



República Federativa do Brasil
Ministério da Economia
Instituto Nacional da Propriedade Industrial

(11) PI 0313681-7 B1



(22) Data do Depósito: 31/07/2003

(45) Data de Concessão: 03/09/2019

(54) Título: SISTEMA DE ENTREGA E APRESENTAÇÃO DINÂMICA DE RECURSOS DE MÍDIA DE GRANDE PORTE EM REDES RESTRITAS POR LARGURA DE BANDA

(51) Int.Cl.: H04L 29/06; H04L 29/08; H04N 5/00; H04N 7/167.

(30) Prioridade Unionista: 17/08/2002 US 60/403,995; 30/07/2003 US 10/632,003.

(73) Titular(es): DISNEY ENTERPRISES, INC..

(72) Inventor(es): WATSON, SCOTT, F; HASELTINE, ERIC, C; FREEMAN, ELISABETH, M.; LABERGE, AARON, P.; FRITZ, ADAM, T.; FREEMAN, ERIC.

(86) Pedido PCT: PCT US2003024175 de 31/07/2003

(87) Publicação PCT: WO 2004/017602 de 26/02/2004

(85) Data do Início da Fase Nacional: 17/02/2005

(57) Resumo: SISTEMA DE ENTREGA E APRESENTAÇÃO DINÂMICA DE RECURSOS DE MÍDIA DE GRANDE PORTE EM REDES RESTRITAS POR LARGURA DE BANDA. Conteúdo de mídia, baseada em um conjunto pre-determinado de restrições a partir de um provedor de conteúdo é entregue a uma cache local de um dispositivo do usuário antes de visualizar a mídia. O processo gerenciador de recurso do cliente residindo no dispositivo do usuário, a lista de recursos no site do provedor de conteúdo, e os recursos de mídia estão localizados em um local remoto.

5

**“SISTEMA DE ENTREGA E APRESENTAÇÃO DINÂMICA DE
RECURSOS DE MÍDIA DE GRANDE PORTE EM REDES RESTRITAS POR
LARGURA DE BANDA”**

BASE LÓGICA

5 1. Campo da Invenção

A presente invenção relata a entrega e apresentação de alta qualidade de conteúdo baseado em mídia, a usuários ou processos em redes, com Qualidade de Serviço (Quality Of Service – QOS) garantida mesmo se estiver disponível uma largura de banda da rede insuficiente.

10 2. Descrição dos Antecedentes da Invenção

Os inventores reconheceram que mesmo com os avanços nas tecnologias de rede, entregar experiências ricas e de alta qualidade continuará a ser um desafio.

Particularmente, a entrega de recursos de mídia de grande porte – sejam elas áudio, vídeo, flash, jogos, dados ou outros formatos de mídia digital –

15 freqüentemente requer maior largura de banda/produzividade da rede do que a disponível. Por exemplo, no caso de áudio e vídeo, um recurso de alta taxa de transmissão de bits somente pode ser entregue em tempo real se uma largura de banda efetiva do usuário for pelo menos igual à taxa de transmissão de bits do recurso, senão o resultado é uma experiência de usuário sub-ótima junto com

20 interrupções, paradas e armazenamento intermediário (buffering) de conteúdo.

Por outro lado, um executável de jogo grade pode não ter as mesmas limitações de tempo real (ou a qualidade de serviço requerida) como um vídeo, porém o download do recurso requer uma quantidade significativa de tempo e custo para o usuário, mesmo nas redes mais rápidas. Enquanto um número de “gerenciadores

25 de download” no mercado cuidarão disso para o usuário, um provedor de

6
conteúdo poderá querer gerenciar de forma inteligente e adaptável ao download
de recursos para o dispositivo de usuário (p.ex., um computador, um
conversor/decodificador com memória e/ou processador, um dispositivo) de forma
elegante e transparente, sem a necessidade da atenção do usuário.

5 Dado o exposto, existe a necessidade de gerenciar e entregar recursos de mídia
de alta qualidade e grande porte a usuários que utilizam sua limitada largura de
banda em turnos. Isto é, existe a necessidade de conseguir entregar conteúdo
discretamente aos usuários por meio da largura de banda e ciclos de inatividade
10 disponíveis, de forma que quando o conteúdo de alta qualidade for necessário,
esteja prontamente disponível sob demanda e seja prestada uma experiência de
usuário invariável. Isto, por sua vez, dá a impressão de que o usuário possui
maior largura de banda efetiva do que está realmente disponível. Para este fim,
também existe a necessidade desta tecnologia integrar-se transparentemente às
15 plataformas de entrega e apresentação (incluindo, mas não limitada a
navegadores da internet, flash e outras plataformas) e sistemas de publicação de
conteúdo. A presente invenção consegue isto e outras funcionalidades e também
supera as limitações da invenção anterior.

Para facilitar o entendimento, as seguintes definições se aplicarão em toda esta
aplicação; porém, nenhuma definição deverá ser considerada como anulando
20 qualquer entendimento aceito pela invenção, dos termos listados.

Glossário:

1. Produtividade – A quantidade de dados transferidos de um lugar para outro em
uma quantidade de tempo específica. Tipicamente, produtividades são medidas
em kbps, Mbps e Gbps.

25 2. Qualidade do Serviço (Quality of Service – QOS) – Termo que determina um

nível de produtividade garantido.

3. Processo-Cliente – O processo no cliente que recebe instruções ou advertências de gerenciamento de cache/exibição de um processo-servidor e depois executa instruções para trazer o estado atual da cache de acordo com o estado desejado e poderá disparar um ou mais notificações aos usuários ou outros processos conforme executa.

4. Cache (memória cache) – Um armazém de recursos com disponibilidade ou QOS “conhecidas”. Uma (memória) cache neste contexto é um mecanismo de armazenamento de recurso onde a QOS atende as exigências de conteúdo e geralmente é maior do que a mídia utilizada para obter os recursos. As mudanças de estado dentro da cache podem resultar em notificações.

5. Processo-Servidor – Fornece ao cliente a informação requerida para ao cliente gerenciar o estado da cache. Em sua implantação mais simples é quase similar a uma lista de execução gerada por servidor dinâmico. Implantações mais elaboradas (todas as nossas) também fornecem instruções de controle para o cliente informar outros processos do progresso em relação a conjuntos específicos de recursos.

6. Data de validade – Data de validade do recurso, e indica quando o recurso deverá ser removido da cache local.

7. URL (Localizador Uniformizado de Recursos) de Rechamada – URL recuperada quando o item de recurso foi baixado.

8. Autenticação no Cliente – autenticação ou cookie para configurar quando o item for baixado. Isto permite que uma aplicação de cliente ou servidor determine a presença de um recurso no sistema local.

9. Data de Embargo – Indica a última data em que o recurso será utilizado.



10. Deletar – Indica que o recurso será marcado para remoção explícita (para atingir a data de validade). Isto inclui a retração de um recurso.

11. Taxa de atualização – determina com que frequência o cliente verifica uma lista de recurso para alterações.

5 12. Caminho do recurso – a localização na rede de qualquer número de recursos associados à lista de recurso.

13. Recursos de Mídia – pelo menos um destes, texto, áudio, vídeo ou arquivo/dado binário.

14. Item – um único arquivo de mídia.

10 15. Link – URL para o arquivo de mídia.

16. hitCountUrl (URL de contagem de cliques) – URL para executar o comando ping (envio de sinal e espera de resposta para medição do tempo de resposta) após o arquivo ter sido descarregado com sucesso. Um parâmetro, duração, será anexado ao final da URL indicando, em segundos, quanto tempo levou o
15 download.

17. helpUrl (URL de ajuda) – URL para a ajuda do processo-cliente que deve ser exibida quando o usuário selecionar o item de menu ajuda.

18. trackWithCookie (Trilha com cookie) – elemento opcional que, se presente, indica que este recurso será adicionado à lista de recursos no cookie especificado
20 pelo cookieName (nome do cookie).

19. cookieName (Nome do cookie) – nome do cookie que lista todos os recursos baixados que possuem presente o elemento trackWithCookie (Trilha com cookie). Este cookie é essencial para anúncios por servidor, de forma que o servidor de anúncio sabe quais anúncios foram baixados. O formato do cookie vai consistir do
25 nome somente dos arquivos (sem extensão ou caminho) separado por vírgulas.

20. cookieDomain (Domínio do cookie) – domínio em que é colocado o cookie de recursos baixados. Podem ser especificados múltiplos domínios se forem separados por pontos e vírgulas ou vírgulas.

21. /regserver (registrar servidor) – registra os controles ActiveX no sistema e adiciona o processo-cliente na pasta inicializar.

22. /shutdown (desligar) – pára outra instância em execução do processo-cliente, se houver.

23. /unregserver (remover registro do servidor) – remove o registro dos controles ActiveX e remove o atalho da pasta inicializar. Também pára a instância em execução do clientprocess.exe e remove entradas de registro de objeto COM.

24. CDN – Content Distribution Network (Rede de Distribuição de Conteúdo). Um grupo federado de servidores de conteúdo de propriedade e operados por terceiros. Na prática um serviço CDN fornece capacidade adicional utilizando uma coleção de servidores altamente descentralizados.

RESUMO DA INVENÇÃO

A presente invenção fornece um sistema e método pelo qual o conteúdo de mídia é entregue de um provedor de conteúdo a uma cache local de um dispositivo de usuário, baseado em um conjunto pré-determinado de restrições, antes de visualizar a mídia. Uma lista de recurso abrange informações relacionadas aos recursos de mídia a serem baixados para o dispositivo de cliente, e é transmitida do provedor de conteúdo para o dispositivo de usuário. A lista de ativo, por exemplo, contém URL's ou informações relacionadas à localização dos recursos de mídia.

Um processo gerenciador de recurso do cliente reside no dispositivo de usuário e é responsável por baixar os recursos do provedor de conteúdo. O gerenciador de

recursos utiliza a lista de recursos para solicitar recursos de mídia que estão localizados em local remoto. O processo-cliente gerencia a entrega de recursos ao dispositivo do cliente, periodicamente, quando são atendidas restrições específicas. Por exemplo, os recursos são entregues ao usuário quando há disponibilidade de largura ideal de banda de rede, memória do dispositivo de usuário, qualidade de serviço garantida, etc.

A presente invenção fornece assim uma exibição de conteúdo de mídia contínua, ininterrupta e substancialmente transparente (visual e auditiva), através da entrega eficiente dos recursos de mídia aos usuários. Integrando estes recursos aos meios de visualização (p.ex., um navegador da web), o usuário recebe um fluxo de conteúdo de mídia ininterrupto e contínuo que não requer armazenamento intermediário em tempo real (real-time buffering).

A presente invenção ainda fornece métodos melhorados para entregar um ou mais recursos de mídia de grande porte, por exemplo, conteúdo de áudio, conteúdo de vídeo, filmes, jogos, etc., de forma inteligente e adaptável, através de rede para um armazém de recursos locais. Assim. Como tal, o armazém de recursos locais está disponível ao dispositivo do cliente, ou usuário final, e onde uma qualidade de serviço relativamente alta deve ser garantida. A invenção também inclui um método adaptável de combinar estes recursos em uma apresentação essencialmente transparente baseada na disponibilidade local dos recursos.

