Formalismo com autômato adaptativo em mecanismo de privacidade e personalização

Paulo Roberto Massa Cereda
Universidade Federal de São Carlos, Departamento de Computação,
São Carlos - SP, Brasil, 13565-905
paulo_cereda@dc.ufscar.br

e

Sérgio Donizetti Zorzo
Universidade Federal de São Carlos, Departamento de Computação,
São Carlos - SP, Brasil, 13565-905
zorzo@dc.ufscar.br

Abstract

In the Web context, the user concern with his/her privacy has increased considerably. To assist him/her in the task to protect his/her personal information, some privacy protection mechanisms had been considered. However, these mechanisms have performance of difficult comparison among them, becoming arduous the attainment of quantitative values for analysis and results. This paper proposes a formalism, using adaptive automata, to define a privacy and personalization mechanism. The adaptive automaton is suitable for this purpose, because of the self-modifications capacity and complex languages recognition. A comparative analysis is performed between an existing privacy and personalization mechanism and the one described in this paper to evaluate the convergence of the obtained solutions relating to the one which better represents the user request. Thus, the bigger the solution convergence degree is, greater will be the user satisfaction about the mechanism use.

Keywords: Privacy, Personalization, Web, Adaptive Automata.

Resumo

No contexto da Web, a preocupação do usuário com sua privacidade tem aumentado consideravelmente. Para auxiliá-lo na tarefa de resguardar suas informações pessoais, alguns mecanismos de proteção à privacidade foram propostos. Entretanto, estes mecanismos possuem desempenho de difícil comparação entre si, tornando árdua a obtenção de valores quantitativos para análise e resultados. Este trabalho descreve um mecanismo para proteção de privacidade e que também ofereça alguma personalização ao usuário, utilizando-se de um autômato adaptativo. O autômato adaptativo se mostra adequado para tal finalidade, devido à sua capacidade de auto-modificação e reconhecimento de linguagens complexas. É realizada uma análise comparativa entre um mecanismo de privacidade e personalização existente e o descrito neste trabalho para avaliar a convergência das soluções obtidas em relação àquela que de fato representa melhor a requisição do usuário. Desta forma, quanto maior o grau de convergência da solução, maior será a satisfação do usuário quanto à utilização do mecanismo.

Palavras-chave: Privacidade, Personalização, Web, Autômatos Adaptativos.

1 Introdução

Com o crescimento e abrangência da Internet, os sites provedores de serviços passaram a oferecer diversos serviços personalizados aos usuários. Assim, as informações relativas aos acessos desses usuários passaram a ser valiosas para tais sites, a ponto de realizarem a coleta destas informações, muitas vezes sem um consentimento explícito. Ao mesmo tempo, a privacidade do usuário passou a ser um elemento muito importante durante a navegação e, na maioria das vezes, decisivo para a permanência ou não de um usuário em um determinado site.

A coleta de dados na Internet, em sua maioria, passa de modo não observado pelos usuários, acarretando em riscos à privacidade. É comum a existência de grandes bancos de dados contendo todo o tipo de informação coletada acerca dos usuários; o comércio destes bancos de dados é praticado frequentemente, devido à demanda crescente por esse tipo de informação [8].

A personalização é um recurso que oferece conforto aos usuários, mas depende de algumas informações acerca da utilização do serviço. Assim, ao restringir o envio de suas informações pessoais, não será possível ao usuário usufruir dos serviços personalizados. As técnicas de personalização são úteis quando utilizadas de um modo correto e de acordo com as informações descritas na política de privacidade do site. Porém, ao serem utilizadas de um modo indevido ou desenfreado, acarretará em conseqüências desastrosas à privacidade dos usuários.

Há alguns mecanismos que visam prover uma navegação com privacidade e, ao mesmo tempo, garantir que o usuário possa receber algum serviço personalizado, buscando assim equacionar a incompatibilidade entre privacidade e personalização.

Os mecanismos existentes possuem execução de difícil comparação e avaliação entre si. Diante deste panorama, este artigo apresenta um formalismo, utilizando um autômato adaptativo, para representar um mecanismo de privacidade e personalização. Para tal, o autômato adaptativo foi escolhido devido à sua simplicidade, capacidade de auto-modificação e de reconhecimento de linguagens mais complexas. Ao definir um formalismo para a representação de um mecanismo de privacidade e personalização, obtém-se um modelo teórico para a realização de análises de execução e comparação, além da possibilidade de ser utilizado como referência para a criação de novos mecanismos.

2 Mecanismos de privacidade

Existem várias propostas na literatura para tentar fornecer privacidade ao usuário na Internet. Algumas delas são baseadas em arquiteturas ou mecanismos que visam manter o anonimato do usuário. Outras, porém, baseiam-se em métodos de policiamento dos sites ou de aviso ao usuário sobre as políticas de privacidade adotadas [10]. De um modo geral, as propostas existentes objetivam manter as informações pessoais dos usuários seguras e confidenciais. Alguns destes mecanismos são apresentados a seguir.

2.1 Anonimato

O anonimato pode ser caracterizado como a situação de não-identificação de um determinado indivíduo, entre um conjunto [23]. Na Web, através do anonimato, o usuário pode navegar por um determinado site sem que seja possível identificá-lo. A privacidade é garantida devido à ocultação do endereço IP de seu computador e de outras informações presentes nos cabeçalhos dos protocolos.

O anonimato ocorre por intermédio de um servidor, chamado *proxy*, que realiza o mascaramento de todas as requisições do usuário como se fossem suas, através de alterações nas informações presentes nos cabeçalhos dos protocolos. Apesar de esta solução ser eficaz na maioria dos casos, existem sistemas mais complexos, que garantem maior segurança sobre as questões de privacidade do usuário.

Os mecanismos de anonimato podem ser divididos em duas vertentes: os *proxies* de anonimato de um único nó e os de vários nós.

Os *proxies* de anonimato de um único nó funcionam do seguinte modo: o usuário faz a requisição da URL ao *proxy*, que a processa e envia uma requisição ao site-alvo; o site-alvo, por sua vez, processa essa requisição, devolve ao *proxy*, e este por fim a encaminha ao usuário [28]. A maioria dos sites que oferecem esse tipo de serviço também disponibiliza recursos adicionais, como filtragem de *cookies*, reescrita de códigos *JavaScript*, bloqueio de pacotes HTTP, entre outros [8].

