ENSINET/NAV: UMA FERRAMENTA PARA ESTRUTURAÇÃO DE CURSOS BASEADOS EM OBJETOS DE APRENDIZAGEM *

Diego Lemos de Souza **
Graçaliz Pereira Dimuro ***
Antônio Carlos da Rocha Costa ****
Raquel Mello de Miranda *****

Resumo: *Este artigo tem como objetivo apresentar o ENSINET/NAV, uma ferramenta para a criação e gerenciamento de cursos on-line baseados em objetos de aprendizagem. Os cursos são realizados no ambiente ENSINET e estruturados através do conceito de Autômato de Navegação. No artigo são apresentadas também uma visão geral dos Hyper-Autômatos, estrutura que foi utilizada em ferramentas semelhantes que estimularam a criação do ENSINET/NA,V e uma descrição de como está implementada essa ferramenta.* *
Palavras-Chave: ENSINET/NAV, PYTHON, ZOPE, AUTÔMATOS DE NAVEGAÇÃ, OBJETOS DE APRENDIZAGEM.

Abstract: *This paper aims the presentation of the ENSINET/NAV, a tool to help the creation and management of on-line courses based on learning objects. Such courses are created in the context of the ENSINET environment and structured according to the concept of Navigation Automata. In the paper, we also prsent an overview of the Hyper-Automata, a structure that was used in similar tools that stimulated the development of the ENSINET/NAV, a well as a description of how this tool was implented.*
Keywords: ENSINET/NAV, PYTHON, ZOPE E NAVIGATION AUTOMATA, LEARNING OBJECTS.

1. Introdução

Este artigo tem como objetivo apresentar o ENSINET/NAV, uma ferramenta baseada em Autômatos de Navegação e Objetos de Aprendizagem, que permite a criação e gerenciamento de cursos on-line dentro do ambiente ENSINET (Miranda, 2001).

O trabalho se baseou, em um primeiro momento, no estudo feito sobre os projetos *Hyper-Automaton (HA)* (Machado, 2002a) e *Extensible Hyper-Automaton (XHA)* (Federizzi, 2002), ambos implementados através da utilização do formalismo dos autômatos finitos com saída. Após o estudo finalizado e alguns testes efetuados, optou-se pela implementação de uma nova ferramenta, o ENSINET/NAV, com a utilização dos conceitos de Autômatos de Navegação (Dimuro, 2002) e Objetos de Aprendizagem (LTSC 2002)(Miranda, 2004).

Os Autômatos de Navegação apresentam as mesmas vantagens existentes no Hiper-Autômatos, ou seja, permite a reutilização dos objetos de aprendizagem em vários

* Projeto Financiado pela FAPERGS
** Bacharel em Ciência da Computação
Escola de Informática/UCPel
Bolsista lyon@atlas.ucpel.tche.br diegolemosdesouza@terra.com.br
http://atlas.ucpel.tche.br/~lyon
*** Doutora
Escola de Informática/UCPel
Orientadora liz@atlas.ucpel.tche.br
**** Doutor
Escola de Informática/UCPel
PPGIE/UFRGS
Co-Orientador rocha@atlas.ucpel.tche.br
***** Mestranda PPGC/UFRGS
Bolsista CNPq raquel@atlas.ucpel.tche.br

contextos, possibilitando também a formalização da navegação, com isso eliminando a possibilidade de existência de *links* quebrados.

Entretanto, a implementação da ferramenta ENSINET/NAV é totalmente diferente das implementações estudadas. Isso se deve ao fato de que a ferramenta é implementada através do paradigma de orientação a objetos, utilizando-se para tal a linguagem de *Python* (Lutz, 1999) e o gerenciador de aplicações Zope (Brochmann, 2002).

O ENSINET/NAV dispõe de uma estrutura de gerenciamento dos objetos de aprendizagem com métodos e classes que possibilitam ao usuário a manipulação desses objetos. No caso de objetos de aprendizagem que são páginas HTML, a ferramenta possibilita aos usuários a edição direta de tais objetos, a fim de facilitar a adequação dos mesmos às necessidades dos cursos. Todos os objetos de aprendizagem que são páginas HTML são criados através da utilização de um editor HTML acessado via *browser* na própria ferramenta.