A presente invenção inclui um método para entregar recurso por rede. O método inclui o fornecimento de uma lista de recursos por rede para um dispositivo de usuário. O método inclui ainda um cliente que opera no dispositivo de usuário. O cliente consulta a lista de recursos ao baixar e entregar o recurso ao dispositivo

de usuário. O cliente também gerencia o download dos recursos baseado em quando pelo menos uma das restrições pré-determinadas for satisfeita.

Um provedor de conteúdo pode posicionar um recurso digital em um dispositivo de usuário a priori, de forma que esteja imediatamente disponível para uso, sem baixar pela rede, quando o usuário necessitar do mesmo. Isto pode ocorrer quando explicitamente solicitado por um usuário ou processo, ou ser iniciado por um provedor de conteúdo à base de serviço de assinatura.

A invenção é ainda descrita com referência aos desenhos que acompanham esta.

BREVE DESCRIÇÃO DOS DESENHOS

A FIG. 1 mostra uma visão geral do processo-cliente, de acordo com uma variante da presente invenção, interagindo com um site provedor de conteúdo e um servidor de mídia remoto;

A FIG. 2 é uma outra vista do processo-cliente que mostra os scripts utilizados para o fluxo de integração e a mídia armazenada na cache nos navegadores da web;

A FIG. 3 mostra um código de programa de amostra para o arquivo;

A FIG. 4 é uma representação de amostra das classes de C++ no aplicativo do processo-cliente;

A FIG. 5 é uma representação de amostra de um dispositivo de tempo para baixar recursos de mídia pelo cliente;

A FIG. 6 é um quadro de erro que afeta a quantidade de tempo de espera pelo download (descarregamento) da mídia.

A FIG. 7 representa um aspecto da natureza hierárquica dos atributos em relação às informações do recurso na lista de recursos; e

A FIG. 8 representa um outro aspecto da natureza hierárquica dos atributos em

relação às informações do recurso na lista de recursos.

DESCRIÇÃO DAS PRINCIPAIS VARIANTES

A. Descrição Geral

Uma exibição de conteúdo de mídia contínua, ininterrupta e substancialmente transparente (visual e auditiva) através da entrega eficiente dos recursos de mídia é fornecida aos usuários. Especificamente, os recursos são entregues a um dispositivo de usuário, periodicamente, quando restrições específicas forem atendidas (p.ex., disponibilidade de largura de banda, memória do dispositivo de usuário, QOS garantido, etc.). Integrando estes recursos aos meios de visualização (p.ex., um navegador da web), é fornecido ao usuário um fluxo de conteúdo de mídia ininterrupto e contínuo que não requer armazenamento intermediário em tempo real (real-time buffering).

Um ou mais recursos de mídia de grande porte, por exemplo, conteúdo de áudio, conteúdo de vídeo, filmes, jogos, etc., são entregues de forma inteligente e adaptável, através de rede a um armazém de recursos locais. Como tal, o armazém de recursos locais está disponível ao dispositivo do cliente, ou usuário final, e onde uma qualidade de serviço relativamente alta deve ser garantida. A invenção também inclui um método adaptável de combinar estes recursos em uma apresentação essencialmente transparente baseada na disponibilidade local dos recursos.

Um exemplo de um armazém de recursos locais, onde o QOS é garantido, poderia ser uma cache ou uma instalação de armazenamento de dados oferecida por uma Rede Local (Local Area Network – LAN) ao qual o dispositivo de usuário está conectado. Por disponibilidade local entende-se um recurso disponível em um local para o usuário final ou sistema de armazenamento do processo (ou um

armazém local, com rede de alta velocidade entre o dispositivo do cliente e o
armazém local).

Em uma variante, um processo ou aplicativo é executado em um dispositivo de
usuário, referido aqui como o cliente. O cliente é responsável por gerenciar a
5 cache de conteúdo no armazém local do usuário. Aplicativos, páginas da web e
apresentações multimídia podem depois confiar nesta cache para incorporar
recursos de mídia de grande porte que já estão residentes. Aplicativos, páginas
da web e apresentações multimídia podem consultar a cache sobre seu conteúdo
e elaborar dinamicamente e sob medida as experiências do usuário sobre o
10 conteúdo disponível. Estes aplicativos podem ser aplicativos por parte do cliente
ou do servidor.

O cliente poderá ser um serviço que é baixado da Internet e instalado no
dispositivo de usuário. O cliente pode alternativamente receber um componente
central de um sistema operacional de computador, ou vir no pacote com outros
15 softwares aplicativos.

O cliente gerencia uma cache de recursos de mídia para o usuário final. Em uma
implementação, os recursos incluídos são ditados por uma lista de recursos
fornecida por um provedor de conteúdo. Nesta implantação, o cliente recupera a
lista de recursos em intervalos de tempo específicos e ajustáveis de um local de
20 rede. Baseado nas informações das listas de recurso, o cliente gerencia o
download de recursos de um local remoto quando as restrições pré-determinadas
forem satisfeitas.

É permitido que um cliente esteja associado a mais de uma lista de recurso, de
forma que muitos provedores de serviço ou múltiplas unidades de negócio de um
25 provedor de serviço possam manter separadamente suas próprias listas de

recurso e fazer uso do mesmo cliente para gerenciar os downloads. Além disso, o cliente poderá saber da localização de uma lista de recurso quando o cliente for instalado ou atualizado. A instalação inicia a partir de uma página da web. Se o cliente ainda não tiver instalado no dispositivo de cliente, o software na página da web detecta isto e avisa o usuário para baixar e instalar o cliente. Também é possível adicionar uma lista de recurso ao cliente através da criação de scripts (combinado com controles ActiveX) em uma página da web.

O cliente gerencia o download de recursos baseados em restrições pré-determinadas. Conforme mencionado até agora, as restrições pré-determinadas incluem, por exemplo, disponibilidade de largura de banda da rede, memória do dispositivo de usuário, hora do dia, e qualidade de serviço garantida. Por exemplo, em uma implantação do cliente, os downloads ocorrem somente quando o dispositivo de usuário está inativo. Em uma outra implantação, o processo-cliente baixa somente quando a atividade da rede está abaixo de um certo nível. Em outro exemplo, o processo-cliente mede a utilização da CPU e da memória no dispositivo de usuário, e a restrição pré-determinada é atendida quando a utilização está abaixo de certos níveis de desempenho. Ainda em outro exemplo, o processo-cliente gerencia o download de recursos baseado na hora do dia, e estatisticamente quando a utilização da rede for a mais baixa. O processo-cliente pode, claro, utilizar adicionalmente qualquer combinação de restrições para gerenciar o download de recursos.

A capacidade de download pode ser ativada e desativada pelo usuário. O usuário também pode especificar que os downloads ocorram somente quando o dispositivo estiver em certas redes (por exemplo, em uma LAN versus uma conexão discada). Por exemplo, um usuário poderá assinar um serviço de

conteúdo por onde os recursos são baixados (pelo cliente) nas últimas horas da noite para visualizar pela manhã.

Além disso, o gerenciamento de recursos inteligente e adaptável é possível sem a necessidade de um “gerenciador de downloads” explícito sob o controle de um usuário. Por inteligentemente entende-se que os recursos sejam entregues ao cliente quando algumas restrições pré-determinadas forem satisfeitas (p.e.x., largura de banda da rede, hora do dia, QOS, etc). Por forma adaptável entende-se que as listas de recurso, no local do provedor de conteúdo ou no local do dispositivo do usuário, sejam automaticamente atualizados baseado nas preferências de visualização do usuário.

As listas de recurso são utilizadas como meio para os produtores de conteúdo publicarem e gerenciarem conteúdo no armazém local do cliente. Assim, esta funcionalidade permite aos produtores de conteúdo publicarem recursos de mídia que serão, em algum momento posterior, integrados à apresentação de um Aplicativo, página da web ou apresentação em flash. Em um exemplo, os produtores inserem informações sobre um recurso, incluindo sua localização, datas de apresentação e informações de texto e relacionadas que possam ser associadas ao recurso no sistema de publicação “back-end”. Estas informações são depois publicadas e incorporadas às listas de recurso relevantes. O sistema de publicação também pode (imediatamente ou em determinada data e hora no futuro) publicar uma página da web ou apresentação que incorpore um recurso.

Um método para entregar recurso em rede inclui: (i) fornecer uma lista de recurso por um provedor de conteúdo pela rede para um cliente, na qual o processo-cliente opera sobre um dispositivo de usuário; e (ii) entregar o recurso que corresponda à lista de recurso, através da rede para o dispositivo de usuário

quando as restrições pré-determinadas forem satisfeitas. Como exemplo, o recurso pode ser pelo menos um destes: dados binários, um conteúdo de áudio, um conteúdo de vídeo, um conteúdo de texto ou um arquivo multimídia. As restrições pré-determinadas podem incluir a hora do dia, o status do dispositivo do usuário (ou seja, se o dispositivo está sendo utilizado ou está inativo), utilização da largura de banda (ou seja, se a utilização da largura de banda está abaixo de um nível operacional pré-determinado), a utilização da CPU do dispositivo de usuário, a utilização da memória do dispositivo de usuário (ou seja, se a utilização da memória está abaixo dos níveis operacionais pré-determinados). Além disso, o recurso poderá ser armazenado na cache local do dispositivo de usuário. Ademais, o cliente poderá integrar o recurso armazenado ao conteúdo em tempo real do provedor de conteúdo de forma transparente para prover uma apresentação de conteúdo ininterrupta e transparente ao usuário. Além disso, a lista de recursos pode ser atualizada freqüentemente, fornecendo portanto ao processo-cliente (ou cache) a capacidade de receber uma ampla variedade de conteúdo periodicamente, baseado em restrições pré-determinadas.

O sistema fornece ao usuário um fluxo de conteúdo ininterrupto incluindo: (i) uma lista de recursos disponível por um provedor de conteúdo pela internet ao processo-cliente, na qual o processo-cliente opera em um dispositivo do usuário; (ii) um recurso, disponível de um local remoto, pela rede para o dispositivo de usuário, quando restrições pré-determinadas forem satisfeitas, na qual as informações do local remoto são obtidas da lista de recursos; e (iii) uma ferramenta integradora para integrar o recurso entregue ao fluxo de conteúdo que está sendo recebido pelo dispositivo de usuário do local remoto, pela internet. É apresentado ao usuário um fluxo de conteúdo ininterrupto e contínuo.

O produtor pode inserir recursos de mídia alternativos de baixa largura de banda e informações de texto correspondentes, que depois podem ser combinadas no “back-end” com as informações nos recursos de mídia ricos para produzir duas versões diferentes da página (uma para usuários que por qualquer motivo não conseguem baixar os recursos de grande porte, e outra para os recursos de mídia ricos). Os usuários poderão não conseguir baixar os recursos de mídia ricos por inúmeros motivos, tais como (1) sistema incompatível, (2) seus sistemas não estavam online para receber os downloads, (3) não pagaram por um serviço premium, (4) possuem espaço de disco insuficiente, (5) utilizavam o computador e não queriam que seus recursos fossem utilizados para download. Desta forma, a capacidade de entregar uma experiência alternativa em ambos os casos é importante.

