EDIC

José Paulo Leal   Nelma Moreira

Technical Report Series: DCC-01-05

Departamento de Ciência de Computadores – Faculdade de Ciências

&

Laboratório de Inteligência Artificial e Ciência de Computadores

Universidade do Porto

Rua do Campo Alegre, 823 4150 Porto, Portugal

Tel: +351+2+6078830 – Fax: +351+2+6003654

http://www.ncc.up.pt/fcup/DCC/Pubs/treports.html

EDIC:UMA ABORDAGEM PARA APRESENTAÇÃO DE CONTEÚDOS PEDAGÓGICOS NA WEB

J.P.Leal, N.Moreira, P.Ribeiro

DCC-FC& LIACC, Universidade do Porto
R. do Campo Alegre 823, 4150 Porto, Portugal
email: {zp,nam}@ncc.up.pt

Resumo

Edic is a Web courseware for an introductory course on the UNIX operating system. In this research we developed a XML language for describing pedagogical content and an exploratory environment, as means for structuring the courseware. In this paper we describe the XML language and its convertion to HTML, introducing graphical layout and navigation. We describe also the exploratory environment and its features for assisted execution, visualization of the file system and automatic grading of exercises.

Palavras-chave: E-Learning, ICT, XML, exploratory environment

1 Motivação e abordagem pedagógica

A colocação de alguns alunos do primeiro ano nos cursos da Universidade, ocorre frequentemente depois do início do ano lectivo. Por esse facto esses alunos perdem um número considerável de aulas onde são abordados conceitos fundamentais necessários para a compreensão das restantes matérias. Apesar de muitos dos docentes disponibilizarem na Web conteúdos pedagógicos usados nas aulas presenciais, como apontamentos, folhas de execícios, sumários, bibliografia, etc, tem-se verificado a insuficiência desse tipo de informação na recuperação do atraso dos alunos.

Esta foi a nossa principal motivação para desenvolver conteúdos pedagógicos na Web para o módulo inicial duma disciplina introdutória de Ciência de Computadores. Este módulo é uma introdução ao sistema UNIX e aos seus comandos básicos de manipulação de ficheiros e directórios. Neste trabalho exploramos novas formas de organização de conteúdos na Web por forma a melhorar a sua eficiência em termos pedagógicos.

Habitualmente são destacadas as vantagens da utilização das Tecnologias de Informação e Comunicação (ICT na sigla inglesa) sendo esquecidos os problemas associados [7]. O tipo de interacção usada nos computadores é bastante diferente da tradicional em papel, exigindo que os conteúdos em formato electrónico se adaptem às suas características, nomeadamente:

A leitura no écran é bastante mais lenta e menos agradável do que a leitura em papel. O leitor duma página num écran perde cerca de 79% do tempo no reconhecimento da estrutura do texto, (títulos, listas itemizadas palavras destacadas) e apenas 16% do tempo na leitura de texto corrido[2].

A escrita com teclado é mais restritiva que a escrita em papel, nomeadamente na utilização de notação complexa (como a notação matemática) e diagramas.

A navegação usando o rato conduz a uma maior dispersão do aluno que constantemente procura um apontador que o leve a uma nova página.

Contudo, os computadores abrem novas possibilidades no desenvolvimento de conteúdos educativos, principalmente pela capacidade de processamento: avaliação automática, simulação de processos e animação. Também as capacidades de navegação, quando guiadas pelos objectivos do conteúdo, podem ser uma mais valia do uso destas tecnologias.

Para tirar partido destas vantagens, minimizando as dificuldades da interacção, é necessário a reformulação das formas de exposição, transmissão e avaliação dos conhecimentos. Não é suficiente simplesmente converter os materiais pedagógicos disponíveis em papel para uma versão HTML. Esta abordagem resulta em textos longos, de difícil leitura, que os alunos rapidamente abandonam por falta de elementos interactivos. A nossa abordagem para resolver este problema baseou-se em:

Seccionar o conteúdo de modo a reduzir a quantidade de texto por página e guiar a interacção, introduzindo automaticamente a formatação e navegação num texto denso.

Usar ambientes de exploração para ilustrar e consolidar os conceitos apresentados no texto, fazendo uso de animação e correcção automática de exercícios.

Na secção seguinte descrevemos a organização dos conteúdos e a sua interligação com o ambiente de exploração e outros elementos interactivos. Na Secção 3 apresentamos um ambiente de exploração desenvolvido para o módulo de introdução ao interpretador de comandos do `UNIX`.

2 Organização dos conteúdos

Para a apresentação dos contéudos é necessário por um lado a definição da estrutura das unidades didácticas e seu desenvolvimento, e por outro lado a sua ligação com as componentes interactivas e de formatação.

