Monitorando alunos de cursos à distância via Internet

Gustavo Pinheiro, Dilvan de Abreu Moreira, Dorival Leão Pinto Junior

[1]Pós-graduação em Ciências da Computação – Universidade de São Paulo (USP)
Av. do Trabalhador São-Carlense, 400 Centro – Cx. Postal 668
São Carlos – São Paulo – Brasil CEP 13560-970

{Gustavo, dilvan, leão}@icmc.sc.usp.br

Abstract. *Nowadays, it is considerably easy to provide knowledge to a huge audience through the Web. Add to that its multimedia capabilities and what we have is a powerful tool for e-Learning.*

Probably, the aspect that differentiates the most e-Learning from the traditional form of education is the lack of physical contact between teacher and students. It sacrifices the richness of their communication. Then, comes up the idea of developing tools to minimize that sacrifice.

This article discusses how to monitor the usage of an e-Learning Web site so that the teacher could have a better perception of his students' characteristics.

Resumo. *A facilidade com que se pode disponibilizar conhecimento a um grande número de pessoas, associada às capacidades multimídia da* World Wide Web*, vieram a criar uma infraestrutura tecnológica muito útil na área de Ensino à Distância Mediado Por Computador.*

Talvez, a característica que mais diferencia o Ensino à Distância do tradicional (em sala de aula) é a falta de contato físico entre as partes. Essa característica empobrece a comunicação aluno-professor. Surge, então, a idéia de desenvolver ferramentas que compensem essa perda.

Neste artigo discute-se como monitorar o uso de um Web site *voltado ao Ensino à Distância de modo a dar ao professor maior percepção das características de cada aluno.*

Introdução

A Internet vem se expandindo em grande velocidade, e se tornando cada vez mais acessível. A facilidade com que se pode disponibilizar conhecimento a um grande número de pessoas, associada às capacidades multimídia da *World Wide Web*, vieram a criar uma infraestrutura tecnológica que impulsionou a pesquisa na área de Ensino à Distância via Internet, ou *e-Learning*.

Talvez, a característica que mais diferencia o Ensino à Distância do tradicional (em sala de aula) é a falta de contato físico entre as partes. Essa característica empobrece a comunicação aluno-professor, que tem no tipo presencial sua forma mais rica [KEE, 1991]. Além disso, pela *Web* pode-se atingir um número de pessoas (alunos) muito grande a um custo (por usuário) relativamente baixo, e a audiência numerosa

certamente dificulta a atuação do professor. Surge, então, a idéia de desenvolver ferramentas que auxiliem o professor nesta tarefa.

Parte-se do pressuposto que um dos principais pontos de interação entre o aluno e o professor é o material didático disponibilizado na *Web*. Sendo assim, uma boa maneira de se saber algo sobre o aluno é acompanhar seus movimentos no *site*. Numa sala de aula (contato físico), o professor poderia facilmente notar quais alunos estão dispersos e quais estão prestando atenção na aula. Num curso via Internet, o professor poderia saber quem acessou o material didático e quanto tempo gastou em cada página se o *site* for devidamente instrumentado. Este artigo trata das diferentes técnicas que podem ser empregadas nesta tarefa.

Instrumentação

Os dados que são tipicamente coletados quando se monitora o uso de um *Web site* são:

- **Endereço I.P. (Internet Protocol) do visitante**: todos os computadores conectados à Internet são identificados por um número (I.P., 127.0.0.1). Esse I.P. geralmente é relacionado a um nome de domínio (www.internet.com). É uma informação importante na medida em que pode auxiliar na identificação de um usuário;
- **Data e hora do pedido**: é preciso atentar para o formato adotado. Esse dado pode ser usado para estimar o tempo que um usuário gastou em cada página visitada;
- **Método HTTP do pedido (GET, POST, HEAD)**: indica o tipo do pedido feito. Útil apenas para se detectar atividades suspeitas. Por exemplo: um hacker pode enviar um pedido com o método DELETE;
- **URL requerido**: é o endereço do recurso pedido ao servidor pelo navegador, ou seja, qual documento o usuário acessou;
- **Status HTTP**: é um código que indica a situação da transação entre o navegador e o servidor. Pode ser usado para identificar problemas no fornecimento do serviço;
- **URL do referente (*Referrer* URL)**: Identifica a página que estava sendo visitada antes do atual pedido. Pode ser um *Web site* diferente, ou uma URL do mesmo *site*. Neste caso, este dado é usado para se determinar a sequência de acessos executada por um usuário;
- **Agente do cliente (*User agent*)**: geralmente é um navegador. Contém tipicamente informação sobre o tipo e versão do navegador e do sistema operacional do usuário. É útil na identificação de robôs;
- ***Cookies***: São pequenos arquivos de texto enviados junto com a resposta de um pedido do navegador. Pode conter informações que identifiquem o usuário;

Esses dados podem ser coletados basicamente de duas maneiras: usando-se o *log* do servidor *Web* ou *softwares* independentes [KOH, 2001]. Nesta última categoria incluem-se códigos ECMAScript (JavaScript) adicionado aos hiperdocumentos, *applets* Java, *servlets* Java, programas CGI, etc.

A solução mais simples e mais adotada é sem dúvida aquela que se vale dos arquivos de *log* do servidor *Web*. É uma funcionalidade padrão da maioria desses *softwares*. E no mercado existem inúmeras opções de programas que analisam esses *logs* e geram estatísticas de uso do *site* em questão. Entretanto, os dados que podem ser obtidos dos *logs* são um tanto restritos, uma vez que só registram pedidos de arquivos e nada dizem sobre a interação do usuário com o documento.

Escrever programas que coletam os dados independentemente do servidor *Web* é uma solução mais versátil. No mercado encontram-se várias empresas que fornecem soluções desse tipo, sendo a mais comum aquela que usa código JavaScript. A empresa fornece um trecho de código que deve ser adicionado à todas as páginas a serem monitoradas. Esse pequeno programa coleta os dados (rodando no navegador) e envia os dados coletados através de uma *string* de consulta (*querystring*). O truque é usar o atributo *src* de uma *tag* HTML **<img>** para fazer um pedido de arquivo e passar os dados coletados através da URL (ver **Figura 1**). É uma solução interessante pois permite a coleta de dados que são impossíveis de se coletar no servidor. E permite que os dados sejam enviados para uma máquina diferente da que roda o servidor *Web*. Entretanto, se usada a mesma máquina do servidor *Web*, esse método pode duplicar o número de conexões no pior caso. Mas, geralmente, uma página é composta por mais de um arquivo (cada um gera uma conexão) e por isso o custo computacional desta técnica tende a ser baixo.

Para receber os dados podem usados códigos *script* (PHP, ASP ou JSP), programas CGI ou *servlets* Java. *Servlets* têm a melhor portabilidade e escalabilidade e são, portanto, a principal solução a ser considerada.

