Autômatos Finitos: um Formalismo para Cursos na Web[1]

Júlio P. Machado, Leonardo Penczek, Carlos Tadeu Q. Morais, Paulo B. Menezes
Universidade Federal do Rio Grande do Sul - Instituto de Informática
Av. Bento Gonçalves, 9500. Caixa Postal 15064. CEP 91501-970.
Porto Alegre - RS - Brasil
E-mail: {jhapm, penczek, cmorais, blauth} @inf.ufrgs.br

[1] Trabalho parcialmente apoiado por CAPES (Projeto TEIA), FAPERGS (Projeto QaP-For) e CNPq (Projetos HoVer-CAM, GRAPHIT)

Resumo

Este trabalho apresenta a modelagem de cursos disponibilizados via WWW utilizando-se construções formais conhecidas como Autômatos Finitos Determinísticos com Saída. É introduzido o conceito de "cursos são autômatos" como uma estrutura em camadas que permite fácil implementação e criação de material hipermídia independente do esquema de navegação. O modelo permite projetar cursos hipermídia com a mesma estrutura, reutilizando seu conteúdo em outros cursos e possibilitando o reuso do material didático. O objetivo fundamental deste projeto é criar um sistema semi-automatizado para o suporte de cursos na Web através da aplicação de conceitos inerentes à Ciência da Computação, em especial da Teoria de Autômatos e Teoria das Categorias, juntamente à tecnologia de hiperdocumentos, reunindo os benefícios de ambas.

Palavras-chave: WWW, cursos hipermídia, autômatos finitos, projeto de cursos

1.Introdução

Com a expansão da *WWW*, tornou-se possível oferecer a estudantes um acesso ilimitado a materiais instrucionais de forma independente de tempo e espaço [3,4,24,32]. Entretanto, tais materiais estão dispersos através de inúmeros servidores *Web* e não é fácil encontrar a informação que se procura quando tentamos aprender ou ensinar algum tópico específico. Torna-se essencial que a Engenharia de Software pesquise e desenvolva modelos [10,12,28], linguagens e ferramentas [6,13,14,15] para este novo domínio de aplicação que é a educação e treinamento baseados em tecnologia *Web* a fim de solucionar e contornar problemas como o mencionado anteriormente.

Neste trabalho, é apresentado uma forma de modelagem de cursos na *Web* utilizando Autômatos Finitos Determinísticos com Saída [17,35]. Esta técnica está direcionada para o suporte de cursos remotos [18,20,34] em um sistema semi-automatizado para o ensino de Informática Teórica no Instituto de Informática da Universidade Federal do Rio Grande do Sul. Neste modelo, cursos são autômatos com saída e *links* entre as páginas são transições de um autômato.

Autômatos têm uma história de uso em muitas disciplinas como uma técnica de diagramação formal [7,17,19]. Os autômatos podem ser representados por uma estrutura de grafo, a qual pode ser manipulada sobre vários aspectos. Grafos com uma semântica formal podem ser usados para prover interfaces de programação para o controle de material hipermídia. Diversos trabalhos relacionados ao uso de representações gráficas na modelagem de aplicações hipermídia, principalmente modelagem de cursos, utilizam outras construções,

como Mapas Conceituais [9,11,26,29], Workflow [1,24], modelos Entidade-Relacionamento e Orientação a Objetos [10,25].

Uma das vantagens da solução utilizada é que a estrutura dos autômatos com saída permite 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 parte ou íntegra das páginas *Web* em diversos cursos (autômatos) - eliminando a redundância na criação de páginas - bem como a modularização que facilita a expansibilidade do sistema.

Alguns aspectos específicos de implementação do modelo proposto são discutidos e as soluções empregadas são apresentadas. Com especial enfoque no desenvolvimento de sistemas modulares [10,12], onde a base de hiperdocumentos é projetada de forma independente da estrutura de controle da aplicação hipermídia.

2.Cursos são Autômatos

Neste sistema, um curso pode ser visto sob dois enfoques principais:

A) A Máquina de Moore [35] é um autômato finito com saídas associadas aos estados. É representada por uma 7-upla $M = (\Sigma, Q, \delta, q_0, F, \Delta, \delta_S)$, onde

Σ alfabeto de símbolos de entrada;

Q conjunto de estados possíveis do autômato, o qual é finito;

δ: Q x $\Sigma \rightarrow$ Q função programa ou função de transição, a qual é uma função parcial que, dependendo do estado corrente e do símbolo lido, determina o novo estado do autômato;

q_0 estado inicial tal que q_0 é elemento de Q;

F conjunto de estados finais tal que F está contido em Q;

Δ alfabeto de símbolos de saída;

δ_S: Q $\rightarrow \Delta^*$ função de saída, a qual é uma função total que determina a geração de uma palavra de saída para cada estrado.

