Automaton, Distance Learning, Evaluation and Exercises, World Wide WEB.

6.2 Introduction

This work has as main objective to generate a tool to support the educators and students, in the elaboration of their exercises in an automatic or semi-automatic way using resources of the WEB.

Its development has as objective to integrate existing courses, so the educators can generate and create their own database of questions.

In order to test and validate the system we worked close with educators in order to build a description of their needs so that the system could assist most part of their tasks.

The educators' experience and the proposed methodology conclude with a practical application that stimulates the reuse of the developed materials, independent from WEB platforms.

The system is itself a structural basic mechanism for presentation and organization of hyperdocuments in hypertext systems based on the World Wide WEB. It is a solution that facilitates the implementation, maintenance and reuse of exercises or exams, since the links among the pages are transitions among Automaton's states and *not* hard coded in html pages.

This article presents a model of automatic or semi-automatic generation of exercises and evaluations based on the Finite Automaton with Output in order to the innovate the support tools for teaching. The main goal of the proposed model is to validate the modeling techniques and to develop a semi-automated system for the application of exercises and exams on-line, for supporting courses in the WEB. The emphasis of the article is in the development, implantation and real case of exams and exercises for graduated students and masters' degree students. The work is organized in the following way: section II presents a general vision of the current situation of the

tools for the educator's work. Section III describes the proposal; its advantages and working techniques improving the teaching activities. Section IV discusses the nonexistence of the technical restrictions. The section V describes the applicability of the model for specifying automatic generation of exercises and evaluations based on automata. Section VI presents an application for exercises – an automaton applied to a real case. Section VII illustrates the conception of the system describing their main modules. In section VIII some important aspect for modeling and implementation of the automatic generation are described.

6.3 At the Present Time

The educators, that today teach in several disciplines and have a great number of students for each class, constitute a situation of high cost of time for the school/university, as well as for the educator because:

- a) The educator is restricted in only one kind of model of exams/exercises;
- b) The time spent for the corrections of evaluations are multiplies by the number of students in the group;
- c) A conventional process that can reduce the time of correction of the exams indeed doesn't exist;
- d) A great number of students' that are anxious waiting the result;
- e) The gap between the evaluations and corrections let the student anxious and full of doubts.

6.4 Proposed Situation

The foreseen situation is to make possible to the educator the generations of his own exercises in an automatic or semi automatic way, totally configured according to his needs. fig 37 demonstrates the relationship among the basic processes for elaboration of the exercises/exams.

In the relationship presented in the illustration we have several processes in which the administrator registers the educators, and the educator registers the course, disciplines, topics, exams/exercises and students.

The relationship between course and discipline induces information for buildings the input alphabet, transitions and outputs of the topics of the course as well as exam, transitions and output exercises. In the table 4 we exemplified its functionality.

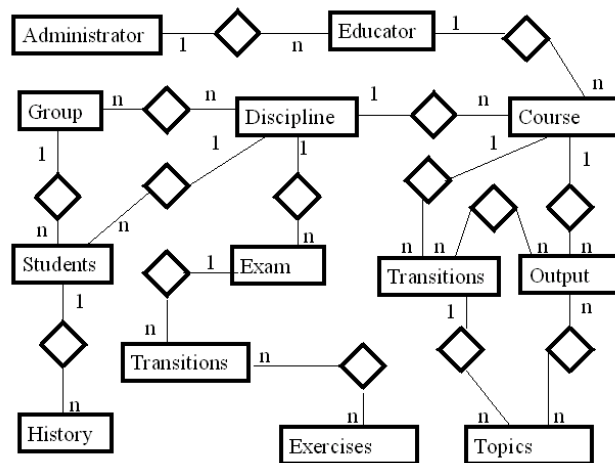


FIGURA 37 - DER Relationship basic processes of the exercises/exams

TABLE 4 - System Modules

| Administrator | Educator | Course | Exam |
|-------------------|-------------------------|----------------|-------------|
| Register Educator | Register Course | Description | Description |
| | Register Discipline | Input Alphabet | Transitions |
| | Register Student | Transitions | Exercises |
| | Register Topics | Output | |
| | Register Exam/Exercises | | |

Advantages:

- An editor of questions for edition and correction;
- More time available for students, once it significantly reduces the time for the preparation and correction of exercises;
- Instantaneous feedback for exams (no more anxious students);
- The automatic or semi-automatic generation is only available to registered educators through passwords and some other personal information to ensure the safeness of the data;
- History of the system navigation;
- Reports of complete and detailed statistical usability data;
- Integration with the course, making possible to use course contents together with the exercises such as clues.

Main functions of the system

- An editor of subjects;
- Research's in the questions database;
- Registration of parameters (attributes) for exercises/exams;
- Generation of exercises and exams in an automatic or semi automatic way.

Performance

The system should make an automatic list of exercises in a fast way (few seconds). A query to the database (questions) should obtain an answer in at most 3 seconds. (Except in the case of on line traffic jam).

6.5 Nonexistence of Technical Restrictions

The system has no theoretical restrictions, since it is fully designed to execute in any WWW platform, what should be confirmed during the real tests.

The main result of the project is, therefore, a clear and integrated vision of the stages of conceptual and practical construction of the system that intends to exhibit such robustness qualities, usefulness and usability.

6.6 Specification of the System

The description of a system of automatic generation of exercises and evaluation (see fig. 38): the system supplies information of the course with registered disciplines; each discipline has several topics so that the automatic generation module can generate subjects for Exams and exercises; the educator, through a local or remote computer, accesses a database of questions of its disciplines, allowing the generation of new questions through an editor of questions.

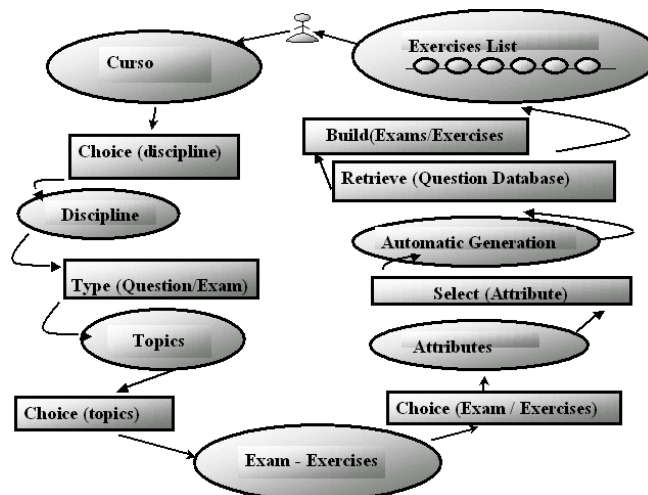


FIGURA 38 - Architecture of the System

Architecture of the Proposed System

From the users' point of view, we accomplished a questionnaire where they answered which was their needs for elaborating exercises. The system is being design so that the Automaton's structure is represented in a friendly way, facilitating its use by the educators. The main characteristic of the system is the automaton's automatic or semi-automatic construction, including the generation of questions, the retrieve of pages of exercises in HTML format from the database and the possibility of integration with the database of WEB courses.

The system is based on a client/server using the WWW as interface. The system uses resources that facilitate the graphic representation of the exercises. The model is described in details in [MAC 99]. The Automaton's use for the definition of a system of exercises and evaluation is being integrated in a on-line course for the teaching of theoretical computer science disciplines. The site provides instructional material for courses of Theory of Computation, Formal Languages and Category Theory based on class notes and in books already published [MEN 2000,MEN 99b,MAC 99a] by educators of the Institute of Computer Science of UFRGS.

The WEB server is extended with CGI programming [GAR 97b]. Basically, the CGI allows a server WEB to execute external programs and incorporate the output data of those programs to send things back to the navigator. Arguments are supplied to the program through special variables and standard input channel (stdin). The results of the execution are returned through the output channel (stdout). With that mechanism, WEB server has his functionality extended [DWI 96], [GAR 97b]. The fig. 39 below show, the operation of a CGI program [TIT 96]: The relationship between a HTML page and a CGI program happens through links or, usually, through HTML forms, that contain graphic elements of interface. One of those elements is the special " button SUBMIT ", those sends the content of the fields of the form to the CGI program specified in the definition of the form. That passage of arguments can be accomplished through two methods: GET or POST. The method GET concatenates the arguments in the own string, URL (Uniform Resource Locator) requested to the server. The method POST creates an object (a file, for instance) that is passed to the program [DWI 96], [TIT 96].

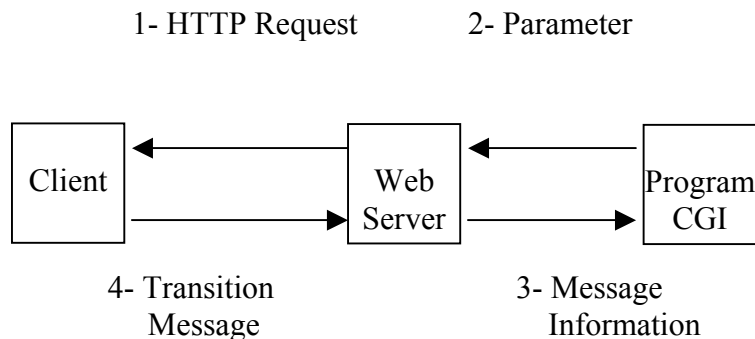


FIGURA 39 - CGI Event

The automatic or semi-automatic generator is a information system based on knowledge that uses a model of Finite Automaton with Output, generating exercises and exams in the normal form.

