

O Uso de *Templates* na Customização de Ambientes Colaborativos

Pedro David Netto Silveira

Instituto Federal do ES
Rodovia ES-482 Km 47, Rive - Alegre
Caixa Postal 47. CEP 29500-000
+55 28 3552 8131
pedro.silveira@ifes.edu.br

Crediné Silva de Menezes

Davidson Cury
Universidade Federal do ES
Av. Fernando Ferrari, 514,
Goiabeiras, Vitória - ES, 29075-910
+55 27 4009 2222
{credine,dede}@inf.ufes.br

Flavio Pavesi Simão

Instituto Federal do ES
Rodovia ES-482 Km 47, Rive - Alegre
Caixa Postal 47. CEP 29500-000
+55 28 3552 8131
fpsimao@ifes.edu.br

ABSTRACT

This article describes a solution to the problems of flexibility in the presentation layer of virtual environments on the Web. In many cases, for a better cognitive adaptation to the environment a custom view is required. In current virtual environments, there is a lack in this aspect. We then will demonstrate an architecture description for customizing the interface and providing a visual perception of contents tailored to the participants of the environment.

RESUMO

Este artigo descreve uma solução para os problemas de flexibilidade na camada de apresentação de ambientes virtuais na web. Em muitos casos, para uma melhor adaptação cognitiva ao ambiente é necessária uma visualização personalizada. Nos ambientes virtuais correntes, nota-se carência nesse aspecto. Demonstraremos então, uma arquitetura para descrição e personalização da interface visual que proporcione uma percepção adaptada do conteúdo aos participantes do ambiente.

Categories and Subject Descriptors

H.1.2 [User/Machine Systems]: Human factors, Human information processing, Software psychology.

General Terms

Design, Experimentation, Human Factors, Standardization, Verification.

Keywords

Customization, template, flexibility, adaptation.

1. INTRODUÇÃO

A cada dia cresce na web a formação de grupos de usuários com o intuito de compartilhar informações, formar opiniões, adquirir conhecimento e colaborar à distância, das mais diversas formas. Um dos principais focos atuais de pesquisa nessa área busca criar ambientes computacionais para apoiar o trabalho em grupo (CSCL). Nesta direção, inúmeras iniciativas de sucesso podem ser encontradas na literatura [9]. Contudo, existe uma demanda singular que se refere ao atendimento particular das necessidades ou das preferências dos participantes. Nesse aspecto, os espaços virtuais colaborativos existentes podem frustrar seus participantes, justamente pela ausência de flexibilidade e adaptabilidade necessárias para atender a diferentes propostas de trabalho que exijam recursos diversos.

Com base em [1], que destaca a importância da adaptação das interfaces para usuários individuais e em [3], que relatam como o uso do computador pode ser menos cansativo e mais eficaz, apresentaremos o Editor de *Templates* (ET). ET é uma ferramenta do Projeto MORFEU [7] que proporciona edição inteiramente customizada sobre a aparência resultando em diferentes visões do ambiente. Sua intenção é proporcionar aos usuários interações mais simples e agradáveis a fim de facilitar a colaboração.

O artigo está organizado da seguinte forma: na Seção 2 será apresentada uma breve revisão bibliográfica e alguns trabalhos correlatos. Na Seção 3 descreveremos o meta-ambiente Projeto MORFEU. A Seção 4 descreverá o Editor de *Templates*. Na Seção 5 será apresentado o protótipo do editor e, finalmente, na Seção 6 serão descritas as conclusões e os trabalhos futuros.

2. REVISÃO BIBLIOGRÁFICA E TRABALHOS CORELATOS

Um dos objetivos do projeto da interface é alcançar alto grau de qualidade de interação entre o usuário e o ambiente. Para se obter interações produtivas é necessário o projeto da interação humano-computador (IHC). Segundo [11], o projeto de interação é um processo no qual o objetivo é desenvolver um espaço de trabalho por meio de uma interface, onde o modelo mental do usuário seja correspondido pelo modelo do programa. A interface gerada a partir deste projeto é basicamente uma resposta às necessidades do usuário dentro do contexto abordado. Nesse sentido, citamos alguns trabalhos que têm como foco a interface com o usuário e que tratam o objeto “apresentação” como moldável.