A mesma estrutura de programação é aplicada nas classes e métodos que criam os cursos na ferramenta. Cada curso é visto como um autômato de navegação, ou seja, cada curso que for criado tem em sua estrutura a definição dos estados do autômato de navegação, que são constituídos de *frame-sets* de páginas HTML, contendo possíveis *links* para outros estados (páginas).

Este artigo descreve sucintamente os projetos *Hyper-Automaton (HA)* e *Extensible Hyper-Automaton (XHA)* na seção 2. Na seção 3 apresentam-se sumariamente os Autômatos de Navegação.

Na sessão 4, mostra-se como a linguagem de programação Python e o ambiente de gerenciamento de aplicações Zope, foram utilizados na implementação do ENSINET/NAV.

Na sessão 5, faz uma breve recapitulação sobre os objetos de aprendizagem assim como a forma de utilização dos mesmos dentro do ENSINET e do ENSINET/NAV.

A ferramenta ENSINET/NAV é abordada na sessão 6, mostrando suas funcionalidades, sua estrutura gerencial, a modelagem desenvolvida, sua arquitetura, e outras informações.

2. Hyper-Automaton e Extensible Hyper-Automaton

A ferramenta *Hyper-Automaton (HA)* (Machado, 2002a) foi desenvolvida com o propósito de auxíliar o usuário na criação de materiais hipermídia, onde a navegação está a cargo do processamento de diversos parâmetros que retornam como resultado uma página do curso. Essa funcionalidade é obtida através da formalização da navegação dos hipertextos através de um autômato finito com saída.

Essa ferramenta foi implementada da seguinte forma: cada estado do autômato possui, vinculada à sua estrutura, uma sequência de unidades de informação que formam a palavra de saída daquele estado. Essas unidades de informação são concatenadas e mostradas no navegador como uma página. Também associado ao estado está a palavra de saída que contém os *links* para outros estados do autômato. O modelo (Figura 01)

utilizado para essa implementação possui as seguintes camadas: camada A (repositório de unidades de informação), camada B (definição dos cursos em forma de autômatos) e camada C (permite a criação de módulos de acesso às estruturas em B).

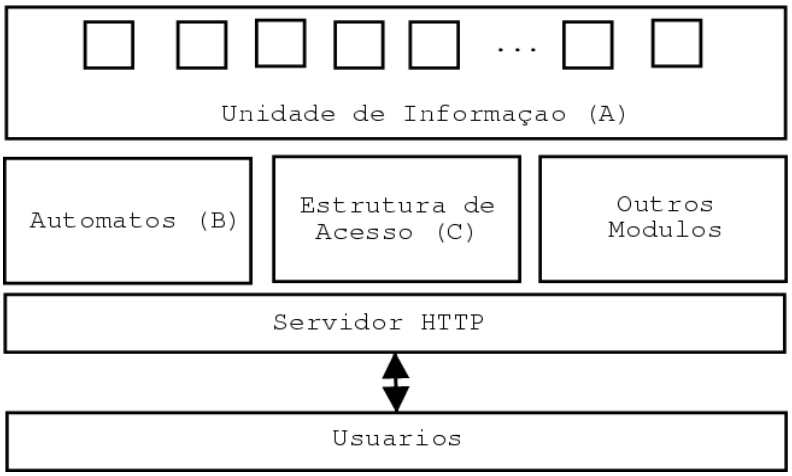

Figura 01 – Modelo Hyper-Automaton

As máquinas de estado dessa ferramenta foram implementadas como programas CGI, em linguagem PERL. Foram implementados dois simuladores de autômatos, um para a máquina de *Moore* e outro para a máquina de *Mealy* as quais se diferem pelos parâmetros recebidos e na forma de descobrimento da palavra de saída.

A ferramenta *Extensible Hyper-Automaton* (Federizzi, 2002) foi baseada na primeira versão (HA), mas totalmente reestruturada para poder eliminar os problemas ocorridos na primeira implementação. Essa ferramenta herdou todas as funcionalidades da primeira, contudo, adicionando novas funções e procedimentos que possibilitam um melhor aproveitamento de seu objetivo.