Conteúdo e anúncios podem ser entregues, por exemplo, de forma seqüenciada, somente permitindo que o conteúdo seja executado quando também existir um anúncio correspondente na cache local. Em uma variante isto é conseguido pela utilização (por parte do cliente ou servidor) da criação de scripts para determinar se ambos recursos de mídia existem no dispositivo de usuário (a peça de conteúdo ou o anúncio). Em outra variante, esta funcionalidade é conseguida através da utilização de pacotes de lista de recursos, que configuram uma autenticação quando os recursos estiverem no dispositivo local.

O usuário pode especificar ativamente a mídia que é baixada e armazenada na cache no dispositivo local. Em uma implantação, isto é conseguido através de um banco de dados por parte do servidor que gerencia uma lista de recursos para cada usuário. Por exemplo, no meio da semana, um usuário poderia especificar que gostaria de assistir a um filme de qualidade de DVD na íntegra naquele final

de semana. Isto resultaria em um recurso ser adicionado à sua lista de recursos e

no cliente baixar o recurso com o tempo. O usuário poderá ser cobrado para adicionar o recurso a esta lista, ou (vide abaixo) o gerenciamento de direitos

digitais poderá ser utilizado para controlar o recurso. Em outro exemplo, este

5 método é combinado com um “empurrão” do provedor de conteúdo, como segue:

para assistir de manhã os noticiários transmitidos de madrugada, o usuário assina um serviço. Como resultado, o serviço insere um novo recurso na sua lista de recursos de usuário cada noite para identificar o novo programa a ser baixado.

Além disso, o sistema pode trabalhar com recursos de entrega de co-locação de

10 provedor ISP e a cabo. Em uma implantação, quando o produtor publica um

recurso, este é então movido para um recurso ou sistema de entrega de

disposição de provedor ISP e a cabo (que é fechado para o usuário na geografia

da rede). O cliente recupera aquele recurso e o coloca na cache local,

recuperando seu recurso ou sistema de entrega de cobrança de provedor ISP e a

15 cabo ao invés dos servidores de mídia de fonte de provedores de serviço.

Além disso, o provedor de conteúdo pode otimizar a experiência do usuário

baseado nos recursos próximos. Por exemplo, um web site relacionado a notícias

poderá fornecer a foto de uma matéria se não estiver disponível nenhum recurso

de mídia rico, porém se um recurso de vídeo estiver armazenado localmente, a

20 página da web poderá substituir o vídeo no lugar da imagem na página da web.

Isto pode incluir a “cópia” textual do vídeo correspondente, já que o vídeo e a

imagem não podem ser da mesma matéria e assim requereriam diferentes textos

e legendas de apoio. Em uma representação, isto se consegue se a detecção de

presença do recurso por meio de sua autenticação de recurso correspondente

25 (vide atributos de listas de recursos acima). Dependendo se a autenticação está

presente ou não, uma página diferente é criada para o usuário final. Em outra variante isto é realizado no cliente por meio da criação de scripts no navegador da web. Ou seja, utilizando a criação de scripts na página da web, uma página diferente é criada via HTML dinâmico baseado na existência de autenticação de recursos de mídia.

Uma vantagem oferecida pela invenção é que ela permite que um provedor de conteúdo otimize sua utilização de largura de banda da rede. Especificamente, aumentando a entrega de conteúdo quando a utilização cai abaixo do pico, os vales de utilização da largura de banda são preenchidos de forma que a utilização da largura de banda seja otimizada.

Em outro aspecto, os desenhos do gerenciamento de direitos digitais (digital rights management – DRM) podem ser aplicados aos fluxos de dados. Por exemplo, o recurso protegido pelo DRM pode ser publicado através da invenção. Assim, quando o usuário “toca” o recurso, eles são avisados para adquirirem a licença apropriada como um “direito de uso” do conteúdo.

Em outro aspecto, os recursos de JOGO podem ser distribuídos na forma de imagens ISO reais ou virtuais que podem ser utilizadas no cliente.

B. Exemplo de Conversor/Decodificador

O cliente poderá baixar e gerenciar recursos de mídia relacionados a conteúdo de mídia de diferentes fontes tais como um conversor/decodificador de TV, no qual os recursos são entregues em uma banda paralela de um sinal da operadora ou através de uma rede de dados. Tais como, a rede a cabo ou satélite, ou Internet ou intranet. Por exemplo, em uma variante, os expectadores em casa têm acesso a uma biblioteca de filmes, ou qualquer outro conteúdo de áudio/vídeo disponível para assistir a qualquer hora. Especificamente, o método envolve a transmissão

de recursos de mídia, tais como filmes, para um conversor/decodificador na casa de uma pessoa e permitir que acumule os filmes.

Uma unidade de disco rígido no conversor/decodificador é utilizada para armazenar filmes. Os filmes são transmitidos utilizando a tecnologia datacasting que transmite grandes quantidades de dados por sinais de televisão padrão. Informações relacionadas à tecnologia datacasting se encontram na Solicitação número WO9955087 para Hartson et al. arquivada em 16 de abril de 1999 e publicada em 28 de outubro de 1999, a qual está aqui anexada como referência.

Por exemplo, em um período de 24 horas, esta tecnologia datacasting pode distribuir 20 filmes na íntegra de alta qualidade. O conversor/decodificador é facilmente conectado à televisão dos usuários como qualquer outro dispositivo externo, tais como videocassete ou reproduutor de DVD. O conversor/decodificador também é conectado a uma linha telefônica para fins de cobrança.

Em uma variante de exemplo, o conversor/decodificador está localizado na casa de uma pessoa, conectado à sua televisão utilizando cabos de vídeo padrão. O conversor/decodificador possui uma antena que recebe dados por meio do sinal de televisão. O conversor/decodificador possui, preferivelmente, pelo menos uma unidade de disco rígido de 80 Gigabytes para armazenamento de uma diversidade de filmes. O conversor/decodificador possui um modem que o conversor/decodificador utiliza para contatar periodicamente o provedor de serviço. As informações passadas entre o provedor de serviço e o conversor/decodificador incluem: o "Histórico de Visualização/Aluguel" ("Viewing/Rental History") dos usuários, que é utilizado para fins de cobrança, os registros históricos (logs) de desempenho do conversor/decodificador, que são utilizados para monitorar o desempenho do sistema e as "Chaves de Filme"

("Movie Keys"), que são utilizadas para decriptografar os filmes.

O conversor/decodificador possui um processador que é capaz de receber o fluxo de dados do sinal de televisão, remontar os dados e gravar os dados no disco rígido. O processador também precisa ser capaz de simultaneamente reproduzir um filme e responder aos sinais infravermelhos do controle remoto, bem como a atividade do modem. O disco rígido é junto com o conversor/decodificador para fins de segurança, tornando-se inútil se removido e utilizado em qualquer outro lugar. O gabinete do conversor/decodificador preferivelmente não possui uma ventoinha e é projetado para ser muito silencioso. O gabinete do conversor/decodificador também compreende um processador seguro como parte de seu sistema de segurança. O processador seguro é o componente ativo de um cartão smart (com chip) que é fisicamente anexado ao PCB com epóxi, para tornar fisicamente difícil de falsificá-lo.

Filmes não são "transmitidos" ao conversor/decodificador em tempo real, ao invés disso os arquivos de conteúdo são "empacotados" e estes "pacotes" são continuamente transmitidos ao conversor/decodificador onde são crescentemente remontados. O usuário não sabe quais dados estão sendo enviados ao seu conversor/decodificador. Os filmes são descarregados em lista inversa pelo provedor para residirem passivamente no conversor/decodificador para um período de tempo limitado. A transmissão dos dados é controlada pelo provedor de conteúdo ou serviço. Para garantir que os filmes sejam recebidos na íntegra, o mesmo filme pode ser transmitido para o conversor/decodificador diversas vezes. Quaisquer pacotes de dados que não forem recebidos na primeira tentativa de transmissão serão recebidos nas transmissões subseqüentes. Na perspectiva da presente invenção, um processo-cliente reside no conversor/decodificador e os

recursos de mídia (conteúdo do filme) são recuperados do local remoto quando as restrições pré-determinadas (p.ex., largura de banda de rede, QOS) forem satisfeitas.

Os filmes transmitidos ao conversor/decodificador também podem ser associados a informações que definem certas características do filme. Por exemplo, um filme poderá ter uma data ou hora de início e final associados, que limite o período de tempo em que um filme pode ser assistido. Por exemplo, um filme poderá chegar e ser armazenado no conversor/decodificador, porém poderá ter uma data de início associada a ele que não permita que seja visto até aquela data. Isto inclui quaisquer discrepâncias nas horas de transmissão de filmes que possam variar de um local para outro, e também inclui filmes a serem “pré-carregados” e imediatamente disponíveis na data de liberação oficial. Similarmente, a lista de recurso poderá ter uma data final associada ao filme, após aquela data o filme não poderá mais ser visto, e é automaticamente removido do conversor/decodificador.

Os filmes armazenados no conversor/decodificador são criptografados. Na seleção de um filme para assistir e na satisfação de regras de negócio (isto é: o usuário possui crédito suficiente), o conversor/decodificador inclui o filme a ser decriptografado e reproduzido. O conversor/decodificador não precisa se conectar ao provedor de serviços antes de permitir que um filme seja assistido, desde que as chaves para decriptografar os filmes sejam tipicamente pré-recuperadas e residentes no conversor/decodificador juntamente com o status da conta corrente. Obviamente, os filmes criptografados não podem ser vistos sem decriptografá-los. Toda decriptografia é registrada em histórico (log) e este log é utilizado para determinar a conta do usuário. A lógica acerca da decriptografia e o status da

conta do usuário é manipulada pelo processador seguro. Na principal variante da presente invenção, é cobrada uma taxa do usuário na seleção do filme para assistir, um diálogo de "confirmação do aluguel" ("rental confirmation") subsequente e antes de ver o filme.

- 5 Em outra variante, o usuário é cobrado pela visualização de um filme, uma vez que uma parte substancial do filme tenha sido assistida. Uma vez selecionado um filme, ele pode ser assistido novamente sem cobrança por um período de tempo limitado (p.ex. 24 horas), ou por um número limitado de reproduções. Em outra variante, ao invés de pagar por cada filme, é cobrada do usuário uma taxa
- 10 mensal. A quantidade de taxa mensal dependerá de várias opções tais como o número de filmes que podem ser vistos, o período de tempo que um filme selecionado fica disponível para assistir, e o número de reproduções permitidas de cada filme selecionado. Uma linha telefônica é utilizada para efetuar a transferência do usuário da informação de que uma chave está sendo vista pelo
- 15 usuário e assim deverá ser feita uma cobrança. Assim, mesmo que o usuário tenha uma biblioteca de filmes armazenados no conversor/decodificador do usuário, não há cobrança a menos que um filme esteja realmente sendo assistido. Em outras incorporações, os dados podem ser transmitidos para o conversor/decodificador por cabo, satélite, internet, etc. Embora a transmissão sem fio seja uma variante principal, a presente invenção não deve ser limitada à
- 20 transmissão sem fio.

