

Desenvolvimento de Aplicações WEB

Conforme já é sabido, os serviços WEB utilizam a infra-estrutura de redes existente para disponibilização de informações e aplicações para o usuário. Para o cliente e o servidor WEB a topologia da rede é irrelevante pois isto é tratado pelo protocolo TCP/IP.

No nível de serviços, os elementos são organizados da seguinte forma: de um lado está o cliente WEB, ou browser, que solicita dados ao servidor WEB, recebe as respostas, formata a informação e a apresenta ao usuário. Do outro lado, está o servidor WEB que recebe as requisições, lê os dados (páginas HTML) do disco e as retorna para o cliente.

Esta é a forma original de funcionamento da WEB e que proporciona apenas páginas de conteúdo estático, ou seja, uma vez programadas, apresentam sempre a mesma informação.

A forma encontrada para modificar esta situação e permitir a criação de páginas dinâmicas foi a seguinte: o usuário entra com informações através do browser utilizando formulários HTML. O browser repassa as informações ao servidor WEB que executa um programa transferindo-lhe as informações vindas do cliente. O programa remoto (serve-side gateway program) trata as informações e retorna uma página HTML criada dinamicamente. Esta página é passada ao servidor que a entrega ao cliente. O padrão para comunicação entre o servidor WEB e o "Server-side gateway program" é conhecido como CGI (Common Gateway Interface).

A Interface CGI explicita como os dados devem ser passados do servidor WEB para o programa e como o este deve devolver os dados ao servidor WEB. Além disto, especifica variáveis de ambiente que devem ser criadas pelo servidor e que podem ser acessadas pelo programa. Estas variáveis contêm o endereço IP do usuário remoto, o tipo de browser que ele utiliza, dados para autenticação do usuário, dentre outros.

Os programas escritos para ser executados por servidores WEB são também conhecidos como scripts CGI. Os scripts CGI implementam a lógica do negócio e, muitas vezes, o acesso aos dados de uma aplicação WEB, pois podem acessar dados armazenados no servidor ou fazer chamadas a um servidor de banco de dados local ou remoto, permitindo o acesso às informações da empresa.

Nesta arquitetura, tem-se programas sendo executados no servidor WEB e em outras máquinas da rede, como o servidor de banco de dados. Só falta agora serem colocados programas para executar no cliente. A execução de programas no cliente torna o browser mais independente do servidor. Sem isto, qualquer operação realizada pelo cliente tem que ser validada e executada pelo servidor, pois o browser não possui nenhuma inteligência.

Os programas que executam no browser, normalmente, são scripts (VBScript ou JavaScript) que tem a capacidade de perceber os eventos causados pelo usuário e responder de forma apropriada. Eles são embutidos no código HTML e o seu código fonte pode ser visualizado pelo usuário, pois não é um código compilado. Os scripts interagem muito com todos os elementos que formam uma página HTML.

Outro tipo de programa que pode ser executado no cliente são os programas escritos na linguagem JAVA, que será descrita em maiores detalhes mais a frente.

Em JAVA, é possível escrever aplicações completas, inclusive acessando bancos de dados relacionais independentemente do servidor WEB, bem como é possível implementar os níveis de interface com o usuário e de lógica do negócio, utilizando um servidor de banco de dados para implementar o nível de acesso aos dados.

A Plataforma Ativa da Microsoft

A Microsoft, como parte da sua estratégia para Internet/Intranet, está lançando no mercado a idéia da Plataforma Ativa Microsoft. Segundo a própria fabricante, pelo menos na visão do seu departamento de Marketing, trata-se de uma arquitetura aberta de software, para disponibilizar conteúdo rico e aplicações através da Internet e Intranets.

Tecnicamente, a plataforma ativa combina uma série de tecnologias para criar sistemas de software distribuídos baseados em componentes, ou objetos. Cada componente é um pedaço de software que possui informações e implementa serviços necessários a uma ou mais aplicações, podendo residir em

uma máquina da rede e ser acessado por outros de forma independente da sua localização. Os componentes implementam serviços dos vários níveis de uma arquitetura multitier: apresentação, lógica do negócio e acesso aos dados.

O objetivo maior desta tecnologia é permitir, com o passar do tempo, o desenvolvimento de aplicações em escalas de tempo bastante reduzidas. Todo o desenvolvimento será calcado na idéia da reutilização de componentes já existentes para a criação de novas aplicações. Antes de implementar algo novo, os projetistas de software devem procurar verificar se já não existem componentes prontos para atender aquela situação específica. Estes componentes podem estar disponíveis dentro da própria organização ou podem ser adquiridos de terceiros.

Os componentes podem estar presentes nas interfaces com o usuário (browser), implementando a lógica do negócio junto ao servidor WEB ou distribuídos pela rede e realizando atividades de acesso aos dados, utilizando bases de dados em qualquer formato.

