

Desenvolvimento de Sistemas e Serviços para a Internet com Interface WWW

José Luis Borbinha^{*}, Alberto Silva^{**}, Pedro Ribeiro^{***}, José Delgado^{****}

INESC - Instituto de Engenharia de Sistemas e Computadores^{*****}

Grupo de Sistemas e Serviços Telemáticos

Rua Alves Redol, 9

Apartado 13069 - 1000 Lisboa

SUMÁRIO

O sucesso da Internet permite afirmar que esta será, provavelmente, a infra-estrutura ideal para as *auto-estradas globais* de informação que na prática venham a ser construídas.

Estão assim pois criadas as condições técnicas, de oportunidade e de motivação para a divulgação e introdução das tecnologias Internet no meio empresarial, onde os benefícios serão o seu menor custo comparativamente a soluções proprietárias e a sua flexibilidade e adaptabilidade natural a condições de heterogeneidade de equipamentos, dados e processos.

Por outro lado o campo da exploração de serviços comerciais “*on-line*” é uma nova realidade, onde apesar de todas as promessas e expectativas continuam a faltar os verdadeiros casos de inovação e sucesso.

É com esta perspectiva e motivação que o Grupo de Sistemas e Serviços Telemáticos do INESC tem vindo ultimamente a adquirir experiência e a desenvolver tecnologia própria para a concepção e concretização de sistemas e serviços para a Internet, com interface WWW. O resultado dessa actividade é o que adiante se apresenta.

1. INTRODUÇÃO

O espectacular crescimento da Internet em todo o mundo deve-se principalmente às características da tecnologia WWW. A facilidade de utilização do seu modelo de navegação hipermédia, associada à simplicidade da linguagem de descrição de documentos

HTML (*HyperText Markup Language*) e ao protocolo de transporte HTTP (*HyperText Transfer Protocol*), a disponibilidade de clientes WWW com interface gráfica para uma grande gama de plataformas e ambientes (Mosaic, Netscape Navigator, etc.), foram aspectos decisivos para o sucesso e conseqüente crescimento da Internet.

A informação acessível por WWW, distribuída por todo o mundo, encontra-se basicamente em ficheiros no formato HTML. Esta informação é, regra geral, criada manualmente, usando-se para tal simples editores de texto, ou eventualmente uma ferramenta especializada na edição de documentos HTML. A estes documentos dá-se o nome de páginas HTML, e pelo facto de serem criadas manualmente dizem-se estáticas. Por oposição, as páginas HTML geradas em tempo real por programas específicos (*gateways*), designam-se por páginas dinâmicas ou virtuais. Estas podem trazer grandes vantagens por permitirem o acesso interactivo e selectivo a repositórios de dados geridos independentemente (como bases de dados tradicionais, por exemplo).

O trabalho descrito neste documento relaciona-se assim com a problemática do desenvolvimento de sistemas e serviços para a Internet com interface WWW, o que compreende a construção de *gateways* para acesso a esses sistemas com a geração dinâmica de páginas HTML.

Apresentam-se de seguida os conceitos genéricos da biblioteca GENESIS (*Generic Library for WWW Services Authoring*), uma biblioteca de classes C++

* Telefone: 310 03 07 - Email: Jose.Borbinha@inesc.pt

** Telefone: 310 03 07 - Email: Alberto.Silva@inesc.pt

*** Telefone: 310 00 00 (Ext. 2568) - Email: rib@crack.inesc.pt

**** Telefone: 310 02 11 - Email: Jose.Delgado@inesc.pt

***** Telefone: 310 00 00 (Geral) - Fax: 52 58 43

desenvolvida para auxiliar a criação de sistemas e serviços do tipo dos referidos.

2. O MODELO COMPUTACIONAL DO WWW

O modelo computacional do WWW baseia-se no modelo cliente-servidor conforme esquematizado na figura 1. É utilizado o protocolo HTTP, um protocolo de transporte sem conexão e executado sobre a pilha TCP/IP. Por sua vez a informação enviada dos servidores para os clientes obedece ao formato HTML

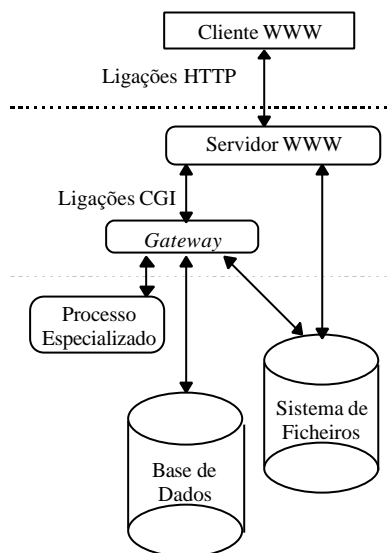


Figura 1: O modelo computacional WWW.

O HTTP é bastante adequado para a pesquisa e navegação de informação hipermédia, mas apresenta alguns inconvenientes quando usado para aplicações interactivas que à partida exigiriam a manutenção de conexões abertas.

Os clientes WWW têm em geral interface gráfica, são multi-plataforma (e.g. X-Windows, MS-Windows, Macintosh, etc.), multi-protocolo (e.g. HTTP, FTP, Gopher, News, File, etc.), e apresentam convenientemente informação no formato HTML. Os clientes mais recentes têm capacidades adicionais, por exemplo: capacidade de invocarem aplicações multimédia externas; visualização de diferentes formatos de imagens incluídos nos próprios documentos (GIF, BMP, JPEG, etc.); disponibilização de um interface de programação básica CCI (“Common Client Interface”); ou mesmo suporte a transações seguras.