C. Descrição Em Relação Aos Desenhos

- A FIG. 1 mostra uma visão geral do sistema 10 utilizado para entregar e apresentar um fluxo de mídia, sem interrupções, em um dispositivo de usuário.
- 25 Especificamente, o dispositivo de usuário 20 inclui pelo menos um processo-

cliente 24 (p.ex., um processo gerenciador de recurso cliente) que interage com a lista de recurso do cliente 42 em um local do provedor de conteúdo, representado como 40. Também presente, no dispositivo de usuário 20, está uma cache local 26 para armazenamento de informações. A cache 26 pode ser pelo menos uma de uma memória de acesso randômico (random-access-memory – RAM), uma memória somente de leitura (read only memory – ROM), ou um disco rígido. Além disso, poderia estar presente um local remoto 60 que contenha recursos 62, tais como conteúdo de programa (p.ex., um evento esportivo, programa de culinária, etc). Como alternativa, os recursos 62 poderiam estar localizados no local do provedor de conteúdo 40, ou no provedor de serviço a cabo/internet (não mostrado). O recurso é disponibilizado no provedor a cabo/ISP, ante do cliente começar a recuperá-lo. Além disso, o cliente poderá recuperar os recursos simultaneamente de uma diversidade de localidades fisicamente separadas (p.ex., de um provedor a cabo/ISP, provedor de conteúdo, etc).

O processo-cliente 24 gerencia uma cache de recursos de mídia 26 para o usuário final. Em uma implantação, os recursos 62 incluídos nesta cache 26 são ditados pela lista de recursos 42 fornecida por um provedor de conteúdo 40 via caminho de dados 80. Nesta implantação, o processo gerenciador de recurso do cliente 24 recupera a lista de recursos 42 em intervalos específicos (e ajustáveis) de tempo, a partir de uma localidade da rede (tais como o site de provedores de conteúdo 40) via caminho de dados 80. Especificamente, o processo-cliente 24 faz uma solicitação, por via do caminho do sinal de controle 84, para o provedor de conteúdo 40 entregar uma lista de recursos atualizada 42 via o caminho de dados 80.

Igualmente, em outra implantação, a lista de recursos 42 pode ser enviada ao

cliente usuário final 24 por um servidor (não mostrado) do provedor de conteúdo, de forma periódica. A lista de recursos 42 é um arquivo de dados que contém, no mínimo, a lista de recursos de conteúdo. Cada elemento na lista de recursos 42 tipicamente identifica uma localização da rede e o protocolo necessário para obter o recurso. Por exemplo, em uma implantação esta localização da rede é uma URL – Universal Resource Locator e o protocolo é HTTP. Uma lista de recursos 42 também poderá conter outros atributos (conforme explicado na seção Resumo e definido na seção Glossário) associado a cada recurso, tais como:

(i) Data de validade, (ii) Remover;

(ii) URL de Rechamada – Em uma variante, esta URL de rechamada é utilizada para iniciar ações em um download de recursos, tais como um aplicativo de rastreamento (para permitir rastrear quais recursos são baixados e com que frequência) e um aplicativo para notificar o usuário via correio eletrônico (email) ou mensagem instantânea de que um dado recurso está pronto para assistir;

(iii) Autenticação por parte do Cliente – Em uma variante, é utilizada para adaptar a apresentação com base na disponibilidade de recursos;

(iv) Produtividade – Em uma variante, este número é uma porcentagem da produtividade disponível do cliente e é menor de 100%. O cliente pode reduzir a velocidade do download, tentando obter somente pedaços de dados por vez ao invés de tudo de uma vez. Em outra variante, este número é uma taxa de dados de download em bytes por segundo;

Data de Embargo – Qualquer conteúdo pode estar sujeito a uma janela limitada de disponibilidade. Uma vez atingida a data de embargo, o conteúdo é removido pelo cliente de forma que não fica mais disponível.

Em um aspecto, estes recursos podem ser combinados em pacotes de conteúdo.

Os atributos acima também podem ser atribuídos a um pacote. Por exemplo, isto permite que uma autenticação por parte do cliente seja criada, uma vez que um conjunto de recursos de mídia tenha sido entregue ao dispositivo de usuário.

Certos atributos podem ser associados a toda a lista de recursos 42, sendo tais atributos:

(a) Taxa de atualização: Em uma implantação isto pode ser controlado no servidor (p.ex., no local do provedor de conteúdo 40), que empurraria a lista de recursos 42 para um gerenciador do cliente 24 via caminho de dados 80.

(b) Caminho do Recurso, (c) Recursos de Mídia.

A FIG. 7 representa um aspecto da natureza histórica dos atributos em relação às informações dos recursos na lista de recursos. Conforme mostrado, a lista de recursos fornecida ao cliente, pelo provedor de conteúdo/ISP, possui atributos para todas informações do recurso. Como exemplo, Atributo (1,2) podem se referir à data de validade do Recurso 1. Os atributos, nesta representação, são locais para cada recurso.

Em um aspecto alternativo, os atributos podem ser globais para uma lista de recursos conforme mostrado na FIG. 8. Nesta situação, o cliente pode gerenciar o download dos recursos, na lista de recursos, baseado nos atributos globais, se necessário, a todos recursos.

O cliente 24 também é capaz de gerenciar muitos parâmetros de seu comportamento de download de recursos a partir de um local remoto 60. Por exemplo, em uma implantação, o gerenciador de recursos do cliente 24 inicia uma solicitação para o local remoto 60, via sinal 86, para transferir recursos de mídia 62 para a cache local 26 via caminho de dados 90. O processo-cliente 24 pode enviar estas solicitações para transferência de dados (downloads) para o local

remoto 60 dependendo se um conjunto de restrições pré-determinadas for
satisfeito ou não. Por exemplo, os downloads podem ocorrer quando o dispositivo
de usuário 20 estiver inativo, quando a largura de banda da rede estiver abaixo de
um certo nível, quando a utilização da CPU e/ou memória no dispositivo de
usuário 20 estiverem abaixo de certos níveis de desempenho.

A capacidade de download pode ser ativada ou desativada pelo usuário. O
usuário também pode especificar que os downloads ocorram somente quando o
dispositivo estiver em certas redes (por exemplo, em uma LAN versus uma
conexão discada).

Os recursos de mídia 62 podem ser regulados ou descarregados pelo local
remoto 60, dependendo das condições de rede (p.ex., tráfego, largura de banda
disponível, hora do dia, etc), sem uma solicitação de início do processo-cliente 24.

As listas de recursos 42 são utilizadas como meio para os produtores de conteúdo
publicarem e gerenciarem conteúdo em armazém/cache locais do cliente 26. Por
exemplo, esta funcionalidade permite aos produtores de conteúdo publicarem
recursos de mídia 62 que serão em algum momento transparentemente
integrados na apresentação de uma página da web ou apresentação em flash
(não mostrada).

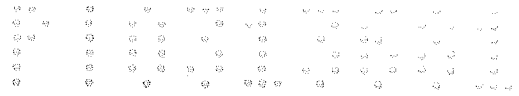
Conformemente, os produtores podem inserir informações sobre um recurso
incluindo sua localização, datas de apresentação e informação de texto &
relacionadas que possam estar associadas ao recurso em um sistema de
publicação "back-end". Estas informações são depois publicadas e incorporadas
nas listas de recursos relevantes. O sistema de publicação também poderá
(imediatamente ou em determinada data e hora no futuro) publicar uma página da
web ou apresentação que incorpore o recurso. O produtor insere recursos

alternativos de mídia de baixa largura de banda e informações de texto

correspondentes, que depois podem ser combinados com as informações nos recursos de mídia ricos para produzir duas versões diferentes da página (uma para usuário que por qualquer motivo não conseguem baixar recursos de grande porte e outra para os recursos de mídia ricos).

O gerenciador de recurso do cliente 24 processa um aplicativo que pode ser executado no dispositivo de usuário 20 para prover download e armazenamento em cache. Os usuários conseguiriam optar, possivelmente através do website do provedor de conteúdo, para obter o gerenciador/processo-cliente. Como variante de exemplo, o gerenciador do cliente poderia ser instalado pra iniciar automaticamente e ser executado em segundo plano.

A FIG. 2 exibe uma outra visão geral do fluxo de informações para o sistema 100 onde o processo gerenciador de recursos do cliente 102 está sendo executado em um dispositivo de usuário. Em uma representação de exemplo, uma lista de recurso do cliente 110 (p.ex., um arquivo XML) é periodicamente baixada de um provedor de conteúdo 120, e os recursos 130 listados naquele arquivo são baixados de um local remoto/provedor de conteúdo 140 e conformemente armazenados em cache. O usuário poderá navegar em uma página da web que pode exibir recurso(s) , incluindo um script 150 que se comunica com o processo gerenciador de cliente 160. Visto o processo gerenciador cliente 102 poder ser também um Servidor ActiveX, somente um objeto proxy poderia estar sendo executado na página da web 180, e o controle real poderia ser parte do processo gerenciador do cliente 102. Em um aspecto, o processo-cliente 160 poderia ser incorporado na página da web 180 ou parte do processo gerenciador do cliente 102.



29

C(i). Perspectivas do Projeto

As entradas podem ser: (i) arquivo de configuração, tais como um arquivo XML que permite o controle remoto sobre o comportamento do processo gerenciador do cliente 102; (ii) arquivos de mídia remotos a serem armazenados em cache no cliente. As saídas podem ser caminhos locais para arquivos de mídia.

Qualquer página da web pode exibir um recurso que seja armazenado na cache pelo cliente 102. Utilizando a interface ActiveX do cliente, uma página pode conter código JavaScript 150 para ver se o processo-cliente 102 está instalado, e solicitar ao cliente informações a respeito do caminho local para um arquivo de mídia. Se um caminho for retornado, o script pode depois passar esta informação do caminho para qualquer reprodutor de mídia baseado na web que aceite parâmetros.

O servidor de configuração 120 pode ser qualquer servidor que hospede o arquivo de configuração do cliente 110. O servidor de mídia 140 pode ser qualquer servidor da web que hospede os arquivos ou recursos de mídia 130.

C(ii). Fluxo de Execução

O processo-cliente 102 pode ser empacotado em um arquivo .cab e assinado com um certificado digital para identificar o criador do programa (p.ex., o Walt Disney Internet Group). Um arquivo .cab separado, xyzvideo.cab, pode ser criado para diferentes provedores de conteúdo. Por exemplo, o nome do arquivo poderia ser espnvideo.cab para a ESPN e ele poderia ser assinado com um certificado digital ESPN. O Internet Explorer utiliza o recurso Internet Component Download (Download de Componente da Internet) para baixar e rastrear a aplicação. Um arquivo.INF poderia ser incluído no .cab para instruir o baixador de componente sobre como instalar o processo-cliente 102.