No projeto SUPPLE [2] existe a premissa que a renderização de uma interface deve refletir as necessidades e padrão de uso de usuários individualmente. Dada variedade de dispositivos, necessidades pessoais e estilos de interação, é inviável para os programadores humanos criar interfaces para cada tipo de dispositivo e todo perfil de usuário. Em vez disso, uma solução automatizada é necessária. Como *templates* no Projeto MORFEU, SUPPLE oferece possibilidade de diferentes interfaces para uma mesma estrutura, mas a geração delas é tratada como um problema de otimização, no qual um algoritmo decide a interface que o usuário deve usar. No Editor de *Templates* o usuário é quem define a apresentação de seus dispositivos.

O MyXML [5] é um sistema baseado em tecnologias XML/XSL, sendo o conteúdo representado em documentos XML bem estruturados. Toda informação de layout é adicionada em arquivos XSL separados, o que proporciona separação entre layout e conteúdo. MyXML também faz a separação da lógica de negócios

do sistema oferecendo uma interação dinâmica para os usuários. É uma solução interessante, porém, ao contrário do ET, não apresenta uma solução gráfica para as operações dos indivíduos, ou seja, tudo é feito com codificação, uma tarefa inacessível ao usuário final não familiar com técnicas de programação.

De acordo com [11], ao centrarmos o projeto no usuário final, concedemos a ele poder que este não tinha antes: de influir qualitativamente no processo de projetar interfaces de usuário, como um amador projetando um jardim ou a melhor maneira de se colocar uma cerca. Com a colaboração ativa dos usuários no processo, ganhamos um importante aliado para a melhoria da qualidade do produto final.

Flexibilidade em interface somada ao conceito de consideração da experiência individual, resumem uma nova definição: adaptabilidade. Adaptação refere-se à capacidade de reagir conforme o contexto e em consonância com as necessidades e preferências do usuário.

Os utilizadores não devem ser obrigados a trabalhar com sintaxe; deve-lhes ser permitido brincar com imagens, formas, cores e sons. Os utilizadores de computadores não deverão ter que se preocupar com a complexidade de uma linguagem de programação; devem ser-lhes fornecidos objetos virtuais que possam ser manipulados tão diretamente quanto possível. [13]

3. O PROJETO MORFEU

Tendo em vista a plasticidade dos ambientes virtuais e as novas possibilidades de automatização de tarefas mecânicas, a meta é buscar a concepção de novos ambientes, pautados pelos seguintes aspectos: plasticidade, ergonomia, adaptabilidade ao perfil do usuário, redução da repetição de trabalho e redução da sobrecarga cognitiva [12]. É neste contexto que surgiu o MORFEU. O objetivo é dar forma ou moldar os espaços de acordo com os interesses, idéias e afetividades dos usuários, sob a premissa de que emoções afetam a aprendizagem [10].

A concepção primordial de MORFEU é o suporte à autoria, à publicação e à socialização das produções intelectuais. Em resumo, suas duas grandes preocupações são a autoria e a veiculação pública do produto da autoria, mas em profundo acordo com o gosto estético do usuário. Assim, ao invés de colocar o foco no uso de ferramentas, dentro de um determinado contexto, sua base é a manifestação dos sujeitos por meio do seu elemento central de autoria, conhecido como Unidade de Produção Intelectual (UPI), usada para registrar as produções dos usuários. Sua ferramenta principal é um editor de UPI. Cada UPI possui um autor, um título e um conteúdo (corpo). O conteúdo pode usar todo e qualquer elemento de representação digital suportado em HTML e disponibilizado no acervo de mídias do ambiente. A qualquer instante o usuário pode criar ou editar uma UPI, sem a preocupação do destino que dará a ela [7].

Tudo que o usuário produz (UPI) estará registrado e versionado pelo ambiente, ou seja, cada edição de uma UPI resulta em uma nova versão independente de sua publicação.

Uma versão de UPI pode ser posteriormente usada e reusada em situações de interação. Por exemplo, uma UPI que foi usada em forma de mensagem para um colega pode também ser usada para publicação em um fórum. De modo contrário, nos ambientes convencionais, o conjunto das produções individuais fica atrelado às ferramentas. Uma mensagem enviada por email é armazenada pelo servidor de email, uma mensagem postada em um fórum faz parte do acervo daquele fórum. Então, se o fórum for excluído, o

autor perde a mensagem. Da mesma forma acontece com as mensagens escritas em um chat. No MORFEU todos estes tipos de mensagens são UPIs e ficam associadas a cada indivíduo.