Este conjunto de tecnologias que vão permitir o desenvolvimento de aplicações distribuídas baseadas no conceito da WEB é normalmente chamado de ActiveX. Os componentes que aderem a estas especificações são conhecidos como componentes ActiveX.

O padrão para construção de componentes ActiveX é especificado pelo Component Object Model (COM). O COM especifica uma série de serviços (interfaces) que devem ser implementadas pelos componentes para que eles possam se integrar com facilidade em qualquer tipo de aplicação. Um objeto COM pode, por exemplo, ser incorporado dinamicamente a uma aplicação como o Microsoft Word ou Internet Explorer, que são conhecidos como "containers" de objetos.

A Plataforma Ativa Microsoft Utiliza Recursos já Existentes na WEB

A arquitetura proposta é a seguinte: de um lado está o chamado cliente ativo, que é um browser, com a capacidade de apresentar páginas HTML, executar programas em JAVA, conter componentes ActiveX e executar scripts (VBScript ou JavaScript) que controlam todos estes elementos. As linguagens de script tem um papel extremamente importante, pois elas formam a cola que une os componentes (ActiveX e JAVA) para criar uma aplicação para o usuário. São elas que ativam os componentes ActiveX/JAVA solicitando-lhes serviços de acordo com as escolhas dos usuários.

Do outro lado, está o servidor ativo, que é um servidor WEB com a capacidade de ler e entregar páginas HTML estáticas e executar scripts que utilizam componentes ActiveX para realizar atividades.

Deve ser observada a simetria entre o cliente ativo e o servidor ativo. Ambos tem a capacidade de lidar com os mesmos elementos, sendo que o servidor não executa nenhuma atividade de apresentação de interface para o usuário. Os componentes que são utilizados pelo servidor implementam a lógica do negócio e o acesso aos dados. Já no cliente ativo, executam os componentes que implementam a interface da aplicação.

Os componentes ActiveX podem residir no mesmo servidor ou podem ser distribuídos pela rede. O padrão para criação de objetos distribuídos é conhecido como DCOM (Distributed Component Object Model).

Espera-se que esta tecnologia permita aos desenvolvedores de software criar aplicações distribuídas, utilizando um browser, que é independente de arquitetura, e reutilizando componentes pré-programados.

Linguagens de Script

As linguagens de script (script languages) são assim chamadas por serem linguagens de programação simples sem muitas das restrições associadas às linguagens mais complexas. Nas linguagens de script, o tratamento dos tipos das variáveis é bastante simplificado para o programador, pois estas não possuem tipos fixos e assumem aqueles dos valores que lhes são atribuídos. Elas são interpretadas em tempo de execução ao invés de compiladas para posterior execução.

Muitas vezes, as linguagens de script são subconjuntos mais simples de linguagens existentes, como é o caso do VBScript e do JavaScript. Ambas tem o mesmo potencial de utilização na WEB apesar de usarem sintaxes diferentes.

Programas feitos em VBScript/JAVAScript podem ser inseridos em qualquer página HTML através do comando

Tratam-se de objetos programados de acordo com o padrão COM (Component Object Model) já mencionados. Este padrão define uma serie de interfaces (serviços) que os objetos devem implementar para que possam ser reutilizados em aplicações diversas. O padrão COM utiliza o formato de objetos OLE para implementar as suas interfaces.

Os objetos ActiveX expõem para o mundo externo um conjunto de atributos e operações (serviços) que ele pode realizar. Pelo fato de seguirem o padrão COM eles podem ser executados em qualquer aplicação que suporte este padrão. Hoje em dia, é possível utilizar objetos COM no Internet Explorer, Servidor WEB IIS ou no desktop do próprio WINDOWS.

Nem todas as interfaces especificadas no padrão COM são implementadas nos componentes ActiveX. Assim, estes tornam-se menores e mais rápidos, de forma que a sua transferência na rede e execução são otimizadas.

Os componentes ActiveX podem ser embutidos em páginas HTML e seus atributos e serviços podem ser acessados através das linguagens de script, como o VBScript. Os componentes ActiveX também podem ser utilizados no servidor WEB através das linguagens de script. A diferença maior entre os componentes ActiveX que rodam no browser dos que rodam no servidor é que os primeiros podem possuir interface com o usuário, enquanto que os últimos não devem possuir este recurso.

Cada objeto ActiveX, inserido em uma página HTML, recebe uma identificação e é acessado pelo VBScript através desta identificação, assim como qualquer campo de um formulário. O tratamento dos eventos que ocorrem nos componentes ActiveX também pode ser feito em VBScript na própria página HTML. O tratamento destes eventos é similar ao tratamento de eventos no Visual Basic: basta definirmos funções que tenham o mesmo nome do componente seguido do evento a ser tratado.