O arquivo .cab é colocado em um servidor da web e quando o navegador carregar

uma página da web com um tag OBJECT (marcação OBJETO) contendo a identificação da classe do controle ActiveX apropriado, ele baixará o arquivo .cab utilizando a URL especificada pelo atributo CODEBASE (BASE DE CÓDIGO). O certificado digital é exibido para notificar o usuário que nosso controle foi baixado e solicita permissão para executar. Uma vez que o usuário aceita, esta janela não será exibida novamente a menos que seja baixada uma versão atualizada.

O módulo, clientprocess.exe, é adicionado à pasta Downloaded Program Files (Arquivos de Programa Baixados) e o cliente poderá criar um atalho para ele na pasta Inicializar. O nome do atalho depende de quem criou o arquivo (p. ex., Disney, ESPN, etc). O processo-cliente também poderá criar um cookie no domínio go.com chamado ClientProcess (Processo-Cliente), que é configurado para verdadeiro se o processo-cliente for instalado. Este cookie poderá ser regularmente verificado para impedi-lo de ser acidentalmente removido por um usuário.

O programa é inicialmente executado pelo sistema operacional e a partir daquele ponto executará sempre que o dispositivo de usuário iniciar.

O certificado digital mencionado anteriormente poderá ser necessário, desde que a maior parte dos usuários possui sua configuração de segurança do navegador alta o suficiente para rejeitar a instalação, a partir da Internet, de aplicativos que não sejam de confiança. A presença do certificado permite ao navegador identificar a origem do software e perguntar ao usuário se deve-se confiar naquela empresa ou pessoa.

O atributo CODEBASE da tag OBJECT suporta a adição opcional da versão desejada do software a ser instalado. A versão está anexada à URL do local do

cab, com um '#' utilizado como separador. Se, por exemplo, a versão atual do processo-cliente no campo for 1,0,0,1 e uma nova versão 1,0,0,2 foi colocada no servidor da web, anexando '#,1,0,0,2' no final da CODEBASE, a URL direcionará o baixador de componente para baixar e instalar a nova versão.

- 5 O arquivo .INF contém um gancho para executar primeiro clientprocess.exe com o parâmetro /shutdown (/desligar) para fechar primeiro o clientprocess.exe de forma que a atualização na requeira a reinicialização.

O instalador de componente poderá adicionar informações ao registro utilizando uma chave pré-definida.

- 10 O menu para o ícone da bandeja do sistema processo-cliente conterá uma opção de Desinstalação (Uninstall). Quando clicado, o processo-client se desinstalará executando uma outra instância do clientprocess.exe que passa pelo parâmetro /unregserver (servidor de desregistro), (ii) limpar as entradas de registro adicionadas pelo processo-cliente, (iii) remover item do processo-cliente da pasta
- 15 Inicializar.

Quando o sistema operacional inicia e o Explorer executa os itens na pasta Inicializar, o processo-cliente 102 iniciará. Ele pode não terminar a menos que o usuário feche-o a partir do ícone na bandeja do sistema ou o sistema operacional desligue.

- 20 O processo-cliente 102 pode não ser visível ao usuário, exceto um ícone na bandeja do sistema contendo alguns itens de menu. Se, por exemplo, o aplicativo BottomLine da ESPN estiver executando, a versão ESPN do processo-cliente não exibirá seu ícone na bandeja porque a BottomLine fornecerá ao processo-cliente 102 algumas opções em seu menu. Isto é útil já que diminui o número de ícones
- 25 que podem ser colocados na bandeja em um único cliente. Programas como o

BottomLine e o processo-cliente 102 podem precisar estar em comunicação periódica um com o outro para garantir que saiba se o outro desliga e que o processo-cliente sabe se um de seus itens de menu é selecionado. O ícone será ou um ícone ESPN ou um outro ícone, dependendo da criação.

- 5 Os seguintes itens estarão no menu de ícone de bandeja do processo-cliente: (i) Ajuda – exibe a ajudar para o processo-cliente 102, (ii) o arquivo do grupo XML pode ter um item helpUrl opcional. Quando o usuário seleciona Ajuda, um navegador será aberto com esta URL, (iii) Sobre – mostra a caixa de diálogo de direitos autorais/informações, (iv) Saída – pára o programa (não o remove da pasta inicializar), (v) Desinstalar – desinstala o programa e remove as entradas de registro COM.

Pelo menos uma instância do processo-cliente 102 poderá ser executada por vez. A única exceção a isto é se clientprocess.exe for executado com o parâmetro /unregserver ou /shutdown, aquela instância terminará uma instância que já estiver executando, se existir. Clientprocess.exe poderá conseguir esta funcionalidade de única instância criando um mutex* compartilhado que é verificado pelas instâncias subseqüentes do programa. Assim em algumas situações, se for encontrada uma instância existente, a segunda terminará imediatamente.

- 20 O modelo de tarefas para o processo-cliente poderá ser o modelo apartamento. Isto significar que os controles ActiveX do processo-cliente são projetados de forma que esperam que somente seja acessado a partir de uma única tarefa. Porém, o processo-cliente como um todo é multithread (multitarefa) e iniciará uma tarefa quando ocorrer o próximo download. Poderá ocorrer só um download por vez.

Durante o modo de operação regular, o processo-cliente executará sem nenhum parâmetro. Porém, os parâmetros linha de comando seguinte podem ser suportados: (i) /regserver; (ii) /shutdown – Para outra instância em execução de clientprocess.exe, se houver; e (iii) /unregserver.

- 5 O processo cliente é configurado utilizando arquivos de Grupo XML baixados do servidor de configuração. A uRL do arquivo de grupo é obtida através da tag PARAM xml dentro da tag OBJECT para o controle ActiveX ClientProcessGroup em uma página da web. Cada vez que um controle ClientProcessGroup é inicializado com um parâmetro xml, o processo cliente verificar para ver se aquele
- 10 grupo já foi adicionado. Se não, é adicionado ao registro e o arquivo é baixado e analisado sintaticamente. As entradas de registro para os arquivos de grupo não são removidas até que o processo-cliente seja desinstalado.

O arquivo de Grupo é baixado regularmente de acordo com sua taxa de atualização configurável. Se o servidor suporta o registro inicial If-Modified-Since (Se-Modificado-Desde), o processo-cliente não poderá baixar o arquivo se não

- 15 tiver sido modificado desde o último download.

O XML no arquivo de grupo utiliza o formato RSS com alguns elementos específicos do processo-cliente adicionais. RSS é um formato aberto para distribuição que é utilizado para muitos dos feeds** da Disney. A idéia é tentar

- 20 utilizar o máximo possível do padrão de forma que os arquivos de grupo possam ser potencialmente utilizados por outros sistemas.

A FIG. 3 mostra um DTD de exemplo para o arquivo de grupo.

Na inicialização, o processo-cliente se registra no sistema como um Servidor Local COM para os controles ActiveX seguintes. Somente ClientProcessGroup

- 25 (Grupo de Processo-Cliente) é marcado como seguro para inicialização e

somente ClientProcessLocator (Localizador de Processo-Cliente) está marcado como seguro para criação de scripts. Isto é para evitar a utilização dos controles para fins malignos. O protótipo inicial do ClientProcess (Processo-Cliente) foi projetado para ter uma interface ActiveX para cada componente principal para permitir que uma página da web exiba o status. Estas interfaces também são muito úteis para solução de problemas durante as fases de desenvolvimento/teste.

ClientProcessSvc (Svc Processo-Cliente):

Esta é a classe principal do aplicativo e está localizada na instalação da página da web com uma tag OBJECT para forçar o download e instalação do aplicativo. Há uma instância de classe singleton de CClientProcessSvc (Svc Processo-Cliente C) criada para o processo que rastreia todos objetos de recurso e de grupo criados. A interface IClientProcessSvc (Svc Processo-Cliente I) fornece métodos para reportar o status do programa.

ClientProcessGroup:

O controle ActiveX ClientProcessGroup é utilizado pelo JavaScript para atribuir um arquivo de configuração ao processo-cliente. Através da interface IPropertyBag (Bolsa de Propriedade I), o ClientProcessGroup aceita um parâmetro da tag OBJECT, chamado xml ou fonte, que é configurado para a URL do arquivo XML. Um único arquivo de configuração (ou grupo) XML é representado por um objeto ClientProcessGroup. A interface IClientProcessGroup fornece o status da funcionalidade.

ClientProcessLocator:

ClientProcessLocator é um controle simples que é seguro para criação de scripts e possui somente um método, GetLocalPath (Obter Caminho Local). Ele é

utilizado pelo JavaScript na página da web que exibe os arquivos de mídia para determinar o caminho do arquivo local, se tiver sido baixado.

Além disso, se o usuário parou o processo-cliente e depois navega pela página da mídia, a tag OBJECT contendo a identificação de classe ClientProcessLocator fará clientprocess.exe executar novamente. Se o usuário desinstalou o produto, esta tag forçará a reinstalar ou uma atualização se o parâmetro CODEBASE estiver especificado.

ClientProcessAsset (Recurso Processo-Cliente):

ClientProcessAsset representa qualquer objeto que deve ser armazenado localmente na cache (arquivo de grupo ou mídia). Este Objeto COM é devolvido pelo método get_asset (obter recurso) no ClientProcessSvc ou get_queuedAsset (Obter Recurso na Fila) no ClientProcessDownloader (Baixador de Processo-Cliente).

ClientProcessDownloader

Um ClientProcessDownloader singleton é criado quando o aplicativo inicia. Ele manipula a fila do download de recursos. Os recursos se adicionam à fila com o método QueueDownload (Download em Fila) e quando um recurso chega ao topo da fila de download, é retirado para a instância CAsyncCacheDownloader (Baixador da Cache Assíncrono C) para processamento.

ClientProcessReference (Referência Processo-Cliente)

Os objetos ClientProcessReference representam arquivos de mídia. Cada grupo pode ter um ou mais referências. Este objeto COM é devolvido pelo método get_reference (obter referência) no ClientProcessGroup.

Outras Classes de C++

Esta seção define classes de C++ de exemplo no aplicativo do processo-cliente e

resume seus comportamentos e interações uns com os outros. Algumas das classes possuem objetos wrapper COM para comunicação com outros aplicativos. A Fig. 4 apresenta uma tabela das várias classes.

CAsyncCacheDownloader – `asynccachedownloader.h/.cpp`:

- 5 Esta classe encapsula o download de recursos para o dispositivo local via HTTP. CAsyncCacheDownloader subclasses CWindowImpl para permitir que receba mensagens visto que a maior parte de suas operações ocorre em uma tarefa separada. O processo-cliente utiliza mensagens de janela para comunicação inter-tarefa.

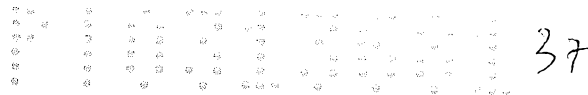
- 10 A maioria de seus métodos é executada na tarefa principal, mas o StartDownload() cria uma nova tarefa que por sua vez chama o método DoDownload(). O DoDownload() utiliza WinInet APIs para baixar um único recurso e criar uma entrada de cache do Internet Explorer para ele.

- O método DoDownload() baixará o arquivo em pequenos pedaços por vez. Ele também ficará inativo após cada pedaço se a produtividade configurada estiver abaixo de 100%. Por exemplo, se a produtividade for 50%, a tarefa de download ficará inativa pela mesma quantidade de tempo que levou para baixar o pedaço para atingir uma média de 50% de produtividade.