Do outro lado do modelo encontram-se os servidores WWW, servidores do protocolo HTTP, em geral *multi-thread* (o que significa que aceitam, de forma concorrente, múltiplos pedidos), com ficheiros de configuração, de históricos de utilização (*logs*) e de protecção/autorização, suportando ainda o protocolo CGI (“Common Gateway Interface”) para aplicações.

O CGI estende a funcionalidade de um servidor HTTP ao permitir ligá-lo a outros processos específicos, designados por *gateways*, e que são lançados em tempo

de execução pelo processo servidor de HTTP. Inicialmente os *gateways* realizavam apenas processamentos simples (e.g. filtros para comandos do sistema operativo como o *ps* ou o *ls* do Unix), mas a introdução da funcionalidade de *forms* na linguagem HTML veio aumentar as possibilidades de utilização de *gateways*, sendo agora possível conceber serviços que requerem uma maior interactividade.

Sistemas de informação e serviços construídos neste modelo podem comunicar com vários repositórios de dados (e.g. sistemas de gestão de bases de dados, sistemas de ficheiros, directórios distribuídas X.500, sistemas de informação geográfica, etc.) ou mesmo com outros processos especializados. Em cada acesso, e de uma forma genérica, o sistema evolui nos seguintes passos:

1. O *gateway* é criado pelo servidor de HTTP;
2. Os dados provenientes do servidor de HTTP são recebidos através do canal CGI (i.e. através de variáveis de ambiente e do *standard input* nos sistemas Unix);
3. Analisam-se os dados recebidos e verifica-se a sua validade;
4. Executa-se processamento dependendo da informação recebida;
5. Produzem-se resultados (geralmente no formato HTML) para o *standard output*, que serão lidos pelo servidor HTTP e redireccionados para o cliente.

3. A BIBLIOTECA GENESIS

A biblioteca GENESIS é uma biblioteca de classes C++ criada para apoiar o desenvolvimento de sistemas de informação e serviços interactivos com interface WWW.

As suas características principais são providenciar ao programador uma estrutura de classes já definidas, suportando as seguintes funcionalidades principais:

- **Uniformização do serviço**, sendo oferecidas facilidades para uniformização, ao nível da estrutura de navegação e da aparência física das páginas;
- **Independência do HTML**, escondendo-se ao programador tanto quanto possível a sintaxe e semântica do HTML, principalmente no que diz respeito à manipulação de funcionalidades mais complexas, como sejam por exemplo *forms* e tabelas;
- **Apoio à interactividade**, através de simulação de conexões cliente-servidor, possibilitando assim a construção de serviços interactivos baseados em máquinas de estados (suportando-se a passagem de parâmetros entre dois acessos);
- **Interface para repositórios “não HTML”**, oferecendo-se funcionalidades para a criação dinâmica de documentos em formato HTML cujos dados pode estar acessíveis a partir de vários repositórios (sistemas de ficheiros, bases de dados, etc.).

Descreve-se nos pontos seguintes a biblioteca GENESIS, sendo privilegiada uma descrição conceptual em detrimento de uma descrição especializada e exaustiva dos detalhes de concretização da mesma.

CONCEITOS GENÉRICOS

Um sistema de informação baseado na biblioteca GENESIS estrutura-se em torno de cinco conceitos:

- *Gestor de Ligação* - classe responsável pela gestão do interface CGI e tratamento das variáveis de estado a manter em acessos sucessivos;
- *Agentes* - um conjunto de classes representativas das funcionalidades oferecidas pelo serviço;
- *Entidades* - conjunto de classes representando os dados a processar (dados armazenados em bases de dados ou outros repositórios);
- *Páginas* - um conjunto de classes onde será criada, por cada acesso, uma instância para retornar ao cliente, permitindo-lhe obter respostas aos seus pedidos e ao mesmo tempo a possibilidade de continuar a interagir com o sistema;
- *Elementos HTML* - conjunto de classes representando os elementos que compõem as páginas, os quais podem ir de desde simples tipos HTML até complexas estruturas compostas dos mesmos.

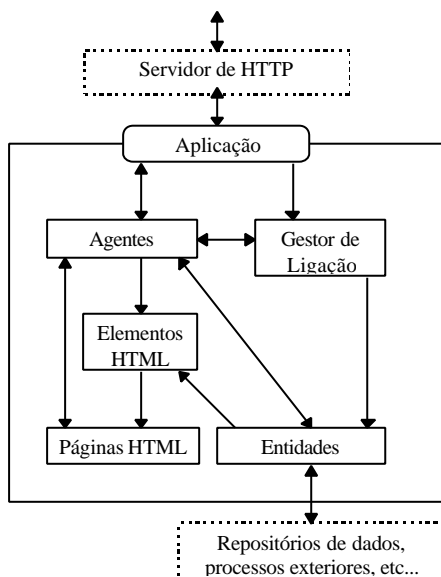


Figura 2: Relacionamento, numa aplicação baseada na biblioteca GENESIS, entre os conceitos apresentados

Cada um destes conceitos é suportado por uma respectiva estrutura de classes, as quais poderão ser directamente instanciadas ou derivadas, para acrescentar funcionalidade.

Na figura 2 apresenta-se, num diagrama de blocos informal, uma descrição do relacionamento entre os conceitos apresentados.

