

# **UM SISTEMA DE APOIO AO WEB DESIGNER EM UM CONTEXTO EDUCACIONAL**

**Marcus Vinicius Maltempi**

mmaltempi@uol.com.br

Universidade Estadual de Campinas, Núcleo de Informática Aplicada à Educação.  
Bloco V da Reitoria, CEP 13083-970 – Campinas (SP). Tel: (14) 6821-5698

## **RESUMO**

A construção de páginas Web pode ser uma rica atividade a ser explorada em ambientes de ensino-aprendizagem. Nela os alunos tornam-se produtores de conteúdo para Web, numa abordagem que privilegia a aprendizagem por meio da criação de produtos pessoais significativos. Neste trabalho examinamos o processo de criação e depuração (*debugging*) de páginas Web, com objetivo de especificar um sistema computacional que, associado a uma ferramenta de autoria de páginas Web e inserido em um ambiente de ensino-aprendizagem, favorecerá a construção de conhecimento por parte dos alunos. Uma síntese do funcionamento e arquitetura deste sistema é apresentada.

**MODALIDADE:** Pôster.

**PALAVRAS-CHAVE:** Desenvolvimento de Ferramentas de Suporte a Autoria; Web Design; Sistema Especialista.

# UM SISTEMA DE APOIO AO WEB DESIGNER EM UM CONTEXTO EDUCACIONAL

## INTRODUÇÃO

A construção de páginas Web<sup>1</sup> (*Web design*) pode ser uma rica atividade a ser explorada em ambientes de ensino-aprendizagem que fazem uso de computadores, Internet e ferramentas computacionais. Nela os alunos deixam de ser simples usuários para tornarem-se desenvolvedores de conteúdo para Web. Esta abordagem baseia-se na premissa de que a aprendizagem é mais efetiva quando os alunos estão engajados na construção de artefatos pessoais significativos, sobre os quais possam refletir e mostrar a outras pessoas, tais como programas de computadores ou aplicativos (Papert, 1980; Resnick, 1996).

A fim de investigarmos o potencial do *Web design* em ambientes de aprendizagem, realizamos um estudo prático em uma escola com crianças de 5<sup>a</sup> e 6<sup>a</sup> séries. Durante nove semanas estas turmas trabalharam em grupos no desenvolvimento de *Web sites* sobre temas diversos. Pesquisas anteriores relataram resultados positivos em experiências semelhantes com o desenvolvimento de programas de computadores (Harel & Papert, 1991; Kafai, 1996) e aplicativos multimídia (Lehrer *et al.*, 1994; Liu & Rutledge, 1997), em vez de construção de páginas.

Certamente os alunos necessitam de apoio e incentivo para que a tarefa de construir páginas não se resume a recortes e colagens de informações, resultando em ocasionais situações de aprendizagem. Sendo assim, é pertinente levantarmos a seguinte questão: *o que fazer para auxiliar o aluno a se engajar em um processo interativo de criação de páginas que favoreça a construção de conhecimento?*

Não há uma única ou trivial resposta para essa questão. Ela envolve desde a ferramenta de autoria de páginas (que características deve ter para favorecer a construção de conhecimento?) até a forma de atuação do professor junto aos alunos. Entretanto, acreditamos que o ponto central para a solução do problema subjacente à questão formulada esteja na criação de situações que favoreçam a atividade de depuração (*debugging*). Entendemos depuração como sendo a detecção e correção de erros gerados no processo de criação e que conduz à finalização consciente do artefato que está sendo criado. Segundo Valente (1994), o processo de programação de computadores pode ser visto como um ciclo consistindo de descrição, execução, reflexão e depuração. Neste ciclo depuração constitui uma oportunidade singular para o aluno construir seu conhecimento e para entender o que está fazendo, justamente o que queríamos para a atividade de *Web design*. Entretanto, diferentemente da programação, no *Web design* as fases de descrição e execução não favorecem a depuração, pois o processo de pensamento do aluno não fica registrado para consulta (como o código fonte de um programa) e o computador executa a seqüência de informação organizada pelo aluno, e não a informação correspondente ao conteúdo da página (o *feedback* torna-se pobre).

