

UM ESTUDO COMPARATIVO DE SIMULADORES WAP E SEU USO PARA UMA APLICAÇÃO DE ACESSO A BANCO DE DADOS

R.V.V. Rosa, R.B.K.A. Antunes e M.A.R. Dantas
Departamento de Ciências da Computação,
Universidade de Brasília, 70.910-970
mario@cic.unb.br

Resumo

Nestes últimos anos tem-se verificado uma tendência na direção do uso de dispositivos menores e uma maior necessidade de obtenção de informações em movimento. Estes fatos abriram o caminho para uma nova abordagem, o *Wireless Application Protocol* (WAP). Esta tecnologia faz uso dos mecanismos do mundo *web* e da telefonia móvel.

O WAP é uma arquitetura de protocolo para aplicações sem fio, que serve para a comunicação e também como um ambiente de aplicações para distribuição de recursos de informação, serviços avançados de telefonia e acesso à Internet a partir de dispositivos móveis como telefones celulares, *paggers* e PDAs (*Personal Digital Assistants*) [1]. Este padrão é global e aberto, surgido no final dos anos noventa [3], quando as empresas Ericsson, Motorola, Nokia e Phone.com fundaram o WAP Fórum. O objetivo deste Fórum era definir uma especificação para o desenvolvimento de aplicações sobre as redes de comunicação sem fio. Essa especificação define um conjunto de protocolos nas camadas de aplicação, sessão, transação, segurança e transporte, que possibilita aos operadores, fabricantes e provedores de aplicações enfrentarem os desafios dos diferenciados e avançados serviços sem fio e criarem serviços rápidos e flexíveis [5].

Neste trabalho apresentamos um estudo comparativo de utilização de quinze diferentes simuladores WAP. Nossa contribuição visa uma certificação de facilidades e adequação destes pacotes quanto ao suporte para os desenvolvedores de aplicações WAP. Em adição, desenvolvemos uma aplicação de acesso a banco de dados que nos auxilia a certificação empírica do nosso trabalho.

1. Introdução

Imagine um grande empresário indo para uma importante reunião de negócios em outro país. A caminho do aeroporto, ele se lembra de que não anotou o telefone do hotel no qual ficará hospedado. Rapidamente, ele acessa o banco de dados de seu computador pessoal por meio de um dispositivo móvel, seja ele um celular (ou *palm top*) e obtém o telefone procurado, teclando somente alguns dados.

Agora, suponha que este mesmo empresário está atrasado para o seu vôo internacional e, no meio do caminho, fica preso em um engarrafamento. Utilizando-se novamente do seu aparelho móvel, ele acessa as informações em tempo real a respeito do horário dos vôos procedentes de sua cidade, confirmando se existe algum atraso (ou não) do seu vôo.

Estes são apenas dois exemplos do uso da tecnologia WAP é capaz de trazer inúmeros benefícios para o nosso dia-a-dia. Serviços cada vez mais úteis e ágeis deverão aparecer com a evolução prevista para a tecnologia sem fio. Para se ter uma idéia do alcance desta tecnologia, o Yankee Group previu que, por volta de 2003, mais de um bilhão de pessoas utilizarão algum tipo de equipamento móvel para conectar-se à Internet [2]. Além disso, segundo previsões do Meta Group, mais da metade do acesso à Internet será através de *não-PC's* [1]. Estes números são surpreendentes, mas se levarmos em conta o dinamismo o qual o mundo está presenciando e que, como consequência, as pessoas estão cada vez mais ocupadas e ávidas por informações de todos os tipos, tamanha expansão não é impossível. E é justamente essa necessidade constante de informações atualizadas, a qualquer hora e em qualquer lugar, que o mercado *wireless* busca satisfazer.

Com isso, o objetivo deste trabalho é realizar, a partir das informações colhidas de um vasto conjunto de simuladores WAP, uma análise comparativa do comportamento desses pacotes dentro do padrão de programação WAP 1.1, estipulado pelo WAP Forum. Vale aqui ressaltar que, na verdade, um simulador WAP é um software/aplicativo que simula a existência de um dispositivo WAP dentro do ambiente computacional. E a partir dele, é possível acessar páginas WAP como em qualquer dispositivo real de acesso a esta tecnologia. Dessa forma, visamos facilitar o trabalho de futuros programadores, mostrando os recursos e as características melhor aceitas nos dispositivos móveis atuais. Em adição, implementamos um aplicativo de

acesso a banco de dados buscando mostrar um pouco da realidade que a tão recente tecnologia WAP é capaz de trazer aos seus usuários.