Toda produção composta de UPIs é tratada por um Veículo de Comunicação (VCom). Cada VCom possui suas diretrizes de composição. Por exemplo, um blog é um VCom no qual as produções são organizadas em forma de pilhas, ou seja, o mais recente está no topo. Na mesma linha, um fórum é um veículo de comunicação no qual as produções estão organizadas em forma de árvore, sendo que uma UPI “responde” a outra UPI ou inicia um novo ramo de publicação.

O principal neste trabalho é a maneira como as UPIs poderão ter sua apresentação projetada nos veículos de comunicação sem a necessidade de conhecimentos de programação.

O *template* é a forma de apresentar os VComs. São responsáveis pelo estilo do espaço virtual, incluindo questões de *design*, *layout* e navegação das UPIs dentro do veículo de comunicação. A ferramenta (integrada ao projeto MORFEU) para construir *templates* é o Editor de *Templates*.

4. O EDITOR DE TEMPLATES

De acordo com [3], construir interfaces agradáveis significa disponibilizar estímulos visuais, tais como cores, formas, texturas e fontes, de maneira equilibrada, com o intuito de não saturar a visão e nem sobrecarregar a capacidade de assimilação das informações manipuladas pelos usuários. O Editor de *templates* foi desenvolvido com a intenção de proporcionar ao sujeito a opção de inventar e descobrir a melhor opção de visualização dos ambientes colaborativos criados no MORFEU, com a intenção de vencer as limitações que dificultam sua participação, ou a tornam monótona

4.1 Contribuição do uso dos *templates*

Para [3], a interface criativa faz surgir uma “filosofia visual” associada, inevitavelmente, à beleza. Em nossa opinião, essa nova filosofia fica mais visível sempre que associada à eficácia e à satisfação do usuário. Entretanto, o conceito de beleza é bastante subjetivo. Como exemplo, citamos uma arquitetura pedagógica chamada de Debate de Teses [6].

No debate de teses, existe a figura do mediador, que insere inicialmente as teses para que outro utilizador do sistema (participante) escreva um posicionamento inicial sobre as teses. Posteriormente, outro participante pode fazer uma revisão sobre esse posicionamento a fim de acrescentar idéias ou causar certo desequilíbrio nas certezas do revisado. Ao findar a etapa de revisão, todos os participantes devem escrever seu posicionamento final sobre as teses.

Da forma como o ambiente foi elaborado, sua apresentação para todos os participantes restringe-se a um formato de grade, ou matriz (Figura 1.a) o que pode não ser “belo” para um indivíduo, gerando desconforto cognitivo. Uma melhor maneira de navegar e enxergar essa arquitetura pode acontecer como na Figura 1.b, onde se vê um debate realizado seguindo uma estrutura de árvore. Em uma terceira opção, o debate poderia ainda ser feito de forma linear vertical, seguindo um curso sequencial.

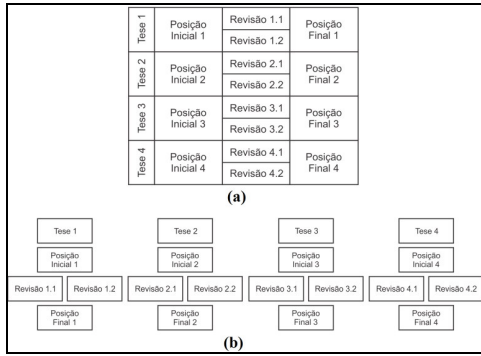


Figura 1. Diferentes formas de visualização da arquitetura pedagógica Debate de teses.

Dessa forma a visualização do ambiente se torna flexível no que tange aos meios colocados à disposição do usuário, permitindo-lhe personalizar a interface a fim de levar em conta as exigências da tarefa, de suas estratégias ou seus hábitos de trabalho.

Uma interface não atende, ao mesmo tempo, a totalidade das pessoas. Então, para que aumente a sintonia entre usuário e ambiente, a apresentação do espaço virtual precisa poder ser adaptada a ponto de não criar barreiras que venham a dificultar o uso do sistema. Então, quanto mais variadas são as maneiras de efetivar uma tarefa, maiores são as chances de o usuário escolher e dominar uma dessas maneiras no curso de seu aprendizado. Devemos, portanto, fornecer-lhe procedimentos, opções e comandos diversificados para a realização de sua tarefa [3].