O objeto acima é identificado pelo nome CommandButton3. A função abaixo trata o evento do clique do mouse sobre o botão.

Quando um componente ActiveX é encontrado em uma página HTML, o browser verifica se aquele componente já encontra-se instalado na máquina do cliente e se a versão existente é compatível com a versão especificada na página. Se estas condições são satisfeitas, o browser utiliza o componente já instalado, sem a necessidade de transferências pela rede.

Caso contrário, o browser inicia a transferência do objeto. Os componentes ActiveX podem ser disponibilizados em arquivos DLL (Bibliotecas de Ligação Dinâmica) ou arquivos OCX (ActiveX Component Object). Este arquivo é instalado na máquina do usuário, e o objeto é instanciado e colocado na apresentação da página HTML.

Os componentes ActiveX podem realizar qualquer operação na máquina do usuário. Por isto, algumas medidas de segurança devem ser tomadas para evitar possíveis desastres. Os componentes ActiveX possuem uma assinatura que identifica a sua procedência. O usuário tem a capacidade de aceitar ou rejeitar componentes ActiveX de procedência duvidosa. Este controle é feito através de opções de configuração do browser.

Os componentes ActiveX podem ser implementados em qualquer linguagem de programação, seja JAVA, C++ ou Visual Basic, desde que atendam aos padrões especificados e implementem as interfaces necessárias.

Os componentes ActiveX podem ser reutilizados em inúmeras aplicações, trazendo vantagens não só no desenvolvimento de software, como também na sua manutenção. O desenvolvimento de aplicações tende a tornar-se uma atividade de reutilização do que de criação de novos componentes.

A manutenção destes sistemas pode ser feita de forma localizada através da alteração de seus componentes. A alteração de um componente que é utilizado em varias aplicações faz com que todas as aplicações sejam automaticamente atualizadas. O browser garante que serão utilizadas sempre as ultimas versões de cada componente.

Os componentes ActiveX podem ser distribuídos pela rede e utilizados como se fossem objetos locais. O padrão DCOM (Distributed Component Object Model) especifica as interfaces que os objetos devem implementar para poder funcionar de forma distribuída. A localização dos objetos na rede deverá ser transparente para as aplicações que os utilizam.

Cada componente ActiveX pode ser especializado para a realização de atividades referentes a cada um dos níveis da arquitetura cliente servidor multitier. Podemos ter componentes que implementam serviços de interface, lógica de negócios e acesso aos dados.

Os componentes de interface são objetos com representação visual para interação com o usuário. Os objetos de interface serão utilizados no browser.

Os objetos que implementam a lógica do negócio serão utilizados por diversas aplicações, pois é neles que reside a "inteligência" dos programas. Alterações nas regras do negócio da empresa deverão ser feitas nestes objetos e estarão, automaticamente, disponíveis para todas as aplicações. Os objetos de negócio poderão residir no browser, no servidor Web ou poderão estar distribuídos pela rede.

Os objetos que implementam o acesso aos dados deverão esconder os detalhes de formato do armazenamento dos dados para os objetos de negócio, que serão os seus principais usuários. Os dados poderão vir de fontes diversas, tais como bancos de dados relacionais, arquivos texto, arquivos VSAM e outros, de forma transparente. Os objetos de acesso aos dados poderão residir no servidor WEB ou poderão estar distribuídos pela rede.

JAVA é uma linguagem de programação orientada a objetos que permite a criação de aplicações que são independentes de arquitetura. Ou seja, é possível executar o mesmo programa JAVA (código binário) em máquinas com hardware e sistemas operacionais diferentes.

Uma aplicação em JAVA executa em cima de uma máquina virtual, que é chamada de Java Virtual Machine. A máquina virtual JAVA especifica os componentes de um computador, incluindo memória, registradores, pilha, etc. Os programas em JAVA, quando compilados, geram código que seria executado nesta máquina. Este código compilado é conhecido como byte code (Java Byte Codes).

Como não existe nenhum hardware que seja igual a máquina virtual JAVA, esta máquina é implementada em software e embutida nos principais browsers da atualidade. Existem também implementações da máquina virtual JAVA, independente de browsers, para diversas plataformas. Isto faz com que um programa JAVA execute hoje nas principais plataformas existentes.

Para encaixar as applets JAVA (aplicações em JAVA que rodam em um browser) no paradigma ActiveX, elas estão sendo tratadas como componentes ActiveX. As applets podem ter atributos e operações disponíveis para ser utilizadas pelas linguagens de script.

Até aqui, foram discutidos os elementos que estão presentes do lado do cliente WEB, o browser. Foram vistas as linguagens de script, representadas pelo VBScript, os componentes ActiveX e as applets JAVA. A partir da próxima seção, serão vistos os elementos que trabalham do lado do servidor para compor as aplicações WEB.