CCacheAsset – `cacheasset.h/.cpp`:

- 20 Uma instância CCacheAsset representa um único arquivo que necessita ser armazenado em cache no sistema local. Isto poderia ser tanto um arquivo de grupo quanto um arquivo de mídia (referência). Ambas instâncias CCacheGroup e CCacheReference instanciam uma instância CcacheAsset para manipular o download e o armazenamento em cache do arquivo. O método GetInterface devolve o ponteiro da interface ClientProcessAsset de seu respectivo objeto
- 25

wrapper COM.



Se o download não for bem sucedido, CCacheAsset programará um temporizador par atentar novamente de acordo com a tabela da Fig. 5. Se o download for bem sucedido, CCacheAsset assiste o sistema de arquivo para certificar-se de que o item de cache não foi removido da cache. Se foi, o item é baixado novamente.

CcacheGroup – cachegroup.h/.cpp:

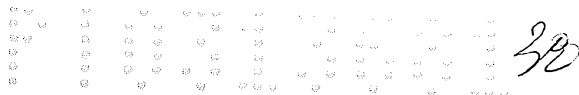
Esta classe encapsula o arquivo de grupo XML. Cada arquivo XML de Grupo é passado para um instância de CCacheGroup que o analisa sintaticamente e cria instâncias CCacheReference para cada item de arquivo de mídia. O método GetInterface devolve o ponteiro da interface ClientProcessGroup de seu respectivo objeto wrapper COM. Uma instância CCacheAsset é criada para representar este arquivo de grupo e é programado um temporizador com o intervalo configurado para o refreshRate especificado no XML. Quando o timer dispara, o CCacheAsset para o arquivo de grupo é adicionado à fila de download.

CCacheReference – cachereference.h/.cp:

Como o arquivo de grupo XML é analisado sintaticamente pelo CCacheGroup, uma instância CCacheReference é criado para cada item de arquivo de mídia. CCacheReference cria uma instância CCacheAsset para manipular aa funcionalidade de download/armazenamento em cache. Esta classe também é responsável pela verificação os tempos de validade do item e remoção dos itens do disco que expiraram.

CCacheTime – cachetime.h/.cpp:

CCacheTime classifica em subclasses a classe ATL Ctime para fornecer a funcionalidade de conversão de tempo. Todas variáveis membro data/hora no processo-cliente são armazenadas utilizando CCacheTime.



CClientProcessModule – clientprocess.cpp:

Esta classe representa o processo executável em si. CClientProcessModule classifica em subclasses a classe padrão ATL CAtlExeModuleT. O motivo para isto é adicionar funcionalidade à inicialização do processo, bem como ao registro e remoção do registro. Na inicialização, um mutex compartilhado é criado utilizando o ClientProcessSvc GUID como o nome (para garantir singularidade). Se o mutex já existe, isto significa que o aplicativo já está sendo executado, de forma que a instância atual sai. Toda funcionalidade extra de instalação e desinstalação também são realizadas nesta classe.

Instâncias de CException são criadas e lançadas quando ocorrem erros. CException manipula a formatação de um erro.

CRegistryVirtualDeviceX – registryvirtualdevicex.h:

Subclasse de CRegistryVirtualDevice em ATL. Esta classe sobrecarrega AddStandardReplacements para que o ATL pare de checar a extensão do módulo para determinar se devemos colocar InProcServer32 ou LocalServer32 no registro. O processo-cliente sempre é um .exe e, portanto, um LocalServer, de forma que o chamado pode ser desnecessário e adicionados requerimentos externos no aplicativo.

CTimer e CTimetable – timer.h/.cpp:

CTimer encapsula a funcionalidade temporizador. Qualquer objeto que queira utilizar um timer precisa somente herdar do CTimetable, implementar OnTimer, criar uma instância de CTimer e chamar seu método Start (Início). Stop() pára o temporizador.

CUrlMap – urlmap.h/.cpp:

CUrlMap é uma classe de modelo CAtlMap que permite que as classes de mapa

hash que mapeiem objetos para URLs sejam criadas.

Mais especificamente, o sistema ainda é melhorado pela: (i) adição do suporte da taxa de download (regulagem) ao código do download, (ii) removendo recursos quando eles expiram, (iii) implantando o controle ClientProcessLocator e removendo a lógica do cookie, (iv) utilizando um ícone na bandeja do sistema e menu, (v) fornecendo desinstalação (remover executável, remover atalho de inicialização, limpar registro), (vi) fornecer suporte HitCountURL, (vii) fornecer informações de programa (quase um diálogo).

Foi tomado o cuidado de garantir que: (i) todas configurações de segurança apropriadas do ActiveX sejam aplicadas a cada controle, (ii) os métodos e propriedades COM sejam escondidos (desta forma não aparecerão nas interfaces de usuário de projeto de controle como Visual Basic), (iii) a função estática seja utilizada para receber o sistema de arquivo de forma que somente um temporizador seja utilizado, (iv) a lógica para BottomLine seja adicionada para detectar ClientProcess e fornecer o submenu de ícone da Bandeja do Sistema, (v) o atalho do menu de inicialização é utilizado ao invés das janelas executarem a entrada de registro, (vi) regularmente, verificar o cookie que indica se o ClientProcess está instalado e o re-cria se necessário.

C(iii). Melhorias do BottomLine

BottomLine (Marca registrada da ESPN) pode precisar ser modificado para detectar o processo-cliente e, se estiver presente, exibir os itens de menu do processo-cliente em seu menu Bandeja do Sistema. Isto impede ter múltiplos ícones na bandeja se o usuário possuir ambos aplicativos instalados e em execução.

O processo-cliente definirá suas identificações de item de menu dentro do



40

intervalo CLIENTPROCESS_MENU_START e CLIENTPROCESS_MENU_END.

Estas constantes serão definidas em um arquivo de registro inicial que é acessível a BottomLine quando é criado.

BottomLine e o processo-cliente precisa saber se cada processo está sendo executado de forma que necessitarão realizar os seguintes passos: (a) quando BottomLine inicia, ele tenta criar o Mutex do processo-cliente para ver se o processo-cliente está em execução, (b) quando o processo-cliente inicia ou fecha, ele transmite uma mensagem de janela registrada que BottomLine vê e o BottomLine responderá com uma confirmação, (c) quando BottomLine inicia ou fecha, ele transmite uma mensagem de janela registrada que o processo-cliente vê e o processo-cliente responderá com uma confirmação.

As mensagens de janelas registradas podem permitir aplicativos para definir suas próprias mensagens exclusivas para comunicação com outras janelas que também registram a mesma mensagem. Quando o processo-cliente ou BottomLine anuncia sua presença a outro aplicativo ou confirma, ele passará sua manipulação de janela para comunicação futura. Depois o processo-cliente utilizará WM_COPYDATA para enviar ao BottomLine as informações a serem exibidas no menu. Quando um item de menu do processo-cliente for selecionado, BottomLine o encaminhará.

Alguns dos recursos também incluem o sistema que implanta o processo-cliente: (i) facilidade de instalação e atualização (automática), (ii) pouca ou nenhuma interface de usuário no cliente, (ii) configuração no servidor do programa do cliente, (iii) controle sobre a taxa em que os arquivos são baixados, (iv) capacidade de rastrear quantos arquivos de mídia são baixados, (v) capacidade de determinar o status do download de um arquivo de mídia via JavaScript, (vi)

capacidade de enviar uma versão do controle que é específica do provedor de conteúdo (p.ex., específica da ESPN) em sua nomeação.

C(iv). Recursos de Sistema

Programa do Processo-Cliente

- 5 O programa do processo-cliente é responsável por baixar horas de conteúdo de mídia antes de serem exibidas na página da web. Quando o usuário navega para uma página que contém o HTML para exibir aquele arquivo, a cópia local do arquivo pode ser utilizada, o que melhorará significativamente a qualidade da reprodução.

10 Instalação:

O programa-cliente pode ser permitido instalar automaticamente. Qualquer página da web pode conter uma tag referente a processo-cliente e o navegador o instalará automaticamente ou o atualizará, se necessário.

Interface de Usuário:

- 15 A interface de usuário para o programa-cliente pode ser integrada a um ícone da bandeja do sistema com um menu que provê a capacidade de desativar ou desinstalar o produto. Se o aplicativo BottomLine estiver presente, seu menu de bandeja do sistema existente poderá ser utilizado em seu lugar.

Execução:

- 20 O processo-cliente poderá ser instalado na pasta inicializar no dispositivo de usuário de forma que sempre estará executando. O programa poderá baixar regularmente um arquivo XML de um servidor WDIG que conterà informações sobre quais recursos baixar, quando expiram e em qual taxa baixá-los. Também haverá uma interface para os sites da web de provedor de conteúdo se
- 25 comunicarem com o processo-cliente para determinar a localização de arquivos



42

de mídia local.

Hospedagem:

O arquivo .cab de instalação poderá ser hospedado por uma ou mais centrais de distribuição fornecidas pelo Vertical que utiliza este produto. Visto ser um arquivo estático, ele pode ser hospedado por qualquer servidor da web. Conforme novas versões do processo-cliente são liberadas, este arquivo será atualizado no servidor.

Modificações do BottomLine do Provedor de Conteúdo:

Modificações compatíveis poderão ser realizadas no aplicativo BottomLine para fornecer a interface de bandeja do sistema para ao processo-cliente.

Amostra de HTML e Script:

Amostra de HTML e JavaScript poderão ser fornecidas aos desenvolvedores para demonstrar como hospedar e se comunicar com o processo-cliente.

Requerimentos da Interface Externa

Esta seção descreve os sistemas externos para os quais as interfaces precisarão ser criadas ou modificadas.

BottomLine da ESPN:

Algumas alterações menores poderão ser realizadas na BottomLine para detectar a presença do processo-cliente e adicionar itens de menu para parar e desinstalar o cliente.

Deteção do processo-cliente:

Cada vez que o menu BottomLine é exibido, o BottomLine precisar detectar a presença do processo-cliente e, se estiver presente e executando, exibir um submenu contendo os itens de menu do processo-cliente. A deteção do processo-cliente é facilmente realizada pela checagem da existência de um Mutex



63

que o processo-cliente cria na inicialização ou pela procura por uma janela que possui a classe de janela processo-cliente.

Comunicação do processo-cliente:

Se o processo-cliente estiver executando, BottomLine poderá utilizar mensagens de janelas para consultar quais itens de menu exibir. Quando um item de menu do processo-cliente é selecionado, BottomLine utilizará mensagens para dizer ao processo-cliente qual item foi selecionado.

Servidor de Conteúdo de Mídia:

O servidor que hospeda os arquivos de mídia poderão ser qualquer servidor de web HTTP 1.1 que deve suportar baixar intervalo de byte HTTP (206 códigos de sucesso) para possibilitar regular o download. Se o servidor não suporta intervalos de byte, o arquivo inteiro poderá ser baixado de uma vez, independente dos ajustes de configuração.

