

WEB SERVICES PARA AUXÍLIO NA INTEGRAÇÃO DE BANCOS DE DADOS HETEROGÊNEOS

Diego Mancilha¹, Daniela Leal Musa^{1,2}

¹UNILASALLE, Centro Universitário La Salle
Av. Victor Barreto, 2288 - Canoas

²Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

diegomancilha@yahoo.com.br, musa@inf.ufrgs.br

***Resumo.** Este trabalho apresenta um modelo para integração de banco de dados heterogêneos via Web Services, que originou na implementação da ferramenta MAIBHE (Maibhe Assistente de Integração para Banco de dados Heterogêneos). A ferramenta foi totalmente desenvolvida em software livre e gera o código em PHP do Web service que realizará o acesso ao banco de dados. O artigo também apresenta o estudo de caso que foi realizado que visava testar as funcionalidades da ferramenta.*

1. Introdução

Nos dias de hoje é evidente a necessidade da utilização de Sistemas de Gerenciamento de Bancos Dados(SGBD), devido a grande quantidade de dados que necessita ser manipulada diariamente, principalmente por grandes empresas e instituições.

Com a crescente evolução e popularização dos sistemas computacionais, cada vez se torna mais fácil o acesso a esse tipo de sistema, de maneira que esses podem ser utilizados por qualquer pessoa que tenha interesse. O surgimento do Software Livre é um dos fatores que teve bastante influência nisso, pois permitiu o surgimento de diversas alternativas à sistemas, que antes eram restritos a grandes corporações, devido ao seu alto custo.

A existência de sistemas legados, ou mesmo a utilização de sistemas distintos e isolados por departamentos ou setores em uma mesma empresa, pode acarretar em uma dificuldade de se obter uma informação que necessite dados de ambos os sistemas. Uma das formas de facilitar o acesso a essas informações, sem ter que reestruturar os bancos de dados, ou unificar os SGBDs, segundo [HAN 2003], é integrar os sistemas existentes.

Segundo [GAR 2001](p. 636), “qualquer que seja a arquitetura de integração que escolhermos, há problemas sutis que surgem quando tentamos associar significado aos dados brutos nas várias origens”.

No entanto, a integração de dados pode permitir que o acesso a informação seja feito de maneira ágil e confiável, ao diminuir as chances de erro ao longo do processo de construção da informação desejada.

Dessa maneira é desejável que os problemas citados por [GAR 2001] sejam vencidos ou pelo menos amenizados, pois a integração pode evitar um trabalho ainda maior que seria a reestruturação e unificação dos dados em um único sistema.

2. Objetivo

Este trabalho visa apresentar uma maneira de amenizar estes desafios através da definição de um modelo e desenvolvimento de uma ferramenta que possa auxiliar o processo de integração de bancos de dados. A técnica escolhida foi o uso de *Web Services* [W3C 2003], que pode ser descrita como uma tecnologia que permite a interoperabilidade entre sistemas. A comunicação é realizada de maneira padronizada entre os componentes de software dos sistemas através da Internet.

Utiliza-se para troca de dados entre tais componentes protocolos bem conhecidos e padronizados, tais como HTTP, SMTP e FTP. A comunicação entre um ponto da rede e o ponto onde está o *Web Service* é realizada através de SOAP (*Simple Object Access Protocol*) [W3C 2000]. SOAP é um protocolo que estabelece um caminho de comunicação entre aplicações em diferentes domínios e é baseado em XML, que é um formato padrão para troca de dados.

3. Modelo Proposto

O propósito é utilizar a arquitetura de *Web Services* para realizar a mediação entre bancos de dados e aplicação. Este modelo leva em consideração a proposta de [WIE 1991], que separa o modelo em 3 camadas distintas, sendo elas: aplicação, mediação e camada de dados.

A ferramenta implementada comunica-se com o SGBD e extrai a informação sobre a sua estrutura, armazenando-a em um banco de dados próprio. Essa informação é apresentada ao usuário administrador, para que este faça a seleção dos atributos e construa as regras de associação entre atributos, a serem aplicadas para cada SGBD.

Após coletar as informações necessárias, a ferramenta gera um primeiro código da especificação do *Web Service*, referente aos SGBDs relacionados.