4.2 Características

As características do ET foram construídas a partir das necessidades comuns aos sistemas editores/gerenciadores de interfaces. Primeiro, a separação da camada de apresentação em relação ao conteúdo [4] e segundo, a percepção sobre quais elementos de interface deveriam ser afetados pelo editor [5].

De acordo com [4], para um rápido desenvolvimento de *websites* flexíveis é necessária a independência entre layout e conteúdo. Nesse sentido o consórcio *World Wide Web*, provê os padrões XML e XSL que visam proporcionar essa separação. Entretanto, faz-se necessário identificar as propriedades da interface visual, o que nos levou a percepção de elementos de interface sujeitos à edição. Em [1] é mencionado que aspectos gráficos do texto, layout de página e navegação no conteúdo estão propensos a mudar em qualquer instante para adaptação do sistema ao usuário.

A partir da relevância da aparência em ambientes virtuais e da pesquisa de interfaces adaptativas, destacamos os requisitos para compor as principais propriedades do Editor de *templates*:

- Não podemos esperar que os usuários operantes no sistema estejam aptos a adaptar a interface às suas necessidades e/ou preferências no nível de codificação, mas devemos fornecer a eles meios para construção da aparência do ambiente. Essa é a principal característica do editor de *templates*;
- Prover total separação entre conteúdo e interface visual. O que facilita a recriação da aparência do ambiente em cada edição;
- Fornecer opção para alterações básicas sobre as UPIs, como por exemplo: cor e tamanho de fonte, cor de plano

de fundo, espaçamento do texto, margens, tamanho da imagem etc.

- Permitir aos participantes do ambiente determinar a disposição dos elementos na página, modificando a visualização dos componentes da estrutura, e definir como navegar dentro dos espaços de produção coletiva.

4.3 Visão geral sobre a arquitetura

O editor foi projetado de maneira que seja de uso simples, exigindo o mínimo de esforço, uma vez que sua interface é representada por uma linguagem gráfica de alto nível. Para a criação da interface, o usuário terá que passar por três módulos pertencentes à arquitetura: módulo de layout, módulo de design e módulo de navegação.

4.3.1 Módulo de layout

A definição do layout é o primeiro passo da criação da aparência do VCom. O participante do ambiente terá a sua disposição uma interface com a qual deverá interagir movendo e redimensionando os objetos pertencentes ao veículo de comunicação

A Figura 2 mostra um exemplo de alteração de layout em um VCom do tipo blog. As definições de layout seguirão um padrão proposto por folhas de estilos *eXtensible Stylesheet Language* (XSL) que, para este caso, são mais poderosas que CSS.

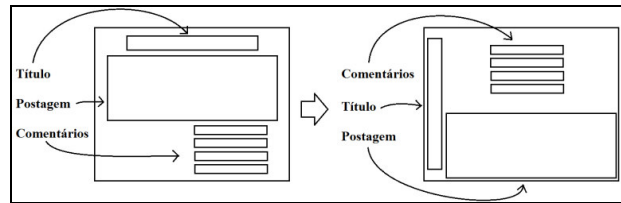


Figura 2. O Layout de Blog que pode ser personalizado em um template no Projeto MORFEU

Posteriormente as folhas de estilo deverão ainda passar por uma transformação XSLT (*eXtensible Stylesheet Language Transformations*). Após a transformação o código será convertido em HTML e poderá ser lido normalmente pelo navegador.

4.3.2 Módulo de design

Design engloba as alterações básicas sobre os componentes da interface, como por exemplo, cor e tamanho de fonte, cor de plano de fundo, espaçamento do texto, margens e bordas, tamanho da imagem etc. Essa funcionalidade pode estar ligada ao gosto pessoal do usuário, ou então a restrição física, como no caso de pessoas com baixa visão ou daltonismo. O design deve seguir às definições de *Cascading Style Sheets* (CSS).

4.3.3 Módulo de Navegação

De acordo com [8] a dificuldade em encontrar informações em um site é, sem dúvida, o problema de usabilidade mais relatado por usuários da web. Isso está relacionado à maneira como a navegação do site em questão foi projetada. Ao elaborar o *template*, o participante do ambiente é quem vai escolher como a navegação será feita e também como ficará a estrutura dos hiperlinks dentro do espaço virtual.