Os *proxies* de anonimato de vários nós baseiam-se no conceito de *MixNets* [5], que são grupos de *proxies* que promovem anonimato, conduzindo o tráfego do usuário através dos vários nós da rede. Esses nós,

também chamados de *Mixes*, realizam operações de atraso, reordenação, adição de dados e encaminhamento de tráfego. Através dessas operações realizadas, o caminho traçado faz com que a origem da mensagem seja desconhecida. Vários mecanismos utilizam-se deste conceito, dentre os quais os mais relevantes são *Onion Routing* [9] e *Crowds* [25].

2.2 Pseudônimos

O método dos pseudônimos consiste em criar apelidos para o usuário, disfarçando assim sua verdadeira identidade. Como o usuário está navegando pela Internet utilizando-se de um apelido, pode usufruir dos serviços de personalização disponibilizados pelos sites.

Com a utilização dos pseudônimos, a identificação é feita sob um apelido, mantendo o anonimato da identidade real do usuário. Assim, um usuário pode ter vários pseudônimos durante sua sessão de navegação. Os pseudônimos são criados por um intermediário entre o usuário e o site-alvo. Esse intermediário atua como um servidor *proxy* e é responsável pela comunicação entre o usuário e o site e pela criação e gerenciamento dos pseudônimos.

Existem alguns mecanismos baseados em pseudônimos, como o *Janus Personalized Web Analyser* [7], que atribui um pseudônimo específico a cada site que o usuário visitar.

2.3 P3P

O Projeto de Plataforma para Preferências de Privacidade foi desenvolvido pelo Consórcio da *World Wide Web*[1], objetivando permitir aos sites Web uma padronização da apresentação das práticas de coleta, de modo que o usuário tome conhecimento das informações coletadas. A P3P permite que site e usuário negociem quais informações serão coletadas, bem como suas utilização e mecanismo de coleta [6].

A P3P introduz as preferências de privacidade ao usuário, podendo este configurá-las como achar melhor. Desta forma, a análise das políticas de privacidade de um site será feita baseada nessas preferências, minimizando o problema da subjetividade do conceito de privacidade.

Alguns sites já oferecem sua política de privacidade no padrão da P3P, oferecendo mais liberdade ao usuário para preocupar-se apenas com seus objetivos de visita, minimizando a necessidade de uma leitura intensa dessa política. Além disso, muitos navegadores já oferecem mecanismos de leitura das políticas de privacidade que estejam no formato da P3P.

2.4 Agentes de privacidade

Os agentes de privacidade são ferramentas que proporcionam ao usuário informações sobre o grau de exposição e os riscos referentes à invasão de privacidade que um determinado site pode conter. A análise é feita através de verificações no site ou mesmo uma leitura da política de privacidade.

O agente denominado *Privacy Critics* [1] visa auxiliar os usuários a resguardar suas informações, através de sugestões e *feedbacks*. Entretanto, não age sem o consentimento do usuário. Sua finalidade é alertar o usuário, para que este tenha um controle maior sobre suas informações pessoais.

2.5 MASKS

O mecanismo MASKS (*Managing Anonymity while Sharing Knowledge to Servers*) visa garantir a privacidade de anonimato do usuário sem deixar de permitir a personalização [12]. Essa arquitetura fornece privacidade aos usuários através de máscaras ou pseudônimos. As máscaras são identificações temporárias que um usuário pode adotar durante a interação com um determinado site, sem ser identificado.

A arquitetura do MASKS possui dois componentes principais: o agente de privacidade e segurança PSA (*Privacy and Security Agent*) e o servidor de máscaras MS (*Masks Server*).

O primeiro componente, o PSA, atua em conjunto com o navegador, sendo um intermediário entre os usuários e o servidor de máscaras. Suas funcionalidades são: cifrar as requisições, manter o usuário informado sobre as máscaras atribuídas, permitir uma interação direta com o site (desligando o processo de mascaramento) e filtrar métodos conhecidos de invasão de privacidade. O segundo componente é o servidor de máscaras, um *proxy* de anonimato entre os usuários e o site. Ele é responsável pelo gerenciamento de máscaras e atribuição destas aos usuários. A atribuição de máscaras baseia-se no conceito de grupo, que

[1] http://www.w3.org

representa um tópico de interesse. Cada requisição do usuário é associada a um grupo, de acordo com sua semântica. Assim, associado às requisições haverá grupos e não mais indivíduos, permitindo a divulgação dos dados sobre o interesse comum dos usuários. Esses dados poderão ser utilizados para oferecer serviços personalizados e, ao mesmo tempo, a privacidade do usuário é preservada.

O servidor de máscaras possui ainda dois componentes: O Seletor, que é responsável pela seleção do grupo de interesse de cada requisição do usuário; e o Gerenciador de Máscaras, que dado um grupo, deverá determinar a máscara correta para o usuário.

A escolha do grupo semântico a ser associado a uma dada requisição é feita pelo componente Seletor do MASKS, utilizando uma árvore de categorias definida pelo ODP (*Open Directory Project*). A árvore em questão, uma vez carregada pelo servidor de máscaras, é utilizada apenas para consulta. Quando não é possível determinar qual grupo associar a uma requisição, o grupo raiz é então selecionado para garantir a privacidade e o mínimo de personalização. O Seletor utiliza um algoritmo de seleção de grupo, o qual pode ser considerado como o item mais importante do MASKS [12]. O objetivo deste algoritmo é garantir que o grupo selecionado seja eficiente e de acordo com os temas de interesse.

O algoritmo de seleção de grupo deve retornar o grupo semântico que mais corresponda à requisição do usuário. Em resumo, suas etapas são: 1) tentar determinar o grupo através dos termos de consulta da URL, através da tabela de termos; 2) caso não existam termos de consulta, é feita a busca pela existência de algum grupo semântico associado à URL da requisição, na tabela de conteúdo; 3) caso a URL não exista na tabela, o algoritmo tenta determinar o grupo utilizando termos presentes na URL, também na tabela de termos; 4) em último caso, o algoritmo retorna o grupo raiz (*Root*).