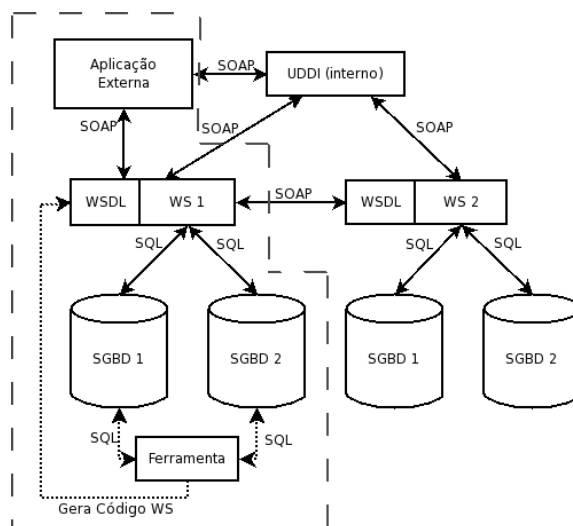


Figura 1. Modelo proposto

A figura 1 apresenta o instante após a integração, tendo cada *web service*, representados por WS 1 e WS 2, acessando os SGBDs relacionados. A parte tracejada na figura 1 representa o que pôde ser implementado neste trabalho. Cada um dos *web services* será encarregado de buscar os dados solicitados pela aplicação através de consultas SQL diretamente ao banco da dados. Um *web service* pode também consultar

outro para obter os dados.

As ligações representadas através de linhas pontilhadas indicam que a comunicação não é obrigatória. Essas ligações só se fazem necessárias no momento em que o *web service* está sendo especificado, ou é preciso realizar alguma alteração em algum *web service* já criado. Um exemplo, seria a adição de um novo banco de dados ao resultado da consulta.

Outra possibilidade vista no modelo apresentado é a geração de um UDDI interno com objetivo de oferecer para uma aplicação externa, a nível de usuário final, a lista de *web services* disponíveis, de modo que este possa então requisitar o serviço desejado. Essas funcionalidades não estão presentes no protótipo, no entanto é algo que pode ser desenvolvido em trabalhos futuros.

As requisições realizadas pela aplicação são feitas através do protocolo SOAP, assim como as requisições entre *web services*. A partir do momento em que a integração tiver sido realizada, não se faz mais necessário a comunicação com a ferramenta, somente se for necessário adicionar um outro banco de dados ao modelo.

A adição de um novo participante a estrutura é mais fácil, devido ao fato de que a ferramenta já possui o registro dos bancos participantes e suas regras, necessitando apenas a adição do novo banco de dados, suas regras, e de sua relação com as demais.

A interface da ferramenta foi desenvolvida para utilização em ambiente Web, visando facilitar a integração, pois torna o acesso independente de plataforma de sistema operacional, satisfazendo assim um dos objetivos citados por [DATE 2000].

A ferramenta foi projetada de forma que permita a sua evolução, aceitando outros SGBDs, sem que se tenha a necessidade remodelá-lo. A utilização dos conceitos de orientação a objetos permitem essa flexibilidade. A linguagem PHP 5, foi utilizada na implementação pois possibilita um uso mais adequado dos conceitos de orientação a objetos em comparação a sua versão anterior.

A respeito da escolha de PHP, outros fatores também podem ser citados, como por exemplo, o suporte à conexão a diversos SGBDs, além de permitir também conexões através de *Open Data Base Connectivity* (ODBC), bem como a existência de bibliotecas para implementação de *Web Services* [ZAV 2004].

Para executar o PHP, foi escolhido, como servidor HTTP, o Apache, sendo este executado em ambiente Linux. A versão utilizada do PHP foi a 5.0.3 e do Apache a 2.0.52.

4. Estudo de Caso

Nesse estudo utilizou-se dois bancos de dados, sendo que ambos possuem apenas uma tabela, contendo cadastros de usuários. Os bancos de dados utilizados pertencem aos projetos AdaptWeb e Claroline, que estão sendo utilizados pelo Instituto de Informática da Universidade Federal do Rio Grande do Sul (UFRGS).