Esta separação tem várias vantagens imediatas:

- quem produz os conteúdos não necessita de estar familiarizado, ou preocupar-se, com as questões de apresentação gráfica e a interactividade
- as diferentes componentes podem ser reutilizadas, desenvolvidas e alteradas em separado
- a possibilidade de extracção de apenas algumas componentes de um documento
- o mesmo contéudo pode ter diversas apresentações: para visualização num ecrã (e na Web) ou para impressão em papel; produção de índices de contéudos, remissivos ou glossários; etc.

Esta abordagem evita que o professor que organiza o conteúdo tenha de conhecer os detalhes da linguagem de formatação (HTML). A introdução da formatação é feita por uma linguagem que permite a *colagem* das diferentes componentes e a sua disponibilização que é desenvolvida por alguem com conhecimentos técnicos, por exemplo um *designer* gráfico.

Especialmente adaptadas a estes requisitos são as linguagens `XML`[1] [4] e as tecnologias associadas. Estas linguagens permitem estruturar a informação de documentos e facilitam a sua portabilidade e tornando-os independentes dos sistemas em que foram produzidos.

As linguagens `XML` são sintaticamente muito parecidas à linguagem HTML pois são baseadas no mesmo conceito de anotação. No entanto o uso duma linguagem especializada `XML` torna-se mais fácil por estar mais direcionada para o domínio do documento e frequentemente ter um número menor de anotações. Depois da linguagem definida é fácil escrever documentos usando editores genéricos que auxiliam a introdução e a validação das anotações.

Na Figura 1, apresentámos um esquema que ilustra as diferentes fases de desenvolvimento:

- O contéudo, isto é, o desenvolvimento da matéria que constituí o curso, é escrito num ficheiro de texto, como que se dum livro se tratasse, mas anotado de modo a reflectir a estrutura das unidades didácticas, os seus objectivos e a possível interligação com elementos interactivos. Para tal foi criada em `XML` uma linguagem, que denominamos `EDIC`, e que define a estrutura das anotações. Será descrita na Secção 2.1.
- Aplicando a transformação ao documento que contém o contéudo, obtém-se finalmente os documentos que serão acessíveis ao utililizador e que neste caso, são documentos `HTML` cujos estilos e organização de apontadores reflete a definida pela transformação `XSL`[2] [5].

A apresentação do contéudo é dividida em duas partes: a estruturação associada às diferentes navegações e a formatação gráfica. Em vez de se construir directamente estas componentes para cada contéudo, foi definido em `XSL` um conjunto de regras de transformação dos documentos `EDIC`. Estas regras permitem a selecção das várias componentes do documento original e a sua manipulação de modo à criação de novos documentos, neste caso, em formato `HTML`. Serão apresentadas na Secção 2.2.

As especificações de formatação gráfica foram definidas usando a linguagem de descrição de folhas de estilo `CSS`[3] e que são habitualmente interpretadas pelos navegadores de `HTML` actuais.

[1] *Extensible Markup Language*
[2] *Extensible Stylesheet Language*
[3] Cascade Style Sheets

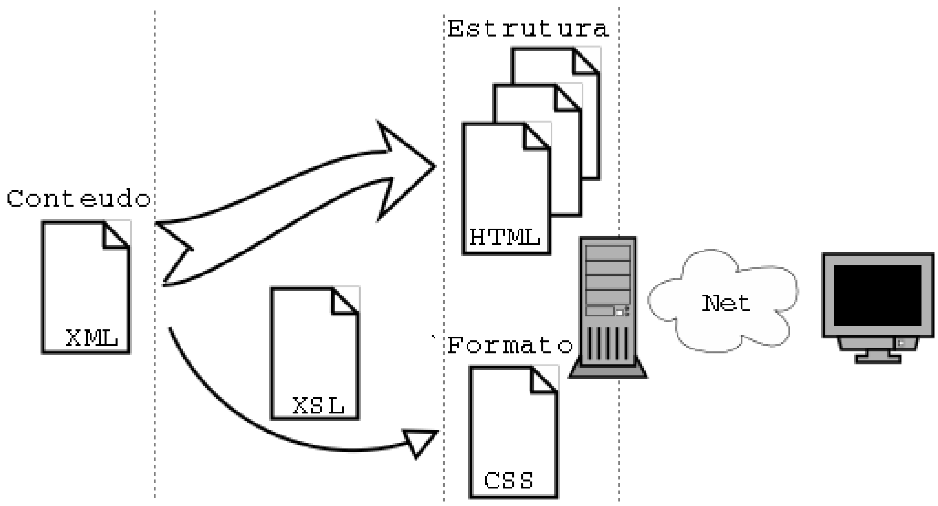

Figura 1: Fases de desenvolvimento

2.1 A linguagem `EDIC`

Como parte do projecto `EDIC` foi desenhada uma linguagem destinada à descrição de conteúdos educativos, focando na sua componente semantica e retirando toda a componente de formatação e navegação estrutural. Para esse efeito foi desenvolvida uma linguagem `XML`. Os documentos desenvolvidos numa linguagem `XML` são facilmente analisados e convertidos noutras linguagens usando utilitários já disponiveis. Assim, a opção por este standard proteje os conteúdos relativamente à obsolescência do *software*.