Pelo fato do servidor de máscaras atuar como um *proxy* entre o PSA e o usuário, é necessário a utilização de métodos de segurança de comunicação para evitar riscos à privacidade dos usuários. Entretanto, o PSA já efetua uma limpeza prévia de informações no cabeçalho HTTP das requisições, aumentando um pouco mais o sigilo de informações.

3 Mecanismos de personalização

Existem diversos mecanismos que oferecem personalização na Internet. Para a coleta de informações, geralmente são utilizados análise de dados em formulários e análise de navegação do usuário. Com os dados obtidos, são aplicadas técnicas de mineração de dados para determinar preferências, tendências, e entre outros.

É importante ressaltar que os próprios navegadores enviam informações, como sistema operacional do usuário, idioma, tipo de navegador, a página de referência, e outros, que podem ser utilizados para serviços de personalização.

A seguir, serão apresentados os mecanismos mais comuns de personalização.

3.1 Cookie

O *cookie*, definido pela RFC 2965 [14], é um grupo de dados trocados entre o usuário e o servidor, armazenados em um arquivo texto criado no computador do usuário, tendo como objetivo manter a persistência de sessões HTTP, mantendo o último estado antes de terminar a conexão, e retorná-lo no próximo acesso. As informações armazenadas nos *cookies* podem evidenciar sobre o perfil do usuário.

A manipulação dos *cookies* consiste em duas etapas: Na etapa 1, o cliente envia a requisição HTTP ao servidor; este, por sua vez, cria o *cookie* – contendo as informações desejadas – e o transmite até o computador do usuário, muitas vezes, sem o seu consentimento. O navegador do usuário recebe o *cookie* e o armazena em um arquivo chamado lista de *cookies*. Na etapa 2, o cliente envia, junto com a requisição, o *cookie* referente ao domínio determinado. O servidor recupera as informações contidas no *cookie* e retorna a resposta HTTP, podendo, inclusive, conter outro *cookie* [27].

Os *cookies* são amplamente utilizados, principalmente para identificação de usuários. Pelo fato deste mecanismo estar presente em todos os navegadores, pode ser usado como uma ferramenta primária para prover personalização [13].

3.2 Clickstream

O *clickstream*, também conhecido como *clickpath* ou seqüência de cliques, representa o caminho que o usuário percorre enquanto está visitando um determinado site. A seqüência de cliques obtida, quando aplicadas de

forma correta, pode proporcionar informações muito importantes para as empresas.

Todos os servidores Web tem a capacidade de registrar as requisições em arquivos de *log* ou em bancos de dados. Os dados de *log* de um servidor Web são a fonte primária de dados do *clickstream*, pois toda vez que o servidor Web responde uma requisição HTTP, uma entrada é anotada no arquivo de *log*. Os dados de *clickstream* também podem ser coletados por Provedores de Serviço de Internet, painéis de acesso ou mesmo manualmente, apesar da coleta através dos arquivos de *log* dos servidores ser mais comum [18].

Ao realizar o *clickstream* manualmente, não adianta colocar cada evento de página ou cada gesto do usuário em um banco de dados, pois um conjunto bruto de dados não é uma descrição útil de comportamento, porque pode levar a conclusões precipitadas [13] [3].

O *clickstream* é importante, do ponto de vista comercial, pois possibilita a identificação das preferências e padrões de comportamento do usuário; isso inclui qual área lhe interessa, a freqüência que a procura e quais as informações úteis para criar estratégias de marketing mais direcionadas ao usuário e, conseqüente, maior chance de sucesso [22].

3.3 Data Mining

Data Mining é um conjunto de técnicas que visam a aquisição de conhecimentos em bancos de dados. A motivação para esse mecanismo vem da dificuldade em obter conhecimento relevante de grandes volumes de dados, logo, é necessário o uso de ferramentas e técnicas que facilitem essa árdua tarefa [29].

Na Web, a descoberta de informações relevantes de padrões de navegação do usuário, a ponto de descrever seu comportamento, é extremamente valiosa para empresas e organizações que trabalham com comércio eletrônico. Segundo Han e Kamber [11], a arquitetura básica de um sistema de *Data Mining* teria os seguintes componentes: Banco de dados contendo todo o tipo de informação, Servidor de banco de dados – responsável por buscar as informações, baseadas nas requisições de mineração de dados que o usuário escolher –, Base de conhecimento contendo o domínio do conhecimento que será usado nas buscas, Mecanismo de *Data Mining* que é o conjunto de módulos funcionais para realizar tarefas de caracterização, classificação, evolução, entre outros, Módulo de Avaliação de Padrões que interage com os módulos do *Data Mining* procurando por padrões que julgar interessantes, e uma Interface Gráfica do Usuário intermediando a comunicação entre o usuário e o mecanismo de *Data Mining*, permitindo a interação do usuário, através de consultas ou tarefas).

As técnicas de mineração podem utilizar as mais diversas abordagens de implementação. As mais comuns incluem uso de técnicas estatísticas, raciocínio baseado em caos, redes neurais, árvores de decisão, algoritmos genéticos, entre outros.

A utilização do *Data Mining* em aplicações de comércio eletrônico é imprescindível, pois pode revelar informações valiosíssimas que não podem ser obtidas pelos métodos tradicionais.

3.4 Web Bugs

Web Bugs são mecanismos que tentam obter algum tipo de identificação de um usuário. Geralmente, estão inseridos em mensagens de e-mail ou páginas da Web. São imagens com o tamanho de 1 *pixel* $\times$ 1 *pixel*[2], transparentes, do tipo GIF (*Graphics Interchange Format*) [16], sendo, desta forma, imperceptíveis ao olho humano.

Uma página Web ou um e-mail podem ser visualizados sem que se perceba a presença de um *Web Bug*. No momento de carregamento da página Web ou do e-mail, a imagem referente ao *Web Bug* é requisitada a um servidor distinto que, por sua vez, obtém as informações do cabeçalho HTTP da requisição, contendo as informações a respeito do usuário (endereço IP, porta de acesso, navegador, sistema operacional, data e hora da exibição da imagem, entre outros) [4].

Os *Web Bugs* são armazenados em servidores distintos, diferentes dos que hospedam os sites de destino. A análise do *Web Bug* pode ser feita através de uma verificação no arquivo de *log* desses servidores.

Os *Web Bugs* podem trabalhar em conjunto com os *cookies*, monitorando quais sites são visitados pelo usuário [2].