It works with the models of Moore and Mealy machines [MEN 2000]. The educator has at his disposal an interface capable to accept entrance parameters and to supply as output a list of exercises or it exams using graphic or textual means.

Fig. 39 illustrates the integrated construction of the interface for the automaton's states and the representation of the automatic or semi-automatic construction, so that the educator has at his disposal an interface capable to request and to supply the mounted Automaton in form of list of exercises.

We use a graph (Automaton) as a model for representing a diagram of Uses Cases, as we show in Fig. 40, with the principal Uses Cases. For instance, in the Uses Cases "Generate / object -state" the user requests the generation of the alphabet (exercises/exams), of an object-state. The system fires the necessary output transitions for the generation of the exercises and it produces an automaton representing a list of exercises.

The diagram (graph) in Fig. 40 also displays the detailed behavior of the Uses Cases "to Generate object-state". The system specifies the use of the services of classes that allow the creation of complex exercises starting from simple components, named service of Course.

We used the compositive pattern [MEN 2000] that describes how to use the recursive composition, so that the users don't have to do distinction between simple components and compositions. This fact allows Automaton's model to set up a process of reutilization of pages.

Besides the service of Course classes, other services are included in the system as for instance: Register that stores the information of the course, disciplines, educator; Properties that put in disposition information of the objects.

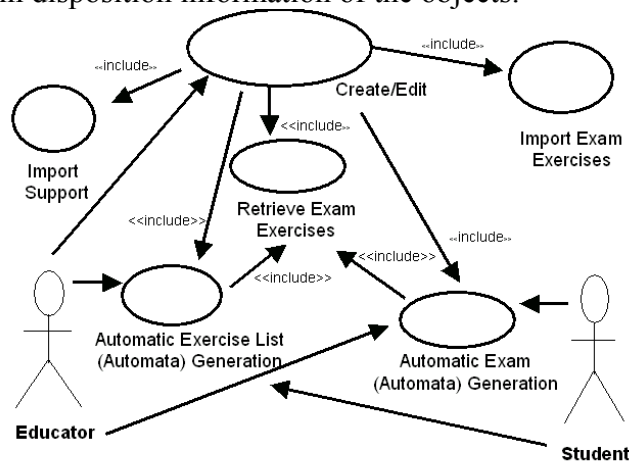


FIGURA 40 - General Architecture of the Automatic Uses Cases

The interface with the server allows access to the remote methods implemented in the server. Class *Exam* for instance, wich key point is the class *List*. That declares methods, as Attributes (fig. 41).

| GASEV-System | | | | |
|----------------------------------|------------------------|------------------|-----------------------------|------------------|
| Maintenance of Attributes | | | | |
| Difficulty Level | Question Number | Clues Add | Feedback | Weight |
| A | | | | |
| B | | | | |
| C | | | | |
| | Fields Clear | Back | Automatic Generation | Main Page |

FIGURA 41 - Maintenance of Attributes Exercises

This class can compose other "Object-exam" recursively. The class *Client* (user) uses the class *Exam* to accomplish his responsibilities, as for instance, to store the objects in the browser. The system of automatic or semi-automatic generation has a central repository for manipulation and storage of data, being responsible for the readiness of the necessary methods for the practical application of a list of exercises or exams. It is therefore, available for the user (student or educator) an automatic or semi-automatic construction of exercises and evaluations.

This service allows the educator the edition of new exercises, using any editor with HTML format. Later, the transference of the list of exercises is made to the database of pages (exams and exercises).

The modeling for automatic or semi-automatic generation of exercises and evaluations is being integrated into a wide education system in the WEB. With this integration it is possible to define intelligent "exercises" with several levels of "clues" for the solution of problems and cross references for the instructional material relative to the presented subject. Besides, the creation of exercises modeled as finite automata is useful for the category of systems capable of offering individualized study, with paths specifically built for each level of the students' ability. This model is described in [MOC 99b]. The system that allows the construction of students' interactive evaluations is part of a model for definition of courses to be available in the WEB using the formalism of Finite Automata with Output (Mealy, Moore or Machine). Such a model is described in details in [MEN 99b,MAC 99a, MAC 99].

6.7 Applying in a Real Case

When we proposed a tool for resolution of exercises and evaluation, we looked for a simple and flexible proposal for immediate use. The system gives freedom to the educators and students, to create and set up their own automaton (exercises), using the database of pages for reuse and its automaton's creation. We applied two on-line exams and other in paper [MAC 99]. With immediate result of the student's evaluation, and the subsequent access to the correct answers, most of the students spontaneously reviewed their answers right after the exam. Also, the educator had an immediate result of the performance of the class, allowing him to work immediately to recover the contents in that the students demonstrated a low profile. Fig. 42 and 43 present models of adaptive questions that we described in the [MOC 99b] [MOC 2000].

6.8 General Conception of an Automatic Generator

Main modules of the automatic generation:

- Course (Educator and Disciplines);
- Exercises (topics / chapters), Type of questions (objective / descriptive), Periodicity, Attributes (level of difficulty, clues, form, weight));
- Exams (topics / chapters); Type of questions (objective / descriptive); Periodicity; Attributes (level of difficulty, clues, form, weight));
- Editing of Questions (subject; special symbols / illustrations; alternatives; right alternative);
- Interface with other systems or processes (process of reuse of pages / navigation; transfer of screens).

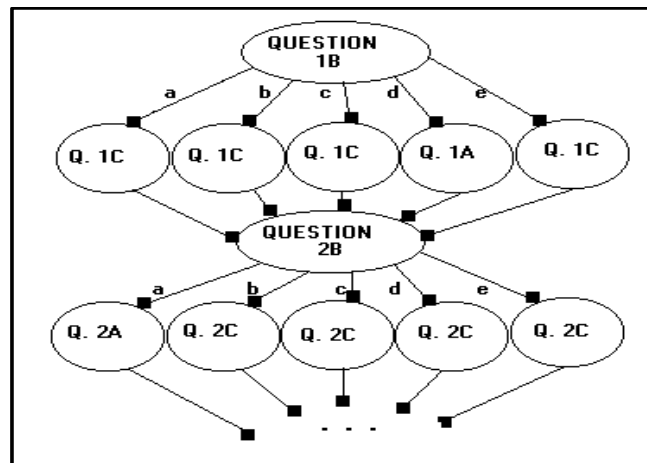


FIGURA 42 - Adaptive Question Moore Machine

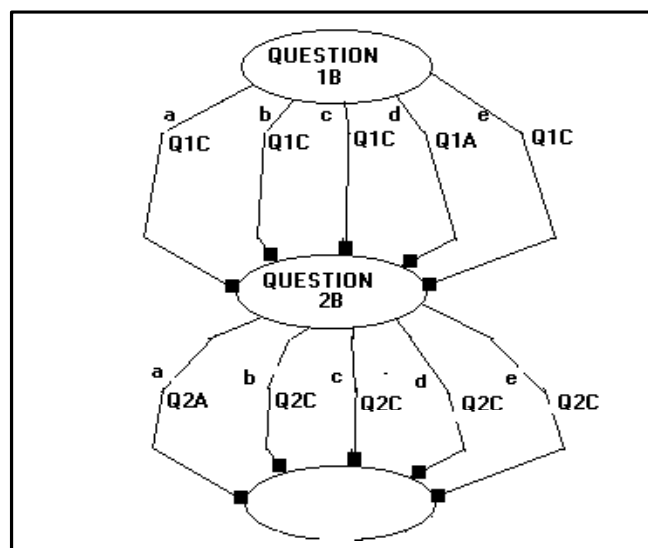


FIGURA 43 - Adaptive Question Mealy Machine

6.9 Important Aspects for Modeling / Implementation

The integration of the characteristics of the usual modeling of conventional information systems with finite automata with output specific concepts, results in the following list of modeling needs:

- Uses Case, identification of the functionality of the system, state and output transition;
- Structure of Documents, modeling of the documents automated organized by the system;
- Contexts of Navigation, definition of the different pages (visions) of documents for each activity and user (educator/students);
- Structure of Navigation, specification of the access path to the functionality of the system to the Automaton's construction;
- Databases of Pages, modeling of the relational database and integration of existent systems (exercises and exams);
- Objects of the Domain of the Application and Executable Components, specification of the other classes of objects and related necessary components as state-object;
- Workflow, specification of the activities to be accomplished by the users of the systems, its sequence and the documents and involved objects;
- Transactions, example of the typical interactions that involve navigation, edition of documents and activation of executable components;
- Implantation, specification of the physical distribution of the components of the system.

To model that concept properly, an Automaton model is described in graph form in which, for every state its transitions are analyzed, comparing with Uses Cases of the modeling language UML.

The integration of those models happens in a quite intuitive way. Three main points exist at the beginning of the modeling process: Uses Cases (Automaton), Documents and Database.

The importance of beginning in Uses Cases is for being an appropriate technique for analysis of requirements with the client/users. Starting from the definition of the functionality of the system through the Uses Cases, documents and databases are then planned to give support to the implementation of the functions and the representation of the necessary information.

Navigational Contexts could be defined to specify visions of each document adapted to the accomplishment of each task and the type of user involved. Based on the structure of the documents and on the contexts, the structure of navigation of the system is defined, specifying the relationship between documents and the access paths to the functions.