Interface de Servidor de Configuração:

Poderia haver um servidor que hospeda o arquivo de configuração XML. O XML poderá aderir ao um formato que seja publicado nas especificações do projeto.

Requerimentos Funcionais

1. Configurar um cookie no dispositivo de usuário que indique que o programa está instalado. Este cookie pode ser periodicamente verificado para certificar-se de que o usuário não o removeu.

2. Atualizações do software de suporte via recurso Download Component (Componente de Download) do Internet Explorer garantirá que o código do cliente seja devidamente atualizado se for colocada uma nova versão na central de distribuição e a página HTML for modificada para solicitar uma versão mais recente. E também, pode-se garantir que nenhuma reinicialização seja requerida



44

na atualização.

3. O programa poderá ser instalado na pasta inicializar e, portanto, sempre estará executando a menos que o usuário a pare especificamente.

4. Somente uma instância do programa será executada por vez.

5 5. A desinstalação será fornecida através de um menu no ícone na bandeja do sistema. A desinstalação poderá ser iniciada fora do programa.

6. Itens de menu fornecidos por um ícone da bandeja do sistema.

7. Ajuda – exibe a ajuda para a Cache da Disney.

10 8. O arquivo de configuração pode ter um item helpURL opcional. Quando o usuário selecionar Ajuda, abrirá um navegador com esta URL.

9. Sobre – mostra o diálogo com os direitos autorais/informações do programa.

10. Saída – pára o programa (não o remove da pasta inicializar).

11. Desinstalar – desinstala o programa e remove as entradas de registro.

15 12. O arquivo de configuração: (i) poderá ser no Formato XML, (ii) poderá fornecer a taxa de atualização para determinar com que freqüência para baixar o arquivo de configuração, (iii) poderá fornecer ajustes por item para a URL, data de validade, URL para rastrear o clique e a taxa de download. Além disso, a prioridade de um item poderá ser determinada pela ordem no arquivo de configuração. E também, dois arquivos de configuração diferentes não poderão
20 conter o mesmo item com a mesma URL. Além disso, as informações sobre a localização do arquivo de configuração XML pode ser localizado no registro sob HKEY_CURRENT_USER.

Download de Arquivo

Cache do Navegador (p.ex., Internet Explorer):

25 O processo-cliente pode utilizar a cache do Internet Explorer para fornecer o

armazenamento em cache dos arquivos de mídia no cliente. Os arquivos armazenados na cache podem ser removidos pelo Internet Explorer, se necessário, criar espaço para outros itens. O arquivo de configuração XML, porém, poderá ser marcado com uma varinha para que permaneça no disco rígido por um certo período de tempo (p.ex., 30 dias). Visto a cache do Internet Explorer ser à base de login do Windows, cada usuário pode ter uma cache separada.

O processo-cliente precisará verificar periodicamente (ou utilizar um observador do Sistema de Arquivo [File System watch]) para detectar quando um item foi removido da cache. Se ainda não expirou, será baixado novamente.

Lifespan do item:

Cada item terá uma data e hora de validade. Periodicamente, o processo-cliente verificará estas horas e apagará qualquer item da cache que tiver expirado. O processo-cliente precisa certificar-se de rastrear recursos que existam no dispositivo de forma que se forem removidos do arquivo de configuração XML antes do cliente ter uma chance de expirá-los, serão apagados do disco rígido do usuário (isto pode ocorrer se o usuário teve seu dispositivo desligado por um longo período de tempo).

Uma vez baixado um item, ele não será baixado novamente a menos que o Internet Explorer o tenha removido da cache.

Atualização de Configuração:

O processo-cliente atualizará periodicamente o arquivo de configuração XML baixando-o do servidor à taxa especificada pelo elemento SML refreshRate.

Erros de Download:

Se ocorrer um erro quando um item estiver sendo baixado, após um intervalo, o processo-cliente tentará novamente. A quantidade de tempo de espera depende

46

do tipo de erro de acordo com o quadro fornecido na FIG. 6.

Além disso, o processo-cliente poderá realizar downloads se a conectividade da rede estiver disponível. Se necessário, poderá não realizar discagem para realizar os downloads.

5 Regulagem:

Cada item no arquivo de configuração pode opcionalmente ser configurado para ter uma porcentagem específica de produtividade do download. Isto implica no processo-cliente poder baixar o arquivo em partes e pausá-lo entre cada parte para conseguir isto. O servidor da web que hospeda o arquivo deve ser capaz de suportar o recurso intervalo de byte HTTP.

Interface da Web

A URL para o arquivo de configuração poderá ser passado para o processo-cliente por meio de uma propriedade do controle ActiveX. O controle ActiveX será seguro para criação de scripts e seguro para inicialização. O processo-cliente poderá verificar a parte do servidor da URL de configuração do XML para verificar se vem do domínio go.com. Isto é para impedir que páginas da web mal-intencionadas inicializem o processo-cliente com dados potencialmente prejudiciais (arquivos grandes baixados freqüentemente com pequenas taxas de atualização). A tag OBJECT não precisa ter um parâmetro CODEBASE de forma que o processo-cliente não é instalado acidentalmente para alguém que não optou por isso. O JavaScript poderá ter código para verificar se o objeto existe antes de fazer qualquer chamada API e verificar também o cookie instalado. Pelo menos uma versão do processo-cliente pode ser instalada no dispositivo do cliente.

25 D. Geral

A descrição das variantes de exemplo e antecipadas da invenção foi apresentada para fins de ilustração e descrição. Não pretendem ser exaustivas ou limitarem a invenção às formas precisas divulgadas.

Muitas modificações e variações são possíveis à luz dos ensinamentos deste. Por exemplo:

(a) Poderá ser possível criar um novo componente que forneça suporte ao ícone da Bandeja do Sistema onde os apps cliente instalados podem se registrar para adicionar itens de menu.

(b) Ajustes do usuário no servidor para determinar o comportamento do processo-cliente. O processo-cliente precisaria de um identificador exclusivo vinculado ao usuário que o instalou. Depois o servidor poderia ser mais sofisticado e enviar diferentes arquivos de configuração XML para usuários diferentes.

(c) Continuação de downloads que não terminaram.

(d) Adicionar regulagens de downloads mais sofisticadas (por exemplo, considerar a medição da largura de banda do cliente como base e utilizá-la para calcular o controle de produtividade).

(e) Smarts no servidor para determinar a carga do servidor baseado nos dados MRTG e utilizá-los para decidir devolver as 503 respostas ao processo-cliente para impedir uma carga pesada no download do recurso.

É evidente que os mais hábeis na invenção poderão agora fazer inúmeros usos e modificação dela, derivadas das variantes específicas aqui descritas sem derivar dos conceitos inventivos. Conseqüentemente, a invenção deve ser interpretada como compreendendo todo e qualquer novo recurso e nova combinação de recursos presentes na mesma ou possuídos pelo dispositivo e métodos aqui revelados e limitados unicamente pelo espírito e objetivo das reivindicações

anexas.

48

REIVINDICAÇÕES

1. Método, compreendendo:

fornecimento de uma lista de ativos de mídia (62) por um fornecedor de conteúdos (40) sobre uma rede para um dispositivo de usuário (20);

distribuição para o referido dispositivo de usuário (20) de selecionados ativos de mídia (62) a partir da referida lista multimídia ativa (42), a partir do fornecedor de conteúdos (40) sobre uma rede para um dispositivo de usuário para armazenamento para prevenir uma memória intermédia em tempo real dos referidos ativos de mídia (62) selecionado para exibição, quando uma restrição predeterminada é satisfeita;

entregar separadamente ao referido dispositivo de usuário (20) um fluxo de conteúdo em tempo real que requer um armazenamento em armazenamento intermediário em tempo real para exibição, depois de entregar completamente o dito recurso de multimídia selecionado (62) ao dito dispositivo de usuário (20);

caracterizado pelo fato de que a entrega do referido fluxo de conteúdo em tempo real requer uma largura de banda de rede mais baixa do que a entrega do referido recurso de multimídia selecionado (62), para facilitar a integração do referido recurso de multimídia selecionado (62) com o referido fluxo de conteúdo em tempo real para formar uma página do navegador da web.

2. Método, de acordo com a reivindicação 1, **caracterizado** pelo dispositivo de usuário (20) selecionar dito recurso de multimídia selecionado (62).

3. Método, de acordo com a reivindicação 1, **caracterizado** pela referida restrição predeterminada incluir uma disponibilidade de largura de banda da rede.

4. Método, de acordo com a reivindicação 1, **caracterizado** pela referida restrição predeterminada incluir uma qualidade de serviço assegurada.

5. Método, de acordo com a reivindicação 1, **caracterizado** pela referida

restrição predeterminada incluir o dispositivo de usuário (20) estando ocioso.

6. Método, de acordo com a reivindicação 1, **caracterizado** pela referida restrição predeterminada incluir um nível de atividade da rede.

7. Método, de acordo com a reivindicação 1, **caracterizado** pela referida restrição predeterminada incluir um nível de utilização de memória do dispositivo de usuário (20).

8. Método, de acordo com a reivindicação 1, **caracterizado** pela referida restrição predeterminada incluir um nível de utilização de CPU do dispositivo de usuário (20).

9. Provedor de conteúdo (40), para distribuir um recurso de multimídia selecionado (62) através de uma rede para ser combinado com um fluxo de conteúdo em tempo real, compreendendo o referido fornecedor de conteúdo:

uma memória; e

uma CPU configurada para:

fornecer a partir de dito provedor de conteúdo (40) uma lista multimídia ativa (42) através da rede para um dispositivo de usuário (20);

entregar ao referido dispositivo de usuário (20) um recurso de multimídia selecionado (62) a partir da referida lista multimídia ativa (42) para armazenamento para prevenir um armazenamento de memória em tempo real do referido recurso de multimídia selecionado (62) para exibição, a partir do referido provedor de conteúdo (40) através da referida rede, se uma restrição pré-determinada está satisfeita;

entregar separadamente ao referido dispositivo de usuário (20) um fluxo de conteúdo em tempo real que requer um armazenamento intermediário (armazenamento intermediário ing) em tempo real para exibição,

depois de entregar completamente o referido recurso de multimídia selecionado (62) para o referido dispositivo de usuário (20);

caracterizado pelo fato de que a distribuição do referido fluxo de conteúdo em tempo real requer uma largura de banda de rede mais baixa do que a entrega do referido ativo de meio de recurso de multimídia selecionado (62), facilitando assim uma integração do referido ativo de meios de recurso de multimídia selecionado (62) com o referido fluxo de conteúdo em tempo real para formar um conteúdo integrado para exibição em uma página do navegador da web.

10. Provedor de conteúdo (40), de acordo com a reivindicação 9, **caracterizado** pelo fato de dito dispositivo de usuário (20) selecionar dito recurso de multimídia selecionado (62).

11. Provedor de conteúdo (40), de acordo com a reivindicação 9, **caracterizado** pelo fato de uma restrição predeterminada incluir uma disponibilidade de largura de banda da rede.

12. Provedor de conteúdo (40), de acordo com a reivindicação 9, **caracterizado** pelo fato de uma restrição predeterminada incluir uma qualidade de serviço.