Neste trabalho focamos na ferramenta de autoria de páginas utilizada pelos alunos (Adobe Page Mill 2.0). Por meio do estudo prático realizado, analisamos o processo de criação e depuração de páginas com o propósito de identificarmos as facilidades e limitações que a ferramenta impõe à atividade de depuração de páginas Web. Os resultados dessa análise contribuíram para a especificação de um sistema computacional de apoio à construção de páginas que favorece a construção de conhecimento por parte do aluno.

## O PROCESSO DE CRIAÇÃO E DEPURAÇÃO DE PÁGINAS WEB

Conforme analisado no estudo prático realizado, tendo por base o ciclo descrição-execução-reflexão-depuração (Valente, 1994) e a Teoria da Ação (Norman, 1986), o processo

<sup>1</sup> Para simplificar, quando conveniente, substituiremos o termo *página Web* por somente *página*.

de construção de páginas se inicia com uma idéia do aluno para a página a ser criada, ou alterada, que corresponde a meta a ser alcançada, em vista do estado atual do sistema. Estabelecendo-se a meta, o aluno forma uma intenção (e sub-intenções) que resultará em uma seqüência de ações a ser executada. Diferentemente da representação das idéias do aluno em uma linguagem de programação, no *Web design* essa seqüência de ações corresponde a atos a serem executados sobre mecanismos do sistema (teclado e *mouse*), não necessariamente todos planejados com antecipação. Assim, o aluno especifica mentalmente as ações a serem executadas, sendo que o computador apenas possibilita a ação do aluno por meio de uma interface de manipulação direta do tipo WYSIWYG (*What You See Is What You Get*).

Após a execução, o aluno deve interpretar a página resultante, e avaliá-la com relação às metas e intenções que foram mentalmente estabelecidas – o que torna a depuração mais difícil de ser realizada. Esta interpretação e avaliação geralmente conduzem a um novo conjunto de metas e intenções (no caso da identificação de *bugs*), e correspondem as fases de reflexão e depuração que, na atividade de programação, são auxiliadas pelo código fonte do programa.

O *bug* é o precursor da depuração. Na programação está relacionado com um comportamento imprevisto do computador, manifestado após a execução do programa desenvolvido, tendo por origem falhas na lógica do algoritmo e/ ou no uso da linguagem de programação. Na atividade de *Web design* identificamos como *bug* os erros relacionados a conteúdo, *link* e *layout*. Nesse último caso, a intenção do aluno é fundamental para que se possa salientar e determinar um *bug*, pois dependendo dela, o que é *bug* em uma página (ou *site*), pode não ser em outra, dependendo dos critérios de *Web design (guidelines)* adotados pelo aluno.

Conforme verificado ao longo do estudo, a inserção de eventos externos que auxiliem os alunos a interpretar e avaliarem suas páginas contribui com a depuração. Os eventos mais eficazes foram aqueles que visavam “executar” as idéias dos alunos, analisando e criticando as páginas criadas, e se baseavam na apresentação destas a outras pessoas, presencialmente ou à distância. Portanto, se desejamos automatizar por meio de um sistema computacional algum mecanismo que favoreça a depuração em um ambiente de aprendizagem baseado na construção de páginas, este sistema deverá “executar” a página do aluno, auxiliando-o a interpretar e avaliar o que fez. A ferramenta de autoria Page Mill é um produto desenvolvido para o mercado, dirigido ao público geral. Como era esperado, essa ferramenta não apresenta mecanismos que favoreçam a atividade de depuração.

## UM SISTEMA DE APOIO AO WEB DESIGNER

Nosso objetivo é construir um sistema computacional que lide com a intenção do aluno e que contribua com a idéia de aprendizado construcionista (Papert, 1986). Um sistema que apóie o aluno quando este achar necessário – de maneira similar ao sistema *Jonas* (Borges, 1997).

Para funcionar como um crítico que avalia e faz considerações sobre as páginas do aluno, considerando a intenção deste, é necessário criar um Sistema Especialista que utilize uma base de conhecimento, informações sobre os critérios de *Web design* assumidos pelo aluno e a página criada. Na interface da ferramenta de autoria de páginas deverá ter um elemento (por exemplo, um ícone) que o aluno aciona toda vez que quiser ajuda do Sistema de Apoio ao Web Designer (SAWeD).