A utilização de linguagens baseadas no `XML` é hoje uma tendência consolidada , existindo várias propostas de linguagens deste tipo para a descrição de conteúdos pedagógicos[6, 3, 1]. No entanto estas linguagens são geralmente mais orientadas para a descrição da meta-informação que caracteriza os conteúdos pedagógicos, ou na descrição de testes de escolha múltipla. Sendo assim optamos pela definição duma nova linguagem adaptada aos nossos requisitos. Embora o nosso objectivo seja produzir documentos para um módulo de introdução ao sistema de operação `UNIX`, ao desenhar a estrutura dos contéudos podemos usar uma abordagem genérica que possa ser refinada em cada caso.

O `XML` permite estruturar um documento como uma árvore. Esta estrutura é conveniente para descrever um conteudo pedagógico. Por exemplo, um curso pode ser dividido em módulos, cada módulo em várias páginas e cada página pode ter texto, imagens ou outros objectos interactivos, como pode ser visto na Figura 2

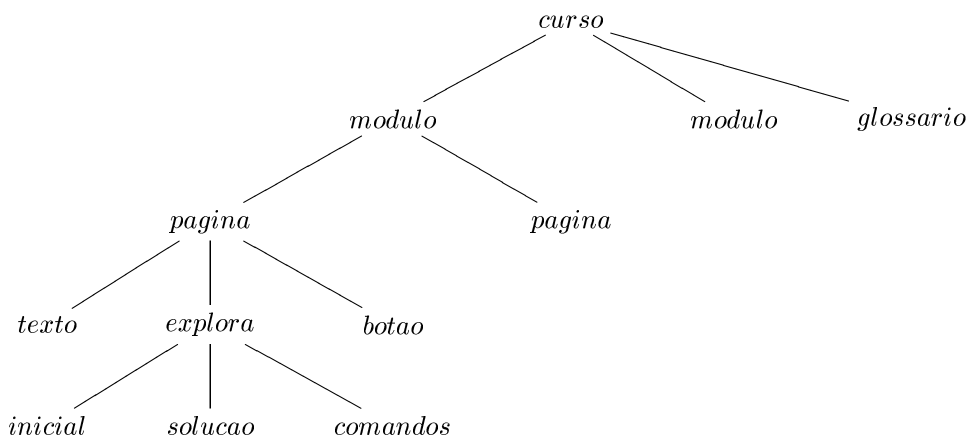

Figura 2: Árvore associada a um documento `EDIC`

Um documento `XML` é uma hierarquia de elementos caracterizado por zero ou mais atributos. Cada elemento é caracterizado por um nome, eventualmente seguido de atributos e definido por anotações de início e fim:

```
<nome_elemento atrib1='val1' atrib2='val2'... > .... </nome_elemento>
```

A estrutura desta hierarquia é habitualmente representada por uma DTD[4]. Na Figura 3 apresentamos um fragmento da DTD usada como base do EDIC e na Figura 4 um documento satisfazendo essa especificação. Na DTD o valor #PCDATA (ou CDATA) indica que o conteúdo do elemento é texto e os operadores pós-fixos *, + e ? indicam, respectivamente, zero ou mais ocorrências do elemento, uma ou mais ocorrências e zero ou uma ocorrência.

```
<!DOCTYPE curso [
<!ELEMENT curso (modulo+,glossario)>
<!ELEMENT modulo (pagina+)>
<!ELEMENT glossario (define*)>
<!ELEMENT pagina (texto?, explora?, botao*)>
<!ELEMENT explora (inicial?, solucao?, comandos?)>
<!ELEMENT define (#PCDATA)>
<!ELEMENT botao (#PCDATA)>
<!ELEMENT texto (#PCDATA)>
<!ELEMENT inicial (#PCDATA)>
<!ELEMENT solucao (#PCDATA)>
<!ELEMENT comandos (#PCDATA)>
<!ATTLIST pagina id CDATA #IMPLIED
         titulo CDATA #IMPLIED>
<!ATTLIST botao tipo CDATA #IMPLIED>
<!ATTLIST define termo CDATA #IMPLIED>
]>
```

Figura 3: Um fragmento do DTD para a linguagem EDIC.