13. Provedor de conteúdo (40), de acordo com a reivindicação 9, **caracterizado** pelo fato de uma restrição predeterminada incluir o dispositivo de usuário (20) a ser ocioso.

14. Provedor de conteúdo (40), de acordo com a reivindicação 9, **caracterizado** pelo fato de uma restrição predeterminada incluir um nível de atividade da rede.

15. Provedor de conteúdo (40), de acordo com a reivindicação 9, **caracterizado** pelo fato de uma restrição predeterminada incluir um nível de utilização de memória do dispositivo de usuário.

16. Provedor de conteúdo (40), de acordo com a reivindicação 9, com uma restrição predeterminada incluir um nível de utilização de CPU do dispositivo de utilizador (20).

17. Método para utilização por um dispositivo de usuário (20), dito método compreendendo:

receber, por meio do dispositivo de usuário (20) de um provedor de conteúdo (40), uma lista multimídia ativa (42);

transmitir um recurso de multimídia selecionado (62) a partir da referida lista multimídia ativa (42) para o provedor de conteúdo (40/60);

receber, pelo dispositivo de usuário (20) a partir do provedor de conteúdo (40/60), o recurso de multimídia selecionado (62) se for satisfeita uma restrição predeterminada;

armazenar o recurso de multimídia selecionado (62) em uma memória do dispositivo de usuário (20) para prevenir um armazenamento intermediário em tempo real do recurso de mídia selecionado (62) para exibição;

receber separadamente, pelo dispositivo de usuário (20) a partir do provedor de conteúdo (40/60), um fluxo de conteúdo em tempo real que requer armazenamento intermediário (armazenamento intermediário ing) em tempo real para exibição, após o armazenamento completo do referido recurso de mídia selecionado (62) selecionado no dispositivo de usuário (20), **caracterizado** pelo fato de que o fluxo de conteúdo em tempo real requer uma largura de banda de rede mais baixa do que a entrega do recurso de mídia selecionado (62);

integrar o recurso de mídia selecionado (62) com o fluxo de conteúdo em tempo real para formar um conteúdo integrado para exibição em uma página de navegador web; e

exibir o conteúdo integrado da página do navegador da web.

18. Método, de acordo com a reivindicação 17, **caracterizado** pelo fato de uma restrição predeterminada incluir um nível de atividade da rede.
19. Método, de acordo com a reivindicação 17, **caracterizado** pelo fato de uma restrição predeterminada incluir um nível de utilização de memória do dispositivo de usuário.
20. Método, de acordo com a reivindicação 17, **caracterizado** pelo fato de uma restrição predeterminada incluir um nível de utilização de CPU do dispositivo de usuário (20).

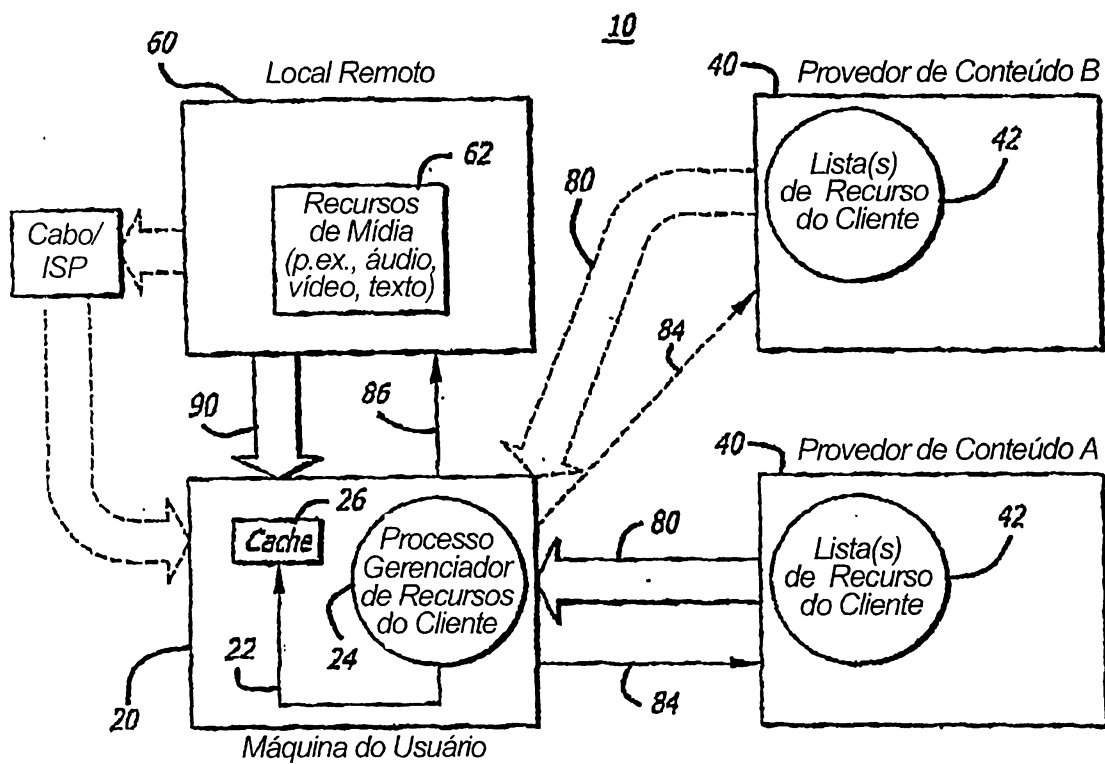
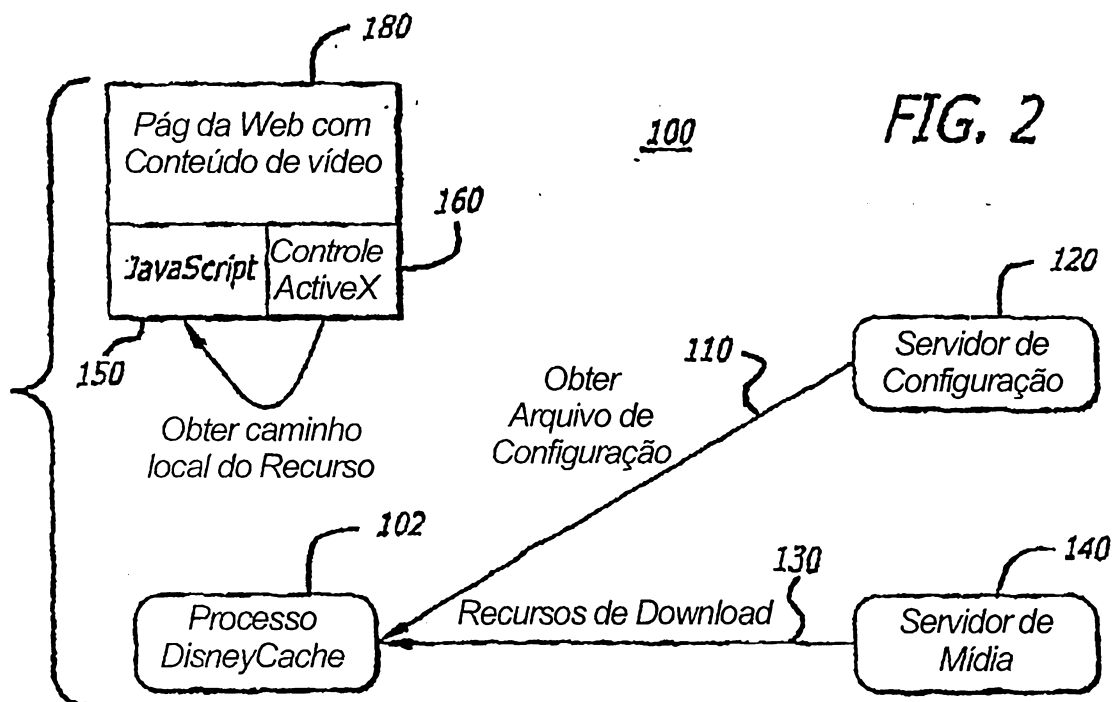


FIG. 1



```

<|ELEMENT channel (Item+ | refreshRate | helpUrl? | cookieDomain? |
cookieName?)>
<|ELEMENT item (link | expires | hitCountUrl? | throughput? | trackWithCookie?
)*>

```

A seguir está um exemplo de arquivo de grupo XML:

```

<?xml version="1.0" ?>
<channel>
  <item>
    <link>http://myserver/myfile.wmv</link>
    <expires>2002-22-03T00:17:15z</expires>
    <hitCountUrl>http://hitserver/hiturl</hitCountUrl>
    <throughput>75</throughput>
    <trackWithCookie/>
  </item>
  <refreshRate>60</refreshRate>
  <helpUrl>http://myserver/help.html</helpURL>
  <cookieName>adAssets</cookieName>
  <cookieDomain>.go.com</cookieDomain>
</channel>

```

FIG. 3

Nome da Classe Interna	Nome da Classe COM/ActiveX
CCacheAsset	CDisCacheAsset
CCacheGroup	CDisCacheGroup
CCacheReference	CDisCacheReference

FIG. 4

Tipo de Erro	Tentativas Realizadas	Intervalo Antes de Tentar Novamente
Invalid Item URL (URL de item inválida)	não	N/D
Server Connection (Conexão com servidor)	sim	5 minutos
Server not found (Servidor não encontrado)	sim	5 minutos
File not on server (arquivo não se encontra no servidor)	sim	15 minutos
Unknown server error (Erro de servidor desconhecido)	sim	15 minutos
Internal Error (Erro Interno)	sim	1 hora

FIG. 5

Tipo de Erro	Tentativas Realizadas	Intervalo Antes de Tentar Novamente
Invalid Item URL (URL de item inválida)	não	N/D
Server Connection (Conexão com servidor)	sim	5 minutos
Server not found (Servidor não encontrado)	sim	1 hora
File not on server (arquivo não se encontra no servidor)	sim	1 hora
Unknown server error (Erro de servidor desconhecido)	sim	15 minutos
Internal Error (Erro Interno)	sim	1 hora

FIG. 6

Asset List (Lista de Recursos)

Asset 1: {Attribute(1,1); Attibute (1,2);...;Attribute (1,M)}
Asset 2: {Attribute(2,1); Attibute (2,2);...;Attribute (2,M)}
•
•
•
Asset N: {Attribute(N,1); Attibute (N,2);...;Attribute (N,M)}

FIG. 7

Asset List {Attribute 1; Attribute2;...;Attribute N}

Asset 1
Asset 2
•
•
•
Asset N

FIG. 8

RESUMO

“SISTEMA DE ENTREGA E APRESENTAÇÃO DINÂMICA DE RECURSOS DE MÍDIA DE GRANDE PORTE EM REDES RESTRITAS POR LARGURA DE BANDA”. Conteúdo de mídia, baseado em um conjunto pré-determinado de restrições, a partir de um provedor de conteúdo é entregue a uma cache local de um dispositivo do usuário antes de visualizar a mídia. O processo gerenciador de recurso do cliente residindo no dispositivo do usuário, a lista de recursos no site do provedor de conteúdo, e os recursos de mídia estão localizados em um local remoto.