As modificações adotadas para o XHA (Figura 02) foram:

- as unidades de informação antes codificadas em HTML foram codificadas em XML (Megginson, 1998), porém o usuário não tem a necessidade de conhecimento de tal linguagem pelo fato de que a ferramenta fornece uma estrutura de apoio que facilita a montagem das unidades de informação.

- Outras modificações foram as gramáticas e *layouts* das estruturas criadas na camada de dados, a primeira tem a função de apoio na criação dos documentos em XML, a segunda tem a função de armazenar os *layouts* criados pelos usuários para os seus cursos.

Assim, tem-se que os *Hyper-Automaton* ou *Extensible Hyper-Automaton* são autômatos finitos com saída processados pelas máquinas de *Moore*, que faz o processamento das informações nos estados do autômato, e de *Mealy*, que faz o processamento das informações nas transições do autômato. Cada estado do autômato possui uma sequência de unidades de informação que são concatenadas em uma página HTML, a fim de mostrar ao usuário uma página do curso.

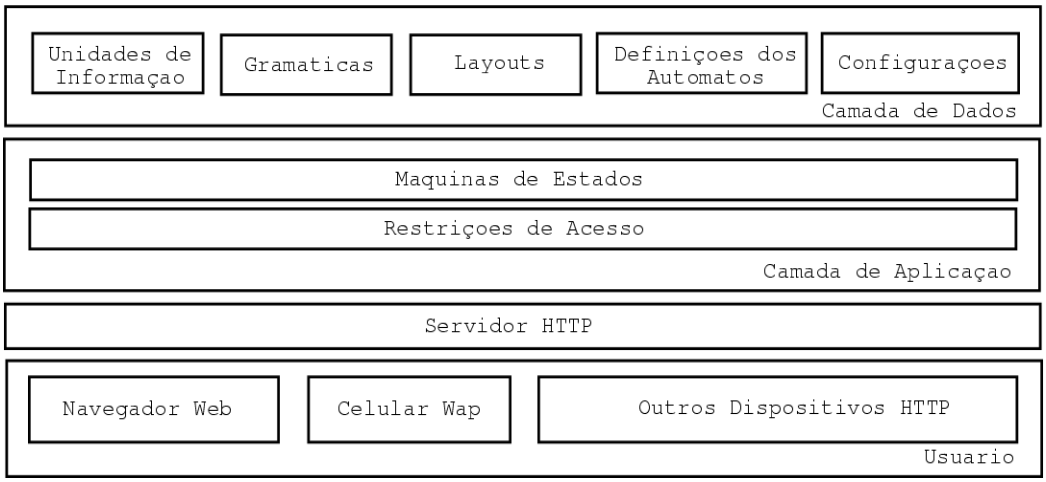

Figura 02 – Modelo Extensible Hyper-Automaton

3. Autômato de Navegação

Para a realização deste trabalho fez-se uso do formalismo dos Autômatos de Navegação (Dimuro, 2002). Os autômatos de navegação (Figura 03) possibilitam a criação de *framesets* de navegação associados a cada estado do autômato. Com isso, cada estado pode conter não somente uma página, mas um *frameset*.

Os *frames* contidos nos *framesets* dos estados do autômato de navegação podem ter associados a eles objetos de aprendizagem quaisquer e, especialmente, objetos de aprendizagem que sejam páginas HTML. Igualmente, podem ter associados *links* para outros estados do autômato, possibilitando assim a navegação.