B) A Máquina de Mealy [35] é um autômato finito modificado de forma a gerar uma palavra de saída para cada transição. É representado por uma 6-upla $M = (\Sigma, Q, \delta, q_0, F, \Delta)$, onde

Σ alfabeto de símbolos de entrada;

Q conjunto de estados possíveis do autômato, o qual é finito;

δ: Q x $\Sigma \rightarrow$ Q x Δ^* função programa ou função de transição, a qual é uma função parcial que, dependendo do estado corrente e do símbolo lido, determina o novo estado do autômato e a saída gerada;

q_0 estado inicial tal que q_0 é elemento de Q;

F conjunto de estados finais tal que F está contido em Q;

Δ alfabeto de símbolos de saída.

As n-uplas dos autômatos de cursos apresentam uma correspondência às estruturas de hiperdocumentos na *Web*. O alfabeto Σ de símbolos de entrada é um conjunto de nomes que identificam os *links* de navegação no hipertexto do curso. O estado inicial q_0 corresponde ao ponto de entrada inicial de um curso, comumente chamado de *homepage*. Note que a noção de palavra permanece, pois as saídas (de um alfabeto Δ) estão relacionadas a unidades de informação (conteúdo do curso) constituídas por páginas *Web* e as palavras de saída Δ^*, presentes na função programa δ e função de saída δ_S, são páginas concatenadas como uma única página *Web* no navegador do usuário. As transições, definidas na função programa δ,

funcionam como ligações lógicas entre os conteúdos dos cursos e definem possíveis *links* HTML a serem selecionados pelos alunos durante a navegação pelo hipertexto.

Optou-se ainda por uma solução não *standard* pois verificou-se que, com freqüência, em cursos pequenos cada estado tem uma saída diferente, ou seja, cada página de conteúdo do curso corresponde a exatamente um estado. Esta solução foi criada para facilitar a especificação da função programa do autômato pelo professor e, conseqüentemente, a utilização de uma interface mais simples e direta.

Uma conseqüência direta da modelagem utilizada é que o conteúdo instrucional deve ser particionado por várias páginas de hipertexto, geralmente pequenas, que no todo formam um tópico a ser apresentado em um curso.

No exemplo da Figura1, é definida uma seqüência (parte de um autômato maior) simples e objetiva de um curso como uma Máquina de Moore. Considere como material instrucional as seguintes páginas de conteúdos, disponibilizadas em um servidor HTTP: definição de autômato finito (A), exemplo de autômato (B), definição de autômato de saída (C), Máquina de Mealy (D), Máquina de Moore (E), exemplo de Mealy (F), exemplo de Moore (G), exercícios de Mealy (H), exercícios de Moore (I). Os *links* de navegação disponibilizados aos usuários são representados pelas seguintes âncoras de texto: próximo (1), anterior (2), exercício (3). As páginas concatenadas de saída, que são os hiperdocumentos apresentados no navegador do usuário: A-B, C, D-F, E-G, H-F, I-G.

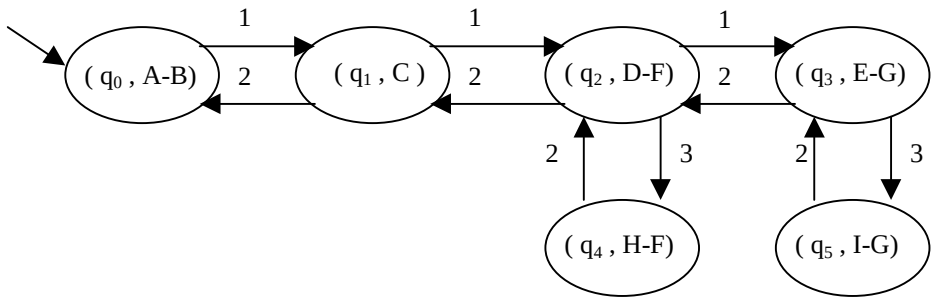

Figura1 - Parte de um curso modelado como uma Máquina de Moore

Um aspecto fundamental do modelo apresentado está na possibilidade da operação de composição sobre nodos/*links* (estados/transições), bem como sobre autômatos. A estrutura formal que permite tal operação de alto nível encontra-se na Teoria das Categorias [2,27]. Utilizando-se construções inspiradas em grafos e categorias com elementos distinguidos, apresentadas em [16], é possível definir a composicionalidade de cursos, facilitando a construção de novos cursos sobre cursos já existentes. Neste contexto, a operação categorial de produto permite a criação e sincronização de caminhos possíveis de navegação, a operação funtorial de restrição (técnica de fibração) permite a seleção de partes específicas de um curso para remoção automática de *links* entre páginas de conteúdo, e a operação de reetiquetação (técnica de cofibração) fornece ferramentas para reetiquetação e encapsulamento de transições (tais operações encontram-se definidas em [36]). Não está no escopo deste trabalho a introdução de conceitos relacionados à Teoria das Categorias, ficando o mesmo restrito à apresentação dos autômatos finitos.