Quando um formulário HTML é preenchido por um usuário e os dados são enviados ao servidor WEB, algum programa deve ser executado para tratar aquela informação específica. O valor do atributo ACTION do tag indica qual programa deve ser utilizado.

O Common Gateway Interface especifica como os dados vindos do cliente WEB devem ser passados ao programa (script CGI) e como este programa deve retornar o resultado ao servidor WEB, para que este repasse as informações ao cliente. O script CGI pode ser escrito em qualquer linguagem de programação, desde que o código gerado possa ser executado na máquina do servidor.

A solução CGI para interatividade na WEB não é adequada para servidores muito requisitados. Cada execução de um script CGI implica na criação de um novo processo no sistema operacional. Para a criação de um novo processo várias tarefas (alocação de memória, criação de estruturas internas etc.) devem ser realizadas, o que consome bastante tempo da máquina. Além disto, a transferência de dados entre processos, no caso o servidor WEB e o script CGI, também não é muito eficiente.

Para minimizar este problema, a Microsoft, criou um servidor WEB que pode carregar módulos adicionais, sob a forma de bibliotecas de ligação dinâmica (DLL). Estas DLLs uma vez carregadas, ficam na memória do servidor e funcionam como se fossem uma parte do software servidor WEB. As novas chamadas para este módulo serão atendidas com mais rapidez e menos consumo de máquina pois os mesmos já estarão carregados e fazendo parte do processo servidor WEB. A transferência de dados também é otimizada, pois tudo acontece dentro de um mesmo espaço de endereçamento.

A criação destes módulos é feita através da implementação de uma interface padrão de comunicação com o servidor WEB. Este padrão de programação é conhecido como ISAPI.

Os módulos ISAPI podem funcionar como scripts CGI, recebendo dados do cliente, processando-os e retornando novas páginas HTML (ou outros tipos de documentos). Para o cliente WEB tudo continua funcionando como se fossem scripts CGI.

Os módulos ISAPI podem também funcionar como filtros, que tratam os dados que transitam entre o cliente e o servidor WEB. Podem ser notificados do acontecimento de determinados eventos e entrar em ação quando estes eventos acontecem.

Um filtro pode, por exemplo, solicitar ser notificado quando um cliente pedir um determinado tipo de arquivo. O filtro pode, então, tratar o arquivo interpretando scripts ou criptografando um documento antes do envio do mesmo ao cliente.

O ASP (Active Server Pages), que será visto na próxima seção, funciona como um filtro que trata todas as solicitações de páginas com a extensão de arquivo .asp. As páginas com extensão .asp são páginas HTML contendo scripts (VBScript/JAVAScript) que são interpretados e executados pelo filtro antes de serem entregues aos clientes. Além disso, este módulo disponibiliza outros componentes ActiveX para que sejam acessados pelos scripts, tais como objetos de acesso a dados.

Este mesmo esquema pode ser utilizado para a criação de outros filtros que interpretem diferentes linguagens de script como, por exemplo, PERL, que é uma linguagem muito utilizada para a criação de scripts CGI e que tem uma grande interação com o sistema operacional e grande capacidade de operar com strings.

Active Server Pages é o nome dado à tecnologia que permite a execução de scripts embutidos em páginas HTML no servidor WEB. Os scripts podem ser escritos em VBScript ou JAVAScript. Os scripts tem acesso a componentes ActiveX que residem no servidor.

A execução destes scripts fica a cargo de um filtro ISAPI que intercepta todas as requisições de arquivos com extensão .asp. As páginas são lidas pelo servidor e, antes de serem enviadas ao cliente, são tratadas pelo filtro ASP, que interpreta e executa os comandos encontrados na página. O resultado final é entregue ao cliente.

Os arquivos com extensão .asp quando solicitados, podem executar programas (server side scripts) embutidos no documento e utilizar serviços de componentes ActiveX. Estes programas geram as páginas HTML de forma dinâmica, podendo acessar bases de dados, que são repassadas ao browser.

Neste material, foi apresentada toda uma nova tecnologia de desenvolvimento de aplicações que está surgindo no mundo da informática. Os paradigmas estão sendo modificados a uma velocidade muito grande. Já encontram-se disponibilizados elementos que nos permitem criar aplicações realmente distribuídas e orientadas a objetos.

As maiores vantagens da utilização desta tecnologia são a possibilidade de reutilização de código em larga escala, utilizando os objetos, e aproveitamento do poder computacional, que está distribuído em redes de computadores de todo o mundo. Neste ambiente, todo software desenvolvido tem a capacidade de ser executado e utilizado em múltiplas plataformas de hardware e software, com total transparência para o usuário.

Em uma próxima oportunidade, será apresentada a proposta da Sun Microsystems e seus parceiros (IBM, Borland, Apple etc.) para o desenvolvimento de software distribuído baseado em componentes, o JAVA Beans.