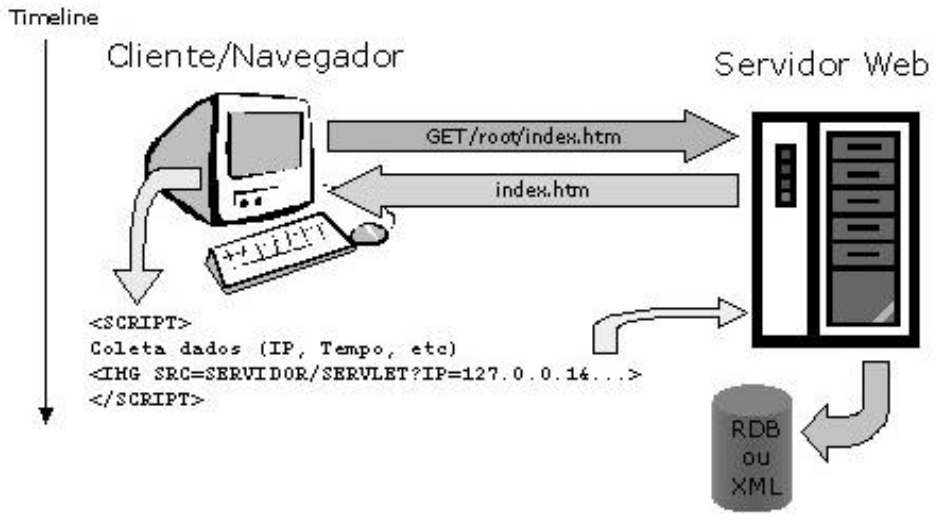

Figura 1 - Coleta de dados com JavaScript

Pode-se usar *applets* Java no lugar de JavaScript, entretanto eles tendem a resultar em *downloads* maiores e pode ser que o navegador do cliente não esteja preparado para executá-los (não possua Máquina Virtual Java instalada).

Uma solução muito versátil é a utilização de JavaScript em conjunto com *servlets* Java. Com essa combinação pode-se coletar dados tanto sobre a interação cliente-servidor, como I.P. do cliente, quanto sobre a interação usuário/hiperdocumento, como o tempo de visualização.

Identificando usuários únicos

O aspecto mais delicado do monitoramento é a identificação de usuários únicos. Há um certo consenso em se usar *cookies* e/ou exigir registro do usuário para se identificar um mesmo usuário. Alguns usam o número I.P. do cliente somado à outro identificador (*cookie*, tipo de *browser*, etc).

Toma-se o caso de uma faculdade da área de computação qualquer a fim de se discutir essa questão. Numa faculdade existem diversos laboratórios ou salas de micros compartilhados por muitos usuários, então deve ser comum a situação em que diferentes usuários acessam um mesmo *Web site* de uma mesma máquina e de um mesmo *browser* (portanto com mesmo I.P. e mesmo *cookie*). Neste caso a única possibilidade de identificação confiável é a exigência de registro (*login*).

Caso o aluno acesse o *Web site* de sua casa, provavelmente ele estará usando uma conexão discada com atribuição dinâmica de I.P. (a cada conexão seu micro pode receber um número diferente). Entretanto, um *cookie* poderia identificá-lo de forma relativamente segura (porém várias pessoas na mesma casa podem compartilhar o micro).

Não há, portanto, forma 100% segura de se identificar um usuário único de forma passiva. É necessário que se estimule a identificação ativa e voluntária do usuário. Um artificio comum é fornecer acesso a determinadas áreas do *site* somente mediante cadastro.

No entanto, a obrigatoriedade de se identificar a cada acesso pode ser desestimulante para o usuário. Com um *cookie* de identificação, podemos tentar "adivinhar" quem é a pessoa por trás do micro. Por exemplo, uma pessoa usando um micro de um laboratório da suposta faculdade acessa um *site* qualquer. Código JSP na página do *site* logo identifica um *cookie* no navegador que fez o acesso, e cruza seus dados com o banco de dados de usuários cadastrados retornando a página principal alterada dinamicamente. O usuário veria em sua tela algo como "Bom Dia José", e logo à frente um pequeno link com a frase "Se você não é o José, identifique-se aqui".

Fontes de imprecisão

Firewalls e memórias *cache* sempre foram uma grande fonte de imprecisão em medidas de audiência. *Firewalls* "mascaram" o I.P. das máquinas por trás dele e geralmente bloqueiam o recebimento de determinados tipos de arquivos que são considerados inseguros (*applets* muitas vezes são bloqueados). Se um cliente pede uma página qualquer e o *proxy* de seu provedor a serve, pois a tem em memória *cache*, esse acesso não será registrado no *log* do servidor *Web* que contém a página. Como isso não é algo incomum, uma considerável parcela dos acessos a um *site* pode ficar fora das estatísticas. O mesmo pode ocorrer por causa da memória *cache* do próprio *browser*. Usando diretrizes em cabeçalhos HTTP, por exemplo **Pragma = No Cache**, pode-se minimizar os problemas com memórias *cache*. Pode-se também definir a data de expiração do documento com valor de datas já passadas.