Entidades

A informação a manipular por um sistema que recorra à biblioteca GENESIS deve ser representada por um conjunto de classes que denominamos de Entidades.

Entidades representam o interface com os repositórios de dados (tipicamente sistemas de ficheiros ou bases de dados), ou outros processos especializados. Sobre esses repositórios deverá ser criada um interface orientado para objectos, em que cada classe será o encapsulamento de uma dada entidade conceptualmente identificada no âmbito do serviço.

Assume-se que cada entidade poderá ter um nome, um identificador (número inteiro), e um conjunto de capacidades que definirão quem lhe poderá ou não pedir o quê (isto é, os resultados da evocação dos seus métodos poderão ser condicionados à identidade do utilizador - ver detalhes mais adiante). A estrutura de classes, correspondente a este conceito de Entidade, e já concretizada na biblioteca, oferece um conjunto de funcionalidades genéricas para manipulação destas características.

Das experiências realizadas encontram-se entretanto desenvolvidas classes gerais para suporte e acesso aos seguintes tipos de repositórios: sistemas de ficheiros, sistemas de base de dados Sybase e Oracle e ainda sistema de directórios distribuídos X500.

Um exemplo de um caso especial de uma classe da hierarquia das classes Entidade, suportada à partida pela biblioteca GENESIS, é o caso da classe Utilizador. Um utilizador é definido à partida como alguém que tem os seguintes atributos (o programador pode acrescentar, por herança, as extensões que desejar):

- um Nome, consistindo numa *string*;
- um Identificador, consistindo num número inteiro;
- um Tipo, que está associado a uma lista de tipos, a qual pode ter elementos como, por exemplo *Anónimo*, *Colaborador*, *Administrador*, *Convidado*, etc.;
- um Modo de acesso, que pode ser por exemplo *Consulta* ou *Alteração*;
- uma Chave, vulgo *password*.

O Tipo de um utilizador permite controlar, da perspectiva do serviço, as funcionalidades e oferecer ao utilizador e o detalhe das respostas às mesmas. Tal consegue-se conjugando esse Tipo com as capacidades associadas às restantes entidades (determinando-se assim quem tem acesso ao quê).

Da mesma maneira o Modo permite controlar, mas agora da perspectiva do utilizador, a forma como este quer que a informação lhe seja apresentada pelo serviço. Assim, se este por exemplo pretender poder alterar todos os dados a que tiver acesso, deve configurar um Modo de *Alteração*, mas se apenas pretende consultá-los pode configurar um Modo de *Consulta*. No primeiro caso o serviço retornará ao utilizador *forms* sempre que este acceda a informação que pode alterar, enquanto que no

segundo caso receberá simplesmente texto (as classes de Elementos HTML tratam esta situação de uma forma transparente para o programador). Esta funcionalidade não se destina apenas a satisfazer os caprichos do utilizador, já que ela pode ser essencial se este pretender por exemplo imprimir, directamente do seu cliente, uma página que se tem presente. Neste caso deve-se configurar previamente um Modo de *Consulta*, para que a página não contenha *forms*, já que de outra forma o conteúdo dos mesmos não será impresso (uma característica lógica de todos os clientes, já que estes *forms* podem ser mesmo janelas de edição contendo textos que podem exceder os limites físicos da janela na página).

A biblioteca oferece já suporte para reconhecimento e gestão da entidade Utilizador e respectivas características apresentadas, sendo no entanto necessário que o programador a configure (ou programe!) para o repositório adequado (base de dados, ficheiro, directório X500, etc.).

Agentes

Conceptualmente, um serviço deverá ser desenhado como um conjunto de funcionalidades distribuídas por vários agentes especializados. Um agente será assim uma classe especializada no processamento de uma determinada componente do sistema de informação (tal como a gestão dos procedimentos de identificação de um utilizador, o acesso e alteração de uma ficha pessoal numa base de dados, etc.).

Os métodos de um agente podem-se agrupar em dois blocos:

- *bloco interno* - consistindo num conjunto de métodos públicos que retornam a informação necessária para requerer posteriormente um serviço ao agente (servindo assim de apoio à formação das páginas de resposta);
- *bloco externo* - consistindo num método público, destinado à identificação dos pedidos dos clientes (método esse evocado por omissão pela função *main* do sistema), e ainda num conjunto de métodos privados destinados ao processamento concreto dos pedidos (devendo existir tipicamente um destes métodos para cada método do bloco interno atrás referido).

Na formação de uma página, se se pretender incluir uma âncora para acesso a uma determinada funcionalidade de um agente, tal deve ser pedido a uma instância do mesmo, evocando o método correspondente no seu bloco interno de métodos. Como resultado recebe-se uma instância de uma classe Estrutura HTML *form*, URL, etc.), a qual tem já todos os parâmetros específicos para esse acesso.

Se esta funcionalidade for agora seleccionada pelo cliente, o *gateway* do serviço receberá um pedido que irá ser processado na seguinte sequência de operações:

1. a função principal (*main*) cria o Gestor de Ligação, o qual lê o CGI e pode ainda criar e iniciar variáveis globais;
2. a função *main* pergunta ao Gestor de Ligação qual o agente requisitado, criando de seguida uma instância deste o qual lhe passa o pedido;
3. o agente pergunta ao Gestor de Ligação qual o pedido que lhe é feito e com que parâmetros vem acompanhado;
4. o pedido é processado pelo método respectivo do bloco interno (de métodos) do agente;
5. no final é composta e enviada a resposta (como resultado a um pedido é sempre gerada uma resposta, na forma de uma página HTML).

