

Integração de Sistemas Embutidos utilizando *Web Services*

Guilherme Bertoni Machado , Frank Siqueira

Universidade Federal de Santa Catarina - Departamento de Informática e Estatística
Campus Universitário - Trindade - 88040-900 - Florianópolis - SC - Brasil

{bertoni, frank}@inf.ufsc.br

Resumo. *Sistemas Embutidos estão cada vez mais integrados à Internet através da interconexão destes dispositivos em redes TCP/IP. Web Services tem se mostrado uma arquitetura eficiente para a interconexão de sistemas através da rede; por outro lado, a integração de aplicações provenientes dos Sistemas Embutidos a outros sistemas vem se mostrando cada vez mais necessária. Este trabalho busca demonstrar a viabilidade de integração de Sistemas Embutidos a outros sistemas utilizando Web Services, através do estudo, da modelagem e da disponibilização de um web service projetado com o toolkit de desenvolvimento gSOAP tendo como ambiente o sistema embutido SHIP e da sua integração com outros sistemas.*

1. Introdução

A *World Wide Web* teve como proposta inicial somente a troca de documentos entre os computadores, utilizando como suporte para comunicação em ambiente distribuído a rede Internet. Porém, com o crescimento e popularização desta, surgem várias aplicações que, ao utilizar a arquitetura de comunicação da *Web* - que é baseada em TCP/IP - necessitam de um método eficiente para intercâmbio de dados (informações). A Arquitetura *Web Services*¹ (WS) busca a solução deste problema, pois ao utilizar padrões abertos de protocolos e linguagens, possibilita a integração das mais diversas aplicações distribuídas sem se preocupar com a heterogeneidade intrínseca dos ambientes distribuídos [1].

WS apresentam-se como uma evolução das tecnologias de comunicação baseadas em Objetos Distribuídos [2], pois ao invés de se referenciar a uma interface de um objeto, um *web service* busca uma mudança deste paradigma para uma Arquitetura Orientada a Serviços (SOA) [3].

Sabendo de toda a potencialidade que WS podem oferecer e analisando a tendência de uma maior integração entre Software e Hardware, nada mais natural em se propor a construção/implementação de *web services* também em sistemas embutidos, uma vez que estes representam a maior fatia do mercado de processadores e que suas aplicações estão cada vez mais sofisticadas, principalmente pelo fato de que estes estão cada vez mais integrados à Internet através da sua interconexão em redes TCP/IP.

O presente trabalho foi realizado com o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq - Brasil e do projeto FunPesquisa-UFSC 2004.

¹Na literatura pesquisada também foram encontrados os termos *WebServices* e *Serviços Web* (termo em português). Optou-se por manter a grafia proposta pelo W3C, e *web service* como uma instância (aplicação) de um *Web Service*.

Como consequência deste processo de integração torna-se possível para as aplicações provenientes dos Sistemas Embutidos interagirem com aplicações disponíveis em outros sistemas. De modo a comprovar essa possibilidade, este trabalho se propõe a realizar a modelagem e a disponibilização de um *web service* num sistema embutido, sendo que a aplicabilidade dessa proposta é apresentada em um cenário (caso de uso) na área médica.

2. Principais Tecnologias Empregadas por *Web Services*

Sabendo que WS não é uma tecnologia específica, e sim um conjunto de protocolos de comunicação e interoperabilidade (consolidados e/ou emergentes) [4], revisamos as principais tecnologias empregadas pelos WS.

A linguagem **XML - *Extensible Markup Language*** [5] consiste em um padrão para especificar e processar as informações (dados). XML é empregada como base de comunicação dos WS [1]. Baseada em TAGs (delimitadores), a linguagem descreve os objetos, seus atributos, métodos e parâmetros de forma que os dados sejam interpretados pelas aplicações.

SOAP - *Simple Object Access Protocol* [5] é um protocolo leve para a troca de dados XML pela *Web* [6]. Serve como um envelope de um documento XML para que este possa ser transmitido pela *Web*. Seu envio pela rede pode ser feito usando diversas tecnologias, incluindo SMTP, HTTP e FTP dentre outros.