4 Autômatos adaptativos

Os autômatos adaptativos constituem um mecanismo formal para a descrição de linguagens sensíveis ao contexto e estruturas de frase; sua simplicidade e facilidade de entendimento em relação ao modelo clássico

[2] *Picture Element*, menor elemento em um dispositivo de exibição, como um monitor, por exemplo.

de reconhecimento destas linguagens, a Máquina de Turing, fazem com que sua utilização seja muito ampla. A seguir, serão apresentados o autômato de pilha estruturado – uma variação do autômato de pilha tradicional, constituindo a base do autômato adaptativo – e o autômato adaptativo.

4.1 Autômato de Pilha Estruturado

O Autômato de Pilha Estruturado é um tipo de autômato de pilha formado por um conjunto de autômatos finitos mutuamente recursivos. Esses autômatos também são chamados de sub-máquinas. A pilha tem a finalidade exclusiva de armazenar os estados de retorno a cada chamada de uma sub-máquina. As chamadas e retornos consistem em transferir o controle entre uma sub-máquina e outra; assim, é efetuada uma operação chamada *lookahead*, que consiste em utilizar o símbolo de entrada apenas para a tomada de decisão do autômato, sendo consumido na próxima transição [19] [20].

Formalmente, o autômato de pilha estruturado pode ser definido como $M = (Q, A, \Sigma, \Gamma, P, q_0, Z_0, F)$, onde Q é conjunto finito de estados, A é conjunto de sub-máquinas a_i, Σ é o alfabeto de entrada, Γ é o alfabeto da pilha, P é o mapeamento, q_0 é o estado inicial, Z_0 é um símbolo especial, indicador de pilha vazia, que é o símbolo inicial da pilha, e F é conjunto de estados finais.

O mapeamento P é definido como uma relação $P \subseteq (Q \times \Sigma \times \Gamma) \times (Q \times (\Sigma \cup \{\epsilon\}) \times (\Gamma \cup \{\epsilon\}))$. A relação de transição é definida por $(q, s, g) \vdash (q', s', g')$, onde q é estado atual, $q \in Q$, s é símbolo a ser consumido, $s \in \Sigma$, g é o topo da pilha, $g \in \Gamma$, q' é o novo estado, $q' \in Q$, s' é o novo símbolo, $s' \in \Sigma \cup \{\epsilon\}$, e g' é o novo topo da pilha, $g' \in \Gamma \cup \{\epsilon\}$.

A linguagem aceita por um autômato de pilha estruturado M é dada por $L(M) = \{w \in \Sigma^* | (q_0, w, Z_0) \vdash^* (q_f, \epsilon, Z_0)$, onde $q_f \in F\}$.

O conjunto de sub-máquinas do autômato de pilha estruturado é representado por A. Cada sub-máquina i é definida como $a_i = (Q_i, \Sigma_i, P_i, q_{i0}, F_i)$, onde $Q_i \subseteq Q$ é o conjunto de estados da sub-máquina i, $\Sigma_i \subseteq \Sigma$ é o conjunto de símbolos de entrada da sub-máquina i, $P_i \subseteq P$ é o mapeamento da sub-máquina i, $q_{i0} \in Q_i$ é estado de entrada da sub-máquina i, e $F_i \subseteq Q_i$ é o conjunto de estados finais da sub-máquina i.

O autômato de pilha estruturado possui o mesmo poder de reconhecimento do autômato de pilha tradicional.

4.2 Autômato Adaptativo

O Autômato Adaptativo, proposto por Neto [19], é uma extensão do modelo do Autômato de Pilha Estruturado que permite o reconhecimento de linguagens mais complexas, dos tipos 1 e 0 segundo a Hierarquia de Chomsky. O termo adaptativo, neste contexto, pode ser definido como a capacidade de um dispositivo em alterar seu comportamento de forma espontânea. Logo, um autômato adaptativo tem como característica a possibilidade de sofrer alterações em sua topologia durante o processo de reconhecimento de uma dada cadeia [20].

Essa capacidade de alteração do autômato faz-se possível através da inclusão de ações adaptativas, que podem ser executadas antes e/ou depois de uma transição. A cada execução de uma ação adaptativa, o autômato tem sua topologia alterada, obtendo-se uma nova configuração. O objetivo de uma ação adaptativa é lidar com situações esperadas, mas ainda não consideradas, detectadas na cadeia submetida para reconhecimento pelo autômato [21]. Uma transição pode ter ações adaptativas associadas, que permitam a inclusão ou eliminação de estados e transições.

Ao executar uma transição que contém uma ação adaptativa associada, o autômato sofre mudanças, obtendo-se então uma nova máquina de estados. Para a aceitação de uma determinada cadeia, o autômato percorrerá um caminho em um espaço de máquinas de estados; em outras palavras, haverá uma máquina de estados inicial E_0, que iniciará o reconhecimento de uma determinada cadeia; máquinas de estados intermediárias E_i, que serão criadas ao longo do reconhecimento; e uma máquina de estados final E_n, que corresponde ao final do reconhecimento da cadeia. Seja a cadeia $w = \alpha_0\alpha_1...\alpha_n$; então o autômato M descreverá um caminho de máquinas de estados $< E_0, \alpha_0 > \rightarrow < E_1, \alpha_1 > \rightarrow ... \rightarrow < E_n, \alpha_n >$, onde E_i representa um autômato correspondente à aceitação da sub-cadeia α_i.

Formalmente, o autômato adaptativo pode ser definido como $M = (Q, A, \Sigma, \Gamma, P, q_0, Z_0, F, E_0, \phi)$, onde Q é o conjunto finito de estados, A é o conjunto de sub-máquinas a_i, Σ é o alfabeto de entrada, Γ é o alfabeto da pilha, P é o mapeamento, q_0 é o estado inicial, $q_0 \in \Sigma$, Z_0 é o símbolo especial, indicador de pilha vazia, que no início será o único símbolo na pilha, $Z_0 \in \Gamma$, F é o conjunto de estados finais, E é o conjunto de

todos os autômatos adaptativos do tipo M (inicialmente, o autômato é E_0, $E_0 \in E$), e ϕ é o conjunto de funções adaptativas.