Definimos a arquitetura do sistema SAWeD da seguinte forma:

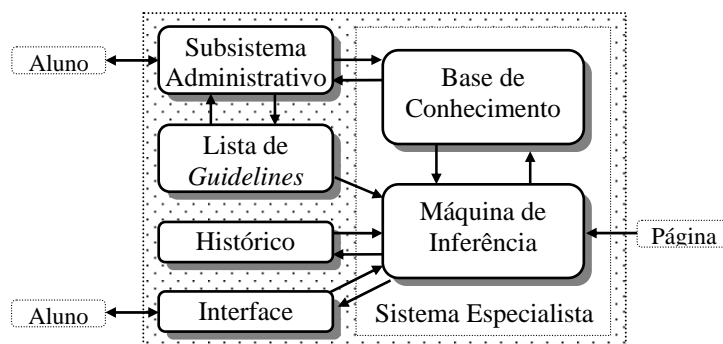
- um Sistema Especialista: máquina de inferência e base de conhecimento;
- uma interface gráfica;
- um subsistema administrativo (para alimentação e atualização do sistema);
- uma lista de *guidelines* (critérios de *Web design*) assumida pelo aluno;

- um histórico contendo as versões da página em desenvolvimento.

A cada solicitação do aluno, a máquina de inferência deverá consultar a página que estiver aberta no sistema de autoria e a lista de *guidelines* assumida pelo aluno. Então, a máquina de inferência usará a base de conhecimento de modo a identificar sugestões e críticas que possam contribuir para que o aluno melhore a página em questão – a figura abaixo apresenta uma representação esquemática da arquitetura do sistema.

Além disso, cada vez que o SAWeD for chamado, ele deverá guardar uma versão da página que está sendo considerada, gerando um histórico do desenvolvimento da página que poderá ser consultado pelo aluno. Além de ser útil no acompanhamento do progresso do aluno, o histórico representa um tipo de descrição que pode servir ao aluno no processo de depuração de suas idéias.

Do modo como está sendo proposto, o SAWeD pode ser considerado um Sistema Especialista, pois há uma clara separação entre sua base de conhecimento e a máquina de inferência (Luger & Stubblefield, 1989). A idéia é que o sistema não gere informações de modo independente, mas colabore com o aluno para que este seja bem sucedido na tarefa de construir páginas Web.



Arquitetura do SAWeD

## REFERÊNCIAS BIBLIOGRÁFICAS

- Borges, M.A.F. (1997). O design centrado no aprendiz no sistema Jonas: uma experiência de desenvolvimento de um sistema para formação na Empresa. Campinas, 101p. (Mestrado – Instituto de Computação, Unicamp).
- Harel, I. & Papert, S. (1991). Software design as a learning environment. In: Harel, I. & Papert, S. (Eds.) *Constructionism*. New Jersey, Ablex Publishing Corporation. p.41-84.
- Kafai, Y.B. (1996). Learning design by making games: children's development of design strategies in the creation of a complex computational artifact. In: KAFAI, Y.B. & RESNICK, M. (Eds.) *Constructionism in practice: designing, thinking and learning in a digital world*. New Jersey, LEA. p.71-96.
- Lehrer, R.; Erickson, J. & Connell, T. (1994). Learning by designing hypermedia documents. *Computers in the Schools*, New York, **10**(1/2): 227-254.
- Liu, M. & Rutledge, K. (1997). The effect of a "Learner as multimedia designer" environment on at-risk high school students' motivation and learning of design knowledge. *Journal of Educational Computing Research, USA*, **16**(2): 145-177.
- Luger, G.F. & Stubblefield, W.A. (1989). *Artificial intelligence and the design of expert systems*. Redwood (CA), Benjamin/Cummings. 660p.
- Norman, D.A. (1986). Cognitive engineering. In: Norman, D.A. & Draper, S.W. (Eds.) *User centered system design: new perspective on human-computer interaction*. LEA, p.31-61.
- Papert, S. (1980). *Mindstorms: children, computers and powerful ideas*. NY, Basic Books.
- Papert, S. (1986). *Constructionism: a new opportunity for elementary science education*. MIT, Epistemology and Learning Group. Proposta para The National Science Foundation.
- Resnick, M. (1996). Toward a practice of "constructional design" In: Schauble, L. & Glaser, R. (Eds.) *Innovations in learning: new environments for education*. NJ, LEA. p.161-174.
- Valente, J.A. (1994). Computers in education: shifting the pedagogical paradigm from instructionism to constructionism. *Logo exchange* (Eugene, OR), **12**(2): 39-42.