Existem quatro formas principais de fazer transição de conteúdo, isto é, navegação: (i) leitura seqüencial em uma página, (ii) árvore de links, (iii) menus ou (iv) paginação. Usando o exemplo de um jornal online como espaço virtual, o indivíduo poderia navegar

pelos cadernos do jornal por meio de links disponíveis em uma página inicial, utilizando menus ou mesmo em forma paginada, por meio de botões do tipo “avançar” e “voltar”. A definição da estrutura de navegação é armazenada em um banco de dados e é montada a cada acesso do usuário.

4.3.4 Visões sobre o espaço virtual

O Sistema MORFEU é multi-usuário que permite a existência de usuários com diferentes visões do sistema. Como por exemplo, a situação de um mediador em relação a um participante do Debate de Teses (citado no item 4.1). Neste caso, o mediador pode criar teses e ver as UPIs de todos os participantes (posicionamento inicial sobre a tese, por exemplo), mas não pode editá-las. Já os participantes estão restritos a ver apenas suas próprias UPIs e as revisões sobre elas (UPIs de outros participantes, por exemplo) e só podem editá-las em um prazo determinado pelo mediador. O mediador em relação aos participantes e os participantes entre si terão diferentes visões sobre o espaço virtual.

A arquitetura do editor de *templates* deverá prover mecanismos que atuem sobre os acessos aos espaços virtuais, incluindo permissões e prazos sobre escrita e leitura de conteúdo, que deverão ser fornecidos na construção/edição do espaço virtual. O *template* terá o papel de interpretar os acessos e tomar a decisão de mostrar ou não determinadas produções e os botões de edição/criação de UPIs, dependendo do vencimento do prazo e permissão de acesso dos usuários.

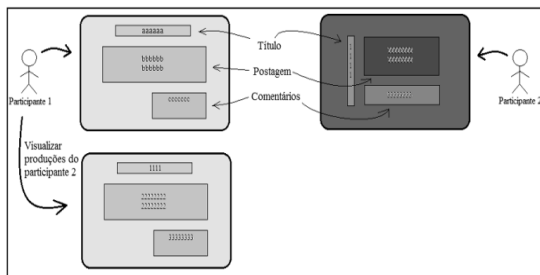


Figura 3. Exemplo do conceitos de visões

Um exemplo sobre visões está apresentado na Figura 3, na qual o participante 1 definiu o seu *template* completamente diferente do *template* do participante 2 e, mesmo assim, ao visualizar as produções do participante 2, tem uma visão inteiramente particular (como ele mesmo definiu) do ambiente, se ele assim desejar. Caso contrário ele também poderá visualizar as produções por meio do *template* do participante 2, ou seja, terá a mesma visão que o participante 2.

5. O EDITOR DE TEMPLATES

Com o objetivo de validar a proposta de arquitetura descrita na seção anterior realizamos uma implementação prototípica do ET. Ele foi desenvolvido essencialmente para funcionar sobre a plataforma web, ou seja, ele funciona em qualquer sistema operacional. Com relação às tecnologias escolhidas para desenvolvimento foram utilizadas as linguagens PHP (servidor), HTML e principalmente *JavaScript* (cliente).

5.1 Funcionamento do editor de templates

O primeiro passo na criação do *template* é a definição do seu layout. O indivíduo acessará uma tela do tipo drag'n'drop/resize que o possibilitará “desenhar” sua página, isto é, escolher a

posição dos elementos do espaço virtual na tela. Definido o layout, um arquivo XSL será gerado codificando essa estrutura.

Uma vez definido o layout, a pessoa acessará outra interface para preparar o design da página, isto é, definir propriedades de cor de fundo, tamanho das fontes etc. Nessa etapa o usuário poderá contemplar uma pré-visualização do design de seu ambiente, obedecendo o paradigma WYSIWYM (*What You See Is What You Mean*). Definido o design, um arquivo CSS será gerado. Mais adiante, esse documento fornecerá os estilos à página HTML.

Na próxima etapa, o usuário será encaminhado a uma nova interface que o permitirá escolher como se dará a navegação no ambiente criado. Definida a navegação, a informação da escolha será armazenada em um banco de dados.

Ao final da edição/criação do *template*, o Projeto MORFEU deverá ter armazenado alguns documentos: (i) XML com a estrutura do espaço virtual (isto é o VCom), (ii) XSL com o layout, (iii) CSS com design. Esses arquivos são gerados de acordo com as regras do núcleo do MORFEU, e serão usados na geração do *template*.