WSDL - *Web Service Description Language* [5] é uma linguagem, baseada em XML, utilizada para descrever WS. Esta descrição inclui detalhes como: definição dos tipos dos dados, operações suportadas pelo serviço, formatos das mensagens de entrada/saída, endereço de rede, mapeamento de protocolos, etc [6].

UDDI - *Universal Description, Discovery, and Integration* - consiste em um serviço de nomeação e localização de *web services* estruturado na forma de repositórios [7]. Após um *web service* ser colocado em operação, este pode ser publicado em um repositório UDDI. A partir desse momento, as informações necessárias para a localização e a utilização do serviço disponibilizado por este *web service* se tornam acessíveis para os clientes na forma de um arquivo WSDL.

3. Sistemas Embutidos e *Toolkits* para Desenvolvimento

Podemos definir um Sistema Embutido como um dispositivo microprocessado, portanto programável, que tem como proposta utilizar o seu poder computacional para uma finalidade específica. Ao contrário das plataformas "genéricas" como o PC, normalmente um sistema embutido é desenvolvido para uma aplicação exclusiva (como por exemplo, decodificadores de TV a cabo, controladores de fornos de microondas, aparelhos de DVD, chips de telefones celulares, dentre outros), logo, possui características bem peculiares tais como [8]: algoritmos complexos, interfaces com o usuário específicas, suporte a tarefas com requisitos temporais, suporte a múltiplas tarefas sendo executadas com prioridades diferentes, custo de fabricação reduzido, monitoração e controle do consumo de energia.

No mercado existem inúmeras soluções existentes que se encaixam, isto é, possuem as funcionalidades (pilha TCP/IP nativa e suporte a HTTP) necessárias para a in-

tegração dos dispositivos ligados a estes sistemas embutidos a ambientes distribuídos via WS.

Quanto ao desenvolvimento de *web services* para sistemas embutidos existe uma quantidade razoável de *toolkits*, sejam eles específicos para esta abordagem, ou então que possuem suporte para este tipo de desenvolvimento. Como principais restrições para a escolha de um toolkit apontamos as seguintes características: que funcione em hardware de porte limitado (processamento, memória, sistema operacional, etc), ambiente de desenvolvimento amigável e boa documentação para o usuário/desenvolvedor.

4. Características da Abordagem Proposta

Um vez que o sistema embutido usado como plataforma de estudo/utilização possui funcionalidades que possibilitam o armazenamento de dados e a troca destes via protocolos Internet, existe a possibilidade da construção de um *web service* neste dispositivo.

A proposta inicial deste trabalho consiste em disponibilizar uma descrição dos serviços disponíveis no *web service* via WSDL e que existam pelo menos dois tipos de serviço, para que numa próxima etapa possamos trabalhar aspectos de QoS e processamento em tempo-real com prioridades distintas. Uma vez definidos os serviços e a descrição destes, a elaboração da aplicação (cliente) pode ser feita de acordo com as necessidades e recursos do usuário, podendo ser uma página, outro sistema embutido, um programa, etc.

Basicamente o sistema proposto segue a estrutura representada na figura 1, na qual um cliente envia suas mensagens XML envelopadas com SOAP de acordo com o tipo de serviço requerido (previamente conhecido através do arquivo WSDL localizado no próprio Sistema Embutido e/ou em um repositório UDDI) através do protocolo HTTP.