3.Modelo do Sistema

Em projetos de *software* para a Internet, a *Web* aparece como uma estrutura básica que pode ser utilizada para a criação e apoio à ambientes de aprendizagem. A estrutura *WWW* é um núcleo de protocolos de transporte, interfaces e armazenamento de dados sobre o qual são definidas camadas de ferramentas para complementar uma estrutura já existente, visando torná-la mais apta para o suporte às necessidades de aplicações mais específicas.

Como um complemento à estrutura Internet, o sistema consiste em um ambiente de apoio interativo ao professor, com recursos que facilitam uma representação gráfica do curso, assim como uma interface amigável para os serviços disponibilizados (criação e manutenção de cursos).

Propõe-se um modelo em múltiplas camadas, as quais podem ser projetadas e implementadas de uma forma independente. Na Figura2 mostra-se os relacionamentos entre os componentes do sistema de controle de acesso ao conteúdo da informação.

A camada A contém o conjunto de conhecimentos disponível aos alunos, sendo composto por material hipermídia disponibilizados em um servidor. O conjunto de autômatos, como definidos na seção anterior, localiza-se na camada B e possibilita selecionar e ordenar as informações que os alunos receberão de acordo com os objetivos de um curso. Na camada C encontram-se os mecanismos que permitem acessar diretamente cada porção de informação, entre eles índices, glossário e ferramentas de busca.

O uso da camada C possibilita uma navegação livre através dos hiperdocumentos, acessando as informações na ordem que se desejar. Enquanto a camada B possui mecanismos que permitem acessar um tópico de cada vez, numa seqüência previamente definida pelo professor.

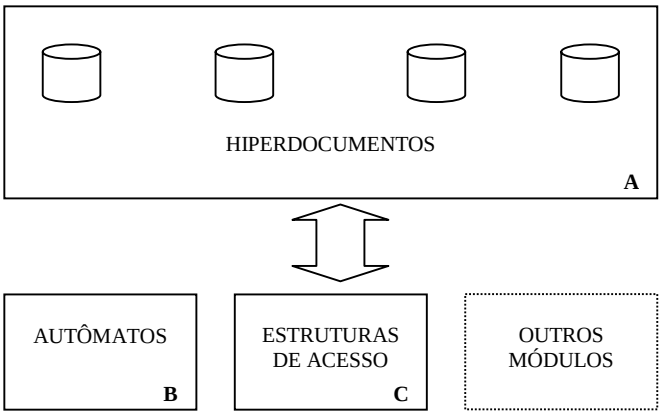

Figura2 - Modelo dos componentes do sistema

A criação de cursos 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 ou com objetivos diferentes. Desta forma, como a criação do material hipermídia é independente do autômato, é possível criar diversos autômatos sobre uma mesma base de materiais, cada um com objetivos específicos ou enfoques diferentes. O professor pode armazenar no servidor um grande número de autômatos diferentes, cada um com um objetivo específico como: introduzir um novo tópico aos alunos, revisar lições previamente apresentadas, preparar o aluno para uma avaliação final, ou até mesmo

seqüências de estudo direcionadas a um grupo particular de estudantes (por exemplo instrução baseada em exemplos ou em resolução de problemas).

Um ponto importante na estrutura do sistema diz respeito a capacidade do modelo de autômatos finitos em mapear as possibilidades de navegação sobre uma hiperbase. A expressividade de um sistema de hipertexto é definida como a capacidade de seguir um *link*, conforme [22]. Ainda segundo os autores [22] (veja seção 6), o autômato com pilha fornece uma capacidade de expressão maior que o autômato finito determinístico, porém tal poder de expressão já está presente no sistema através dos mecanismos de "história" e "botão de volta" implementados pela maioria dos navegadores de hipertextos disponíveis. É importante salientar que o sistema não tem a intenção de mapear com totalidade a expressividade dos sistemas de hipertexto, somente conseguida através de *links* funcionais (Máquina de Turing [7,35]), mas está dirigido à 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, desta forma autômatos finitos cumprem com eficiência o papel de mecanismos de restrição e estruturação dentro do modelo proposto.