O elemento **pagina** é o elemento fundamental da linguagem permitindo a divisão do contéudo em pequenas unidades. Os seus sub-elementos que reflectem a interligação do texto com o ambiente de exploração são:

texto que contém o desenvolvimento da matéria e a proposta de exercícios a realizar.

explora que determina a interacção com o ambiente de exploração. O seu contéudo relaciona-se directamente com os componentes desse ambiente. Neste caso, e de acordo com a descrição do ambiente de exploração da Secção 3, temos três sub-elementos:

- **inicial** inicializa o estado inicial do ambiente de exploração. Neste caso, uma árvore de diretórios do sistema.
- **solucao** contém uma possível solução do exercício proposto nessa página, que será usada no ambiente de exploração quer na correção automática quer para ser visualizada pelo aluno.
- **comandos** lista de comandos que foram introduzidos ao aluno e que poderão ser usados na resolução dos exercicios.

Cada página poderá ainda ter um ou mais elementos de nome **botao** que corresponderão a objectos gráficos interactivos (p.e. botões) que quando accionados fornecerão informações como o resumo ou os objectivos de cada página ou conjunto de páginas.

Os atributos do elemento **pagina** são usados para a estruturação dos ficheiros HTML, sendo o valor do atributo id usado para o nome do directório que conterá a informação da página a visualizar. Este atributo e o atributo **titulo**, também são usados para a criação do índice de conteúdos.

O elemento **glossario** contém as definições dos conceitos básicos as quais poderão ser acedidas anotando, no texto, as palavras associadas a esses conceitos.

2.2 Apresentação dos contéudos

Actualmente, para a apresentação de um documento XML é habitual sua transformação para HTML (para visualização no ecrã) ou para LaTeX ou PDF para a impressão em papel. Essa transformação pode ser feita por programas escritos em linguagens de programação (Perl, Java, Tcl, Python), a

[4] *Document Type Definition*

```
<curso>
<glossario>
<define termo="ambiente de exploração">
Programa de apoio ao curso que permite explorar os conceitos apresentados.
</define>
</glossario>
<modulo>
<pagina id="unix1" titulo="Interpretador de comandos">
<botao tipo="Objectivos">
Introdução à shell de UNIX e aos seus comandos básicos de manipulação
de ficheiros e directórios.
</botao>
<texto>
No <ref>interpretador de comandos</ref> o utilizador pode executar linhas de
comando.  Uma linha de comando pode ser comparada a uma frase em
português que exprime uma ordem para o computador executar.<p/>
Um exemplo dum comando UNIX é o comando <gloss>date</gloss>.  Escreve na
consola a palavra "date" e carrega na tecla RETURN.<p/>
O sistema responde escrevendo a data e a hora corrente do computador.
</texto>
<explora>
<solucao>date </solucao>
<comandos>date</comandos>
</explora>
</pagina>
</modulo>
</curso>
```

Figura 4: Um documento EDIC.

partir bibliotecas básicas. Mas pode ainda ser feita utilizando linguagens de descrição de folhas de estilo como o XSL. Este tem a vantagem de usar a sintaxe do XML e ser subdivida em duas componentes: uma que permite a transformação do documento e outra que formata o documento transformado. Um documento em XML é visto como a *árvore fonte*, cujos nós são os elementos (ver Figura 2), à qual são aplicadas transformações obtendo-se uma *árvore resultado*. Para tal são definidas regras em que aos elementos que satisfaçam determinados padrões são aplicadas funções e instanciados modelos pre-definidos (*templates*, em inglês) para a construção da nova árvore.

Usando estas transformações é possível seleccionar, reordenar e até acrescentar nova informação aos novos objectos produzidos. Assim é possível construir várias navegações do documento original, permitindo a inteligação das várias apresentações: índice de contéudos, índice remissivo, glossário e desenvolvimento do texto anotado. Cada elemento **pagina** dum documento EDIC irá corresponder a um ficheiro HTML que será visualizado numa página.

Para a comunicação com o ambiente de exploração é criado, para cada página, um ficheiro de configuração, a partir do contéudo dos elementos **explora**. Actualmente o ambiente de exploração é activado sempre que é selecionada uma nova página e o respectivo ficheiro de configuração consultado, como é descrito na sub-secção 3.

Na Figura 5 apresentámos uma página do EDIC como é visualizada num navegador HTML. É composta por três areas distintas: o índice de conteúdos, a página de contéudo e o glossário. Em cada página de contéudo, existem apontadores para as páginas anterior e seguinte, para além dos botões, eventuais imagens e apontadores para o glossário e índice remissivo. O ambiente de exploração aparece numa janela autónoma.

3 Ambiente de exploração

O Ambiente de Exploração (AE) é um componente central do sistema cujo principal objectivo é a exploração dos conceitos apresentados no conteúdo do EDIC. O AE tem também um papel importante na estruturação desses conteúdos.