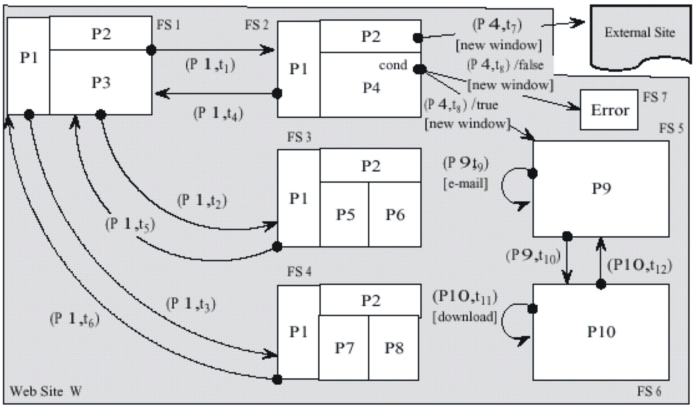

Figura 03 – Autômatos de Navegação

4. Python e Zope

A ferramenta foi desenvolvida para o sistema ENSINET, que está baseado no

gerenciador de aplicações *ZOPE* (Brochmann, 2002), utilizando a linguagem de programação *Python* (Brown, 2001) (Lutz, 1999) e DTML (Machado, 2002b). Com a linguagem *Python* foram desenvolvidas todas as classes que formam a ferramenta, dentre elas a classe de objeto de aprendizagem (Miranda, 2003), a classe de autômato, a classe de estilo de página, etc.

5. Objetos de Aprendizagem

Com a finalidade de padronização do material didático inserido dentro do ENSINET, o sistema contém um *repositório* que tem como função armazenar os objetos de aprendizagem (OA) (Miranda, 2003). Esses objetos são utilizados pelo ENSINET e também pelo ENSINET/NAV.

De acordo com o IEEE, um objeto de aprendizagem (OA) é "qualquer entidade digital ou não digital, que pode ser usada, reusada ou referenciada durante a aprendizagem suportada pela tecnologia" (LTSC, 2002). Conforme essa definição, podemos estabelecer que os objetos de aprendizagem se adequam perfeitamente a um dos principais propósitos do ENSINET/NAV, a reutilização de conteúdo didático sem a necessidade de reescrevê-lo.

6. ENSINET/NAV

A ferramenta ENSINET/NAV é baseada nos autômatos de navegação e nos objetos de aprendizagem. Além desse diferencial, a ferramenta conta com uma estrutura que tem a finalidade de criar cursos contendo sub-cursos (autômatos com níveis), ou seja, no curso (autômato) de primeiro nível cada estado (página) é criado com uma estrutura de documento HTML, sendo que também é possível definir um ou mais estados como um curso (autômato), caracterizando assim cursos com estruturas internas que são sub-cursos. Essa funcionalidade provê a existência de vários níveis de cursos embutidos, caso seja desejado pelo professor.

6.1. Implementação

Este trabalho também foi desenvolvido como Projeto de Graduação na Escola de Informática da UCPel (Souza, 2003). Na primeira versão, se encontravam classes e métodos (Figura 04) que criavam e manipulavam componentes de texto chamados de *unidades de informação*, cursos (autômatos), páginas (estados), estilos de páginas e máquina de estados. Na versão aqui descrita, que está sendo finalizada, as *unidades de informação* foram substituídas pelos *objetos de aprendizagem,* ampliando significativamente as potencialidades da ferramenta.

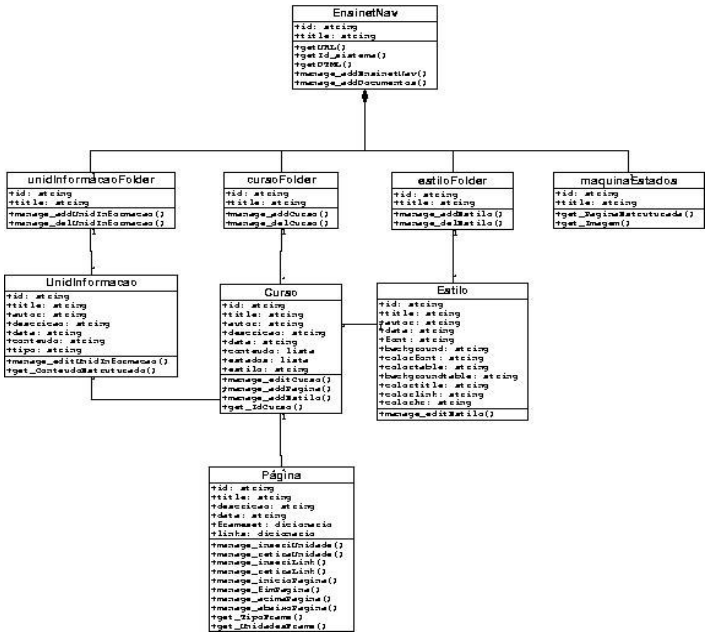

Figura 04 – Diagrama de classes

Na estrutura de gerenciamento dos objetos de aprendizagem encontram-se: a criação dos objetos de aprendizagem que são páginas HTML, que é realizada através da utilização do editor HTML (Figura 05) existente na ferramenta; a visualização, que possibilita ao usuário ver todos os objetos de aprendizagem disponíveis na ferramenta; a edição desses objetos de aprendizagem, que retorna todo o texto escrito pelo usuário no objeto e por consequência possibilitando a sua edição; e a exclusão, que possibilita ao usuário excluir os objetos de aprendizagem.