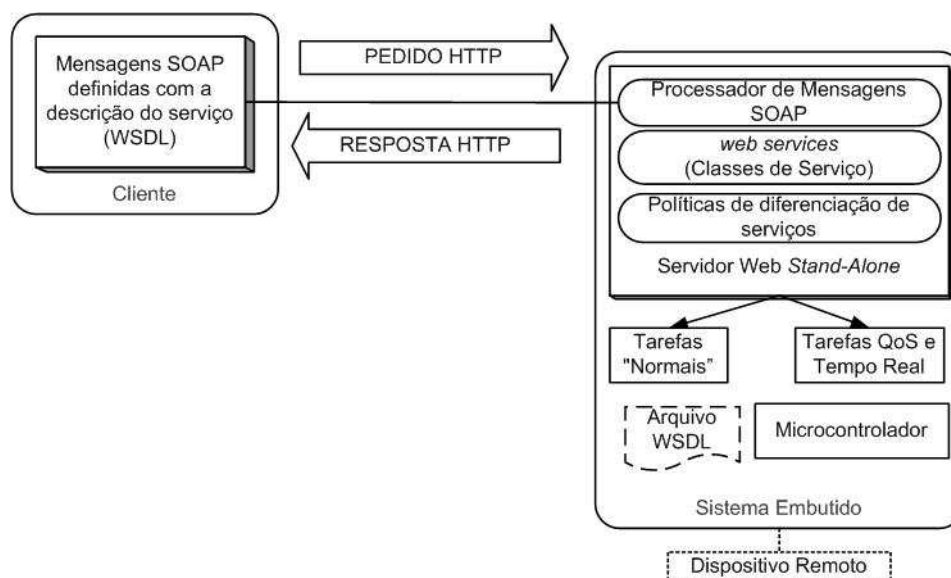


Figura 1. Sistema proposto.

No *web service* há um processamento (categorização, seguindo a metodologia proposta por Serra et al. [9], que consiste na distribuição eficiente dos serviços, ou seja, prioridades de escalonamento, baseado na diferenciação - *gold*, *silver*, *bronze* e *best effort* - destes) dessas mensagens que serão então processadas pelo microcontrolador. Feita a

execução destas mensagens, que pode ser uma resposta do próprio microcontrolador ou do dispositivo remoto ligado a ele, estas são retornadas ao cliente.

5. Cenário de Utilização da Proposta

Um exemplo de uso da nossa proposta é a utilização de sistemas embutidos no monitoramento e controle de pacientes, de acordo com a figura 2. Uma vez que os sinais vitais destes estejam sendo constantemente monitorados/controlados por equipamentos, que na nossa abordagem seriam os dispositivos remotos ligados às plataformas, estas fariam a aquisição dos sinais vitais (seguindo parâmetros pré-programados, isto é, aqueles que possuírem requisitos de tempo-real têm maior prioridade), e disponibilizariam via XML/SOAP as informações para o servidor de aplicação.

O papel do servidor de aplicação seria o de distribuir as diversas informações provenientes de n configurações como essa dentro do ambiente hospitalar para os respectivos clientes (a base de dados do hospital e/ou médicos em casa ou no próprio hospital, utilizando PCs, Palms, etc).

No nosso exemplo podemos ver claramente a diferenciação dos serviços: *gold* para o controle em tempo-real, *silver* para os alarmes enviados à equipe de plantão, *bronze* para a atualização da base de dados e *best effort* para os demais serviços.