Como suporte à entidade Utilizador, é também já oferecido na biblioteca GENESIS um agente específico para gestão de utilizadores suportando operações como alteração de dados (tipo, chave, etc.) e ainda de autenticação (reconhecimento de nome e chave).

Gestor de Ligação

O interface CGI entre o *gateway* do serviço e o servidor HTTP é gerida pelo Gestor de Ligação. O Gestor de Ligação é uma instância de uma classe, definida na biblioteca, que recebe e armazena os parâmetros recebidos por CGI, tornando-os acessíveis ao resto do serviço.

Os parâmetros recebidos por CGI num acesso são de quatro tipos:

- o identificador do agente evocado;
- a identificação da função evocada no agente;
- uma lista de parâmetros para a função evocada no agente;
- uma lista de parâmetros de entidades globais.

Cada parâmetro consiste num par *nome/valor*, sendo *nome* e *valor* sequências de caracteres compostos de números e/ou letras (*strings*).

As entidades globais correspondem a variáveis globais da aplicação, conhecidas pelo Gestor de Ligação, sendo criadas e iniciadas directamente por este. A classe fornecida na biblioteca já reconhece e trata um conjunto de variáveis deste tipo (como seja por exemplo uma instância que identifica o Utilizador), podendo o programador estender a classe para novas variáveis.

Os restantes tipos de parâmetros são da responsabilidade dos agentes, sendo gerados e depois recuperados por estes. Para estes casos o Gestor de Ligação forma internamente uma lista de pares de *strings*, com significados de [nome,valor], pelo que ao pretender recuperar um parâmetro um agente apenas tem de requerer ao Gestor de Ligação que lhe seja devolvido o valor associado ao parâmetro cujo nome se fornece.

Por questões de segurança estes parâmetros circulam na realidade na rede como apenas um único parâmetro CGI (par [nome,valor]), o qual é na prática uma *string* resultante da codificação da sequência de parâmetros reais. A manutenção da ligação com o utilizador é ainda concretizada associando ao parâmetro identificativo do mesmo (que pode ser o seu número), de uma chave baseada no valor da hora de acesso, a qual é “refrescada” em cada acesso efectuado (quer isto dizer que, uma vez identificado, um utilizador vê essa identificação reconhecida pelo sistema enquanto o intervalo de tempo entre acessos não exceder um determinado valor, determinado pelo programador da aplicação ou pelo administrador da mesma).

Páginas

Cada agente gera sempre, como resposta a um pedido, uma página HTML. O conjunto de páginas existente no serviço é independente dos agentes, podendo diferentes agentes retornar a mesma página como resposta.

Uma página é constituída por três partes: um cabeçalho, um corpo e um rodapé. Cada uma destas partes é uma lista de elementos HTML.

O cabeçalho e o rodapé poderão ser comuns a um grupo de páginas, sendo formados no(s) construtor(es) da classe respectiva (sendo deste modo apenas programados uma única vez, dando uma identidade única a todo o serviço ou a determinadas partes deste). O corpo é a parte específica da página, podendo ainda assim, para uma determinada classe, assumir uma ou mais formas, de acordo com os métodos concretizados.

Um agente, ao pretender criar uma página, evoca um construtor que forma transparentemente o cabeçalho e o rodapé. De seguida deve evocar o método desejado para o corpo da mesma.

Na sua formação uma página poderá criar instâncias de entidades ou de outros agentes, as quais interrogará para obter informação (estado das entidades e interfaces dos agentes). Nas respostas a estes pedidos tanto os agentes como as entidades poderão verificar as capacidades do utilizador, decidindo assim que tipo de resposta deve ser retornada para a página (por exemplo, aquilo que para um determinado utilizador pode ser retornado como um URL, dando-lhe assim a possibilidade de continuar a navegar no serviço, pode para outro utilizador ser apenas texto, ou então não ser mesmo nada).

Elementos HTML

Um conjunto de classes básicas encapsula completamente a linguagem HTML, havendo assim classes para representar cabeçalhos, texto, listas, URLs, *forms*, etc. Neste momento é suportada a linguagem HTML nas suas versões 1.0 e 2.0, tendo ainda sido concretizadas algumas das principais extensões

suportadas pelo cliente Netscape Navigator. Para clientes que apenas suportem HTML 2.0, são ainda fornecidas versões específicas de algumas dessas extensões, como por exemplo classes que oferecem funcionalidades de tabelas.

Herdando esta estrutura de classes podem ser ainda construídas classes de elementos mais complexos ou apenas específicos para determinadas aplicações, como por exemplo elementos (em formato *form* ou não), que representem datas (dia, mês e ano), horários (diários, mensais, anuais), etc.

Um outro aspecto extremamente importante desta estrutura de classes é que elas excedem as simples características dos respectivos correspondentes HTML, integrando-se com outros aspectos da biblioteca como por exemplo o Modo do Utilizador ou o Tipo do mesmo e as Capacidades programadas para a funcionalidade requisitada (ver descrição anterior da entidade Utilizador).

4. INTERACÇÃO COM UM SERVIÇO BASEADO NA BIBLIOTECA GENESIS

Em jeito de resumo da descrição apresentada, ilustra-se na figura 3 a sequência de acções resultantes de um pedido genérico feito a um serviço utilizando a biblioteca GENESIS.