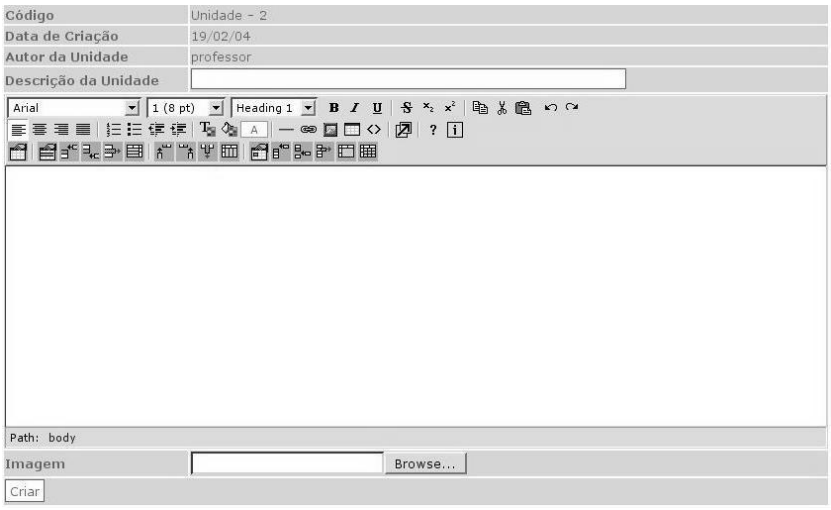

Figura 05 – Tela do editor HTML

O gerenciamento de cursos, onde é possível a criação de cursos, realiza-se através das seguintes funções:

- **criar curso -** que possibilita ao professor a criação do curso informando um título,

uma descrição e selecionando os objetos de aprendizagem que farão parte do curso;

- **editar curso -** que possibilita ao professor editar o título e a descrição do curso assim como adicionar novas unidades de informação ou retirá-las;
- **criar páginas do curso -** onde é possível se criar páginas para o curso selecionado informando-se o nome da página e o tipo de *frame* para essa página;
- **definir conteúdo da página -** através da seleção do curso e posteriormente a seleção da página é possível se inserir o conteúdo (objetos de aprendizagem) que a página deverá conter, essa estrutura prove também mecanismos de manipulação da ordem que esse conteúdo disposto na página;
- **definir *links* da página -** através dessa estrutura é possível se criar *links* de navegação de uma página para a outra, essa função na realidade cria as ligações entre as páginas (estados) de um curso (autômato);
- **definir estilo do curso -** essa estrutura tem como finalidade a associação de um estilo, criado pelo professor, ao seu curso;
- **ver curso -** por fim temos uma aba no sistema que mostra um *link* para todos os cursos o que possibilita a visualização do produto final, o curso propriamente dito, pronto para a utilização.

Existe também a estrutura de gerenciamento de estilos de cursos, onde o usuário tem como criar um estilo para o seu curso caso isso seja necessário. Todos os cursos criados dentro do sistema já são iniciados automaticamento com o estilo padrão existente dentro da ferramenta, isso evita problemas como a não identificação dos vários níveis de títulos existentes dentro dos padrões definidos para a criação dos estilos na ferramenta. Como não haveria de deixar de existir também, foi necessário a criação de uma estrutura para revisar e editar os estilos criados pelos usuários.

Quando um curso (autômato) é criado, ele é iniciado com um título, uma descrição e com os objetos de aprendizagem que farão parte de sua estrutura. Ao criar um objeto que é uma página do curso, o usuário informa o tipo de *frame* que essa página terá, com isso cada estado do autômato retornará para o usuário uma página HTML contendo todos os *frames* referentes ao estado em que o autômato se encontra, por conseqüência, também retornando os objetos de aprendizagem referentes a cada *frame*. Inclui também na página os *links* para os estados seguintes e anteriores ao atual.