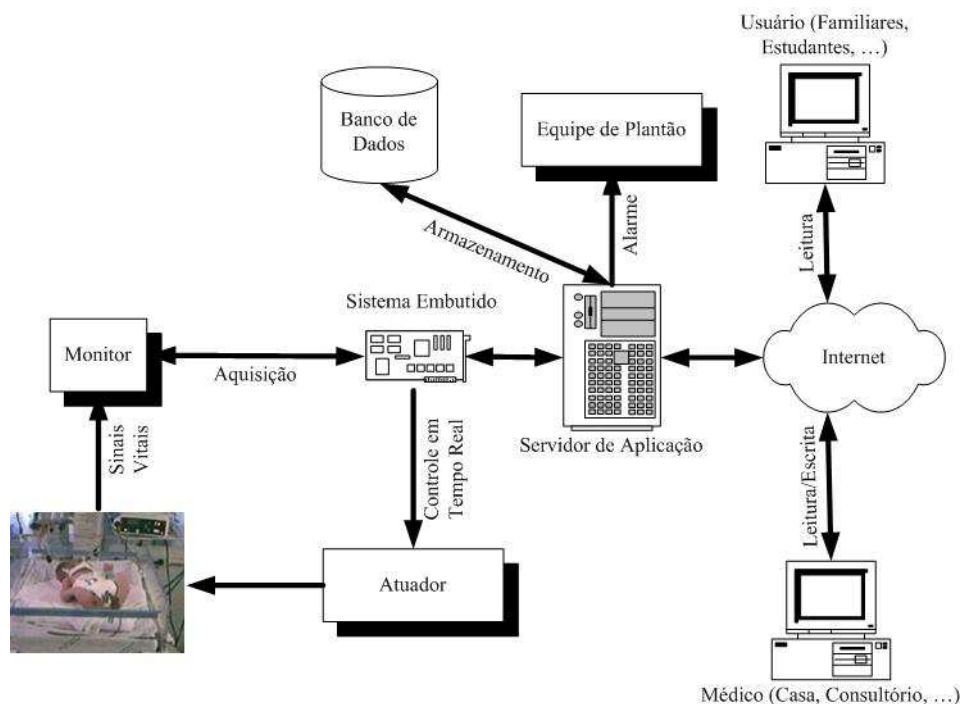


Figura 2. Cenário de Utilização

O grande mérito dessa proposta é que ao utilizar WS não precisamos nos importar com o resto do ambiente distribuído, isto é, quais são os sistemas legados utilizados no servidor de aplicação, como foi feita a modelagem do banco de dados, que tipo de software cliente os usuários estão utilizando, etc.

Todas essas vantagens partem do princípio de que WS utilizam XML sobre uma arquitetura aberta com padrões de *facto*, logo, a utilização dessa estrutura pode ser apli-

cada nos mais diversos cenários, tais como: monitoramento, Controle e Difusão de dados em linhas de transmissão (elétricas, de televisão, etc) e seus diversos sistemas/aplicações envolvidos e ambientes de chão de fábrica com seus diversos sistemas embutidos com interfaces de comunicação específicas.

6. Implementação

O primeiro passo para a verificação das idéias descritas no trabalho consistiu na seleção da plataforma alvo para implantação de WS em sistemas embutidos. Através de um convênio firmado com a empresa Boreste, obtivemos a plataforma SHIP (versão de desenvolvimento) [10], que possui as seguintes características:

- Microcontrolador ARM7TDMI, família AT91X40 com processador RISC;
- Memória flash de 512K Bytes, sendo 448K Bytes livres para as aplicações;
- Sistema Operacional desenvolvido pela Boreste estruturado para dar suporte a conexões *Ethernet* em comunicações TCP/IP. Tem como principais características o μ Boot para carga do sistema e um programa monitor - μ Monitor - para configuração e carga das aplicações na plataforma;
- Diversos dispositivos externos podem ser ligados à plataforma desde equipamentos de automação da manufatura, passando por sensores e atuadores, equipamentos médicos, etc.

Em seguida, foram detectadas as alterações necessárias na plataforma para possibilitar a disponibilização de WS. Foi verificada a necessidade de desenvolver aplicações com suporte a chamadas HTTP, ou seja o próprio *web service* é capaz de prover conteúdo *Web*, além de efetuar os seus serviços.

Em um segundo momento, mostrou-se necessário portar um *toolkit* de desenvolvimento para a plataforma alvo. Optamos pelo *toolkit* gSOAP [11], que se mostrou o mais compatível em relação ao suporte de *hardware/software* da plataforma. Este *toolkit* possui as seguintes características:

- Faz a composição (*binding*) de mensagens SOAP com C/C++ criando *Stubs* e *Skeletons*;
- Cria clientes C/C++ a partir da descrição WSDL;
- É independente de plataforma (possui exemplos de aplicações desenvolvidas em Windows, Linux, Unix, Mac OS X, Pocket PC, Palm OS, Symbian e *embedded Linux*);
- Aplicações podem ser criadas com menos de 100K Bytes, totalizando um consumo total de memória de 150K Bytes.