Being defined the structures of data and of navigation of the system, executable components are introduced to specify where it will be implemented the dynamic behavior of documents and links and how the integration will be accomplished between documents and databases.

6.10 Conclusion

In this article we presented the applicability of the model of Automatic or Semi-Automatic Generation of Exercises and Evaluations in Finite Automaton with Output specification / implementation, with emphasis in the fact of being a support tool to the educators/students. Initially some basic aspects of the educator's work were approached as state of the art, requirements and methods for specification of automatic or semi-automatic generation. It was described a proposal for modeling of exercises and evaluation, pointing out advantages and acting techniques with respect to teaching. We pointed out that any restriction doesn't exist with respect to the WEB platform.

The applicability of the model was demonstrated for specification of an automatic or semi-automatic generation based on the automaton model. We presented a model of exercises - applied automaton in real case; we illustrated the conception of the system describing its main modules. In the modeling and implementation of the automatic or semi-automatic generation the following aspects are important: uses cases, structure of documents, navigation contexts, databases of pages, workflow, transactions and implementation.

The presented approach constitutes an attractive alternative for educators and students. Through this model of automatic or semi-automatic generation of exercises and evaluations the educators/students can set up their own exercises and obtain their results in real time. The use of the model GASEV (Automatic or Semi-automatic Generation of Exercises and Evaluation) allows the potentialities of Automaton's model to be naturally incorporated in other courses. In particular, this approach allows the planner to be the own user, because he can build his own automaton.

7 Sistema Hyper-Automaton para Cursos e Avaliações Mediadas por Computador Baseado em Autômatos Finitos

Este capítulo propõe apresentar a integração dos módulos de exercícios e avaliações de forma automática e semi-automática com o sistema de curso que faz parte do Hyper-Automaton. Com esta integração o sistema torna-se mais completo para aplicação no ensino como ferramenta de apoio ao professor e aluno. Submetido e aceito como artigo curto no seguinte evento:

Evento Nacional: SBIE2000 – Simpósio Brasileiro de Informática na Educação.

Principais Organizadores

- Comissão Especial de Informática na Educação da Sociedade Brasileira de Computação

Local: Alagoas – Maceió **Período:** 9 a 12 de novembro 2000

Categoria: Artigo Curto.

Página do Evento: <http://www.tci.ufal.br/sbie2000>

Tema do Congresso: Tem como objetivo divulgar a produção científica nacional nesta área e proporcionar um ambiente para a troca de experiências e idéias entre profissionais, estudantes, pesquisadores nacionais e estrangeiros. Em seu décimo primeiro ano, a versão 2000 do SBIE teve como tema principal "*As Tecnologias da Informação e Comunicação na Aprendizagem*".

A principal contribuição para a dissertação foi quando o grupo de pesquisa, no qual este trabalho está inserido, realiza algumas avaliações experimentais, usando métodos tradicionais para aplicações de cursos, exercícios e provas e sistema Hyper-Automaton semi-automatizado, projetado para curso, avaliação e exercícios, utilizando redes locais e a WEB. Essa experiência permitiu identificar as vantagens e restrições de trabalhar de forma independente. Porém, as restrições podem ser superadas quando integramos o curso com o sistema de avaliações de um modo cooperativo ao processo de ensino. Cabe reforçar que nessa integração os alunos trocam de módulos de acordo com a sua necessidade, e o professor vai tomar conhecimento quando receber o log dos alunos.

Neste capítulo são apresentados os Cursos no Hyper-Automaton, Exercícios e Provas no Hyper-Automaton, Provas e Exercícios Adaptativos no Hyper-Automaton Adaptação do Sistema com Geração Automática de Exercícios e Provas, Integração do Curso com Geração Automática de Exercícios e Provas e Conclusão.

Sistema Hyper-Automaton para Cursos e Avaliações Mediado por Computador Baseado em Autômatos Finitos

*Carlos Tadeu Q. de Moraes, Júlio P. Machado, Paulo B. Menezes, Ricardo A.L. Reis
SBIE2000*

6.1 Resumo

Este artigo tem por objetivo apresentar um sistema para geração de cursos, exercícios e avaliações interativas possivelmente adaptativas de alunos para o ensino mediado por computador. O sistema contempla o educador de maneira simples e flexível facilitando as suas atividades diárias, como uma ferramenta de apoio tanto para criação de um curso, montagem de lista de exercícios e prova, possibilitando também ao aluno obter automaticamente listas de exercícios específicas. A geração e montagem dos módulos de curso e avaliação são efetuadas de forma independente. A principal função é validar a modelagem e implementar um sistema semi-automático baseado em autômatos finitos com saída.

Palavras-chave: ensino a distância, exercícios e avaliação, WWW, autômatos.

7.2 Introdução

A produção de material instrucional para a WWW é uma tarefa dispendiosa. Para justificar sua construção é necessário ter-se a habilidade de poder utilizá-lo em muitas circunstâncias e dentro de cursos produzidos por diferentes professores. De forma similar, os componentes do *courseware* associado devem ser reutilizáveis sempre que possível. Neste artigo apresentaremos a validação do sistema Hyper-Automaton [MEN 2000] [MAC 99a] utilizando como modelo de estruturação de hiperdocumentos os fundamentos dos autômatos finitos com saída (máquinas de Moore e Mealy) [MOC 99b] para implementação de um protótipo que gera curso e exercícios/avaliação, bem como relatos de testes reais para geração de cursos e exercícios. O modelo é inspirado em pesquisas clássicas na área de hiperdocumentos e recentes iniciativas na WWW, com especial enfoque no desenvolvimento de sistemas de hipertexto onde a base de hiperdocumentos é projetada de forma independente da estrutura de controle da aplicação hipermídia. Essa solução facilita a implementação, manutenção e reuso dos componentes instrucionais. Neste modelo, visões sobre uma base de hiperdocumentos são autômatos com saída e *links* entre as páginas são transições entre os estados do autômato. O que caracteriza como um "curso", um conjunto de hiperdocumentos organizados no conceito de autômato, é a forma didática da apresentação da informação de acordo com premissas pedagógicas e motivacionais direcionadas a facilitar a aprendizagem. O artigo está organizado nas seguintes seções: seção 2 apresenta um modelo de curso implementado de forma semi-automático no Hyper-Automaton. Seção 3 Exercícios e Provas no Hyper-Automaton. Seção 4 Provas e Exercícios Adaptativos no Hyper-Automaton. Seção 5 Adaptação do Sistema com Geração Automática de Exercícios e Provas. Seção 6 Integração do Curso com Geração Simples e Automática de Exercícios e Provas.

7.3 Cursos no Hyper-Automaton

O sistema consiste de três componentes principais, além do servidor *WEB*: uma ferramenta de autoria de autômatos, uma interface de navegação e um controle de armazenamento. A ferramenta de autoria permite ao professor criar, modificar e reutilizar definições de autômatos [MAC 99]. Quando o professor grava uma definição de curso, um processo específico garante o armazenamento dos dados no servidor. Por fim, um usuário acessa os hiperdocumentos pelo módulo de navegação sobre autômatos. A interface do sistema está desenvolvida completamente em *HTML*, de modo que a sua utilização é feita via *Internet* usando um browser. O acesso fácil e multiplataforma que a *Internet* oferece ao usuário são um dos principais motivos para a escolha de uma interface via *WEB*. Essa característica faz com que o sistema possa ser utilizado a partir de qualquer computador conectado a uma rede com acesso ao servidor do *WEB*.

7.4 Exercícios e Provas no Hyper-Automaton

A criação do módulo de exercício e da prova serviu para dar suporte ao módulo de geração de curso do Hyper-Automaton que atendeu ao ensino de Informática Teórica na UFRGS, tendo como base à criação de uma ferramenta de fácil implementação, manutenção e reuso, independente de outros sistemas (como banco de dados, sistema operacional, etc.). Efetuamos um teste real do Hyper-Automaton em um ambiente local, onde foi testada uma lista de exercícios, junto aos cursos de graduação e pós-graduação nas disciplinas de Linguagens Formais e Projeto de Sistemas VLSI - UFRGS, como mostrado em [MOC 99a]. As *n*-uplas presentes nos autômatos para construções de exercícios e avaliações correspondem um hiperdocumento estruturado na *WEB*, de forma a serem vistos sob dois enfoques Máquinas de Mealy e Máquinas de Moore. Um autômato, com saída nas transições (Maquina Mealy), geram uma fita de saída para um novo estado (correspondente a uma pagina html), suas informações de saída são: número da questão, resposta e o gabarito. Na última transição de saída do autômato são armazenados os dados de saída em arquivo no servidor [MOC 2000], demonstrada na fig. 44.