O servidor de HTTP, ao ser contactado por um cliente com um pedido para acesso a um serviço baseado na biblioteca GENESIS (passo 1), limita-se a lançar o *gateway* do serviço e a enviar-lhe os possíveis parâmetros por CGI (passo 2).

Na início do serviço é criada de imediato uma instância da classe Gestor de Ligação (passo 3), a qual se inicia lendo e salvaguardando todos os parâmetros recebidos pelo interface CGI. Se for caso disso, o objecto Gestor de Ligações criado irá de seguida interpretar os parâmetros recebidos e criará uma instância da classe Utilizador (passo 4), a qual ao ser criada irá recorrer a um repositório para se iniciar (base de dados de utilizadores, por exemplo).

A função principal do serviço (*main*) recebe de seguida do Gestor de Ligações a indicação de qual o agente requisitado, criando uma instância deste a quem requer que prossiga com o processamento (passo 5).

O agente evocado dialoga agora com o Gestor de Ligação, questionando qual o pedido que lhe é endereçado e os parâmetros CGI correspondentes que o acompanham (passo 6). Uma vez o pedido identificado inicia-se o seu processamento específico invocando o método privado correspondente (passo 7). Este processamento pode implicar diálogos com a instância do utilizador (passo 8) ou ainda com outras entidades do serviço (passo 9), podendo daí resultar alterações aos estados dos mesmos (isto é, escritas nos repositórios de dados).

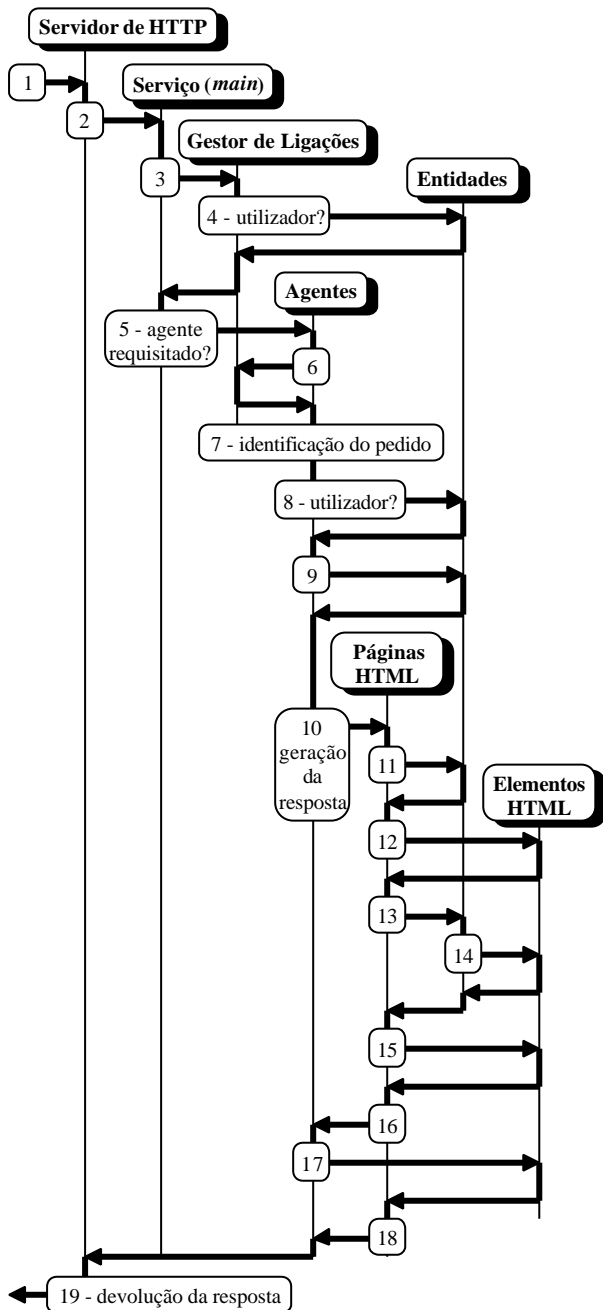


Figura 3: Interação com um serviço construído utilizando a biblioteca GENESIS.

Uma vez o pedido processado, e em função dos seus resultados, o agente cria a instância da página de resposta correspondente (passo 10).

A página de resposta poderá consultar a instância da entidade que representa o utilizador (passo 11) na intenção de saber características deste (privilégios, modo de acesso, etc.), indo de seguida formar a lista de instâncias de elementos HTML a devolver-lhe (passos 12 a 17). Estas instâncias poderão ser criadas directamente pela página (passo 12) ou poderão ser pedidas a outras entidades do serviço (passos 13 e 14) ou agentes (passos 16 e 17).

No caso de se pretender retornar informação sobre entidades, esta pode ser fornecida directamente por essas entidades já como elementos HTML (passos 13 e 14) ou de uma outra qualquer forma que depois terá que ser transformada em elementos HTML pela própria página (passo 15). Já no que respeita aos agentes, a página receberá sempre elementos HTML (passos 16 e 17), pois neste caso trata-se de obter acesso às funcionalidades dos mesmos (URLs, forms,...). Se alguma dessas funcionalidades for posteriormente seleccionada pelo utilizador dar-se-á assim início a um novo ciclo de acesso ao serviço, onde o agente requisitado será agora o mesmo que gerou o elemento HTML.

No final de tudo isto o agente requisitado recebe a página completamente formada (passo 18), a qual é finalmente devolvida ao cliente através do servidor de HTTP (passo 19).