O desenvolvimento das aplicações para a plataforma SHIP é realizado no ambiente Cygwin [12] configurado com o *cross compiler* ARM GNU *toolchain* [13], com o kit de desenvolvimento (bibliotecas e exemplos) da plataforma e com o *toolkit* gSOAP (versão independente de plataforma).

No presente estágio do trabalho foram construídas aplicações cliente-servidor utilizando a plataforma para executar o serviço e PCs como clientes, demonstrando assim a possibilidade da execução de *web services* em sistemas embutidos. Em seguida estaremos nos dedicando ao desenvolvimento de cenários mais complexos, envolvendo um número maior de clientes e servidores e realizando a integração destes com servidores de bancos de dados e de aplicações, como os descritos na seção 5.

7. Conclusão

WS apresentam-se como uma forma de interconexão de aplicações, através da Internet, entre sistemas computacionais. Além disso, por possuírem uma arquitetura eminentemente aberta e padronizada os WS têm um grande potencial de uso para a computação distribuída.

Portanto, nada mais natural em se propor a construção de um *web service* em um sistema embutido podendo assim promover a integração deste em um ambiente distribuído. A partir desse estudo, esperamos possibilitar uma nova área de aplicação dos WS como meio de integração de Sistemas Embutidos a outros sistemas. Além disso, este trabalho se propõe a adicionar suporte a QoS em WS, visto que requisitos de QoS e sua política de utilização ainda não estão bem consolidados nesta tecnologia.

Referências

- [1] BONIATI, Bruno B.; PADOIN, Edson Luiz. Web services como middlewares para interoperabilidade em sistemas. *Revista do CCEI - Centro de Ciências da Economia e Informática*, v. 7, n. 12, p. 17 – 24, Agosto 2003.
- [2] VOGELS, Werner. Web services are not distributed objects. *IEEE Internet Computing*, v. 7, n. 6, p. 59 – 66, Nov. - Dec. 2003.
- [3] CHAPPELL, David; JEWELL, Tyler. *Java Web Services*. First. [S.l.]: O'Reilly, 2002.
- [4] KILGORE, Richard A. Simulation web services with .net technologies. *Proceedings of the Winter Simulation Conference*, v. 1, p. 841 – 846, 8 - 11 Dec. 2002.
- [5] W3C. *World Wide Web Consortium*. 2005. Disponível em: <<http://www.w3.org/>>. Acesso em: 29/04/2005.
- [6] ROY, Jaideep; RAMANUJAN, Anupama. Understanding web services. *IEEE Internet Computing*, v. 3, n. 6, p. 69 – 73, Nov. - Dec. 2001.
- [7] UDDI. *Universal Description, Discovery and Integration*. 2004. Disponível em: <<http://www.uddi.org/>>. Acesso em: 13/08/2004.
- [8] WOLF, Wayne. *Computer as Components: principles of embedded computing system desing*. [S.l.]: Morgan Kaufmann, 2001.
- [9] SERRA, Antônio; GAÏTI, Dominique; BARROSO, Giovanni; RAMOS, Ronaldo; BOUDY, Jérôme. Uma plataforma distribuída com balanceamento de cargas para servidores web baseada na diferenciação de serviços. *XXXI SEMISH - Seminário Integrado de Software e Hardware*, Agosto 2004.
- [10] BORESTE. *SHIP - Software e Hardware Integrados em um Plataforma*. 2005. Disponível em: <<http://www.boreste.com>>. Acesso em: 28/02/2005.
- [11] GENIVIA INC. *gSOAP - C/C++ Web Services and Clients*. 2005. Disponível em: <<http://www.genivia.com/>>. Acesso em: 28/02/2005.
- [12] CYGWIN. *Cygwin Information and Installation*. 2005. Disponível em: <<http://www.cygwin.com>>. Acesso em: 22/03/2005.
- [13] MACRAIGOR SYSTEMS. *Macraigor Systems GNU Tools*. 2005. Disponível em: <http://www.macraigor.com/full_gnu.htm>. Acesso em: 22/03/2005.