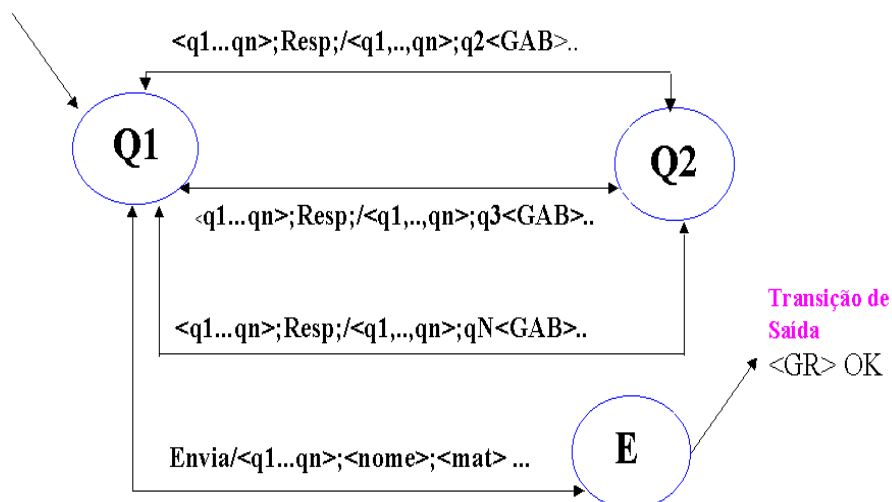


FIGURA 44 - Modelo de AFS nas Transições (Maquina Mealy)

7.5 Provas e Exercícios Adaptativos no Hyper-Automaton

O sistema Hyper-Automaton também fornece funcionalidades para a construção de avaliações adaptativas via *Internet*. [MEN 2000] [MOC 2000] [MOC 2000a] Nesse caso, entende-se como adaptativa, uma prova cuja seqüência de apresentação de questões é dependente da performance do aluno ao responder questões prévias. Esse é somente um modelo de prova que também poderia ser utilizado outras formas de representar, tal comportamento que pode ser modelado por um autômato onde as questões foram construídas com três níveis de dificuldades. O aluno responde primeiramente uma questão de nível médio e, caso acerte a resposta, o sistema providencia uma questão de nível mais alto. Caso contrário, terá uma chance de recuperar o conceito errôneo através de uma questão formulada especialmente para a recuperação. A construção do autômato é demonstrada em [MOC 2000a]. É importante destacar, neste caso, a diferença entre a modelagem como uma Máquina de Moore e como uma Máquina de Mealy. Note que uma Máquina de Mealy (fig. 45) é um formalismo mais adequado para o suporte a avaliações e exercícios. Um dos motivos está na explosão de estados que ocorre na Máquina de Moore, necessário para esconder do aluno o gabarito de respostas (o qual poderia ser deduzido pela análise dos estados destinos das transições), o que não ocorre na Máquina de Mealy, pois as saídas estão associadas às transições e não aos estados.

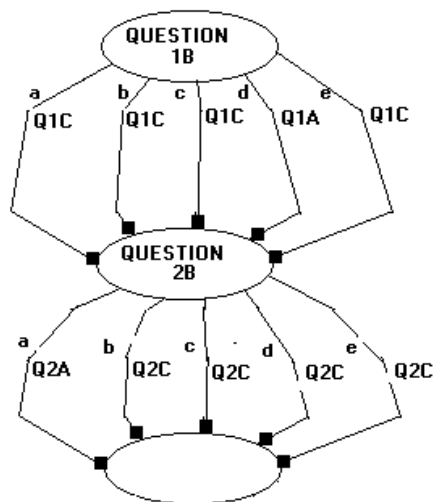


FIGURA 45 - Modelo de Prova Máquina Mealy

7.6 Adaptação do Sistema com Geração Automática de Exercícios e Provas

Durante o processo de teste e validação do sistema semi-automático do Hyper-Automaton, buscamos modelar um módulo de geração automática de exercícios e provas [MOC 2000a], entretanto procurou-se adaptar a ferramenta semi-automática de forma a permitir que qualquer usuário crie suas próprias listas de exercícios e provas de uma maneira automática. Ou seja, o módulo possibilitará que o usuário configure previamente parâmetros importantes como: número de questões, nível de dificuldade, porcentagem de questões por nível de dificuldade e o peso por questões, com o objetivo de gerar automaticamente o autômato para a respectiva prova/exercício (a qual pode ser modificada posteriormente pelo professor/aluno). Ao desenvolver uma modelagem de

geração automática de exercícios e provas aplicado ao autômato finito com saída, buscamos inovar as ferramentas de apoio ao ensino, com o objetivo principal de validar as técnicas de modelagem e desenvolver um sistema semi-automatizado para a aplicação de exercícios e provas on-line, com suporte a cursos na WEB. A ênfase deste módulo é desenvolver uma ferramenta prática de forma a facilitar o dia a dia do professor e do aluno, tanto na geração de listas de exercícios e provas como agilizando o processo de ensino. O professor tem à sua disposição uma interface capaz de aceitar parâmetros de entrada e fornecer como saída uma lista de exercícios ou prova utilizando meios gráficos ou textuais.

7.7 Integração do Curso com Geração Automática de Exercícios e Provas

Quando propomos uma ferramenta para geração de cursos, exercícios e provas, buscamos uma proposta flexível para uso imediato de maneira automática e semi-automática. O sistema Hyper-Automaton dá essa liberdade aos professores e alunos, para criarem e montarem seu próprio autômato, usando o banco de páginas para reuso e criação de novos autômatos, tanto para cursos e exercícios/provas de maneira automática e semi-automática. A integração dos módulos acontece de maneira transparente o uso do sistema, pois o modelo Hyper-Automaton fornece a base para o funcionamento de ambos módulos. Na geração automática de exercícios e provas o sistema Hyper-Automaton fornece informações do curso que possui disciplinas cadastradas, onde cada disciplina possui vários tópicos em que o módulo de geração automático pode obter questões para provas e exercícios. O professor, mediado por um computador local ou remoto, acessa um banco de questões relativas às suas disciplinas, permitindo gerar novas perguntas através de um editor de perguntas, descrito em [MOC 2000a]. Na análise funcional do sistema de Hyper-Automaton, os seguintes aspectos foram observados durante a resolução de um teste integrado com o curso: durante sua aplicação, o aluno utilizou-se do Curso (livro eletrônico) para consultas (utilizando o mecanismo de busca por índice). Na resolução dos exercícios/provas, notou-se que os alunos tiveram a necessidade de um bloco de rascunho, para circular e rabiscar sobre os enunciados das questões relevantes à prova. Como teste de aplicabilidade real do sistema de uso independente, aplicou duas provas: uma *on-line*, utilizando o livro eletrônico, e outra em papel [MOC 99b] utilizando o livro tradicional. Observou-se que: a busca manual do tópico do assunto relevante à questão e a procura automática pelo índice remissivo do livro eletrônico. Na prova on-line os alunos obtiveram o resultado da prova imediatamente, com o acesso às respostas corretas da prova. Após o teste do sistema Hyper-Automaton os professores obtiveram imediatamente uma avaliação do desempenho da turma, com o retorno automático dos resultados dos estudantes, permitindo ao professor iniciar imediatamente a recuperação dos conteúdos em que os alunos demonstraram um baixo aproveitamento. Por outro lado os alunos buscaram efetuar logo a revisão das questões erradas e revisão dos conteúdos.

7.8 Conclusão

Este trabalho apresentou uma alternativa de criação e modelagem de sistemas para o apoio à criação de cursos e avaliações (provas e exercícios) de forma automática e semi-automática na *WEB*, visando fornecer maior flexibilidade e facilidades a professores e alunos. A solução é suportada por uma arquitetura cliente-servidor modular, chamada de Hyper-Automaton, baseada em um modelo de organização de hiperdocumentos definido sobre o formalismo de Autômatos Finitos com Saída. O

modelo apresentado leva a um alto grau de modularização do material instrucional, apresentando as seguintes vantagens: facilidade de reuso de páginas em diversos cursos e avaliações, com eliminação da redundância; independência dos hiperdocumentos da estrutura do autômato, cuja alteração não influi nas páginas e vice-versa; permite que qualquer usuário crie *links* de e para qualquer documento; facilidade de implementação e manutenção; interface gráfica simples e direta; elaboração de seqüência instrucional com enfoques específicos e capazes de oferecer estudo individualizado; construção de caminhos adaptativos de navegação sobre hiperdocumentos; etc.

8 Conclusão

Esta última seção apresenta as conclusões obtidas com o desenvolvimento desta dissertação. Também são sugeridos trabalhos futuros que poderão ser de interesse para o aprofundamento do assunto e aproveitamento dos conceitos criados.

A evolução das idéias desenvolvidas ao longo desta dissertação ocorreu de acordo com a seguinte cronologia, com a publicação dos artigos abaixo:

1. **Study of Rooms for the Teaching Mediated by Computer.** International Conference on Engineering and Computer Education, 1999, **ICECE99**, Rio de Janeiro. [MOC 99].
2. **Exercícios e Avaliações Interativo Mediado por Computador Baseado em Sistema Formal.** Simpósio Brasileiro de Informática na Educação, 10. 1999 , **SBIE99** - Curitiba [MOC 99b].
3. **Avaliação Interativa de Alunos em Cursos na WEB Baseada em Sistemas Formais,** Semana Acadêmica do PPGC, 4., 1999, Porto Alegre [MOC 99a].
4. **A WEB Teaching System Based on Formal Methods,** IFIP WCC - Educational Uses of Communication and Information Technologies.,16.,2000. **ICEUT2000** – China [MOC 2000].
5. **Modelagem para Geração Automática de Exercícios e Avaliações.** International Conference on Engineering and Computer Education, 2000, **ICECE2000** - São Paulo, [MOC 2000a].
6. **Structuring WEB Course Pages as Automata: revising concepts.** Recherche d'Informations Assistee par Ordinateur 2000, Conference on Content-based Multimedia Information Access, 6., 2000, Paris [MAC 2000]
7. **Autômatos Finitos: um formalismo para cursos na WEB.** Simpósio Brasileiro de Engenharia de Software, 13., 1999 Florianópolis. [MAC 99a]
8. **Computer in Classroom: Impact on the Environment and on the Relationships.** International Conference on Engineering and Computer Education, 1999, Rio de Janeiro. [OLI 99].
9. **Sistema Hyper-Automaton para Cursos e Avaliações Mediado por Computador Baseado em Autômatos Finitos.** Simpósio Brasileiro de Informática na Educação, 11. 2000 Maceió [MOC 2000b].