2. Ambiente Experimental

Para a realização dos experimentos, dividimos o ambiente de trabalho em dois grupos distintos. O primeiro foi montado para testar os simuladores WAP que podem funcionar numa rede local ou, até mesmo, em computadores que não estejam ligados a nenhuma rede. Nesse ambiente não existe a necessidade de conexão à Internet. Por outro lado, o segundo ambiente foi montado para os simuladores que trabalham exclusivamente *on-line*. Esses simuladores apenas acessam os *sites* que foram anteriormente registrados em algum domínio válido na Internet.

O primeiro ambiente foi montado sobre o sistema operacional Windows 98, que foi instalado em um computador com processador K6-II da AMD de 500 Mhz e com 128 Mb de memória RAM. Como, neste caso, não houve necessidade, a máquina não foi conectada a nenhuma rede. Além disso, instalamos e configuramos o servidor *web* da Microsoft, conhecido como *Personal Web Server* (PWS), que acompanha o referido sistema operacional. A configuração do PWS fez com que esse ambiente também servisse para testar a aplicação que desenvolvemos para acessar um banco de dados *Access*, através da tecnologia ASP (*Active Server Pages*) que permitiu a geração de conteúdo dinâmico para as páginas WML. É interessante mencionar que este ambiente de teste foi considerado, pois é uma configuração bastante representativa para os usuários convencionais.

Para o segundo ambiente, que serviu exclusivamente para o teste de simuladores *on-line*, fez-se necessária a procura de um *site web* que hospedasse os baralhos (ou *decks*) WML usados para os testes. Dentre os vários hospedeiros gratuitos existentes na *web*, optamos por um serviço brasileiro denominado WAPSite, disponível em [7].

3. Descrição dos Experimentos

O primeiro passo deste trabalho foi escolher os simuladores que iriam fazer parte do estudo. Após extensa pesquisa, escolhemos um grupo de simuladores bastante representativo para iniciarmos a busca de nosso objetivo. Em seguida, instalamos e configuramos cada um dos simuladores escolhidos nos ambientes que utilizamos para realizar os testes. Além disso, dedicamos uma parte do nosso tempo para aprender como manipular cada simulador, já que cada um deles possui uma forma particular de operação.

A fim de obter dados suficientes para uma boa análise dos simuladores WAP, nós resolvemos desenvolver um aplicativo que poderia ser visualizado, pelo menos teoricamente, em qualquer simulador. Para que esse aplicativo fosse genérico o bastante a ponto de ser adequado para testar o maior número de recursos e de simuladores, tivemos que fazer muitas correções e alterações ao longo do desenvolvimento, até mesmo quando já estávamos na fase final de teste dos simuladores, o que acarretou na necessidade de repetição de todos os testes. Com o aplicativo desenvolvido, foi possível visualizar como é que os simuladores implementam a maioria dos recursos WML e WMLScript, e até mesmo verificar se os implementam, ou não. Para melhor estruturar a análise, dividimos os testes em oito grupos distintos, sendo que cada um deles possui recursos e *tags* relacionados (**figura 1**).



Figura 1 – Grupos de teste

A separação do teste em grupos trata-se de uma mera formalização, que visa facilitar a navegação de quem deseje testar um dispositivo WAP por meio do aplicativo desenvolvido. Assim, o objetivo é comum a todos os oito grupos, ou seja, testar a funcionalidade ou não dos recursos disponibilizados pela tecnologia WAP, versão 1.1. Detalhando: o grupo denominado ‘Formatação’ é encarregado das *tags* e comandos de formatação de texto e *layout*, como alinhamento; ‘Decks e Cards’ verifica a aceitação de recursos básicos de paginação e título; ‘Eventos’ testa recursos de navegação, como comandos de *timer* e de *back*; o grupo seguinte, ‘Tarefas e Variáveis’, analisa comandos mais simples de navegação, como *refresh*, além de se ocupar do uso de variáveis; ‘Entradas’, como o próprio nome já diz, é responsável pela entrada de dados; o grupo ‘WMLScript’ testa a possibilidade de uso desta linguagem pelo dispositivo; o próximo grupo, ‘Âncoras e Imagens’, é auto-explicativo; e, por fim, ‘ASP’ verifica a aceitação de código ASP.