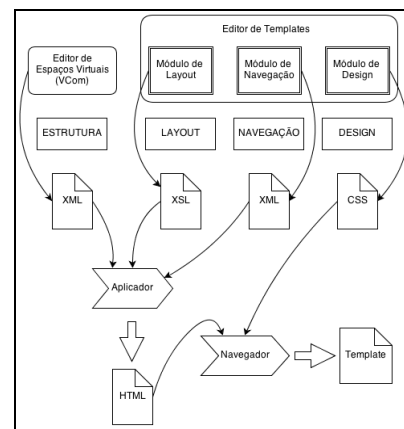


Figura 4. Visão Geral do Editor de Templates

Como se pode observar na Figura 4, existe o elemento aplicador, que será responsável por realizar a transformação XSLT, gerando o documento HTML final. Já as definições de design, que são feitas por meio de um documento CSS, serão aplicadas pelo próprio navegador do usuário.

Os arquivos gerados, XML, XSL e CSS, serão armazenados em uma biblioteca, podendo ser reaproveitados por outras pessoas na personalização de seus espaços virtuais. Por exemplo, havendo alguém que deseje aproveitar um *template* de um usuário, alterando apenas uma propriedade simples de design, um novo arquivo CSS será gerado baseado no CSS original e um novo *template* será criado.

É importante dizer que ao contrário desses elementos, a leitura do contrato das visões é feita em tempo de execução, dinamicamente. Por causa disto, ela não está detalhada na Figura 4, que mostra os passos feitos em tempo de construção da interface.

Nas sessões a seguir, será descrito o funcionamento da implementação prototípica do ET. Para tal demonstração, usaremos como exemplo a construção de um *template* para um VCom do tipo blog, estruturado com um título, uma postagem e uma área para comentários.

5.1.1 Módulo de layout

O módulo de *layout* recebe como entrada o arquivo XML da UPI e se encarrega de dispor os seus elementos, inicialmente como na Figura 5, para que o usuário possa modelar a interface que ele pretende obter.



Figura 5. Interface de Utilização do Módulo de Layout.

Depois de alguns cliques no navegador, arrastando, soltando e redimensionando os elementos, ele poderá fazer com que seu modelo de interface fique como na Figura 6.

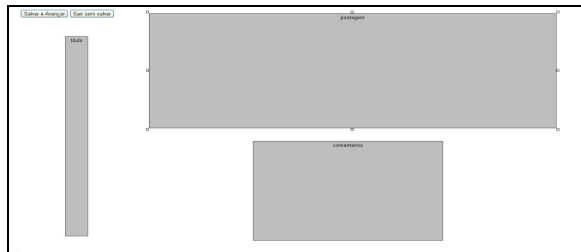


Figura 6. Layout Construído

Uma vez definido o modelo inicial de *layout*, o usuário deverá salvar suas alterações. Feito isso, o servidor se encarregará de coletar os dados e escrever o arquivo XSL responsável por guardar essas informações. Esse arquivo será armazenado no servidor, em uma pasta com a identificação do usuário, para que em momentos futuros ele possa alterar o *layout* quantas vezes forem necessárias.

5.1.2 Módulo de design

Inicialmente, ao carregar os elementos, o usuário terá uma pré-visualização da interface conforme ela foi criada no módulo anterior. Para isso, o servidor interpreta o arquivo XSL montando as estruturas para edição das propriedades de *design* dos elementos, como pode ser observado na Figura 7.

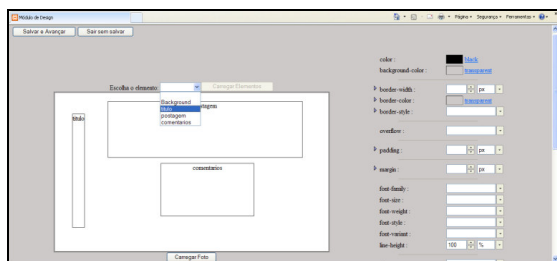


Figura 7. Interface de Utilização do Módulo de Design

Na medida em que as propriedades dos elementos são alteradas, a pré-visualização indica como o modelo do design será apresentado, e isso acontece em tempo real.

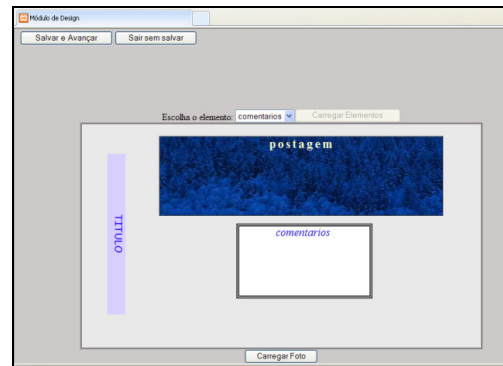


Figura 8. Interface de Utilização do Módulo de Design (alterado).