8.1 Principais Propostas da Dissertação

Foi apresentada nesta dissertação uma estrutura e modelagem de sistemas para o apoio à geração de exercícios/provas na WEB de forma interativa, utilizando os conceitos do autômato finito com saída. O objetivo do sistema Hyper-Automaton é fornecer maior flexibilidade de forma consistente e capaz de suportar uma arquitetura cliente-servidor modular, baseado nos sistemas de apoio à informação organizacional distribuídos através da Internet, definido sobre o formalismo de Autômatos Finitos com Saída, utilizados como ferramenta de apoio aos alunos e professores.

Para que todo esse **processo** aconteça no plano operacional, há necessidade de que, a partir de **fundamentos teóricos** claramente identificados e de **mecanismos de exercícios e avaliações** coerentes e eficazes, se promova à necessária mudança de consciência, de métodos, de instrumentos e de perspectivas. Isto implica, além de **análise crítica**, capacidade de formular **propostas alternativas**. Neste sentido, esta é uma proposta alternativa, que, aplicável para o **ensino**, e para um projeto **EAD**, pode ser considerada também uma visão de avanço do aprendizado, e uma alternativa para contribuir na busca de novos **caminhos** no contexto da **pesquisa e inovação**.

Nem esta dissertação, nem seu autor têm a pretensão de antecipar ou definir uma nova solução para avaliação ou dar uma receita final para a superação dos métodos aplicados até hoje. No entanto, em consequência da inovação tecnológica e com a velocidade da transformação das coisas, no contexto do ensino, uma nova ferramenta necessariamente virá. Sob o aspecto teórico e de sua aplicação prática, esta dissertação constitui, isto sim, uma proposta de interpretação da utilização da ferramenta interativa de apoio, para os professores e alunos e um esforço de identificar seus componentes essenciais, pela raiz, e não apenas conjunturais conseqüentes ou aparentes. Como decorrência, vários testes on-line foram realizados, busca igualmente, refletir e propor à reflexão. Ainda que se possa discordar, pontualmente, de caminhos apontados, deve-se registrar o mérito da proposta apresentada e a dimensão corajosa de proporções das publicações que o autor apresentou durante o período do desenvolvimento do Hyper-Automaton, tocante a exercícios e avaliações. O objetivo nem é fazer sínteses completas, nem obter consensos ou unanimidades, mas levar à reflexão e ao debate de uma proposta alternativa aos modelos em uso, onde valida a modelagem de autômato de forma consistente e completa, através do uso apropriado do formalismo proposto. Também, este processo de amadurecimento foi tomando forma concreta ao autor nos últimos anos, a partir dos primeiros testes on-line realizados. Muitos debates aconteceram, desde então através de exposições, seminários e palestras em universidades, centros de estudos e outros ambientes, surgiram a modelagem para geração automática para exercícios e avaliação, bem como a técnica de Engenharia de Software. A verdade, que esta dissertação, pretende ser fruto e soma de todos esses elementos, que fazem a essencialidade do ensino, de seus processos e de suas circunstâncias. Esta é sua qualidade maior, acredito, propor uma ferramenta interativa para exercício e avaliação como base no autômato finito embora, isso pareça ser seu maior desafio.

8.1.1 Conceituação de Hyper-Automaton

O sistema Hyper-Automaton foi projetado para a construção de hipertextos fortemente hierárquicos de forma a evitar o excesso de liberdade e falta de profundidade do aluno quando explorando uma base de hiperdocumentos. Dessa forma, os autômatos finitos com saída cumpriram com eficiência o papel de mecanismo de restrição e estruturação dentro do modelo proposto, inclusive mostrando-se uma ferramenta mais promissora que sistemas similares disponíveis na Internet.

Esse ambiente apresenta características próprias que condicionam o desenvolvimento e, portanto, a modelagem de sistemas. Essas características são a estrutura hipertextual e as ferramentas próprias de implementação de soluções.

A estrutura hipertextual de um Hyper-Automaton define sua interface com o usuário e a localização de sub-sistemas. A navegação sobre essa estrutura define como o usuário interage com o sistema, seus caminhos de acesso, o acesso a dados e o disparo de ações e transações. Enfim, é através da navegação que o usuário utiliza o Hyper-Automaton, e a estrutura hipertextual deve ser bem planejada e especificada.

Sistemas implementados na Internet devem também se adaptar às novas características das ferramentas disponíveis para desenvolvimento nesse ambiente. Documentos HTML e XML, *links* de navegação, linguagens de *script* de cliente e servidor, *applets* Java e servidores de aplicação possuem peculiaridades que devem ser refletidas em componentes de modelagem adequados.

Em Hyper-Automaton, documentos organizacionais se beneficiam das possibilidades abertas pela tecnologia e se tornam estruturados e dinâmicos. A especificação estrutural dos documentos de um sistema permite o reuso e relacionamentos entre documentos e subdocumentos, modularizando a informação, melhorando a organização do sistema e facilitando o reuso de informações.

O comportamento dinâmico dos documentos é definido pela montagem dos autômatos. Esses autômatos são ativados por eventos como a abertura de um documento, a ativação de um *link* de navegação, o acionamento de um elemento de interface, etc. Autômatos permitem a integração transparente de documentos e bancos de dados, garantindo a sincronia e atualização de informações.

8.1.2 Razões para usar o Hyper-Automaton:

- Para apoiar à criação de cursos, exercícios e avaliação na WEB;
- Por não possuir links fixos;
- Por possuir uma estruturação de links lógicos como componentes externos aos documentos HTML;
- Por permitir o reuso de estruturas existentes (links, nodos, conteúdos e exercícios/provas) através da definição de múltiplos autômatos sobre o mesmo conjunto de materiais;
- Por permitir aos usuários criarem links e alterações do código HTML sem a necessidade de autorização;
- Por ser livre de criação, possibilitando ao usuário construir seu próprio curso e avaliações, principalmente sua estrutura de navegação lógica;
- Por permite a criação de provas adaptativas, possibilitando uma melhor forma de avaliação do aluno.

8.1.3 Abordagem Proposta em Hyper-Automaton

Para auxiliar no desenvolvimento de sistemas com essas características, uma metodologia de desenvolvimento apropriada é importante. **Para alcançar esse objetivo, o Hyper-Automaton estendem técnicas OO**, particularmente a linguagem UML, com novos conceitos fundamentais **para a especificação de Geração Automática** de forma integrada e consistente:

- características específicas de ferramentas de desenvolvimento para Internet;
- conceitos de gerência de documentos estruturados e dinâmicos;

- conceitos e funcionalidade de sistemas hipermídia integrados a sistemas de informação;
- conceitos e funcionalidade de sistemas de *workflow* de documentos.

A integração das características usuais de sistemas de informação convencionais com esses conceitos específicos da geração automática resultará de forma bastante intuitiva.

Existem três pontos principais de início do processo de modelagem:

- Casos de Uso;
- Documentos;
- Banco de Dados.

É aconselhável iniciar por *Casos de Uso*, pois é uma técnica apropriada para análise de requisitos com o cliente/usuário. A partir da definição da funcionalidade do sistema através dos Casos de Uso.

Documentos e bancos de dados são então planejados para dar suporte à implementação das funções e à representação da informação necessária. Contextos Navegacionais podem então ser definidos para especificar visões de cada documento apropriadas à realização de cada tarefa e ao tipo de usuário envolvido. Com base na estrutura dos documentos e nos contextos, é definida a estrutura de navegação do sistema, especificando o relacionamento entre documentos e os caminhos de acesso às funções.

Estando definidas as estruturas de dados e de navegação do sistema, serão componentes executáveis são introduzidos para especificar onde serão implementados o comportamento dinâmico de documentos e links e como será realizada a integração entre documentos e bancos de dados. Essa idéia mais acabada do comportamento dinâmico de documentos possibilita a definição do seu ciclo de vida, especificando em qual estado cada documento pode se encontrar em um dado momento.

Nesse ponto, toda a estrutura estática do sistema está modelada. Parte-se, então, para a definição do *workflow* existente, especificando tarefas, seqüência de execução, documentos e atores envolvidos (professor/aluno). Dependendo do sistema, um modelo de primeiro nível pode especificar a seqüência de execução de cada módulo, como no exemplo apresentado. Em níveis mais detalhados, cada um pode ter seu próprio modelo de *workflow*, ou um modelo pode englobar mais de um Caso de Uso. Cada tarefa definida no *workflow*, cada Caso de Uso ou ainda outras funções relevantes são, então, detalhadas na modelagem de transações, que especifica interações típicas para a execução de tarefas no sistema.

Por fim, um modelo de Implantação define a localização física dos componentes do sistema. Nesse modelo, documentos, componentes executáveis, bancos de dados, servidores de aplicações e sistemas legados podem ser alocados a diferentes computadores, especificando a topologia do sistema.

