República Federativa do Brasil
Ministério da Indústria, Comércio Exterior
e Serviços
Instituto Nacional da Propriedade Industrial

(11) PI 0520186-1 B1

(22) Data do Depósito: 25/08/2005

(45) Data de Concessão: 21/11/2017

(54) Título: MÉTODO IMPLEMENTADO EM COMPUTADOR E SISTEMA PARA EXIBIR UMA JANELA EM UM DISPOSITIVO DE VÍDEO

(51) Int.Cl.: G06F 3/00; G06F 13/00

(30) Prioridade Unionista: 22/04/2005 US 11/111.983

(73) Titular(es): MICROSOFT TECHNOLOGY LICENSING, LLC

(72) Inventor(es): GREG SCHECHTER; JEVAN SAKS; CHRIS FORTIER

Relatório Descritivo da Patente de Invenção para **"MÉTODO IMPLEMENTADO EM COMPUTADOR E SISTEMA PARA EXIBIR UMA JANELA EM UM DISPOSITIVO DE VÍDEO".**

CAMPO DA INVENÇÃO

[001] Os aspectos da presente invenção referem-se a miniaturas de janelas e, em particular, a uma interface e sistema de manipulação de miniaturas em um gerenciador de janelas.

FUNDAMENTOS DA INVENÇÃO

[002] Os sistemas de computador exibem uma variedade de janelas nas quais os usuários podem entrar dados, manipular dados ou ativar procedimentos. Ao executar mais de uma aplicação simultaneamente, diversas janelas podem ser exibidas em um vídeo de computador, cada janela sendo correspondente a uma aplicação. Pode haver também múltiplas janelas para qualquer uma aplicação. Por exemplo, quando um usuário entra um texto em um programa de processamento de texto enquanto também trabalha em um programa de planilhas, pode haver duas janelas abertas no vídeo. Uma das janelas é a janela da aplicação de processamento de texto e a segunda janela é a janela de programa de planilhas. Quando o usuário está também assistindo a um vídeo em uma aplicação de tocador de mídia (media player), haverá uma janela correspondente adicional para a aplicação de tocador de mídia. Conforme o número de aplicações ativas aumenta, assim também aumentará o número de janelas exibidas no vídeo do computador.

[003] Um usuário que utiliza múltiplas aplicações simultaneamente, com freqüência, fica frente a frente com uma multidão de janelas no vídeo, fazendo com que a área de trabalho amontoada leve a uma confusão como também a uma frustração devido à desordem observada no vídeo. Por exemplo, pode haver tantas janelas sobrepostas no vídeo que um usuário terá de desperdiçar tempo para encontrar

uma janela desejada cada vez que ele quiser completar uma tarefa em uma aplicação diferente. Para amenizar este problema, o usuário poderá minimizar janelas ou fechar completamente algumas aplicações a fim de diminuir a desordem (e o número de janelas no vídeo). No entanto, quando as janelas são minimizadas, o usuário não mais terá acesso imediato à aplicação correspondente. Quando uma ação precisa ser feita em uma aplicação em particular, na qual a janela foi minimizada, o usuário deverá primeiro localizar a janela desejada e abrir a janela desejada depois de localizar a mesma. Este processo é muito demorado. Da mesma forma,quando o usuário fecha totalmente a aplicação para limpar a desordem na área de trabalho, a aplicação não mais ficará ativa. Neste caso, o usuário deverá relançar a aplicação a fim de completar a tarefa desejada e gastar mais tempo ainda com isso. Ainda, quando a aplicação realiza uma função em desenvolvimento, esta função se perde durante o tempo que a aplicação não se encontrava ativa.

[004] Tipicamente, quando uma janela de aplicação é minimizada, um botão de barra de tarefa pode ser exibido em uma barra de tarefas a fim de indicar que a aplicação se encontra ativa. Embora o botão da barra de tarefas possa indicar que a aplicação está ativa, com freqüência só há um ícone ou um nome da aplicação no botão de barra de tarefa. Sem informação adicional, o usuário teria de abrir a janela a fim de visualizar os conteúdos da janela. No entanto, ao abrir e ao fechar a janela simplesmente para verificar os conteúdos da janela (por exemplo, certificar-se da identidade da janela), o usuário perde tempo e esforço, o que proporciona uma perda de eficiência e produtividade. Este problema se completa quando o número de aplicações ativas e o número de janelas aumentam conforme o usuário deve abrir e fechar muitas janelas a fim de encontrar a janela desejada.

[005] Sendo assim, existe a necessidade na técnica por um sis-

tema e método para a provisão de um acesso programático conveniente no sentido de manipular janelas ou miniaturas e expressar as miniaturas de maneira a prover flexibilidade na exibição das miniaturas ou do conteúdo das miniaturas. Ainda, existe a necessidade na técnica por uma interface de programação e de um método de implementação da interface de programação de modo a efetivamente gerenciar e implementar janelas de aplicação e miniaturas, e expressar as janelas e as miniaturas de uma maneira flexível.

SUMÁRIO DA INVENÇÃO

[006] Os aspectos da presente invenção provêm uma interface, um sistema ou método para a exibição de uma janela em um vídeo, deste modo tratando de uma ou mais questões acima descritas. Uma miniatura dinâmica correspondente a uma janela de aplicação é ainda provida, na qual as modificações de conteúdo de uma janela de aplicação são refletidas na miniatura dinâmica correspondente. Uma modificação de conteúdo da janela de aplicação pode ainda ser refletida na miniatura dinâmica correspondente em tempo real.