3.1.Reusabilidade e Expansibilidade

A apresentação de conhecimento na *Web* se realiza através da composição de diversos arquivos, trechos de textos em *HTML*, figuras, simuladores [21] em *Java*, e outros recursos hipermídia que são ligados e interligados por *hyperlinks* de acordo com a estrutura definida pelos autômatos do curso. 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 conseqüência, 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 cursos.

A estrutura do sistema de hipertexto que não utiliza *links* diretos permite a estruturação e edição do conjunto de materiais hipermídia sem a necessidade do autor dos documentos se preocupar com a inserção de *links* diretamente nos textos, facilitando a tarefa de edição dos documentos.

Um benefício adicional é que, como páginas de um curso podem ser constituídas por *n* outras sub-páginas concatenadas, o autor pode construir funções de saída/programa diferentes que contenham acesso a páginas de mesmo conteúdo, evitando-se redundância na criação dos documentos *HTML*. Por exemplo, pode existir uma função que gere uma página contendo uma "definição" e um "exemplo", enquanto outra página pode ter o mesmo "exemplo" como "dica" para um exercício.

Por apresentar características modulares, outros módulos e modificações podem ser facilmente adicionados ao sistema de acordo com as necessidades. Novos módulos de controle podem ser adicionados sobre as camadas da hiperbase e autômatos. Por exemplo:

- Uma alteração da função programa do autômato finito permite a inclusão de restrições temporais no modelo, como um subconjunto das possibilidades encontradas na modelagem de *workflow* [24]. Aspectos relacionados ao tempo são de fundamental importância em ferramentas de *courseware*. Um exemplo de restrição temporal seria a necessidade de completar um curso no período máximo de quatro meses. A inclusão de regras condicionais (segundo alguma linguagem de lógica temporal) na definição da função programa está intimamente ligada a existência de agentes automatizados de

monitoração das atividades dos estudantes, os quais controlam a disponibilidade de uma transição de acordo com as restrições impostas.

- Com a representação do autômato armazenada, é possível gerenciar a execução da navegação de alunos de forma individualizada, através da verificação dos caminhos dos autômatos. Também é possível ao aluno retornar ao último ponto de acesso no curso durante sua última sessão de trabalho, bastando o armazenamento do estado atual do autômato dentro de uma base de dados com *logging* das atividades realizadas pelos alunos.
- Com a possibilidade de definição de diferentes autômatos sobre um mesmo curso, um módulo Tutor Inteligente pode ser adicionado juntamente com um Modelo de Aluno. Esta nova camada seria responsável por selecionar o autômato a ser disponibilizado ao aluno de acordo com regras específicas armazenadas em sua base de conhecimento. O modelo de Multi-Agentes pode ser utilizado na implantação de tal solução.
- Um módulo para avaliação de alunos está em desenvolvimento. Este módulo consiste em um banco de dados de provas e exercícios on-line integrado com o módulo de controle dos autômatos. 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.

3.2. Operações Sobre Cursos

O modelo de autômatos utilizado é inerentemente um mecanismo de controle de navegação, provendo ao autor de um curso um controle maior sobre a seqüência na qual os nodos serão visitados pelos alunos. Este mecanismo pode ser implementado através de regras de restrição aplicadas sobre o autômato de um curso. Basicamente, temos 3 diferentes restrições: *transições* - o autor pode selecionar transições específicas que não mais estarão disponíveis no autômato, desta forma removendo *links* entre páginas de conteúdo; *etiquetas* - pode-se escolher etiquetas de determinadas transições para serem removidas de um autômato, restringindo os caminhos de navegação possíveis; *estados* - o autor pode querer remover certos materiais instrucionais de um curso, esta operação pode ser realizada através da remoção de estados do autômato e correspondentes transições etiquetadas.

Outra operação é construída sobre o alfabeto de entrada, o qual representa os *links* entre as páginas do curso, conforme o modelo já introduzido. O autor pode reetiquetar transições de forma a alterar as âncoras de um *link* dentro de uma página através de um procedimento rápido e seguro, sem a necessidade de alterar o código HTML de alguma página de conteúdo do curso.

Durante a fase de desenvolvimento de um curso, o autor pode estar interessado em gradualmente adicionar novos materiais instrucionais a um curso. Usualmente, estes materiais não estão completamente desenvolvidos e o autor pode especificar na estrutura do autômato quais estados estão "incompletos", limitando o acesso de alunos às respectivas páginas. Tal procedimento ilustra uma operação categorial conhecida como encapsulamento.