Para cada um dos grupos testados, uma tabela foi preenchida de acordo com os resultados percebidos nos testes realizados. Com isso, tais tabelas informam se o simulador aceita, rejeita ou ignora o comando ou função testado. De posse de todas as tabelas completamente preenchidas, nos empenhamos em criar um aplicativo, que fosse ao mesmo tempo útil aos usuários de WAP e funcionasse sem problemas na grande maioria dos dispositivos móveis, tomando por base os resultados obtidos com os testes efetuados. Pensando na limitação da agenda de telefones dos aparelhos celulares, decidimos criar um aplicativo WAP que acessa um banco de dados, o qual armazena nome, sobrenome, endereço, telefone e data de nascimento em uma quantidade infinitamente superior aos dos referidos aparelhos. Quanto à programação, primamos por utilizar apenas os comandos que são suportados por grande parte dos celulares, para não limitar o acesso dos usuários em função do dispositivo que possuem.

4. Resultados Experimentais

Seria uma tarefa extremamente exaustiva analisar e comparar os resultados obtidos para cada um dos simuladores e depois compara-los uns com os outros. Por isso, decidimos realizar a análise dos resultados observados utilizando como referência duas frentes distintas: uma qualitativa e uma outra quantitativa.

Qualitativamente, fez-se necessário consultar as tabelas construídas a partir dos testes de cada um dos grupos, quais sejam: formatação, *decks* e *cards*, eventos, tarefas e variáveis, entradas, âncoras e imagens, WMLScript, e, por fim, ASP. Com isso, foi possível analisar o comportamento dos simuladores frente aos recursos e comandos testados e, conseqüentemente, pôde-se chegar à uma conclusão de quais as *tags* e comandos melhor implementados e mais aceitos dentre todos aqueles utilizados no aplicativo de teste.

Para realizar a análise quantitativa, optamos por fazer um gráfico informando quais, dentre todos os simuladores testados, implementam o maior número de recursos e quais devem ser levados em conta quando queremos desenvolver um aplicativo que funciona corretamente num maior número possível de dispositivos clientes. Esse gráfico só pôde ser construído a partir da inferência de uma tabela de pontuação que indica se determinados recursos são ou não aceitos ou implementados pelos simuladores. Assim, de acordo com os resultados obtidos nos testes, decidimos estabelecer a seguinte tabela de pontuação:

Comportamento	Pontuação
Suporta o recurso	3
Ignora o recurso	1
Não suporta o recurso	-1
Suporta ou ignora o recurso, porém com restrições	2

Tabela 1 – Parametrização dos resultados colhidos

O gráfico 1 apresenta um ranking com a soma total de pontos alcançados por cada um dos simuladores testados, mediante o comportamento observado em todos os testes realizados. Esse ranking não tem a função de indicar qual é o melhor simulador, apesar de poder ser usado para esse fim, mas sim de oferecer uma visão de quais simuladores devem ser considerados quando do desenvolvimento de uma aplicação e qual a implementação deve ser feita para alcançar os usuários de determinados clientes. Porém, é notório ressaltar o destaque negativo de dois simuladores, que ficaram com a pontuação muito abaixo da média: um funciona melhor como somente editor de códigos WML e outro não segue as normas padrões do WAP 1.1, determinadas pelo WAP Forum. Dessa forma, é melhor não utilizá-los como base para visualização de aplicativos WAP. Outro ponto importante a ser ressaltado é o fato de que nenhum dos simuladores WAP testados alcançou a pontuação máxima de 130 pontos, ou seja, nenhum deles responde fielmente a todos os recursos e *tags* implementados no programa de testes.