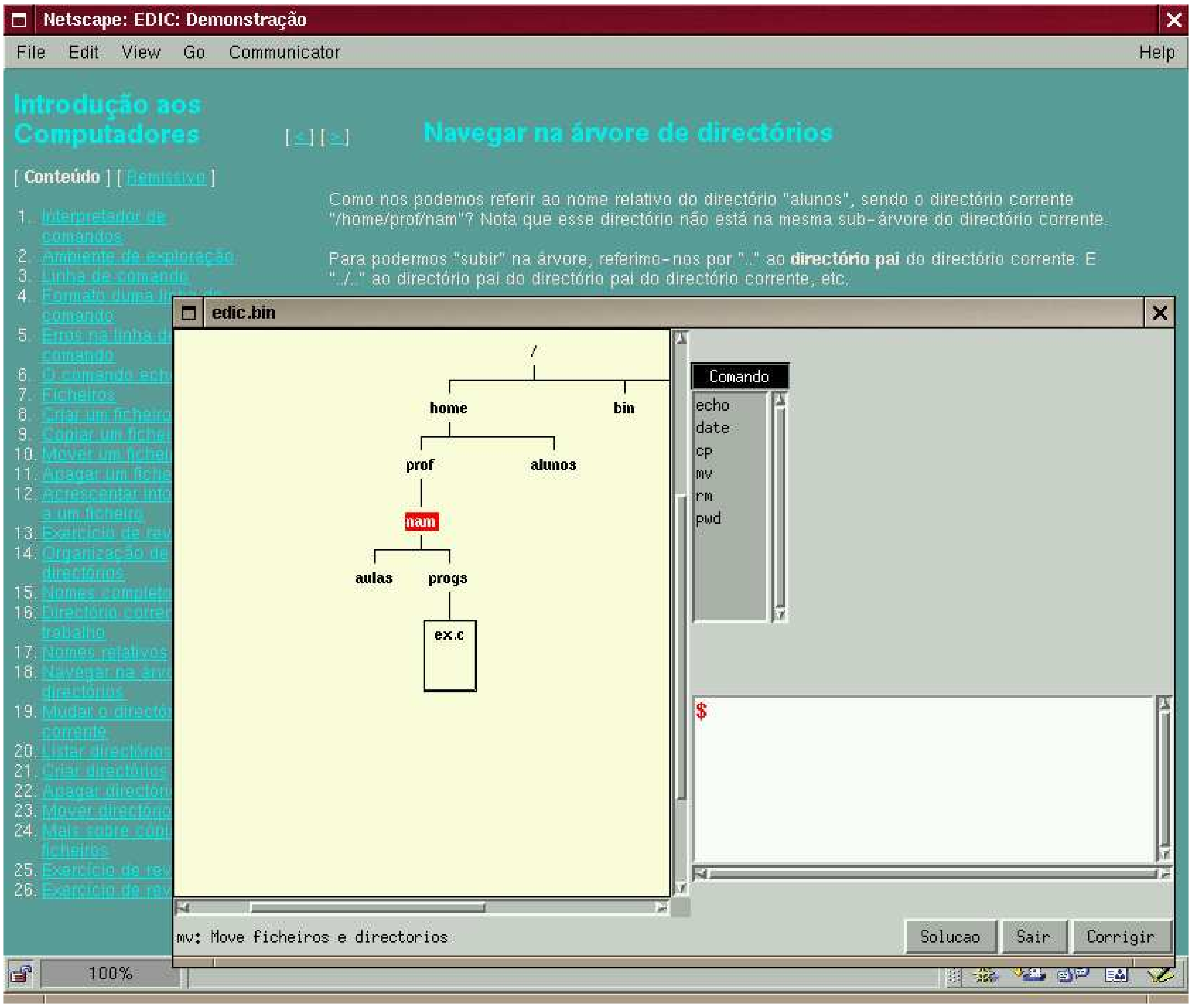

Figura 5: O conteúdo num navegador WWW e o ambiente de exploração numa janela autónoma

Numa primeira análise o `AE` pode ser visto como um programa educativo que poderia acompanhar qualquer conteúdo pedagógico com os objectivos de `EDIC` (em suporte electrónico ou em papel) ou até ser usado isoladamente. Enquanto componente autónomo o `AE` tem por objectivo explorar as possibilidades dum interpretador de comandos `UNIX`, auxiliando na:

- Edição de comandos
- Visualização do seu efeito
- Ajuda na resolução de problemas

Para além da sua possível utilização autónoma, o `AE` foi pensado para complementar os conteúdos apresentados no `EDIC`, tendo também um papel na sua estruturação. Os objectivos do `AE` na estruturação dos conteúdos do `EDIC` incluem:

- Ilustração dos conceitos
- Consolidação dos conhecimentos
- Redução da quantidade de texto por página
- Aumento da interactividade

No restante desta secção começamos por descrever a abordagem seguida no desenvolvimento dum sistema com estas características. Seguidamente descrevemos o próprio programa tendo por base a interface gráfica. Finalmente descrevemos os mecanismos de integração com o restante sistema.