5. CASOS DE UTILIZAÇÃO DA BIBLIOTECA GENESIS

O desenvolvimento da biblioteca GENESIS tem-se realizado ao mesmo tempo que esta tem sido utilizada num conjunto de experiências levadas a cabo pelo Grupo de Sistemas e Serviços Telemáticos do INESC, as quais têm determinado as suas características. Tal estratégia tem-se mostrado ser bastante válida, já que permite por um lado identificar requisitos concretos e por outro testar novas experiências de uma forma incremental e evolucionista.

O SERVIÇO ININWEB: O INESC NA WEB

O InInWeb tem como objectivo informatizar alguns serviços e procedimentos administrativos do INESC, tornando-os deste modo menos burocráticos e mais acessíveis. Pretende-se que cada colaborador possa participar, de uma forma uniforme e discreta, numa gestão mais descentralizada e eficaz da instituição. Esta iniciativa foi a primeira experiência que levou à concepção e desenvolvimento da biblioteca GENESIS, pelo que vamos apresentá-la com algum detalhe.

Passado do InInWeb (Motivação para a biblioteca GENESIS)

Iniciou-se em meados de 1994 no INESC uma experiência que visava criar um sistema onde fosse possível obter informação genérica sobre a instituição, nomeadamente naquilo que fosse comum a pessoas, unidades organizacionais e suas actividades. Ao mesmo tempo, e tentando otimizar a máquina administrativa existente, este serviço deveria suportar dois tipos de funcionalidades:

- Funcionalidade denominada de “Dossier de Organização”, em que o objectivo seria o de dar informação geral sobre a instituição, nomeadamente passando para suporte informático aquilo que no INESC já era conhecido como os

“Dossiers de Organização” das unidades, e que compreende um conjunto de informação relativamente estática (o suporte desta informação era o papel), sobre o historial, objectivos e actividades de cada uma dessas unidades e das pessoas que as compõem;

- Funcionalidade de *Timecards*, em que o objectivo era suportar a tarefa administrativa em que cada elemento do INESC tem que declarar, no final de cada mês, quais as actividades em que andou envolvido e quais as percentagens do seu tempo que lhes dedicou. Este procedimento administrativo, que permite gerar importantes métricas para gestão, é também bastante útil para permitir obter uma visão dinâmica e actual das actividades e do envolvimento de cada pessoa e unidade nas mesmas.

Uma vez que ambas as funcionalidades iriam partilhar informação comum (para ambos os casos é necessário conhecer as pessoas que trabalham no INESC, as suas unidades e as tarefas em que se ocupam), optou-se por um serviço único, com as duas funcionalidades integradas.

A diferença residiria no modo de acesso, em que para o caso do “Dossier de Organização” este seria público, sem qualquer restrição, enquanto que no caso do serviço de *Timecards* teria que existir um acesso controlado e restrito. Isto seria necessário não só para proteger a confidencialidade do mesmo mas também para controlar o acesso dos diferentes tipos de utilizadores (os quais iriam variar de desde o simples indivíduo que apenas tem acesso ao seu *Timecard*, até aos elementos administrativos que têm a tarefa de, globalmente, confirmar, controlar e processar os mesmos).

Tentando tirar ainda partido de algum suporte que já havia no INESC, a experiência foi planeada para ser concretizada em dois *gateways WWW*, ambos a desenvolver em linguagem C e em máquinas Unix. A funcionalidade associada ao “Dossier de Organização” teria como suporte um Directório X.500 [RFC 1309], já utilizado e mantido no INESC, usando os *Timecards* uma Base de Dados Relacional (SQL Server da Sybase [Sybase 94]), também já utilizada e mantida para esse fim, mas com uma funcionalidade limitada e de acesso apenas local.

Cedo no entanto se verificaram algumas limitações a esta solução, de onde se destacam, resumidamente:

- as características do directório X.500, que se veio a mostrar pouco flexível quando se pretendem fazer alterações à sua estrutura (na realidade, sendo um serviço independente e mantido no INESC para outros fins, não se mostrou prático nem conveniente alterá-lo e utilizá-lo como um simples SGBD);
- o facto de os dois sistemas (X.500 e base de dados de *Timecards*) se encontrarem em máquinas

diferentes tornava a consistência de informação entre os mesmos uma tarefa penosa, senão mesmo impossível nalguns casos (a tecnologia *WWW* não tem, neste momento, qualquer suporte à distribuição, e muito menos a transacções...).

Como resultado dos dois pontos anteriores estava-se a caminhar para a concretização de um serviço que na realidade não eram mais que dois serviços separados, com funcionalidades complementares que, apesar de fazerem todo o sentido ser encarados como um só, tal não era possível por questões puramente tecnológicas (o projecto estava assim a assentar em soluções expeditas que de modo algum reflectiam uma visão integrada).