4.Implementação

A implementação do sistema descrito neste trabalho tem como base as premissas da criação de uma ferramenta simples, de fácil implementação, manutenção e reuso, independente de outros sistemas (como banco de dados, sistema operacional, etc). Com base nessas premissas, as escolhas de projeto centraram-se na seleção de linguagens de

programação suportadas em múltiplas plataformas e na utilização de um sistema de armazenamento de dados com formato interno próprio. Deve-se salientar que o sistema pode vir a ser portado para um banco de dados específico, caso seja necessário no futuro.

Foi adotado uma arquitetura cliente-servidor para o sistema na *WWW* (Figura3). 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 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.

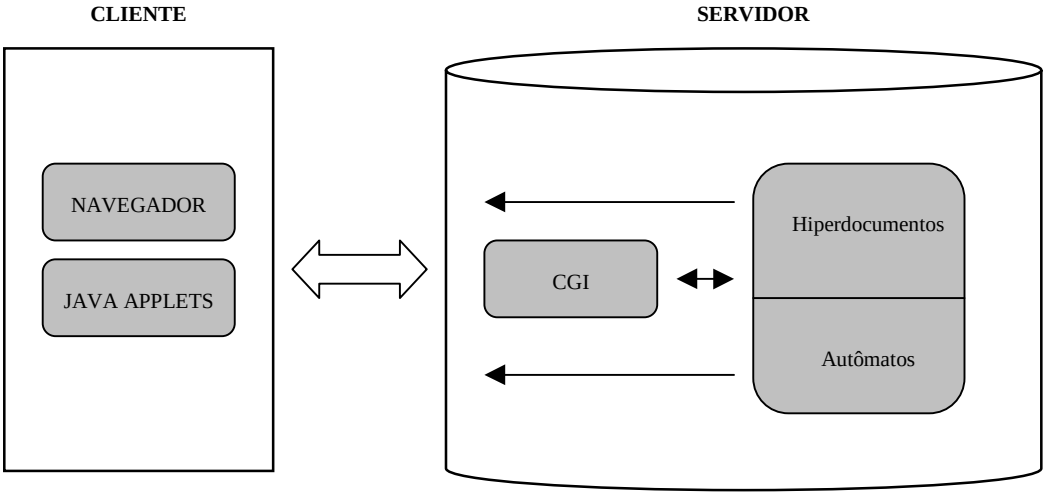

Figura3 - Arquitetura cliente-servidor básica do sistema

O autor de um curso terá acesso a uma interface gráfica com suporte à notação de grafos para a criação e manipulação dos autômatos, utilizando-se as facilidades multiplataforma da linguagem *Java*. Além de facilitar a visualização do autômato, pretende-se que a interface em *Java* implemente operações sobre caminhos do curso, como remoção e inserção de estados no autômato, bem como composição de autômatos com ou sem identificação de partes. Tais operações, construídas formalmente sobre o fecho reflexivo transitivo [17,27] do autômato a nível de transições, serão coerentemente definidas por construções categoriais [2,27], a fim de manter consistente os autômatos. O uso de categorias deve-se ao fato do alto poder de expressividade de suas construções, que permitem definir operações de alto nível como as mencionadas na seção3.2 acima.

Os módulos CGI em Perl são responsáveis pela edição dos autômatos e controle da navegação através do curso. Estes *scripts* são encarregados da manutenção da base de dados dos autômatos (definição do alfabeto de entrada, cadastramento dos documentos HTML, definição das transições entre os estados) e do acesso à hiperbase, realizando a concatenação de vários documentos HTML em uma única página no navegador do usuário.

5.Site de Linguagens Formais: um exemplo

A educação baseada na *Web*, neste site implementada como suporte ao ensino presencial, utiliza a rede como meio para a publicação do material didático, simuladores

interativos, apresentação de tutoriais e trabalhos de alunos, aplicação de exercícios e provas, bem como ferramentas de comunicação para os estudantes.

O estudo da utilização de autômatos para a definição de cursos está sendo realizada sobre um site voltado para o ensino de Informática Teórica (ver Figura4). 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 [7,17] 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.