8.1.4 Ambiente de Desenvolvimento Baseado em Hyper-Automaton

Para comprovar a sua viabilidade, foi também apresentada uma proposta de ambiente integrado de desenvolvimento de Curso e Exercícios/Avaliação baseado em Autômatos Finitos com saídas. Essa proposta especifica algoritmos para geração automática de documentos e componentes a partir dos modelos, abordando todos os aspectos envolvidos na implementação de sistemas para a WEB.

O ambiente proposto facilita a montagem do autômato porque é capaz de gerar automaticamente uma grande parte dos links lógicos necessários para a sua execução. Hyper-Automaton propicia isso porque suas estruturas possuem as informações necessárias para essa geração tanto para Máquinas de Moore e de Mealy, pois são explicitamente voltados para a modelagem de sistemas para a WEB. A integração entre os módulos permite o reuso de documentos entre o curso e exercícios/avaliação. Para implantação de componentes e bancos de dados baseados em propriedades do ambiente e dos modelos e em arquiteturas abertas e padrões, faz com que o programador consiga implementar completamente o Hyper-Automaton a partir de um único ambiente, de forma independente de outros sistemas utilizados.

8.2 Benefícios do Hyper-Automaton

O maior benefício propiciado pela metodologia Hyper-Automaton é a modelagem integrada de sistemas de informação baseados em documentos para ambiente Internet. A união dos conceitos fundamentais da informática teórica, especificamente o autômato finito com saída (Máquinas Moore/Mealy) e as tecnologias que compõem o campo do desenvolvimento para a *WEB*, possibilitam a especificação completa e consistente de todos os aspectos importantes desse tipo de sistema. Como resultado dos testes pilotos, foi aprovado o uso desta modelagem e tem-se o projeto e a implementação de sistemas facilitados. Esta modelagem pode servir como base para o desenvolvimento do módulo do sistema de geração automática como componente do Hyper-Automaton, pois será capaz de integrar todos os seus aspectos.

A decisão de estender a UML através dos conceitos existentes nas diversas tecnologias que compõem o Hyper-Automaton faz com que seja uma notação adequada para o entendimento do problema a ser modelado. Em outras palavras, os diagramas propostos, que têm como objetivo explicitar a modelagem de determinadas visões do sistema, possibilitam o projetista ter uma boa percepção de como os conceitos estão sendo utilizados. Por outro lado, Hyper-Automaton também se baseia nas ferramentas próprias para modelagem de sistemas para a Internet, fazendo com que a notação também seja adequada para a especificação da implementação do módulo de geração automática. Em suma, o Hyper-Automaton reúne qualidades tanto para modelagem conceitual quanto para utilização da implementação.

Outra contribuição importante deste trabalho é na direção da modelagem de sistemas baseados em autômatos finitos, nos quais esses autômatos são mais que interface para dados em bancos de dados. Na metodologia Hyper-Automaton, o grande poder obtido pelo uso da tecnologia de hipertexto por causa das vantagens da Internet é combinado às potencialidades organizacionais da gerência de documentos. Essa união possibilita a disseminação de princípios de controle de documentos. Também é

importante a especificação das interações entre documentos e bases de dados, pois ambos representam formas de armazenamento idealmente ortogonais em sistemas baseados em documentos. Além disso, dados exibidos em documentos muitas vezes têm origem em consultas a bases de dados. Por serem construídas na Internet com o uso de padrões abertos, soluções baseadas nesta metodologia são, também, portáteis, expansíveis e integráveis com outros sistemas, protegendo investimentos já realizados e permitindo novas escolhas no futuro.

As características das arquiteturas abertas disponíveis para desenvolvimento em Internet também são abordadas, colaborando para a sua adoção e para a especificação de sistemas que aproveitam suas vantagens. A proximidade de elementos de modelagem e ferramentas de implementação auxilia a validação da metodologia e a formalização de técnicas de desenvolvimento para *WEB*. Essa proximidade também contribui para a construção de ferramentas de geração automática de código.

8.3 Trabalhos Futuros

Apesar de já ter sido testada em alguns casos reais, a utilização da metodologia deve ser intensificada para melhor avaliação e refinamento. Para isso, outros testes devem ser realizados com o uso de Hyper-Automaton, tanto no sistema existente, para testar a capacidade da metodologia proposta, quanto em novos, para explorar possíveis especificações não apresentadas até o momento. Essa ampliação de uso pode dar-se através da experimentação em disciplinas de graduação ou em projetos específicos do Laboratório de Fundamentos da Computação.

Entretanto, um dos módulos importantes do trabalho e o que pode ser realizado a partir desta dissertação é a implementação de um sistema de geração automática e semi-automática de exercícios e avaliação. O capítulo 6 apresenta uma proposta para este fim, oferecendo soluções para integração entre os módulos especificados no Hyper-Automaton.

No contexto da exploração de novas tecnologias de sistemas hipertexto e aplicações de ferramentas para trabalho colaborativo entre professores e alunos, as ferramentas integram algumas funcionalidades, propiciando aos autores, a possibilidade de desenvolverem seu próprio estilo. Assim, os conjuntos de ferramentas, atuam como uma camada que separa os usuários da tecnologia. Com o objetivo de minimizar as limitações impostas pela linguagem HTML, o sistema Hyper-Automaton passará por uma transformação do módulo de sistema de geração automático e semi-automático de exercícios e avaliações [MOC 2000b] para linguagem XML. Portanto, o XML ao contrário de HTML, permite criar novas *tags* (novas gramáticas) em um documento, o que o torna capaz de se autodescrever. [BOS 97], [HOL 98]. Outra importante diferença entre HTML e XML é o propósito a que servem as *tags* de cada linguagem. Em HTML, elas são usadas para especificar o formato de apresentação do texto (tamanho e cor da letra, por exemplo), enquanto em XML elas descrevem o significado semântico do texto. Essa capacidade do XML torna a especificação dos dados muito mais formal, possibilitando a interpretação de um arquivo por qualquer ferramenta. É possível usar *tags* HTML em documentos XML. As *tags* podem ser definidas manualmente em uma DTD (Document Type Definition) feita pelo usuário ou pode ser usada uma pronta, disponibilizada por terceiros [XML 98]. O sistema tem como objetivo a exploração de textos didáticos, exercícios e testes de avaliação especificados para o padrão XML. Na

figura 46, pode-se visualizar a integração das ferramentas, as quais são armazenadas no servidor HTTP na forma de arquivos HTML. A arquitetura é estendida com a ferramenta que permite a construção do autômato (exercícios e exames) e apresentadas como documentos XML com Java e JavaScript. Assim o Hyper-Automaton associa, a cada sessão de exercícios, um documento estruturado com informações pertinentes a lista de exercícios interagindo com a ferramenta HTML.

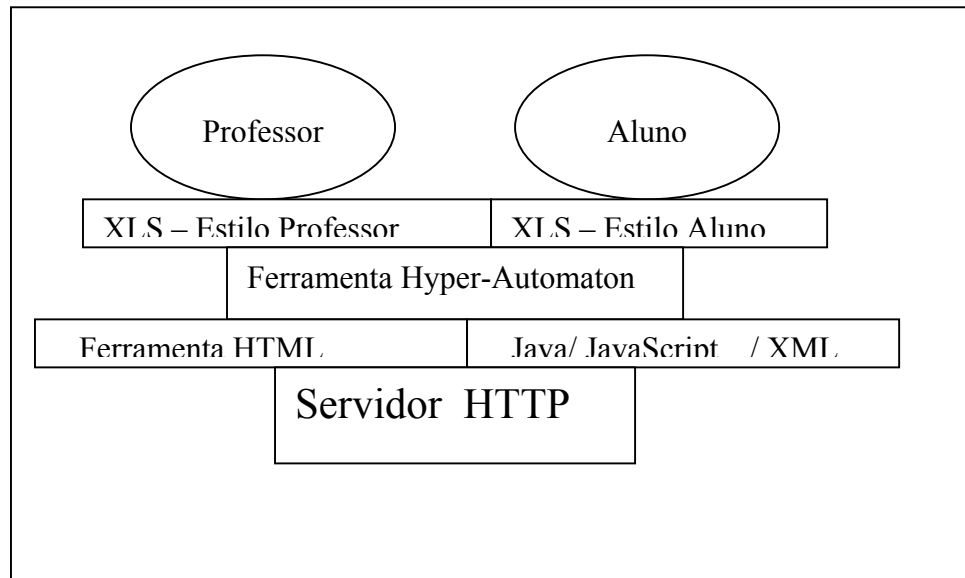


FIGURA 46 - Integração das Ferramentas.