Todo o processamento das requisições dos usuários para acessar a navegação dos cursos é feito dentro de um objeto que recebe por meio de parâmetros o identificador do curso (autômato) e página (estado), fazendo, através dos parâmetros recebidos, a busca dos objetos de aprendizagem referentes a esse curso e página no repositório do ENSINET e também os *links* que serão disponibilizados no estado informado.

Cada curso criado dentro da ferramenta pode ser alterado sem que haja a necessidade de alteração dos objetos de aprendizagem que compõem o conteúdo do curso. Também é possível a alteração dos objetos de aprendizagem sem necessidade da alteração do curso (autômato). Todas os objetos de aprendizagem criados podem ser utilizados quantas vezes forem necessárias dentro do curso, além de poderem ser utilizados em outros cursos. Isso é possível porque o repositório que armazena os objetos de aprendizagem é uma base totalmente liberada para os usuários do tipo professor. Essa funcionalidade do sistema é o que permite a reutilização dos objetos de aprendizagem.

7. Conclusão

O ENSINET/NAV faz uso da estrutura de autômatos de navegação e da tecnologia dos objetos de aprendizagem para possibilitar a estruturação de cursos fundada na noção de reutilização de recursos didáticos. O suporte de adequados sistemas de gerência de repositórios de aprendizagem se torna essencial para que a busca e recuperação desses objetos se faça de modo organizado e significativo (Miranda, 2003). A combinação de tecnologias como a do ambiente ENSINET, que suporta cooperação on-line; ENSINET/NAV, que suporta estruturação de cursos baseados em objetos de aprendizagem, e gerenciadores de repositórios de objetos de aprendizagem representa um avanço significativo na construção de ambientes de suporte à aprendizagem cooperativa.

8. Referências Bibliográficas

MIRANDA, R. M.; DIMURO, G. P.; COSTA, A. C. (2001). **Um Ambiente de Suporte ao Ensino Integrado dos Fundamentos Matemáticos da Ciência da Computação utilizando o ZOPE**. In Anais do WSL 2001, pages 40—42. WORKSHOP DE SOFTWARE LIVRE 2001.

MACHADO, J. P.; MENEZES, P. B. **Hyper-Automaton: hipertextos e curso na web usando autômatos finitos com saída,** no Estado do Rio Grande do Sul. Porto Alegre: UFRGS, 2002. Dissertação de Mestrado.
FEDERIZZI, G. L. **Extensible Hyper Automaton,** no Estado do Rio Grande do Sul. Porto Alegre: UFRGS , 2002. Monografia de Graduação.

DIMURO, G. P.; COSTA, A. C. da R. Towards na Automata-Based Navegational model for the Specification of Web Sites. In: **WORKSHOP ON FORMAL METHODS,** 2002, Porto Alegre. Anais. Porto Alagre: 5th WORKSHOP ON FORMAL METHODS, p.36—51.

LUTZ, M.; ASCHER, D. **Learning Python.** O'Reilly, Sebastopol: 1999.

BROCHMANN, M.; KIRCHNER, K.; LÜHNSDORT S.; PRATT, M. **ZOPE - Kit de Construção de Aplicativos de Web.** Alta Books, Rio de Janeiro: 2002.

MEGGINSON, D. **Structuring XML Documents.** Prentice Hall do Brasil, Rio de Janeiro: 1998.

BROWN, M. C. **The Complete Reference Python.** McGraw-Hill, Berkeley: 2001.

MACHADO, C. C. **eXtensible Hyper- Automaton,** no Estado do Rio Grande do Sul. Porto Alegre: UFRGS, 2002. Monografia de Graduação.

MIRANDA, R. M. Groa: Um sistema de gerência de repositórios de objetos de aprendizagem. In: X Semana Acadêmica do PPGC-UFRGS. Porto Alegre: [s.n], 2003.

LTSC. **Learning Technology Standasds Committee**. Final 1487.12.1 LOM draft

standard document. Disponível em: http://ltsc.ieee.org/wg12. Jan. 2002.