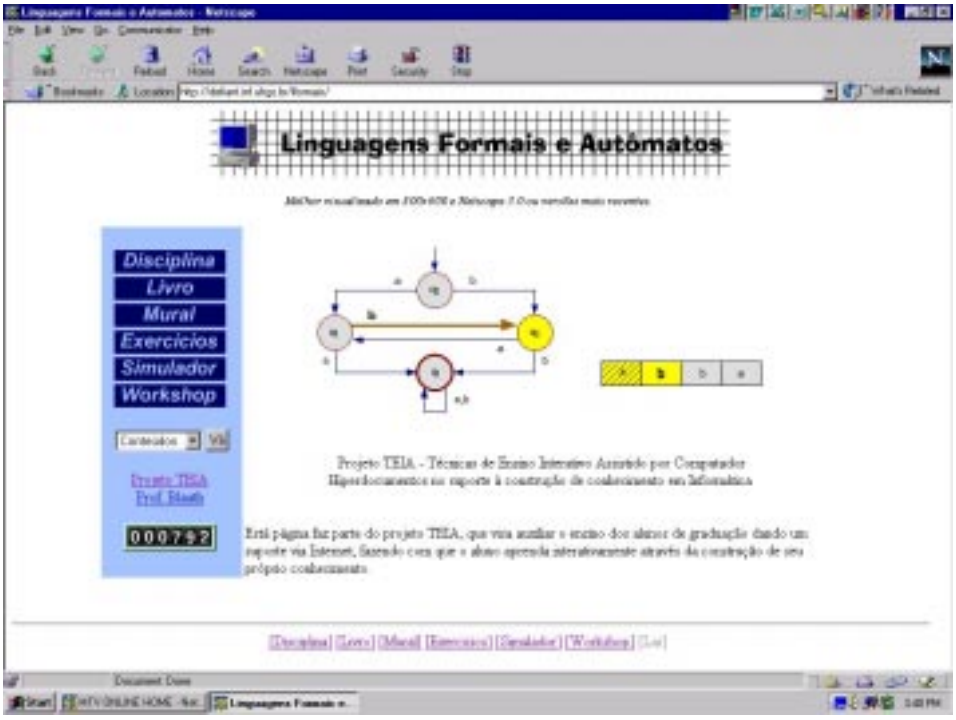

Figura4 – *Site* da disciplina de Linguagens Formais

6.Trabalhos Relacionados

O propósito deste trabalho é estudar um mecanismo básico para a estruturação de um sistema hipertexto. Para tal, foi apresentado o formalismo de autômatos finitos como suporte aos conceitos de nodos e *links* de um hipertexto [23]. Nossa meta difere dos trabalhos apresentados a seguir no sentido que estamos interessados na definição de um formalismo básico sobre o qual construímos um sistema de gerência de cursos na *Web*, enquanto os demais trabalhos estão direcionados para modelos formais no estudo da semântica de sistemas hipertexto [33].

No trabalho [22], os autores formalizam sistemas de hipertexto como máquinas abstratas. Os autores apresentam três classes de sistemas modelados como autômatos finitos, autômatos com pilha e máquina de Turing. O estudo está centralizado nos diferentes mecanismos de *links* hipertexto.

Uma discussão sobre o uso de *links* como parte de documentos (por exemplo em *HTML*) ou armazenados de forma separada dos documentos é encontrado em [5]. Os autores descrevem os benefícios e problemas advindos da utilização de uma solução ou outra.

Outros trabalhos são complementares à visão apresentada neste texto. Em especial [31] formaliza hipertextos como grafos e o trabalho [30] define um "link-automaton" como uma máquina abstrata sobre a qual podem ser provadas várias propriedades sobre um dado hipertexto.

Além de autômatos, a especificação de um sistema hipertexto também pode ser representado por Redes de Petri (o modelo Trellis [8] é um exemplo). Embora Redes de Petri seja um formalismo mais poderoso que um autômato finito, a implementação do último requer menos complexidade e portanto foi escolhida como base para o sistema aqui apresentado. O autômato finito também pode ser utilizado como uma representação mais natural para a metáfora de "livro", comumente utilizada na implementação de material instrucional em hipertexto pela maioria dos sistemas de autoria multimídia.

7.Conclusão

Este trabalho apresentou uma alternativa para o apoio à criação de cursos na *Web*, visando fornecer maior flexibilidade e facilidades a professores e alunos. A solução é suportada por uma arquitetura cliente-servidor multicamada baseada em um modelo de cursos para a Web definida sobre o formalismo de Autômatos Finitos com Saída. É importante salientar que embora o modelo proposto esteja baseado em sistemas *CAI (Computer Assisted Instruction)* ou Instrução Assistida por Computador, ele visa fornecer apoio ao processo de ensino/aprendizagem independente de uma teoria pedagógica específica.