A modelagem de documentos fica mais simplificada pelas possibilidades abertas pelo DTD. Atributos e métodos de documentos podem ser especificados em um nível não muito alto de abstração, o que facilita a geração automática de código. Além disso, o uso de modelagem orientada a objetos se torna bem mais apropriada e intuitiva, mesmo se tratando de um documento texto. [LIM 99]

Bibliografia

- [BAI 91] BAIRD, P. et al. Hypertext and Added Value. In: **Hypermedia/Hypertext and Object-Oriented Databases**. London: Chapman & Hall, 1991. p.71-90.
- [BAR 90] BARR, M.; WELLS, C. **Category Theory for Computing Science**. Englewood Cliffs: Prentice Hall, 1990.
- [BEN 99] BENNETT, F. **Computadores como Tutores: Resolvendo a Crise em Educação**. 1999. Disponível em: <<http://www.concentric.net/~Fabem1>> Acesso em: 2001.
- [BOR 98] BORONI, C. et al. A Paradigm Shift! The Internet, the WEB, Browsers, Java and the Future of Computer Science Education. **SIGCSE Bulletin**, New York, v.30, n.1, p.145-152, 1998.
- [BOS 97] BOSAK, Jon. **XML, Java, and the Future of the WEB**. Disponível em: <<http://www.xml.com/xml/pub/w3j/s3.bosak.html>>. Acesso em: jun. 1999.
- [BRU 94] BRUILLARD, Eric; VIVET, Martial. In: BALACHEFF, Nicolas; VIVET, Martial. **Didactique et Intelligence Artificielle**. Grenoble: La Pensée Sauvage, 1994. Concevoir des EIAO pour des situations scolaires approche méthodologique.
- [BUS 45] BUSH, V. As We May Think. **Atlantic Monthly**,[S.I.], p.101-108, 1945.
- [CAN 96] CANDAU, Vera Maria. Pluralismo cultural, Cotidiano Escolar e Formação de Professores. In: ENCONTRO NACIONAL DE DIDÁTICA E PRÁTICA DE ENSINO, 8., 1996, Florianópolis. **Formação e Profissionalização do Educador: anais**. Florianópolis: Universidade Federal de Santa Catarina,1996. p. 295-304.
- [CON 87] CONKLIN, Jeff. Hypertext: An Introduction and Survey. **IEEE Computer**, New York, v.20, n.9, p. 17-41, Sept. 1987.
- [DIV 99] DIVERIO, Tiarajú A.; MENEZES, Paulo F. B. **Teoria da Computação: Máquinas Universais e Computabilidade**. Porto Alegre: Sagra-Luzzato, 1999.
- [DWI 96] DWIGHT, Jeffrey; ERWIN, Michael. **Special Edition Using CGI**. Indianapolis: Que Corp., 1996.
- [ESC 99] A ESCOLA do Futuro. São Paulo: Centro de Capacitação Profissional, Centro Universitário Maria Antônia Terceira Geração, 1999. Disponível em: <<http://www.futuro.usp.br/ef/menu/menu.htm>>. Acesso em: 1999.
- [FUR 89] FURUTA, Richard; STOTS, P. D. Programmable Browsing Semantics in Trellis. In: ACM CONFERENCE ON HYPERTEXT, 2., 1989, Pittsburgh. **Proceedings...** New York: ACM Press, 1989. p.27-42.
- [GAF 93] GARZOTTO, F. et al. HDM: A Model Based Approach to Hypertext Application Design. **ACM Transactions on Information Systems**, New York,v.11, n.1, p.1-26,1993.
- [GAM 95] GAMMA, E. et al. **Design Patterns – Elements of Reusable Object-Oriented Software**. Reading: Addison Wesley Longman, 1995.
- [GAR 97] GARFINKEL, Simson; SPAFFORD, Gene. **WEB Security & Commerce**. Cambridge: O’Reilly & Associates, 1997. 483p.
- [GAR 97a] GARRIDO, Alejandra et al. Disponível em:

- <<http://jerry.cs.uiuc.edu/plop/plop97/Workshops.html>> Acesso em: 1997.
- [HEA 97] HEATHER, M. A.; ROSSITER, B. N. Object-awareness in multimedia documents. In: INTERNATIONAL WORKSHOP PRINCIPLES OF DOCUMENT PROCESSING, 3., 1997. **Proceedings...** Palo-Alto: Springer-Verlag, 1997. (Lecture Notes in Computer Science, 1293).
- [HOL 98] HOLZSCHLAG, E. **Special Edition Using HTML 4**. 5th ed. [S.l.]: Que Education & Training, 1998.
- [HOP 79] HOPCROFT, J.; ULMANN, J. **Introduction to Automata Theory, Languages and Computation**. Massachusetts: Addison-Wesley, 1979.
- [ISA 95] ISAKOWITZ, T.; STOHR, E.; BALASUBRAMANIAM, P. RMM, a Methodology for Structured Hypermedia Design. **Communications of the ACM**, New York, v.38, n.8, Aug. 1995.
- [KAW 96] KAWASAKI, E. I.; FERNANDES, C. T. Modelo para Projeto de Cursos Hiperídia. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCACAO, 7., 1996, Belo Horizonte. **Anais...**Belo Horizonte: SBC, UFMG, 1996. p.227-240.
- [LIM 99] LIMA, Flávio Azevedo de; PRICE, R. T. Defining UML Stereotypes for Modelling Document-based WEB Information Systems. In: SEMISH – SEMINÁRIO INTEGRADO DE SOFTWARE E HARDWARE, 26., 1999. **Anais...** Rio de Janeiro: [s.n.], 1999.
- [MAC 2000] MACHADO, Júlio Pereira; MORAIS, Carlos Tadeu Q. de; MENEZES, Paulo Blauth; REIS, Ricardo. Structuring WEB Course Pages as Automata: revising concepts. In: RECHERCHE D'INFORMATIONS ASSISTEE PAR ORDINATEUR, 2000, Paris. **Content-Based Multimedia Information Access**: conference proceedings. Paris: Centre de Hautes Etudes Internationales d'Informatique Documentaires, Center for the Advanced Study of Information Systems, 2000. v.1, p.150-159.
- [MAC 2000a] MACHADO, Júlio Pereira. **Hyper-Automaton**: hipertextos e cursos na web usando autômatos finitos com saída. 2000. Dissertação (Mestrado em Ciencia da Computacao) – Instituto de Informatica, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [MAC 98] MACHADO, Júlio Pereira; MENEZES, Paulo Blauth. Sistemas de Gerenciamento para Ensino a Distância. In: CONGRESSO INTERNATIONAL DE EDUCACÃO A DISTÂNCIA, 5., 1998, São Paulo. **Anais...** São Paulo: ABED, 1998. Disponível em: <<http://www.abed.org.br/artigos2/artigos/24/sgead971.html>>. Acesso em: 1998.
- [MAC 98a] MACHADO, Júlio Pereira. **Sistemas de Gerenciamento para Ensino a Distância**. 1998. Trabalho Individual (Mestrado em Ciencia da Computacao) – Instituto de Informatica, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [MAC 99] MACHADO, Júlio Pereira; MENEZES, Paulo Blauth; PENCZEK, Leonardo. A System for WEB Based Instruction Using Sequential Automata. In: INTERNATIONAL CONFERENCE ON ENGINEERING AND COMPUTER EDUCATION, 1999, Rio de Janeiro. **Proceedings...** Rio de Janeiro: IEEE, 1999. p.187-191.

- [MAC 99a] MACHADO, Júlio Pereira; PENCZEK, Leonardo; MORAIS, Carlos Tadeu Q. de; MENEZES, Paulo Blauth. Autômatos Finitos: um formalismo para cursos na WEB. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 13., 1999, Florianópolis. **Anais...** Florianópolis: UFSC: Instituto de Informática da UFRGS, 1999. p.213-223.
- [MAD 94] MAÇADA, Débora Laurino. Software no Ensino. In: COMPÊNDIO de Trabalhos sobre Informática na Educação. Porto Alegre: CPGCC-UFRGS, 1994. v.1.
- [MED 98] MENASCÉ, Daniel A. Educational Challenges and Opportunities in the WEB Era. In: WORKSHOP SOBRE EDUCAÇÃO EM INFORMÁTICA, 6.; CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 18., 1998, Belo Horizonte. **Anais...** Belo Horizonte: SBC: UFMG, 1998. p.433-444.
- [MEN 2000] MENEZES, Paulo Blauth. **Linguagens Formais e Autômatos**. 3.ed. Porto Alegre: Sagra Luzzato, 2000.
- [MEN 98] MENEZES, Paulo Blauth; DIVERIO, Tiarajú A. et al. Desenvolvimento da Área Formal da Computação no Instituto de Informática da UFRGS. In: WORKSHOP DE MÉTODOS FORMAIS, 1., 1998, Porto Alegre. **Anais...** Porto Alegre: Instituto de Informática da UFRGS, 1998. p.1-12.
- [MEN 98a] MENEZES, Paulo Blauth; SERNADAS, A.; COSTAS, J. F. Nonsequential Automata Semantics for a Concurrent Object-Based Language, Revised and Extended Version. **Electronic Notes in Theoretical Computer Science**, n.14, 1998. Disponível em: <<http://www.elsevier.nl/locate/entcs/volume14.html>>. Acesso em: jun. 1999.
- [MEN 99] MENEZES, Paulo Blauth. Mathematical Methods in the Computing Institute of UFRGS. In: INTERNATIONAL CONFERENCE ON ALGEBRAIC METHODOLOGY AND SOFTWARE TECHNOLOGY, 7., 1999, Rio de Janeiro. **Proceedings...** [S.l.: s.n.], 1999.
- [MEN 99a] MENEZES, Paulo Blauth; MACHADO, Júlio Pereira. WEB Courses are Automata: a Categorical Framework. In: WORKSHOP ON FORMAL METHODS, 2., 1999, Florianópolis. **Proceedings...** Florianópolis: UFSC: Instituto de Informática da UFRGS,SBC, 1999. p.79-88.
- [MOC 2000] MORAIS, Carlos Tadeu Q. de; MACHADO, Júlio Pereira; MENEZES, Paulo Blauth; REIS Ricardo. A WEB Teaching System Based on Formal Methods. In: ICEUT: INTERNATIONAL CONFERENCE ON EDUCATIONAL USES OF COMMUNICATION AND INFORMATION TECHNOLOGIES; IFIP WORLD COMPUTER CONGRESS,16.,2000,Beijing. **Proceedings...** Beijing – China: IFIP, 2000. p.221-224.
- [MOC 2000a] MORAIS, Carlos Tadeu Q. de; MENEZES, Paulo Blauth; REIS Ricardo. Modeling for Automatic Generation of Exercises and Evaluations. In: ICECE INTERNATIONAL CONFERENCE ON ENGINEERING AND COMPUTER EDUCATION, 2000, São Paulo. **Cooperative Network for Engineering and Computer Education Development: Proceedings...** São Paulo SENAC/SP, IEEE. 2000. 1