Como pode ser observado na Figura 8, algumas propriedades do *design* do espaço virtual foram definidas (cor de plano de fundo, imagem de fundo na postagem, título na vertical etc.). Quando o usuário clicar em “salvar e avançar”, a aplicação escreverá as propriedades escolhidas em um arquivo CSS estruturado e encaminhará o usuário para a interface de definição da navegação.

5.1.3 Modulo de navegação

Na Figura 9 está representada a tela de escolha do tipo de navegação: Estrutura Sequencial Vertical, Árvore de *Links*, Paginação ou Estrutura de Menus. Caso seja escolhida a primeira opção, não deverá haver transição de páginas, ou seja, o espaço será todo descrito em uma única página. Nas outras três opções há transição, sendo que em “árvore de *links*” o acesso a determinado conteúdo é direto, apenas com um clique, diferente da opção de paginação, na qual o acesso é paginado com mecanismos de avanço e volta. Ainda se podem acessar os *containers* de dados dos *templates* por meio da opção de menus, que fornecesse acesso direto a conteúdo, como na árvore de *links*.

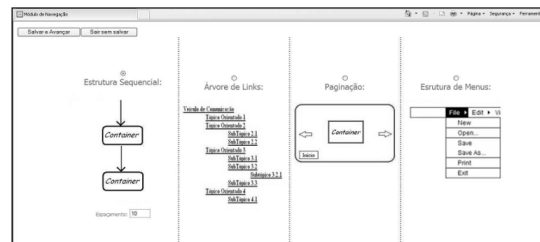


Figura 9. Interface de Utilização do Módulo de Navegação

5.1.4 Um exemplo de uso

Imaginemos que pessoas se dividam em dois grupos a fim de discutir sobre um assunto, sendo que um grupo se posiciona a favor e outro contra uma determinada idéia. Os grupos têm um prazo para postar suas argumentações e apresentar duas provas que a sustentem (citações de referência de trabalhos na área, por exemplo). No momento, não existe ferramenta computacional voltada para esse tipo de ambiente de discussão. Porém, com o MORFEU poderemos facilmente modelar o espaço virtual e cada grupo poderá descrever seu *template*, como mostrado nas Figuras 10 e 11. Nomeemos esse ambiente como “Confronto de Opiniões” ou simplesmente “Confronto”.

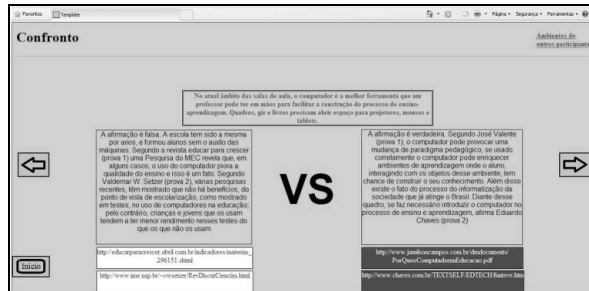


Figura 10. Confronto – Template do Grupo 1

Nota-se que o *template* do Grupo 1 apresenta o símbolo “VS” que, na verdade, é uma imagem. Já o Grupo 2 optou por não utilizar o símbolo, deixando-o escondido na página. Essa opção é dada na utilização do módulo de *design* do editor.

Outra diferença está demonstrada na forma de navegar pelos *containers* de dados dos dois grupos. O primeiro grupo optou pela utilização da navegação paginada, enquanto o segundo grupo está navegando pela árvore de *links*.

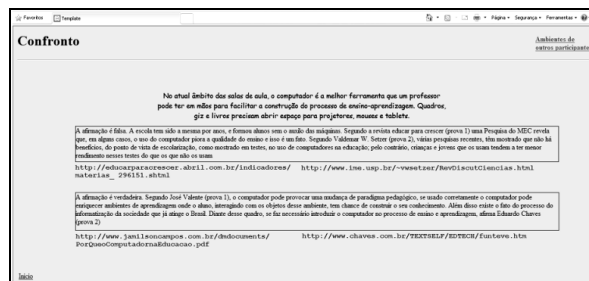


Figura 11. Template do Grupo 2

Neste exemplo pode-se ver como uma interface gráfica em um ambiente virtual colaborativo pode ser estabelecida de forma totalmente individual e como o suporte computacional oferecido para a proposta do ET contempla os requisitos estáticos e dinâmicos na montagem da apresentação do ambiente Confronto.