O AdaptWeb é um Ambiente de Ensino-Aprendizagem Adaptativo na Web [ADA 2005]. Foi desenvolvido por um consórcio de pesquisa envolvendo a UFRGS e a Universidade Estadual de Londrina (UEL). O Claroline é um ambiente colaborativo que permite a professores e instituições de ensino criar e administrar cursos através da web [CLA 2005], e é desenvolvido pela Université Catholique de Louvain. O quadro 1 apresenta mais características sobre os bancos de dados utilizados no estudo.

	adaptweb	claroline
SGBD	MySQL 4.1.7	PostgreSQL 8.0
Nome da Tabela	usuario	usuarios
Lista de Campos	id_usuario, nome, senha_usuario, email, tipo_usuario, instituicao_ensino, observacao, idioma, tema, status_usuario	nome, endereco, email, telefone
Quantidade de Registros	92	2693

Quadro 1. Informações dos bancos utilizados.

Para seu gerenciamento, ambos os projetos utilizam o SGBD MySQL, no entanto para realização dos testes do protótipo implementado neste trabalho, havia necessidade de que um dos gerenciadores fosse o PostgreSQL. O banco claroline foi convertido e importado para que pudesse ser gerenciado por esse. Essa necessidade deve-se ao fato de que o trabalho visa integração de bancos de dados heterogêneos e o gerenciador é algo que pode diferir.

O teste realizado integrou os registros dos dois bancos de dados, sendo que para isso foi criado um *web service* denominado “integraUsuarios”. Através do protótipo, na tela de configurações do *web service* (figura 2) foi especificado que esse possui os atributos “nome”, “email”, “local” e “telefone”.

Figura 2. Tela de criação do *web service*.

Os atributos especificados, servem como um alias para as colunas de cada banco, e são utilizados na criação da regra que associa e indica, que por exemplo, a coluna *instituicao_ensino* do banco de dados *adaptweb* contém valores que podem ser considerados equivalentes aos valores da coluna *endereco* do banco de dados *claroline*.

Além de servir para essa associação, o alias informado é utilizado também na criação da consulta SQL que busca as informações do banco. O código é gerado em PHP, utilizado-se da biblioteca NuSOAP para tratamento dos *web services* que já possui suporte ao WSDL.

Como aplicação cliente, foi implementado um programa, que também utiliza a biblioteca NuSOAP. No entanto, deve ficar claro que no caso do cliente, este pode ser implementado em qualquer linguagem que dê suporte a *web services*, no caso do

exemplo, este foi escrito em PHP portanto optou-se pela utilização dessa classe. O cliente implementado faz a chamada ao *web service* e recebe os dados, já integrados, exibindo-os na tela. O total de registros retornados foi exatamente a soma dos registros de cada banco de dados participante ou seja 2785 registros. Desses 2785, os primeiros 92 são referentes ao banco adaptweb e os 2693 restantes pertencem ao claroline.

5. Resultados e Trabalhos Futuros

A partir desse estudo pode-se identificar dois aspectos que podem ser tratados em trabalhos futuros, sendo eles: a questão da ordenação dos registros e a identificação e resolução de como serão tratados registros duplicados.

No caso dos bancos estudados havia repetição de registros onde todas as colunas continham o mesmo conteúdo e outros em que, por exemplo o nome da pessoa foi informado de uma forma diferente, sendo possível identificar que ambos tem relação.

Além disso há a questão das regras para formação do *web service*, que nesse trabalho, levou em consideração associação simples entre atributos, essa questão pode evoluir permitindo por exemplo a consulta a outros *web services* e seleções que envolvam mais de uma tabela para uma mesma conexão. No protótipo implementado é possível envolver uma ou mais conexões no resultado, no entanto, cada uma faz seleção em apenas uma tabela.

Como sugestão, referente ao retorno dos dados em formato XML, poderia ser realizada a compactação dos dados a serem transmitidos. Para isso seria necessário que o cliente fosse capaz de descompactá-lo. Como teste foi realizada a compactação do arquivo retornado como resultado, tendo esse 406 kb. Foi utilizado o compactador zip disponível no sistema Linux utilizado. Nesse caso, o arquivo de 406 kb ficou em 55 kb. No entanto seria necessária a realização de testes mais complexos para se afirmar que é uma solução viável.

6. Considerações finais

A evidência da necessidade de integrar informações é bastante clara, até mesmo pela quantidade de técnicas e propostas existentes relacionadas ao problema. A utilização de *web services*, permite que a integração seja feita de forma facilitada garantindo a interoperabilidade entre os sistemas, podendo em muitas vezes utilizar-se de estruturas já existentes.