Abordagem

O desenvolvimento do `AE` do `EDIC` foi baseado em três conceitos principais, com os quais pretendemos responder aos objectivos definidos anteriormente. Esses conceitos principais são:

- Visualização do estado
- Escrita assistida de comandos
- Correcção automática de exercícios

Os interpretadores de comandos permitem executar comandos sobre o sistema de operação, em particular sobre o sistema de ficheiros, mas não fornecem nenhuma forma de visualização do estado do mesmo. É conveniente referir que a visualização do estado é desnecessária, e até inconveniente, para uma utilização efectiva dum interpretador de comandos. No entanto, para um utilizador inexperiente a visualização do estado do sistema de ficheiros permite compreender melhor as alterações provocadas pela execução dos comandos.

O estado do sistema de ficheiros pode ser representado graficamente na forma duma árvore de directórios. Nesta visualização são incluídos directórios e ficheiros bem como as ligações destes aos respectivos directórios progenitores.

Esta representação estática do sistema de ficheiros pode ser complementada com o uso de **animação** para visualizar as transições entre estados sucessivos. Desta forma são destacados os efeitos dos comandos que alteram o sistema de operação (por exemplo, que criam directórios ou removem ficheiros).

Os comandos textuais reconhecidos pelo interpretador são uma forma concisa e poderosa de interagir com o sistema de operação. No entanto, a sua utilização exige a memorização dum vocabulário vasto, incluindo não só o nome dos vários comandos mas também as opções e argumentos de cada um.

Numa fase inicial de aprendizagem, é necessário recordar ao aluno o nome e sintaxe dos comandos disponíveis, e assistir na escrita de linhas de comando. No entanto é importante que esses mecanismos não substituam completamente o texto e a edição textual da linha de comandos, já que é essa forma de interacção que constitui o objectivo da aprendizagem.

Um mecanismo de escrita assistida reduz os erros sintácticos mas não impede a escrita de comandos errados. Para isso é necessário introduzir a correcção automática dos exercícios propostos. A correcção dos exercícios permite ao aluno verificar se assimilou correctamente os conceitos expostos.

A correcção dos exercícios não poderá ser feita simplesmente analisando o textos das linhas de comando escritas pelo aluno, dado que geralmente existem diferentes formas de resolver o mesmo problema. No entanto o objectivo final, o estado do sistema de ficheiros, está bem definido e para corrigir o exercício basta analisar o estado final do sistema de ficheiros.

Programa

O AE é um programa autónomo, que funciona como um interpretador de comandos de UNIX sem requer a instalação deste sistema de operação. O AE tem por base um **simulador** dum sistema de ficheiros que permite a execução de comandos básicos, como se fossem realizados num verdadeiro interpretador, e serve de base à visualização do estado do sistema.

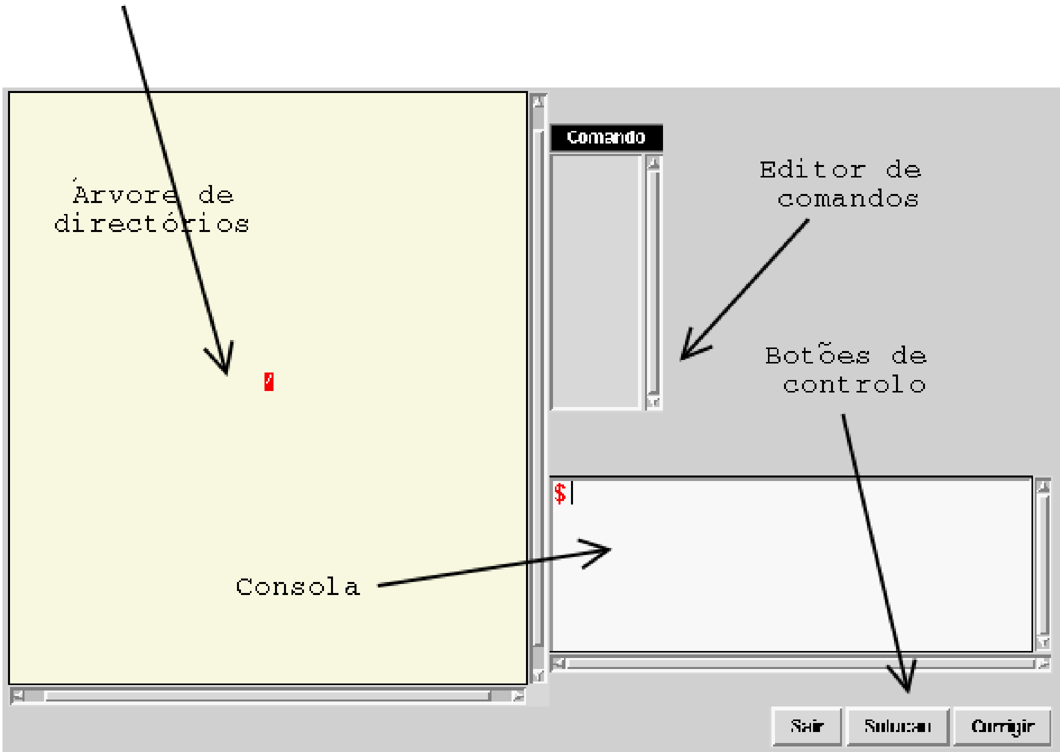

Figura 6: Interface do Ambiente de Exploração

A interacção com o simulador, quer para a execução de comandos quer para a visualização do estado, e feita através da interface gráfica do programa. A interface gráfica do AE, conforme pode ser visto na Figura 6, é composto pelos seguintes componentes:

- Consola
- Árvore de directórios
- Editor de comandos
- Botões de controlo

Seguidamente cada um destes componentes é descrito individualmente, destacando-se a sua forma de visualização e mecanismos de interacção. No entanto é conveniente frisar que estes componentes estão interligados, no sentido em que a interacção realizada com cada um afecta os restantes.

Consola A consola é o componente central da interface do AE pois simula o funcionamento dum interpretador de comandos UNIX. Tem por objectivo precisamente ajudar o aluno a usar interpretadores de comandos cuja interface se reduz a um equivalente da consola. Na interface gráfica do AE ocupa um rectângulo no canto inferior direito. Na consola o aluno pode digitar comandos directamente, sendo o resultado idêntico ao que obteria num interpretador de comandos. Para maior veracidade foi tomando como referência a bash (*Bourne Again Shell*) [9], em particular em relação às mensagens de erro.

A consola relaciona-se com os restantes componentes da interface. A linha de comandos, em vez de digitada directamente na consola, pode ser escrita através do editor de comandos. Depois de executada a linha de comandos as alterações ao estado do sistema de ficheiros são visualizadas na árvore de directórios.

Árvore de directórios A árvore de directórios apresenta na área à esquerda da interface todos os directórios e ficheiros definidos no simulador. Esta área é actualizada a cada execução duma linha de comando: primeiramente é executada uma sequência de animação que destaca graficamente as alterações introduzidas e finalmente toda a árvore é redesenhada.

Na árvore os directórios são representados graficamente por uma etiqueta com o seu nome, com ligações aos seus sub-directórios e ficheiros (ver Figura 5). Os ficheiros são representados por rectângulos, dentro dos quais é apresentado o seu conteúdo, encimado pelo respectivo nome.

O directório corrente de trabalho é destacado na árvore de directórios apresentando o seu fundo a vermelho. Cada vez que é redesenhada, a árvore de directórios é centrada no directório corrente; caso esta não possa ser integralmente visualizada no rectângulo em torno da directório corrente podem ser usadas *scrollbars* para navegar na totalidade da área de directórios.

Embora o principal objectivo da árvore de directórios seja a visualização do estado do sistema de ficheiros, é também usada na edição de comandos. Por exemplo, o aluno pode apontar (com o rato) para qualquer ficheiro ou directório para indicar um argumento. Os caminhos na árvore de directórios construídos no editor de comandos são também visualizados na árvore de directórios.

Editor de comandos O editor de comandos situa-se no canto superior direito do `AE` e permite a escrita assistida de comandos `UNIX`. Este componente permite ao aluno escrever uma linha de comandos simplesmente apontando para os elementos dum conjunto de listas de selecção.

Uma linha de comandos `UNIX` é constituida por uma sequência de palavras em que a primeira palavra corresponde ao comando (a acção a realizar) e os seguintes aos elementos sobre os quais este opera. O editor de comandos permite sucessivamente seleccionar de listas construídas dinamicamente, cada uma dessas palavras.

Num dado momento são apresentados ao aluno apenas os comandos já introduzidos no texto. Quando um dos comandos é apontado pelo aluno é apresentado no rodapé do `AE` uma breve descrição da sua função.

Botões de controlo No canto inferior direito existem três botões que permitem o controlo do `AE`, permitindo sair do programa, obter uma solução do último exercício proposto e corrigir a resolução do aluno face a essa solução. A abordagem seguida para a correcção automática foi desenvolvida pelos autores como parte dum outro projecto de investigação [8].

Internamente o `AE` possui duas árvores de directórios. Uma é modificada pelos comandos executados pelos aluno e é visualizada no interface. A outra é modificada pela solução do sistema e não é visualizada. Quando é pressionado o botão de correcção estas duas árvores são comparadas e se existirem diferenças estas são comunicadas ao aluno, que poderá assim corrigir a sua resolução.

Integração

Como foi referido, o `AE` funciona em coordenação com o conteúdo do `EDIC`, permitindo a sua utilização na ilustração dos conceitos apresentados e a resolução assistida dos exercícios propostos. Esse facto requer do `AE` a capacidade de comunicar em ambos os sentidos com um navegador Web onde são apresentados os conteúdos, permitindo ao navegador alterar o estado no `AE` e vice-versa, em função da interacção do aluno.