O mapeamento P é definido como $P : Q \times \Sigma \times \Gamma \rightarrow Q \times (\Sigma \cup \{\epsilon\}) \times (\Gamma \cup \{\epsilon\}) \times (E \rightarrow E) \times (E \rightarrow E)$. A transição é da forma $(q, s, g) \vdash (q', s', g', A, B)$, onde q é o estado atual, $q \in Q$, s é o símbolo a ser consumido, $s \in \Sigma$, g é o topo da pilha, $g \in \Gamma$, q' é novo estado, $q' \in Q$, s' é o novo símbolo, $s' \in \Sigma \cup \{\epsilon\}$, g' é o novo topo da pilha, $g' \in \Gamma \cup \{\epsilon\}$, A representa as ações adaptativas a serem executadas antes da nova configuração do autômato, sendo opcional, e B representa as ações adaptativas a serem executadas após a nova configuração do autômato, sendo também opcional.

O conjunto de sub-máquinas do autômato adaptativo é representado por A, sendo cada sub-máquina i definida como $a_i = (Q_i, \Sigma_i, P_i, q_{i0}, F_i, \phi_i)$, onde Q_i é o conjunto de estados da sub-máquina i, Σ_i é o conjunto de símbolos de entrada da sub-máquina i, P_i é o mapeamento da sub-máquina i, q_{i0} é o estado de entrada da sub-máquina i, F_i é o conjunto de estados finais da sub-máquina i e ϕ_i é o conjunto de funções adaptativas da sub-máquina i.

As ações adaptativas são definidas através de funções adaptativas, contendo ou não argumentos. Formalmente, uma função adaptativa pode ser definida como $(F, P, V, G, C, E, I, A, B)$, onde F é o nome da função adaptativa, P é uma lista de parâmetros formais $(r_1, r_2...r_n)$ desta função, V é o conjunto de variáveis, G é o conjunto de geradores, C é o conjunto de padrões a serem consultados para o preenchimento das variáveis, E é o conjunto de padrões a serem consultados e removidos, I é o conjunto de padrões a serem inseridos na máquina de estados atual, A é a ação adaptativa a ser executada antes de F, e B é a ação adaptativa a ser executada após a execução de F. As variáveis são preenchidas uma única vez na execução da função adaptativa. Os geradores são tipos especiais de variáveis, usados para definir nomes a estados recém-criados; os valores definidos são únicos e identificados pelo símbolo $*$, por exemplo, g_1^*, g_2^*.

Uma ação adaptativa, definida através de funções adaptativas, é descrita por (F, P), onde F é o nome da função adaptativa, e P é a lista de parâmetros $(r_1, r_2...r_n)$ a serem passados a F.

Os autômatos adaptativos possuem três tipos de ações adaptativas elementares utilizadas para edição: ação adaptativa elementar de consulta (realiza uma busca no autômato por produções cujos componentes sejam correspondentes aos valores definidos), ação adaptativa elementar de remoção (remove uma produção de acordo com os valores definidos) e ação adaptativa elementar de inclusão (inclui uma produção de acordo com os valores definidos).

A linguagem aceita por um autômato adaptativo M é dada por $L(M) = \{w \in \Sigma^* | (q_0, w, Z_0) \vdash^* (q_f, \epsilon, Z_0)$, onde $q_f \in F\}$.

O autômato adaptativo, graças à sua capacidade de auto-modificação, possui o poder de representação de uma Máquina de Turing [26].

5 Autômato Adaptativo em Mecanismo de Privacidade e Personalização

Os autômatos adaptativos, devido à capacidade de auto-modificação, têm sua aplicação em diversas áreas, por exemplo, na robótica [30], onde dois autômatos adaptativos são responsáveis pelo mapeamento e navegação de um robô em um ambiente desconhecido; em sistemas de decisão e aprendizado de máquina [24], onde é apresentado um sistema que visa possibilitar a construção de árvores de indução adaptativas; no processamento de linguagem natural [17], onde é proposto um etiquetador morfológico treinado de acordo com um determinado corpus; na otimização de código em compiladores [15], onde é proposto um otimizador que faz uso de técnicas adaptativas para gerar um código objeto com tamanho reduzido.

Este artigo descreve um mecanismo de para proteção de privacidade e que também ofereça alguma personalização ao usuário, utilizando-se de um autômato adaptativo. Utilizando o MASKS como referência, o autômato em questão tem a característica de adequar-se ao contexto, de modo que a requisição do usuário possa sempre corresponder ao melhor grupo disponível, evitando inclusive que o nó raiz seja sempre selecionado para um determinado assunto, caso este não conste na árvore de categorias.

A utilização de um formalismo adaptativo para a representação de um mecanismo de privacidade e personalização na Web se torna adequada, devido à sua capacidade de auto-modificação e da possibilidade de representação de linguagens mais complexas. Logo, o formalismo permite a obtenção de resultados relevantes para fins de comparação e avaliação com os mecanismos existentes.

A Figura 1 mostra uma representação do algoritmo de seleção de grupo do MASKS como um autômato adaptativo:

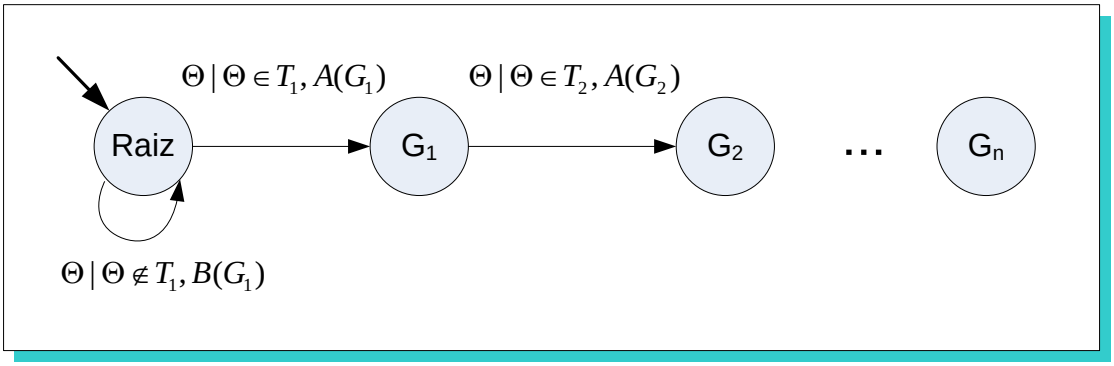

Figura 1: Representação do algoritmo de seleção de grupo do MASKS como um autômato adaptativo.