Perante esta realidade resolveu-se interromper a experiência (se bem que se encontrassem bastantes funcionalidades já concretizadas), e repensar todo o projecto. Dessa reflexão resultaram, sucintamente, as seguintes conclusões:

- concretizar ambas as funcionalidades integradas no mesmo serviço, assentando toda a funcionalidade sobre uma única Base de Dados Relacional;
- quando confrontadas a dimensão e características do projecto com as limitações que a tecnologia *WWW* ainda apresenta no seu estado actual, cedo se mostrou aconselhável dividir a experiência em duas iniciativas complementares, a desenvolver em paralelo:
 - uma iniciativa deveria visar desenvolver uma tecnologia de base que permitisse suportar, genericamente, serviços desta natureza, já que entretanto outros casos potenciais começaram a ser identificados (ver outros casos de estudo adiante). Esta plataforma de suporte deveria “puxar” para mais alto nível alguns dos aspectos da tecnologia *WWW*, tornando esta mais agradável e simples de usar em projectos de certa dimensão;
 - o serviço em si deveria ser então concretizado recorrendo à tecnologia de suporte a desenvolver, devendo ao mesmo tempo os seus requisitos funcionar como motivação e ponto de partida para os objectivos do ponto anterior;
- finalmente foi decidido complementar ainda o serviço com uma terceira funcionalidade, consistindo num tratamento especial para as tarefas relacionadas com projectos europeus em que o INESC se encontre envolvido, já que este é um dos mais importantes campos de actividade da instituição.

Presente e Futuro

O serviço InInWeb encontra-se neste momento em fase de integração nos processos administrativos do INESC. A sua primeira versão está neste momento pronta e a ser testada, sendo no entanto de realçar, como aspecto

negativo, a fraca qualidade dos elementos HTML (versão 2.0) disponíveis para a composição de páginas genéricas de interrogação da base de dados e de apresentação de resultados, principalmente quando estes são muito volumosos ou requerem algum detalhe. Recentemente decidiu-se suportar directamente a versão 1.1N do Netscape Navigator (designadamente do mecanismo de descrição de tabelas), o que se traduziu num acréscimo de qualidade significativo.

Como aspectos positivos tem no entanto sido realçado o aspecto do acesso heterogéneo ao sistema (o qual pode ser feito a partir de computadores pessoais IBM compatíveis ou Macintosh, ou ainda de máquinas Unix), e ainda a relativa facilidade de utilização e de adaptação dos utilizadores ao serviço.

SISTEMA DE INFORMAÇÃO AO PÚBLICO DA CP

Uma outra experiência recentemente iniciada visa disponibilizar um serviço de informação ao público para a CP (Companhia de Caminhos de Ferro de Portugal), onde estarão disponíveis todas as linhas e horários de comboios, assim como informação geral sobre a empresa. Este serviço está a ser desenvolvido numa máquina Sun, recorrendo-se a um gestor de bases de dados Oracle.

SISTEMA DE GESTÃO DE HORÁRIOS PARA O IST

Este serviço tem como núcleo um sistema pericial, o qual recorre a uma base de dados Oracle como repositório. Em termos de funcionalidades este serviço deverá gerar e fornecer os horários para todas as turmas do IST, isto de acordo com os currículos das mesmas, com as disponibilidades das salas e com vários conjuntos de restrições susceptíveis de serem configuradas pelo interface. Outras funcionalidades suportadas serão ainda a marcação de salas para eventos extraordinários ou para provas de avaliação.

SISTEMA DE INFORMAÇÃO DE RESULTADOS ELEITORAIS

Foi desenvolvido, em colaboração com o jornal “O Público” e o núcleo do INESC na Faculdade de Ciências da Universidade de Lisboa, um sistema de informação para a Internet com os resultados dos actos eleitorais decorridos em Portugal deste 1974. O interesse tecnológico deste projecto é a geração automática de informação estatística em forma gráfica e de ter um sistema de consulta de informação baseado em imagens sensíveis com base nos mapas geográficos de Portugal.

SISTEMA DE GESTÃO DE MANUAIS DE QUALIDADE

Encontra-se em desenvolvimento, por contratação pelo Centro de Ferramentas de Software (CFS) do INESC, um sistema de gestão de manuais de qualidade, cujo objectivo é permitir às entidades certificadas construir os seus próprios manuais de qualidade e poderem utilizá-los, recorrendo a qualquer cliente WWW. Este

sistema é o primeiro a ser desenvolvido para ambiente MS-Windows e também será o primeiro, da sua classe, a ser efectivamente comercializado no mercado nacional pelo consórcio constituído pelo CFS e pela Partex SA.

6. TRABALHO RELACIONADO

O desenvolvimento de *gateways* CGI, que geram dinamicamente informação em páginas HTML e recorrendo a repositórios de dados heterogéneos, é actualmente uma importante área de interesse industrial e de investigação. Existem em todo o mundo projectos e trabalhos, cujos objectivos são, de uma forma ou outra, similares ao nosso, de onde podemos referir:

- OMNIWeb [Matheus 95], da National Space Science Data Center (NSSDC), dos EUA. OMNIWeb providencia acesso a uma grande variedade de informação das missões da NASA, permitindo aos utilizadores pesquisá-la e analisá-la..
- Hyper-G [Andrews 95], do Institute for Information Processing and Computer Supported New Media (IICM), dos EUA. Hyper-G é um sistema de informação hipermédia, distribuído, multi-protocolo e à escala mundial, que usa uma base de dados orientada para objectos para providenciar mecanismos de manutenção de Ligações e de estruturação de informação.
- Servidor OMNIS-WWW [Clausnitzer 95], da Universidade Técnica de Munique, Alemanha. OMNIS é um sistema de pesquisa de informação multimédia para a gestão de documentos em bibliotecas e escritórios. O servidor OMNIS-WWW é um *gateway* entre o servidor OMNIS e qualquer cliente WWW.
- O2Web [O2Web 95], da empresa O2 Technology - SA, França. O2Web pretende integrar todos os benefícios da tecnologia das bases de dados com um conjunto completo de ferramentas para desenvolver servidores WWW baseados no sistema proprietário O2.
- GIB [Lonzewski 95], do Instituto de Computação da Universidade Técnica de Munique, Alemanha. GIB (*Generation of Intelligent User Interfaces*) é um projecto suportado pela Siemens cujo objectivo é definir um mecanismo geral de geração de interfaces de utilizador baseada em modelos. Foi utilizado o ambiente WWW e adoptado a componente PLUG-IN (*Plan based User Guidance for Intelligent Navigation*) com suporte para geração dinâmica de páginas HTML. Apresentaram um exemplo interessante de um interface de gestão e utilização de um telefone RDIS.
- Investigadores do Departamento de Informática da Universidade de Trento, em Itália, apresentaram em [Ronchetti 95] propostas para a utilização da tecnologia WWW como forma de acesso mais fácil e mais agradável a sistemas de informação antigos.