No desenvolvimento do `AE` foram considerados dois modelos de integração com navegador: a integração nas páginas do navegador como um componente um externo ("`plugin`") e a colaboração como uma aplicação externa ("`helper application`"). Foram feitos protótipos de ambas as abordagens e optou-se por esta última. A primeira abordagem (usando o "`plugin`") foi abandonada por apresentar vários inconvinientes: maior instabilidade, demora na comunicação de alterações vindas do navegador e excessiva ocupação de espaço na formatação da página. O maior inconveniente da abordagem escolhida é a necessidade dum passo prévio de instalação do `AE` antes da utilização do `EDIC` na Web.

A comunicação entre o navegador e o `AE` é feita abrindo no navegador um ficheiro com o tipo MIME `application/edic`. O navegador deve estar previamente configurado para enviar este ficheiro para a aplicação instalada no sistema. Se já existir um processo `AE` em funcionamento este será reconfigurado com os dados enviados pelo ficheiro. Caso contrário, um novo processo do `AE` entrará em funcionamento. Tipicamente é enviado um novo ficheiro para o `AE` sempre que uma página de conteúdo é visualizada.

A informação comunicada por esta via influência o `AE` de várias formas. Primeiro, é reconfigurado o estado inicial do `AE` actualizando a árvore de directórios. Desta forma o interface do `AE` é usado para ilustrar conceitos expostos no texto. Seguidamente é alterada a lista de comandos disponíveis para selecção no interface do editor de comandos. Desta forma evita-se que o `AE` sugira a utilização de comandos que ainda não foram introduzidos. Finalmente é actualizada a resoluçao do exercício definido na página corrente. Esta resolução pode ser obtida na consola do `AE` usando os botões de controlo e também é usada na correcção automática, conforme explicado anteriormente.

A comunicação entre o `AE` e o navegador é feita pela utilização de apontadores. Nas mensagens de erro dos comandos são usados apontadores para referir as páginas relevantes do conteúdo do `EDIC`. Por exemplo, uma mensagem de erro resultante duma má utilização do comando `ls` inclui um apontador para a página do `EDIC` onde este comando é explicado, permitindo ao aluno consultar novamente esta matéria.

4 Conclusões

Neste artigo apresentamos um trabalho ainda em curso que visa o desenvolvimento dum módulo para ensino via Web. No estado actual da investigação não ainda possível efectuar qualquer avaliação dos objectivos propostos. Essa avaliação só será possível de concretizar no início do próximo ano lectivo, comparando os resultados dos alunos que utilizarem este sistema com os alunos colocados tardiamente nos anos anteriores.A fase actual dos sistema está disponível na Web em `http://www.ncc.up.pt/ zp/edic`.

No entanto pensamos que a metodologia apresentada pode ser usada com vantagem na adaptação para cursos Web outros conteúdos pedagógicos. Em particular a linguagem XML desenvolvida facilita o trabalho de estruturação dos conteúdos em páginas Web menos densas e de fácil navegação. A utilização de ambientes de exploração também nos parece recomendável, requerendo em geral a implementação de raiz por um programador. Todavia é cada vez mais frequente encontrar ambientes deste tipo disponíveis na Internet que poderão ser adaptados para o efeito.

Agradecimentos Agradecemos ao Gabinete de Ensino à Distância da Reitoria da Universidade do Porto o apoio na contratação dum assistente de investigação para o desenvolvimento deste trabalho.

O tabalho apresentado foi parcialmente suportado por fundos atribuidos ao LIACC pelo do Programa de Financiamento Plurianual, Fundação para a Ciência e Tecnologia e Programa PRAXIS.

Referências

[1] Adl scorm. `http://www.adlnet.org/`.

[2] How users read on the web. `http://www.useit.com/alertbox/9710a.html`.

[3] Ims content packaging. `http://www.imsproject.org/`.

[4] Página www da especificação de xml. `http://www.w3.org/TR/REC-xml`.

[5] Página www da especificação de xsl. `http://www.w3.org/Style/XSL/`.

[6] Universal learning format. `http://www.saba.com/standards/ulf/Overview/Frames/overview.htm`.

[7] Piet A. Kommers. Multimedia environments. In Piet A. Kommers, Scott Grabinger, and Joanna C. Dunlp, editors, *Hypermedia learning environments*. LEA Publishers, 1996.

[8] José Paulo Leal and Nelma Moreira. Using matching for automatic assessment in computer science learning environments. In Francisco Restivo and Lígia Ribeiro, editors, *Web-Based Lerning Environments*. Merlin 2000 Project, FEUP, June 2000.

[9] Cameron Newham and Bill Rosenblatt. *Learning the bash shell*. O'REILLY, 2 edition, 1998.