Os elementos presentes na Figura 1 são: Θ é o termo consultado, G_n representa o Grupo n, *Raiz* é o nó raiz, T_n é o conjunto de termos do grupo G_n, A é a função adaptativa a ser executada se $\Theta \in T_n$, e B é a função adaptativa a ser executada se $\Theta \notin T_n$.

O autômato adaptativo da Figura 1 terá como configuração final todos os grupos possíveis, dado um termo Θ. Os estados sem transições (não-alcançáveis) representam os grupos que não possuem o termo Θ. A Figura 2, apresentada a seguir, mostra um exemplo de configuração final do autômato:

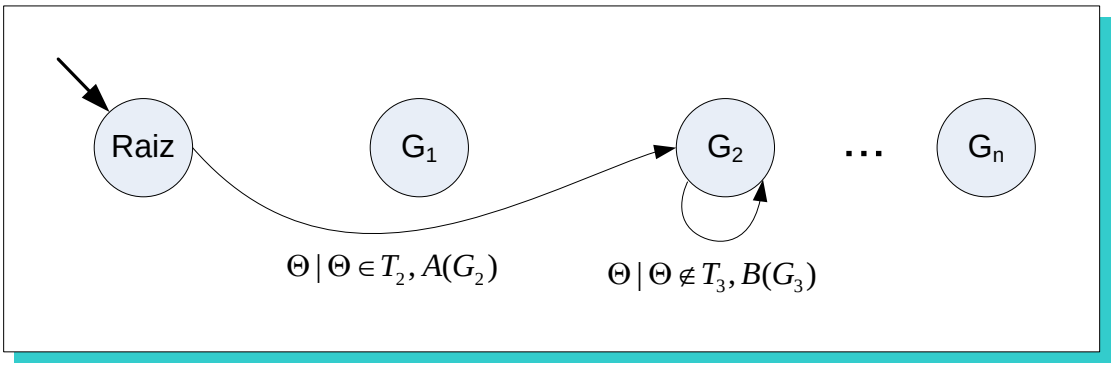

Figura 2: Exemplo de configuração final do autômato adaptativo.

Na Figura 2, partindo da raiz, o único estado acessível é G_2, portanto ele será o grupo escolhido. Caso existam mais estados, será necessário fazer uma escolha para decidir qual deles é o que melhor representa a requisição do usuário. Por outro lado, se nenhum estado for acessível, o grupo escolhido será o nó raiz.

O mecanismo ALFA, baseada no MASKS, visa incluir uma função adaptativa que, de acordo com o grupo selecionado, pode criar um outro grupo com um determinado termo associado. A Figura 3 ilustra a função adaptativa aplicada no grupo selecionado para mascarar uma requisição do usuário.

Os elementos presentes na Figura 3 são: Υ é o símbolo especial indicador de criação de um novo grupo a partir do grupo atual, G_i representa o Grupo i (escolhido para mascarar a requisição do usuário), Ψ é o conjunto de termos associados ao novo grupo, e C é a função adaptativa. A função adaptativa C é responsável pela criação de um novo grupo a partir do grupo selecionado. Esse novo grupo será uma especialização semântica de seus ancestrais.

Na Figura 3, o grupo G_i foi escolhido como o melhor grupo associado à requisição do usuário. Entretanto, de acordo com os termos pesquisados, pode ser viável a criação de um novo grupo, uma especialização de G_i, a ser utilizado. Para isso, o símbolo Υ é utilizado para indicar a criação de um novo grupo semântico a partir do estado atual.

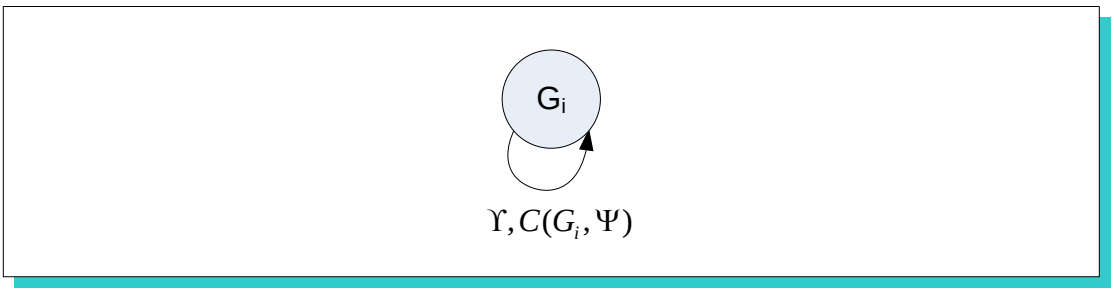

Figura 3: Função adaptativa aplicada no grupo selecionado para mascarar uma requisição do usuário.

A Figura 4 ilustra a nova situação, onde um novo grupo semântico G_j, uma especialização semântica de G_i, foi criado.

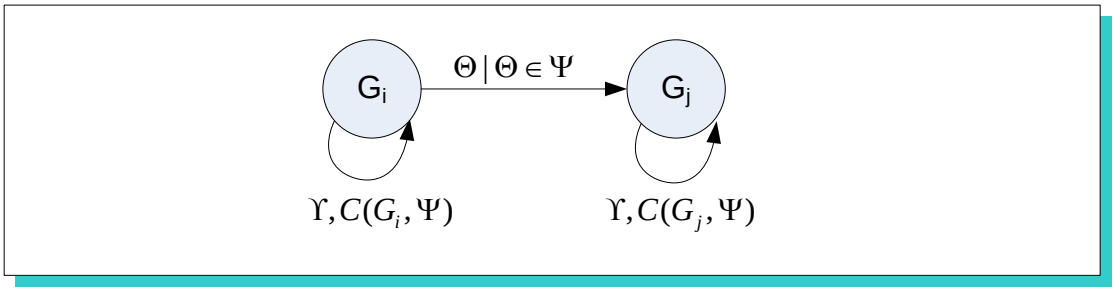

Figura 4: Exemplo de uma situação de criação de um novo grupo semântico.