O modelo apresentado leva a um alto grau de modularização do material educacional, apresentando as seguintes vantagens: facilidade de reuso de páginas em diversos cursos, com eliminação da redundância; independência dos hiperdocumentos da estrutura do autômato, cuja alteração não influi nas páginas e vice-versa; facilidade de implementação e manutenção; interface gráfica simples e direta; elaboração de seqüências instrucionais com enfoques específicos e capaz de oferecer estudo individualizado. Os cursos foram modelados sob três enfoques: Máquina de Mealy, Máquina de Moore e um autômato onde cada estado é constituído por uma única página.

Acredita-se que este esforço empregado na criação de ferramentas para o suporte à disponibilização de cursos na *Web* utilizando uma estrutura padrão fornecerá aos profissionais do ensino universitário (em especial da área de Informática Teórica) meios de oferecer aos estudantes um ambiente consistente e integrado aos diversos centros de pesquisa do Brasil, através do compartilhamento e reuso de material hipermídia desenvolvido para a rede.

8.Referências

[1] Heimo H. Adelsberger, Frank X. Körner, Jan M. Pawlowski. *A Conceptual Model for an Integrated Design of Computer Supported Learning Environments and Workflow Management Systems*. In Proceedings of the XV IFIP World Computer Congress - Teleteaching 98. Vienna, 1998. p.55-64.

[2] Michael Barr, Charles Wells. *Category Theory for Computing Science*. Prentice Hall. USA, 1990.

[3] Thierry Beltran. *Hypermedia and Computer Aided Instruction (CAI): an Architecture for Hybrid Systems*. In Proceedings of the CALISCE91. Lausanne, September 1991. p.167-176.

[4] Christopher M. Boroni, Frances W. Goosey, et al. *A Paradigm Shift! The Internet, the Web, Browsers, Java and the Future of Computer Science Education.* In SIGCSE Bulletin, March 1998, v.30, n.1, p.145-152.
[5] P. J. Brown, H. Brown. *Embedded or Separate Hypertext Markup: is it an issue?* In Electronic Publishing, 1995, n.8, p.1-13.
[6] F. Casati, B. Pernici. *The Design of Distance Education Applications Based on the WWW.* http://www.epcc.ed.ac.uk/t+lwkshp/submissions/barbara/
[7] Tiarajú A. Diverio, Paulo F. B. Menezes. *Teoria da Computação: Máquinas Universais e Computabilidade.* Editora Sagra-Luzzatto. Brasil, 1999.
[8] R. Furuta, P. D. Stots. *Programmable Browsing Semantics in Trellis.* In Proceeding of the Hypertext'89, 1989, p.27-42.
[9] Brian R. Gaines, Mildred L. G. Shaw. *Concept Maps as Hypermedia Components*. 1996. http://ksi.cpsc.ucalgary.ca/articles/ConceptMaps/
[10] F. Garzotto, D. Schwabe, P. Paolini. *HDM: A Model Based Approach to Hypertext Application Design*. In ACM Transactions on Information Systems, January 1993, v.11, n.1, p.1-26.
[11] D. Jonassen, R. Marra. *Concept Mapping and Other Formalisms as Mindtools for Representing Knowledge*. 1993. http://www.icbl.hw.ac.uk/~granum/class/altdocs/dav_alt.htm
[12] E. I. Kawasaki, C. T. Fernandes. *Modelo para Projeto de Cursos Hipermídia*. In Proceedings of the SBIE 96, Simpósio Brasileiro de Informática na Educação. Brasil, Novembro 1996. p.227-240.
[13] Anthony Kelly. *Designing Instructional Hypertext for Use in Lecture Note Review: Knowledge Engineering and Preliminary Testing*. In Journal of Educational Multimedia and Hypermedia, 1990, v.2, p.149-176.
[14] Júlio P. Machado, Paulo F. B. Menezes. *Sistemas de Gerenciamento para o Ensino a Distância*. In Proceedings of V Congresso Internacional de Educação a Distância. São Paulo - Brasil, October 1998. http://www.abed.org.br/artigos2/artigos/24/sgead971.html
[15] Júlio P. Machado. *Sistemas de Gerenciamento para o Ensino a Distância*. Instituto de Informática UFRGS. Brasil, January 1999. 80pp.
[16] Paulo F. B. Menezes. *Reificação de Objetos Concorrentes*. Tese de Doutorado. Instituto Superior Técnico. Universidade Técnica de Lisboa. Portugal, 1997. 148pp.
[17] Paulo F. B. Menezes. *Linguagens Formais e Autômatos*. Segunda Edição. Editora Sagra-Luzzatto. Brasil, 1998.
[18] Paulo F. B. Menezes, Tiarajú A. Diverio, et al. *Desenvolvimento da Área Formal da Computação no Instituto de Informática da UFRGS*. In Proceedings of the I Workshop Brasileiro de Métodos Formais. Porto Alegre - Brasil, 1998. p.1-12.
[19] Paulo F. B. Menezes, A. Sernadas, J. F. Costa. *Nonsequential Automata Semantics for a Concurrent Object-Based Language, Revised and Extented Version.* In Proceedings of the First USA-Brazil Workshop on Formal Foundations of Software Systems. Rio de Janeiro - Brasil, 1998. Electronic Notes in Theoretical Computer Science n.14. http://www.elsevier.nl/locate/entcs/volume1.html
[20] Paulo F. B. Menezes. *Mathematical Methods in the Computing Institute of UFRGS*. In Proceedings of the VII International Conference on Algebraic Methodology and Software Methodology. Brasil, Janeiro 1999.
[21] Thiago F. Moesch, Fábio Lutz, Tiarajú M. Diverio, et al. *Simuladores de Máquinas Abstratas: ferramentas de apoio ao ensino de informática teórica*. In Proceedings of the