- CD.
- [MOC 2000b] MORAIS, Carlos T. Q. de; MACHADO, J. P.; MENEZES, Paulo B.; REIS Ricardo. Sistema Hyper-Automaton para Cursos e Avaliações Mediado por Computador Baseado em Automatos Finitos. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 11., 2000, Alagoas. **Anais...** Maceio: SBC, 2000. p.419-422
- [MOC 99] MORAIS, Carlos Tadeu Q. de; MENEZES, Paulo Blauth; MACHADO, Júlio Pereira. Study of Rooms for the Teaching Mediated by Computer. In: ICECE/IEEE - INTERNATIONAL CONFERENCE ON ENGINEERING AND COMPUTER EDUCATION, 1999, Rio de Janeiro **Proceedings...** Rio de Janeiro: IEEE, 1999. p.395-398.
- [MOC 99a] MORAIS, Carlos Tadeu Q. de; MENEZES, Paulo Blauth; REIS Ricardo. Avaliação Interativa de Alunos em Cursos na WEB. In: SEMANA ACADÊMICA DO PPGC,4., 1999, Porto Alegre. **Anais...** Porto Alegre: PPGC da UFRGS, 1999. p.230-233.
- [MOC 99b] MORAIS, Carlos Tadeu Q. de; MACHADO, Júlio Pereira; MENEZES, Paulo Blauth; NOGUEIRA Mauro. Exercícios e Avaliação Interativa Mediada por Computador Baseada em Sistemas Formais. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO,10., 1999, Curitiba. **Anais...** Curitiba: SBC, 1999. p.349-356.
- [MOE 98] MOESCH, Thiago F.; LUTZ, Fábio; DIVÉRIO, Tiarajú A. et al. Simuladores de Máquinas Abstratas: ferramentas de apoio ao ensino de informática teórica. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 9., 1998, Fortaleza. **Anais...** Fortaleza: SBC, 1998.
- [MOL 98] MOREAU, Luc; HALL, Wendy. On the Expressiveness of Links in Hypertext Systems. **The Computer Journal**, Oxford, v.41, n.7, p.459-473, 1998.
- [MOM 97] MORRISON, Michael et al. **Java 1.1 Unleashed**. 3rd ed. Indianapolis: Sams.net, 1997. 1469p.
- [MOR 97] MORGAN, Bryan. **CORBA Meets Java**. Disponível em: <<http://www.javaworld.com>>. Acesso em: out. 1997.
- [NEW 96] NEWMAN, Alexander et al. **Special Edition - Using Java**. Indianapolis: Que Corporation, 1996. 869p.
- [NIE 90] NIELSEN, Jakob. **Hypertext and Hypermedia**. San Diego: Academic Press, 1990. 268p.
- [OIN 97] OINAS-KUKKONEN, Harris. Embedding Hypermedia into Information Systems. In: HAWAII INTL. CONFERENCE ON SYSTEMS SCIENCE, 30., 1997. **Proceedings...** Maui: IEEE Computer Society Press, 1997.
- [OLI 99] OLIVEIRA, Mauro Nogueira; MACHADO, Júlio Pereira; MORAIS, Carlos Tadeu Q. de; MENEZES, Paulo Blauth. Computer in Classroom: Impact on the Environment and on the Relationships. In: INTERNATIONAL CONFERENCE ON ENGINEERING AND COMPUTER EDUCATION,12., 1999, Rio de Janeiro. **Technology-based Education and Globalization**. [S.l.:s.n.], 1999. 1 CD.
- [ORF 97] ORFALI, Robert; HARKEY, Dan; EDWARDS, Jeri. CORBA, Java, and the Object WEB. **BYTE Magazine**, [S.l.], Oct. 1997.
- [ORT 95] ORTIGOSA, Alvaro Manuel. **Proposta de um Ambiente Adaptável**

- de Apoio ao Processo de Desenvolvimento de Software.** 1995. Dissertação (mestrado em Ciencia da Computacao) - Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [PEC 2000] PENCZEK, Leonardo – **HYPER-AUTOMATON** : Implementação e uso. 2000. Projeto de diplomação (Bacharelado em Ciência da Computação) - Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [PER 92] PERIN, Marcelo Gattermann. **Um Sistema de Gerenciamento de Hiperdocumentos para Ambientes de Desenvolvimento de Software.** 1992. Dissertação (Ciência da Computação) - Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [RAP 90] RAPKIEWICZ, Clevi Elena. AA Informatização do Professor no Processo de Informatização da Escola. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 1.,1990,Rio de Janeiro. **Anais...** Rio de Janeiro: SBC,1990.
- [ROS 97] ROSSI, Gustavo; SCHWABE, Daniel; GARRIDO, Alejandra. Design Reuse in Hypermedia Applications Development. In: ACM CONFERENCE ON HYPERTEXT, 1997. **Proceedings...** Southampton: [s.n.], 1997.
- [SCH 95b] SCHWABE, D.; ROSSI, G.; BARBOSA, S. **Systematic Hypermedia Application Design with OOHDM.** Disponível em: <<http://www.inf.puc-rio.br/~schwabe/>>. Acesso em: 1995.
- [SIM 99] COSTA, Simone A.; MACHADO, Julio P.; MENEZES, Paulo Baluth. Teoria das categorias: experiência e proposta de ensino. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO,10., 1999, Curitiba. **Anais...** Curitiba: SBC, 1999. p. 389-391.
- [STO 89] STOTTS, P. D.; FURUTA, R. Petri-Net-Based Hypertext: Document Structure with Browsing Semantics. **ACM Transactions on Information Systems**, New York, v. 7, n. 1,p.3-29, Jan. 1989.
- [STO 98] STOTTS, P. et al Hyperdocuments as Automata: Verification of Trace-Based Browsing Properties by Model Checking.**ACM Transactions on Information Systems**, New York, v.16, n.1, p.1-30, 1998.
- [SUN 98] SUN MICROSYSTEMS. **Java WEB Server - Servlet Tutotial.** Disponível em: <<http://jserv.java.sun.com> >. Acesso em: ago. 1998.
- [THO 96] THOMPSON, J. et al. Disponível em: <<http://www.cti.ac.uk/publ/actlea/issue4/thompson/thompson1.html>>. Acesso em: 1996.
- [TIT 96] TITTEL, Ed; GAITHER, Mark; HASSINGER, Sebastian; MIKE, Erwin. **World Wide WEB com HTML e CGI.** São Paulo: Berkeley Brasil, 1996. 524p.
- [TOM 89] TOMPA, F. W. M. A Data Model for Flexible Hypertext Database Systems. **ACM Transactions on Information Systems**, New York, v.7, n.1, p.85-100, Jan. 1989.
- [VAR 91] VARELA, Francisco J.; THOMPSON, Evan; ROSCH, Eleanor. **The embodied mind: Cognitive science and human experience.** Cambridge, MA, USA: MIT Press, 1991. Disponível em: <http://cogWEB.english.ucsb.edu/Abstracts/Varela,Thompson,Rosch_91.html>. Acesso em: 1991.
- [WAN 98] WANG, Weigang; RADA, Roy. Structured Hypertext with Domain Semantics. **ACM Transactions on Information Systems**, New York,

- v. 16, n. 4, p.372-412, Oct. 1998.
- [WEB 98] WEBER, Taisy et al. Uma Experiência com Hiperdocumentos e Internet no Suporte a Disciplinas de Computação. In: WORKSHOP SOBRE EDUCAÇÃO EM INFORMÁTICA, 6.; CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 18., 1998, Belo Horizonte. **Anais...** Belo Horizonte: SBC: UFMG, 1998. p.532-545.
- [WII 97] WIIL, Uffe; LEGGET, John. Hyperform: A Hypermedia System Development Environment. **ACM Transactions on Information Systems**, New York, v. 15, n. 1, p.1-31, Jan. 1997.
- [WIN 87] WINSKEL, G. Petri Nets, Algebras, Morphisms and Compositionality. **Information and Computation**, San Diego, v.72, p.197-238, 1987.
- [XML 98] FREQUENTLY Asked Questions about the Extensible Markup Language. Disponível em: <<http://www.ucc.ie/xml/faq.html>>. Acesso em: jul. 1998.
- [XML 99] XML.com. **EDMS 98 – Product Info**. Disponível em: <http://xml.com/xml/pub/p/EDMS_98>. Acesso em: jun. 1999.