Os elementos presentes na Figura 4 são: Υ é o símbolo especial indicador de criação de um novo grupo a partir do grupo atual, G_i representa o Grupo i, G_j representa o Grupo j (recém-criado), Ψ é o conjunto de termos associados ao novo grupo, Θ é o termo consultado, e C é a função adaptativa.

T_i é o conjunto de termos associados ao grupo G_i, inicialmente escolhido para mascarar a requisição do usuário. $\Psi \subseteq T_i$, assim o novo grupo G_j será um subconjunto de G_i, representando uma especialização semântica.

A capacidade de inserção e remoção de estados faz com que os grupos possam ser adaptados ao contexto de utilização do sistema, de modo que a escolha do grupo reflita melhor a requisição do usuário.

A utilização do formalismo de autômatos adaptativos no mecanismo de privacidade e personalização permite a análise dos passos necessários para a obtenção da melhor solução para o usuário, bem como sua otimização.

Para análise comparativa dos mecanismos apresentados, considere o caso em que a personalização esteja em um subconjunto do grupo G_3. Os autômatos adaptativos referentes aos mecanismos MASKS e ALFA receberão uma mesma entrada, representando um termo de consulta. Como resultado, será retornado o grupo que melhor represente a requisição do usuário. Para representar as etapas de percurso dos autômatos será empregada a notação apresentada na Tabela 1 para a relação $\vdash$, omitindo a representação da pilha, pois não é utilizada pelos dois autômatos.

Tabela 1: Notação empregada para representar as etapas de percurso dos autômatos.

$a_1)\ (q, s) \vdash (q', \epsilon)$	$a_2)\ (q, s) \vdash (q', s)$
$b_1)\ F_1(p_1), (q, s) \vdash (q', \epsilon), F_2(p_2)$	$b_2)\ F_1(p1), (q, s) \vdash (q', s), F_2(p_2)$
$c_1)\ F_1(p_1), (q, s) \vdash (q', \epsilon)$	$c_2)\ F_1(p_1), (q, s) \vdash (q', s)$
$d_1)\ (q, s) \vdash (q', \epsilon), F_2(p_2)$	$d_2)\ (q, s) \vdash (q', s), F_2(p_2)$

De acordo com a Tabela 1, q é o estado atual, s é o símbolo a ser consumido, q' é o novo estado, ϵ indica que o símbolo foi consumido, F_1 representa a função adaptativa a ser executada antes do consumo do símbolo, p_1 representa os parâmetros da função F_1, F_2 representa a função adaptativa a ser executada depois do consumo do símbolo, e p_2 representa os parâmetros da função F_2. Na primeira coluna, em todas as transições o símbolo s é consumido, enquanto na segunda, é feita uma operação de *lookahead*, onde o símbolo é mantido.

No autômato adaptativo representando o mecanismo MASKS, considere como entrada o termo a, $Q = \{Raiz, G_1, G_2, G_3\}$, T_i o conjunto de termos referentes ao grupo i, $a \notin T_1$, $a \notin T_2$, $a \in T_3$. As transições necessárias para atingir a personalização que esteja em um subconjunto do grupo G_3, a partir do estado de origem, são descritas a seguir (Tabela 2):

Tabela 2: Transições necessárias para atingir a personalização que está em um subconjunto do grupo G_3 no autômato adaptativo representando o mecanismo MASKS.

Origem	**Transição**	**Destino**
$Raiz$	$B(G_1), (Raiz, a) \vdash (Raiz, a)$	$Raiz$
$Raiz$	$B(G_2), (Raiz, a) \vdash (Raiz, a)$	$Raiz$
$Raiz$	$A(G_3), (Raiz, a) \vdash (G_3, \epsilon)$	G_3

$G_3 \in F$ é o grupo escolhido para mascarar a requisição do usuário. Percebe-se que G_3 não é a solução ótima, pois procura-se um subconjunto de G_3, uma especialização semântica do assunto de interesse retornado por G_3.

No autômato adaptativo representando o mecanismo ALFA, supondo como entrada o termo a e o símbolo especial de criação de grupo Υ, $Q = \{Raiz, G_1, G_2, G_3\}$, T_i é o conjunto de termos referentes ao grupo i, $a \notin T_1$, $a \notin T_2$, $a \in T_3$, $\Psi \subseteq T_3$, tem-se as seguintes transições (Tabela 3):

Tabela 3: Transições necessárias para atingir a personalização que está em um subconjunto do grupo G_3 no autômato adaptativo representando o mecanismo ALFA.

Origem	**Transição**	**Destino**
$Raiz$	$B(G_1), (Raiz, a) \vdash (Raiz, a)$	$Raiz$
$Raiz$	$B(G_2), (Raiz, a) \vdash (Raiz, a)$	$Raiz$
$Raiz$	$A(G_3), (Raiz, a) \vdash (G_3, \Upsilon)$	G_3
G_3	$C(G_3, \Psi), (G_3, \Upsilon) \vdash (G_{100}, \epsilon)$	G_{100}

$G_{100} \in F$ é o grupo escolhido para mascarar a requisição. G_{100} é proveniente de G_3, uma especialização semântica do pai; assim, representa com maior precisão a requisição do usuário do que seu ancestral G_3, mais abrangente. $\Psi \subseteq T_3$ é o conjunto de termos associados ao novo grupo, mais específico que T_3, portanto apresenta uma solução melhor.

Ao submeter uma mesma entrada a nos dois autômatos adaptativos, percebeu-se que o grupo retornado pelo mecanismo ALFA representa melhor a requisição do usuário, pois possibilitou uma personalização mais direcionada ao seu interesse.

O resultado apresentado pelo mecanismo ALFA é justificado pela utilização dos conjuntos de termos associados a cada grupo.

6 Conclusões

A utilização de um formalismo com autômatos adaptativos em um mecanismo de privacidade e personalização demonstrou ser adequada para tal, devido à característica inerente da Internet em ser adaptável e dinâmica. Os autômatos adaptativos permitiram que o formalismo pudesse abranger situações esperadas, mas não identificadas até então, sobre o contexto de utilização do mecanismo, tratando-as de acordo com as ações definidas nas funções adaptativas. Uma vez que o mecanismo de privacidade e personalização tem a capacidade de tratar das mudanças de contexto e assim refletir com maior precisão a requisição de um usuário, o grau de satisfação deste em relação à utilização do mecanismo aumenta consideravelmente.