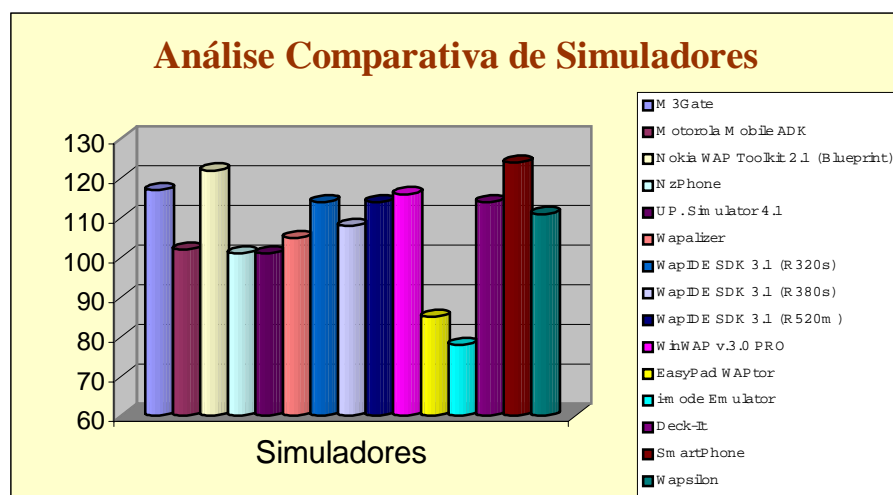


Gráfico 1 – Análise quantitativa dos simuladores WAP

5. Conclusão

O rápido crescimento do mercado da telefonia celular e o aumento do acesso à Internet através de dispositivos móveis têm criado um novo mercado para os programadores de aplicativos *wireless*. Por outro lado, existem numerosos navegadores e numerosos clientes que os executam, o que vem causando problemas no desenvolvimento, pois, conforme fica claro com o estudo aqui desenvolvido, cada dispositivo interpreta o aplicativo WAP à sua maneira. Isso tem ocorrido mesmo com o advento do WAP Forum, buscando criar padrões de protocolos e de especificação – como o WAP 1.1. Assim, como os desenvolvedores terão suas aplicações acessadas por usuários que possuem diferentes dispositivos clientes, eles devem dar muita importância aos resultados apresentados nos testes de simuladores, pois só assim poderão obter um almejado sucesso com as suas aplicações.

A partir daí, surgem duas vertentes bastante distintas a respeito de como se deve encarar tais resultados: uma que nivela os simuladores por cima e outra, por baixo. No primeiro caso, a intenção é criar uma pressão sobre o mercado de dispositivos móveis com acesso à Internet para que ele, então, acompanhe a evolução dos recursos WAP e o desenvolvimento de páginas cada vez mais trabalhadas. Tal fato pode ser

encarado como semelhante ao ocorrido com a *web*, que, no início, era caracterizada especialmente por páginas simples, sem muitos recursos, o que as tornava bastante uniformes. Atualmente, com a imensa expansão da Internet, são raros os *sites* que não utilizam algum aplicativo externo ou algum recurso mais sofisticado a fim de incrementar ou facilitar a navegação. Um ponto negativo nesta abordagem é que, enquanto não houver uma sincronização entre as funções disponibilizadas pelo WAP e o suporte a elas por parte dos dispositivos, a quantidade de usuários pode ficar bastante limitada.

É justamente em cima deste fator que a segunda vertente se apóia. Ao se restringir aos recursos básicos do WAP, o desenvolvedor cria aplicativos suportados por um número maior de usuários, já que haverá uma maior quantidade de dispositivos atuais compatíveis com o código programado. Mais uma vez tecendo um comparativo com a WWW, renomados escritores e estudiosos da usabilidade dentro da *web*, como Jakob Nielsen [18], sugerem que se faça páginas não muito complexas para a *web* – entenda-se: sem *frames*, *plug-ins* e outras coisas do gênero que, segundo eles, apenas tornam a navegação mais lenta e difícil. Páginas complexas, em outras palavras, repelem um grande número de possíveis usuários.

Independente do ponto de vista preferido, uma necessidade latente é que haja um empenho maciço no aperfeiçoamento do protocolo WAP, buscando uma certa homogeneidade no que se refere ao comportamento dos inúmeros dispositivos já existentes, e, é claro, dos que ainda estão por vir.

De qualquer maneira, a sinergia entre a Internet, que nos deu poder de obter informações de todos os tipos, e as redes digitais sem fio, permitindo-nos um acesso independente de local e tempo a estas informações, surge como uma perspectiva imponente acerca do futuro da tecnologia da informação.

Finalizando, acreditamos que nossa contribuição é importante uma vez que além de apresentarmos um estudo com a indicação das facilidades e restrições de quinze pacotes diferentes de software WAP, realizamos testes empíricos que podem auxiliar os programadores no desenvolvimento de suas aplicações.

Referências

- [1] AREHART, Charles et al, **Professional WAP**, São Paulo, Makron Books, 2001.
- [2] NzPhone - <http://www.nzphone.com.br>
- [3] FROST, Martin. **Aprendendo WML e WMLScript.**, Rio de Janeiro, Campus, 2001.
- [4] WAP Forum - <http://www.wapforum.org>
- [5] International Engineering Consortium - <http://www.iec.org/tutorials/wap/topic01.html>
- [6] Phone.com - <http://www.phone.com/industry/wap.html>
- [7] <http://www.wapsite.com.br>