SBIE 98, Simpósio Brasileiro de Informática na Educação. Fortaleza - Brasil, Novembro 1998.
[22] Luc Moreau, Wendy Hall. *On the Expressiveness of Links in Hypertext Systems*. In The Computer Journal, 1998, v.41, n.7, p.459-473.
[23] Jacob Nielsen. *Hypertext and Hypermedia*. San Diego Academic Press. EUA, 1990.
[24] José Palazzo M. de Oliveira, Mariano Nicolao, Nina Edelweis. *Concenptual Workflow Modelling for Remote Courses*. In Proceedings of the XV IFIP World Computer Congress - Teleteaching 98. Vienna, 1998. p.789-797.
[25] Mariano G. Pimentel. *Modelo Orientado a Conceitos (MOC)*. In Proceedings of the SBIE 98, Simpósio Brasileiro de Informática na Educação. Fortaleza - Brasil, Novembro 1998.
[26] Miguel R. Flores Santibañez, Clovis T. Fernandes. *SICH - Uma Ferramenta para a Construção de Cursos Hipermídia no WWW*. In Proceedings of the SBIE 98, Simpósio Brasileiro de Informática na Educação. Fortaleza - Brasil, Novembro 1998.
[27] G. Winskel. *Petri Nets, Algebras, Morphisms and Compositionality*. Information and Computation 72, Academic Press, 1987, p.197-238.
[28] Daniel Schwabe, Gustavo Rossi. *Introdução aos Sistemas e à Autoria Multimídia*. In Proceedings of the VI EBAI, Escola Brasileira-Argentina de Informática. Córdoba, 1993.
[29] Perla Señas, Norma Moroni. *Un Ambiente Computacional para el Aprendizaje Significativo: Mapas Conceptuales Hipermediales*. In Proceedings of the SBIE 97, Simpósio Brasileiro de Informática na Educação. Brasil, Novembro 1997. p.469-483.
[30] P. David Stotts, R. Furuta, C. Ruiz Cabarrus. *Hyperdocuments as Automata: Verification of Trace-Based Browsing Properties by Model Checking*. In ACM Transactions on Information Systems, January 1998, v.16, n.1, p.1-30.
[31] F. W. M. Tompa. *A Data Model for Flexible Hypertext Database Systems*. In ACM Transactions on Information Systems, 1989, n.7, p.85-100.
[32] J. Vassileva. *Dynamic Course Generation on the WWW*. In Proceedings of the Artificial Inteligence in Education. Kobe - Japão, 1997, p.498-505.
[33] Weigang Wang, Roy Rada. *Structured Hypertext with Domain Semantics*. In ACM Transactions on Information Systems, October 1998, v.16, n.4, p.372-412.
[34] Taisy Weber, Raul F. Weber, Ingrid J. Pôrto, Flávio Wagner, Paulo F. B. Menezes. *Uma Experiência com Hiperdocumentos e Internet no Suporte a Disciplinas de Computação*. In Proceedings of the VI Workshop sobre Educação em Informática, Anais do XVIII Congresso da Sociedade Brasileira de Computação. Belo Horizonte - Brasil, Agosto 1998. v.1, p.532-545.
[35] J. E. Hopcroft, J. D. Ulmann. *Introduction to Automata Theory, Languages and Computation*. Addison-Welsey, 1979.
[36] P. Blauth Menezes, Júlio P. Machado. *Web Courses are Automata: a Categorial Framework*. II Workshop de Métodos Formais. Santa Catarina, 1999. (submetido)