A técnica que utiliza JavaScript não enfrenta essas dificuldades. Mesmo se a página que estiver sendo acessada vier de uma memória *cache* (do *proxy* ou navegador), o navegador ainda tentará recuperar a suposta imagem referenciada pelo *tag* <img> uma vez que uma página dinâmica (o documento ao qual a *tag* faz referência) não pode ser

armazenada em *cache*. Essa característica é uma das maiores vantagens dessa técnica. *Firewalls* também não são problema uma vez que o código JavaScript fica embutido no código HTML e, portanto, não sofre restrição.

Geralmente, quando se mede a audiência de um *Web site*, deseja-se atestar que muitas pessoas o visitam. Mas nem sempre há uma pessoa por trás de um acesso, e esse é um problema de grande relevância. Na rede, existem *softwares* chamados de agentes ou robôs que podem acessar um *site*. São eles: *softwares* que catalogam *sites* para mecanismos de busca, agentes que buscam informação, etc. Existem listas disponibilizadas na rede, constantemente atualizadas, dos principais robôs. Entretanto, no caso de um *browser offline*, também caracterizado como agente ou robô, pode-se supor que seu usuário (uma pessoa) está interessada no conteúdo que o robô deve recuperar. Neste caso pode ser legítimo contabilizar o acesso.

Existem casos em que não se pode detectar a interação do usuário usando-se as técnicas anteriormente discutidas. *Applets* Java, ActiveX, scripts e Flash são usados comumente para gerar *Web sites* dinâmicos. É relativamente comum encontrarmos *sites* com menus tipo *rolldown* implementados em uma dessas linguagens. Existem casos extremos onde todo o conteúdo é implementado em formato Flash, por exemplo. É preciso atentar para esses casos.

Finalmente, nem todo dado coletado deve ser utilizado, pois podem ter sido gerados por fontes ilegítimas. É necessário definir quais acessos não devem ser contabilizados. Deve-se definir regras de filtragem que excluam esses dados não qualificados. Geralmente essas regras se baseiam no I.P. do visitante, no URL requerido, no status HTTP ou no tipo MIME. Por exemplo, pode-se usar o I.P. para excluir dados gerados por atividade interna de manutenção. O status HTTP é muito usado para excluir acessos que não foram servidos adequadamente. Além disso, muitas vezes uma conexão lenta pode fazer com que um usuário abandone um *site*, gerando dados atípicos de acesso. É interessante, então, tentar detectar esse problema.

Questionários

Dados de uso de um *site* não são a única forma de se obter informação via *Web*. Questionários são uma ferramenta valiosa na tarefa de conhecer o aluno. Afinal, não há forma mais simples de se obter uma informação do que perguntando diretamente ao dono dessa informação, o aluno. Ao se elaborar um questionário, deve-se atentar para as observações seguintes.

De modo geral, questionários são eficientes na coleta de dados subjetivos, como por exemplo a satisfação do usuário em relação a algo. Esse algo pode ser a qualidade do material didático, ou partes dele.

Para respeitar a privacidade e estimular a participação, geralmente questionários são aplicados a voluntários anônimos. Por isso, deve-se tomar cuidado com questões de cunho pessoal. Via de regra, não se deve perguntar coisas como a renda de um indivíduo sem que se tenha um razão clara para isso. Deve-se ter em mente que as perguntas devem ser o mais diretas, objetivas e impessoais possíveis, a fim de não levar a pessoa respondendo ao questionário a responder tendenciosamente, ou mesmo mentir [QUE, 1997]. Entretanto toda informação que sirva para traçar um perfil sócio-econômico dos alunos é interessante.

Perguntas de formato fechado, ou seja, com número determinado de respostas (múltipla escolha), trazem muitas vantagens em relação à aplicação e interpretação, que pode ser totalmente tratada por computador. Dessa forma, fica fácil calcular porcentagens e outros dados estatísticos em relação ao grupo todo ou a um subgrupo.