Em muitos casos, empresas que tem necessidade de integrar, já tem algum servidor HTTP em funcionamento com a linguagem necessária, podendo-se aproveitar os recursos já existentes. Dependendo das necessidades, da demanda, é possível ampliar as capacidades do servidor ou até mesmo distribuir os serviços. Essa possibilidade de distribuição física dos serviços, também é uma vantagem que a utilização de *Web Services* propicia.

No caso da empresa não ter o servidor implementado, o custo com a implementação deste, poderá ser resumido apenas ao custo com hardware, treinamento, e o tempo necessário para sua implantação, pois podem ser utilizados programas que além serem de licença gratuita, são programas que tem o seu código fonte aberto. Como exemplo podem ser citados o Apache, como servidor HTTP e o PHP como linguagem de programação.

A conclusão que chega-se a partir do estudo de caso realizado, é que mesmo

existam problemas a serem resolvidos, sua utilização é válida, pois auxilia no desenvolvimento dos *web services*, permitindo que o desenvolvedor de ênfase as questões mais relevantes como a integração em si, tornando transparente a forma de como o *web service* é implementado.

Em relação ao protótipo desenvolvido neste trabalho, antes de torná-lo uma ferramenta de código aberto, é necessário uma maior documentação no código escrito e a possibilidade da seleção do idioma da ferramenta. Também seria desejável, um programa que facilite sua instalação em um servidor em ambiente Windows, pois para sistemas linux foi desenvolvido um programa em *shell script(bash)*, que realiza as alterações nos arquivos necessários.

O objetivo de torná-la pública, além de que possa ser utilizada como auxiliar no processo de integração, é contribuir para que, em cima do que já foi construído, mais estudos possam ser feitos para que novas melhorias e facilidades sejam implementadas e disponibilizadas.

Referências

- [ADA 2005] Adaptweb - Ambiente de Ensino-Aprendizagem Adaptativo na Web. 2005 Disponível em: <<http://www.inf.ufrgs.br/adapt/adaptweb/>> Acesso em: 16 mai 2005
- [CLA 2005] Claroline : Open Source e-Learning. 2005 Disponível em: <<http://www.claroline.net>> Acesso em: 16 mai 2005
- [DATE 2000] Date, C. J. Introdução a Sistemas de Bancos de Dados - Tradução de Introduction to Database Systems, Rio de Janeiro: Campus, 2000
- [FER 2003] Ferris, Christopher; Farrell, Joel. What are Web Services? ; Communications of the ACM, Vol 46 n. 6, Junho 2003, p. 31
- [GAR 2001] Garcia-Molina, Hector; Ullman, Jeffrey D. ; Widom, Jennifer. Implementação de Sistemas de Banco de Dados – Tradução de Database System Implementation, Rio de Janeiro; Campus, 2001
- [HAN 2003] Hansen, Mark ; Madnick, Stuart ; Siegel, Michael. Data Integration using Web Services. Working papers4406-02, Massachusetts Institute of Technology (MIT), Sloan School of Management, 2003 Disponível em: <<http://ideas.repec.org/p/mit/sloanp/4406-02.html>> Acesso em: 24 ago 2004
- [W3C 2000] World Wide Web Consortium. Simple Object Access Protocol (SOAP). Maio 2000. Disponível em: < <http://www.w3.org/TR/SOAP/> >. Acesso em: 15 out 2004
- [W3C 2001] World Wide Web Consortium. Web Services Description Language (WSDL) Março 2001. Disponível em: < <http://www.w3.org/TR/wsdl>>. Acesso em: 15 out 2004
- [W3C 2003] World Wide Web Consortium. Web Services Architecture Março 2003. Disponível em: <http://www.w3.org/TR/ws-arch/>> Acesso em: 15 out 2004
- [WIE 1991] Wiederhold, Gio: "Mediators in the Architecture of Future Information Systems"; IEEE Computer, Março 1992, p. 38-49.
- [ZAV 2004] Zavlik, Claudimir; Lacerda, Guilherme; Oliveira, José Palazzo M. de. Implementando Web Services com Software Livre. In: 5o Fórum Internacional do Software Livre, Anais. Porto Alegre: PUC-RS 2004. p. 35-38