6. Conclusões e trabalhos futuros

Durante os estudos sobre criação de interface com o usuário, constatamos que para um ambiente virtual ser melhor aproveitado por seus participantes, ele precisa de uma interface apropriada e adaptável, baseada em princípios ergonômicos, fazendo com que as ferramentas, além de cumprirem sua função específica, com eficiência e eficácia, atraiam por sua beleza e/ou seu estilo.

O Editor de *Templates* apresenta algumas limitações que o torna, mas que não impedem a possibilidade de evolução. Um exemplo disso está no módulo de navegação no qual o usuário escolhe sua navegação, e não a cria como nos outros módulos, quando o propósito principal do MOrFEu é a flexibilidade dos ambientes.

Construímos um protótipo de nossa proposta e o testamos na descrição de interfaces visuais para diversos veículos de comunicação incluindo a arquitetura pedagógica debate de teses, *blogs* e *wikis*. Pensamos como primeiro trabalho futuro a realização de testes em uma turma diversificada.

Em outros trabalhos futuros, planejamos desenvolver um editor de *templates* inteligente que, baseado em estatísticas de escolhas dos participantes, possa fornecer sugestões de *layout* e *design* para

novos espaços virtuais. Temos ainda um plano de construção de uma ferramenta do tipo *Wizard* que permita elaborar os *templates* sob uma ótica descritiva.

7. REFERÊNCIAS

- [1] Brusilovsky, P. (2001) "Methods and techniques of adaptive hypermedia" In: Journal of User Modeling and User-Adapted Interaction, Spring Netherlands.
- [2] Krzysztof, G., Weld, D. S. (2004) "SUPPLE: Automatically generating user interfaces" In: IUI '04 Proceedings of the 9th international conference on Intelligent user interfaces ACM New York, USA.
- [3] Guedes, G. (2009) "Interface humano computador: prática pedagógica para ambientes virtuais", Teresina, Piauí: Ed. Graf. UFPI, pp 59-93.
- [4] Kerer, C., and Kirda, E. (2000) "Layout, content and logic separation in web engineering", In: Proceedings of the 9th International World Wide Web Conference, 3rd Web Engineering Workshop, Amsterdam, Netherlands..
- [5] Kirda, E., Kerer, C., Matzka, G. (2001) "Using XML/XSL to build adaptable database interfaces for web site content management". In: XML In Software Engineering Workshop, 23rd International Conference On Software Engineering. Toronto, Canada.
- [6] Nevado, R. A., Menezes, C. S., Vieira R. M. J. (2011) "Debate de teses – uma arquitetura pedagógica". In: XXII Simpósio Brasileiro de Informática na Educação. SBIE, Aracajú, SE.
- [7] Menezes, C. S., Nevado, R. A., Castro, A. N. Jr., Santos, L. N. (2008) "MOrFEU – Multi-Organizador Flexível de Espaços Virtuais para Apoiar a Inovação Pedagógica em EAD" In: Simpósio Brasileiro de Informática na Educação, Fortaleza-CE.
- [8] Palmer, J. W. (2002) "Web site usability, design, and performance metrics" In: Journal Information Systems Research, USA.
- [9] Pessoa, J M.; Menezes, C S. (2003) "Framework baseado em padrões abertos para construção de ambientes CSCW/CSCL na Web". In: Proceedings of the First Latin American Web Congress.
- [10] Piaget, J. (1989) "Les relations entre 'l'intelligence et l'affectivité dans le développement de l'enfant". In Les Émotions." Niestlé, pag. 75-95.
- [11] Valente, C. E. (2004) "Padrões de interação e usabilidade", UNICAMP, Campinas, SP.
- [12] Romero, P., Du Boulay, B., Cox, R., Lutz, R., Bryant, S. (2007) "Debugging strategies and tactics in a multi-representation software environment". In: International Journal of Human-Computer Studies, pp 992–1009.
- [13] Turkle, S. A. (1997) "Vida no Ecrã: A Identidade na Era de Internet". Lisboa: Relógio d'Água Editores. Tradução: Paulo Faria. 482 p. (Coleção A Sociedade Digital)