- Componentes CGI da linguagem/ambiente Delphi [DelphiCGI 95]. Delphi é o “Visual Pascal” da Borland para ambiente Windows 3.x, Windows NT e Windows 95. Apresenta um compilador bastante eficiente, ambiente integrado e visual de desenvolvimento e um conjunto adicional de características. A empresa de consultoria “Ann Lynnworth. SynchroniciTech” nos EUA tem vindo a desenvolver para o ambiente Delphi um conjunto de componentes para criação rápida de aplicações CGI para ambientes Windows e Windows NT.

7. CONCLUSÕES E TRABALHO FUTURO

Este documento abordou o problema do desenvolvimento de sistemas de sistemas e serviços para a Internet, com interface WWW.

Apresentou-se a biblioteca GENESIS, um conjunto de classes C++ em desenvolvimento com o objectivo de apoiar a construção desse tipo de aplicações, escondendo-se assim ao programador um conjunto de dificuldades que se encontram presentes na estado actual da tecnologia WWW. Deste modo pode-se potencialmente produzir sistemas de informação e serviços comerciais cliente-servidor, distribuídos, multi-plataforma, com informação multimédia e acessíveis à escala mundial (isto mérito da Internet).

Trabalho futuro incluirá:

- Extensão dos suportes à estrutura de classes de Entidades (interface uniforme para acessos a diferentes SGBDs, acessos via ODBC, etc.);
- Extensão para classes “extra-HTML” genéricas e de interesse geral, como por exemplo a geração de gráficos, etc.;
- Suporte ao desenvolvimento de serviços distribuídos, compostos por *gateways* em diferentes máquinas mas que de algum modo se possam conhecer mutuamente (possibilitando assim a concepção integrada dos serviços);
- Escolha (adaptação ?) ou desenvolvimento de uma metodologia para a análise e desenho de sistemas e serviços baseados na biblioteca GENESIS;
- Integrar a biblioteca GENESIS num ambiente de *authoring* mais produtivo, nomeadamente com interface visual (actualmente o interface disponível é bastante rudimentar, resumindo-se ao nível da programação);
- Seguir as tendências tecnológicas, designadamente no que diz respeito a novas versões do protocolo HTTP e das linguagens HTML, VRML e Java.
- Utilização da biblioteca GENESIS no desenvolvimento de novos serviços...

8. BIBLIOGRAFIA

[Andrews 95] Andrews, K. et al. “Serving information to the Web with Hyper-G”. Computer Network and

ISDN Systems, vol.27 n°6, 819-26, April 1995.

[Berners-Lee 94]

T.Berners-Lee, R. Cailliau, A. Luotonen, H.F. Nielsen, A. Secret. “The World-Wide Web”, *Communications of the ACM.*, pp. 76-82, August 1994.

[Clausnitzer 95]

Clausnitzer, A. et al. “A WWW interface to the OMNIS/Myriad literature retrieval engine”. *Computer Network and ISDN Systems*, vol.27 n°6, 1017-26, April 1995.

[Codd 70]

E.F. Codd, “A Relational Model for Large Shared Data Banks”, *Communications of the ACM*, vol.13 n°6, pp. 377-387, June 1970.

[CODASYL 78]

CODASYL, “Data Description Language Committee Report. Information Systems”, vol. 3 n°4, pp. 247-320, 1978.

[DelphiCGI 95]

Ann Lynnworth. SynchroniciTech: “Delphi CGI Components” (<http://super.sonic.net/ann/delphi/cgicomp>)

[Lonczewski 95]

F. Lonczewski, “Using a WWW Browser as an Alternative User Interface of Interactive Applications”, Poster Proceedings of 3rd International WWW Conference, 1995

[Matheus 95]

Matheus, G. J. and Towherd, S. “NSSDC OMNIWeb: The first space physics WWW-based data browsing and retrieval system”. *Computer Network and ISDN Systems*, vol.27 n°6, 801-8, April 1995.

[O2Web 95]

O2Web, “O2 Technology. O2Web - Using object technology to build an Enterprise WWW Server”. *Informação Comercial* (<http://WWW.o2tech.fr>)

[RFC 1309]

C. Weiser, J. Reynolds, S. Heker, “RFC 1309 - Technical Overview of X.500”.

[Ronchetti 95]

M. Ronchetti et al, “Face Lift: using WWW technology for an external reengineering of old applications”, Poster Proceedings of 3rd International WWW Conference, 1995

- [Stroustrup 91] B. Stroustrup, "The C++ Programming Language", 2nd ed., Addison Wesley Publishing Company, 1991.
- [Sybase 94] Sybase, Inc.. "Open Client DB-Library/C Reference Manual Release 10.0", 1994.