O formalismo apresentado pode ser utilizado como modelo de referência para a criação de novos mecanismos e, a partir destes, obter medidas para avaliação e comparação, não somente em relação ao desempenho e execução, mas também em relação à satisfação do usuário.

Referências

[1] Ackerman, M., and Cranor, L. F. Privacy critics – safeguarding users' personal data. *Web Techniques* (September 1999).

[2] Aiello, W., and McDaniel, P. *Lecture 1, Intro: Privacy.* Stern School of Business, NYU, 2004.

[3] Bogo, L. R. *Criação de comunidades virtuais a partir de agentes inteligentes: Uma aplicação em e-learning.* Dissertação de Mestrado – Universidade Federal de Santa Catarina, Florianópolis, 2003.

[4] CERT.BR. *Cartilha de Segurança para a Internet, parte VI: Spam.* Versão 3.0. Comitê Gestor da Internet no Brasil, 2005. Disponível em <http://cartilha.cert.br>. Acesso em 08/03/2007.

[5] Chaum, D. Untraceable electronic mail, return addresses and digital pseudonyms. *Communications of the ACM 24*, 2 (February 1981), 84–88.

[6] Coyle, K. *A social analysis of the platform for privacy preferences (P3P).* Platform for Privacy Preferences (P3P) Project, 1999. Disponível em <http://www.w3.org/P3P/>. Acesso em 04/03/2007.

[7] Gabber, E., Gibbons, P., Matias, Y., and Mayer, A. *How to Make Personalized Web Browsing Simple, Secure, and Anonymous.* Bell Laboratiories, Lucent Technologies, 1997.

[8] Goldberg, I., Wagner, D., and Brewer, E. Privacy-enhancing technologies for the internet. *Proceedings of IEEE Spring CompCon* (1997).

[9] Goldschlag, D., Reed, M., and Syverson, P. *Anonymous Connections and Onion Routing.* Center for High Assurance Computer Systems, Naval Research Laboratory, Washington DC, 1996.

[10] Grande, R. E. *Sistema de Integração de Técnicas de Proteção de Privacidade que permitem Personalização.* Dissertação de Mestrado – Programa de Pós-Graduação em Ciência da Computação, Departamento de Computação. Universidade Federal de São Carlos, São Carlos, 2006.

[11] Han, J., and Kamber, M. *Data Mining: Concepts and Techniques*, 1 ed. Morgan Kaufmann, San Francisco, CA, 2001.

[12] Ishitani, L. *Uma Arquitetura para Controle de Privacidade na Web.* Tese de Doutorado – Departamento de Ciência da Computação. Universidade Federal de Minas Gerais, 2003.

[13] Kimball, R., and Merz, R. *Data Webhouse: construindo o data warehouse para a Web.* Campus, Rio de Janeiro, 2000.

[14] Kristol, D., and Montulli, L. *HTTP State Management Mechanism.* RFC 2965 – Bell Laboratories, Lucent Technologies, 2000. Disponível em <http://www.ietf.org/rfc/rfc2965.txt>. Acesso em 04/03/2007.

[15] Luz, J., and Neto, J. J. Tecnologia adaptativa aplicada à otimização de código em compiladores. *IX Congreso Argentino de Ciencias de la Computación* (October 2003).

[16] Martin, D. *Detecting Web Bugs with Bugnosis: Privacy Advocacy through Education.* Boston University Computer Science Department, 2003. Disponível em <http://www.bugnosis.org/faq.html>. Acesso em 05/03/2007.

[17] Menezes, C., and Neto, J. J. Um método para a construção de analisadores morfológicos, aplicado à língua portuguesa, baseado em autômatos adaptativos. *V PROPOR, Encontro para o Processamento Computacional de Português Escrito e Falado* (November 2000).

[18] Montgomery, A. L. *Using Clickstream Data to Predict WWW Usage.* Working Papers Series, Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA, 2000.

[19] Neto, J. J. *Contribuições à Metodologia de Construção de Compiladores.* Tese de Livre Docência – Escola Politécnica da Universidade de São Paulo, São Paulo, 1993.

[20] Neto, J. J. Adaptive automata for context-dependent languages. *ACM SIGPLAN Notices 29*, 9 (1994).

[21] Neto, J. J. Solving complex problems efficiently with adaptive automata. *Lecture Notes in Computer Science – Implementation and Application of Automata 5th International Conference 2088* (2000).

[22] Nogueira, A. L. B., and Oliveira, L. R. *Uma análise da aplicabilidade de Data Warehouse em ambientes empresariais.* Faculdade Ruy Barbosa, Salvador, 2004.

[23] Pfitzmann, A., and Köhntopp, M. Anonymity, unobservability, and pseudonymity – a proposal for terminology. *Designing Privacy Enhancing Technologies: Proceedings of the International Workshop on the Design Issues in Anonymity and Observability 2009*, 2 (July 2000), 1–9.

[24] Pistori, H., and Neto, J. J. Adaptree – proposta de um algoritmo para indução de Árvores de decisão baseado em técnicas adaptativas. *Anais Conferência Latino Americana de Informática – CLEI 2002* (November 2002).

[25] Reiter, M. K., and Rubin, A. D. *Crowds: Anonymity for Web Transactions.* AT&T Labs – Research, 1997.

[26] Rocha, R. L. A., and Neto, J. J. Autômato adaptativo, limites e complexidade em comparação com máquina de turing. *Proceedings of the second Congress of Logic Applied to Technology – LAPTEC'2000* (2001), 33–48.

[27] Shiflett, C. *Essential PHP Security*, 1 ed. O'Reilly Media, Sebastopol, CA, 2005.

[28] Shubina, A. M., and Smith, S. W. Using caching for browsing anonymity. *Dartmouth Computer Science Technical Report TR2003-470* (July 2003).

[29] Silva, M. P., Boscarioli, C., and Peres, S. M. Análise de logs da web por meio de técnicas de data mining. *I Congresso de Tecnologias para Gestão de Dados e Metadados do Cone Sul* (September 2003).

[30] Sousa, M. A. A., and Hirakawa, A. H. Robotic mapping and navigation in unknown environments using adaptive automata. *Proceedings of International Conference on Adaptive and Natural Computing Algorithms – ICANGA* (March 2005).