Análise dos dados

O processo de monitoramento não se resume à coleta de dados. Ainda é preciso extrair conhecimento desse dados, que podem atingir volumes muito altos. E essa característica logo nos faz pensar em duas áreas da Ciência da Computação: Visualização de Informação e Descoberta de Conhecimento.

Através de técnicas de visualização dos dados de acesso a páginas *Web* é possível analisar os padrões de utilização e navegação dos usuários de forma mais eficiente e precisa [CUG, 2001]. Com técnicas adequadas, pode-se visualizar o fluxo de visitantes por página. Por exemplo, tomando-se uma página por referência, exibe-se as páginas que mais levaram a ela e as que mais foram acessadas a partir dela. Assim, pode-se detectar qual caminho de navegação dentro do *site* atrai maior fluxo. Também pode-se saber por quais páginas a maioria dos alunos deixam o *site*, o que pode indicar que determinada página contém a informação desejada, ou que seu conteúdo é muito pouco interessante. É também possível analisar os caminhos de navegação percorridos pelos alunos individualmente, o pode levar à descoberta de padrões que podem ter significados característicos.

Com técnicas de Descoberta de Conhecimento, particularmente técnicas de agrupamento, pode-se identificar alunos que se desviem de um determinado padrão (dadas as características analisadas) [FAY, 1996]. Por exemplo, faz-se uma análise em função do número de acessos a uma determinada área de um *site* voltado ao Ensino à Distância. Descobre-se que os alunos se dividem em quatro grupos (clusters), que são: os alunos que mais acessaram o material ditático (são, digamos, 60% da turma) e tiveram conceito de aproveitamento B ou superior; os alunos que também acessaram muito, mas tiveram conceito abaixo de B (15%); alunos que acessaram pouco mas tiveram bom aproveitamento (5%) e os que não acessaram e não tiveram bom aproveitamento (20%). Logo, o professor pode dar atenção especial aos 15% de alunos que parecem ser esforçados e que tiveram dificuldades com a matéria. Pode ainda dar um "puxão de orelha" nos outros 20% que não demostraram muito esforço. E, quem sabe, convidar os 5% de ótimos alunos para participarem de um programa de iniciação científica.

Discussão final

Monitorar o uso de um *Web site* pode não ser uma tarefa das mais simples. As soluções disponíveis no mercado dificilmente atendem às necessidades particulares da aplicação. E implementar uma solução customizada pode demandar mais esforços do que se pode dispor. Ainda é um desafio para a pesquisa em Computação desenvolver ferramentas mais versáteis e capazes de realizar análises mais complexas, usando técnicas de Descoberta de Conhecimento por exemplo.

Com ferramentas desse tipo, monitorando o uso do material didático *online*, é possível diminuir a perda da percepção que o professor tem das características de seus alunos. Assim, o professor pode descobrir casos excepcionais (alunos com dificuldades,

por exemplo) e dispender mais efetivamente seus esforços. Isso aumentaria a capacidade de se atender a crescente demanda por conhecimento na *Web*.

Referências

[CUG, 2001] Cugini, J.; Scholtz, J., VISVIP: 3D visualization of paths through Web Sites, IEEE, set. 1999, http://zing.ncsl.nist.gov/webmet/.

[FAY, 1996] FAYYAD, U., PIATETSKY-SHAPIRO, G., AND SMYTH, P.; Knowledge discovery and data mining: Towards a unifying framework. In Proceedings of the Second International Conference on Data Mining and Knowldege Discovery, pages 82 to 88. AAAI Press, Menlo Park, US; 1996.

[KEE, 1991] KEEGAN, D.; The Foundations of the Distance Education, London, Routledge, 1991.

[KOH, 2001] Kohavi, Ron; Minning E-Commerce Data: The Good, The Bad and The Ugly; KDD 2001's Industrial Track.

[QUE, 1997] QUESTIONAIRE DESIGN, Disponível on-line em: http:// www.cc.gatech.edu/classes/cs6751_97_winter/Topics/quest-design/

[SUN, 2002] The Java Tutorial. Disponível on-line em: http://java.sun.com.