BREVE DESCRIÇÃO DOS DESENHOS

[007] A Figura 1A ilustra um exemplo de um sistema para a implementação da presente invenção, incluindo um dispositivo computacional de uso geral na forma de um computador.

[008] As Figuras 1B a 1M mostram um ambiente de computador de uso geral que suporta um ou mais aspectos da presente invenção.

[009] A Figura 2A é um gráfico de cena ou um diagrama em árvore de vídeos ilustrando um exemplo de componentes de um vídeo.

[0010] A Figura 2B demonstra os componentes de um vídeo de amostra correspondente ao da Figura 2A.

[0011] A Figura 3A é um gráfico de cena ou um diagrama em árvore de vídeos ilustrando um exemplo de componentes de um vídeo da presente invenção.

[0012] A Figura 3B demonstra o vídeo correspondente ao da Figura 3A.

[0013] As Figuras 4A e 4B demonstram um exemplo de uma aplicação da presente invenção na provisão de miniaturas de janelas de aplicação.

[0014] A Figura 5 ilustra uma outra aplicação de uma miniatura da presente invenção na qual miniaturas são exibidas em um menu de ALT-TAB.

[0015] A Figura 6 é um fluxograma ilustrando um exemplo de um processo para a provisão de miniaturas.

DESCRIÇÃO DETALHADA DA INVENÇÃO

[0016] O presente documento é dividido em seções a fim de ajudar o leitor. Estas seções incluem: o resumo introdutório, o ambiente do computador de uso geral, e as miniaturas.

[0017] Observa-se que são apresentadas várias conexões entre os elementos na descrição a seguir. Nota-se que estas conexões em geral, e, a menos que de outra forma especificado, podem ser diretas ou indiretas, e que o presente relatório descritivo não pretende ser limitante a este respeito.

RESUMO INTRODUTÓRIO

A) Miniaturas Estáticas

[0018] Em uma abordagem, uma miniatura pode ser associada à aplicação. Com uma miniatura, uma pequena versão da janela de aplicação pode ser exibida de modo a indicar o conteúdo original da janela de aplicação, permitindo ao usuário identificar o conteúdo aproximado da janela. No entanto, estas miniaturas podem não prover o conteúdo atualizado da janela de aplicação. Sendo assim, caso tenham sido feitas modificações à janela de aplicação, a miniatura pode estar desatualizada e poderá não prover a imagem apropriada do conteúdo em questão da janela de aplicação. Deste modo, fica difícil para o usuário

identificar de maneira apropriada a janela de aplicação desejada sem ter de abrir a própria janela de aplicação.

[0019] Ainda, as miniaturas perdem em flexibilidade em termos de exibição de conteúdo, colocação, dimensionamento e/ou efeitos. Uma miniatura de uma janela de aplicação provê uma imagem estática do conteúdo original da janela de aplicação. No entanto, não apenas o conteúdo permanece estático e potencialmente desatualizado, como também as características e a colocação da miniatura podem ser pre-determinadas e podem não variar com base nas necessidades do usuário em um dado momento.

[0020] Os sistemas operacionais de computador empregam uma camada de software responsável pelo gerenciamento dos objetos de interface com o usuário, tais como, miniaturas, e pela provisão dos serviços de interface com o usuário às aplicações de software. A camada de software pode ser referida como o Gerenciador de Janela de Área de Trabalho (DWM). A lógica da renderização, o roteamento de eventos de entrada, e as interfaces de programação de aplicação (API) do Gerenciador de Janela de Área de Trabalho coletivamente incorporam a política de interface, que, por sua vez, define a experiência do usuário geral do sistema operacional. Ao renderizar a área de trabalho, o gerenciador DWM tipicamente processa atualizações de tela, tais como a movimentação das janelas, por meio de uma pintura coordenada diretamente sobre a tela de vídeo. Deste modo, quando uma janela sobreposta é fechada ou movimentada para longe, a janela ou área de trabalho subjacente recém exibida deve ser repintada. Este é um processo lento no qual tempo e recursos são desnecessariamente desperdiçados. Ainda, quando as janelas são pintadas diretamente em uma posição em particular e em uma configuração particular no vídeo, como é tipicamente feito, a imagem correspondente da janela é definida naquela posição e configuração particulares. As modificações

à posição ou à configuração da janela requereriam um grande gasto de recursos para repintar a imagem.

B) Miniaturas Dinâmicas

[0021] Em um outro aspecto da presente invenção, uma interface, um sistema e método são providos para a exibição de uma miniatura em um vídeo no qual um Gerenciador de Janela de Área de Trabalho (DWM) provê uma Interface de Programação de Aplicação (API) que registra uma janela de origem e uma janela de destino. Neste exemplo, a janela de destino é uma janela na qual uma miniatura pode ser renderizada. A miniatura renderizada dentro da janela de destino pode representar pelo menos uma porção de uma janela de origem correspondente. Deste modo, a miniatura pode representar uma associação entre a janela de origem e a janela de destino ao se prover pelo menos uma porção da janela de origem conduzida para pelo menos uma porção da janela de destino. O gerenciador DWM pode receber uma alça (handle) de janela (window) (HWND) correspondente à janela de origem e uma alça HWND correspondente à janela de destino. A aplicação pode ainda receber uma alça representando o registro da miniatura correspondente à janela de origem.

[0022] Em um outro aspecto da presente invenção, uma interface, um sistema e método são providos para a exibição de uma mínimo em um vídeo no qual um Gerenciador de Janela de Área de Trabalho (DWM) atualiza ou modifica uma miniatura associada a uma janela de origem correspondente. Neste exemplo, o gerenciador DWM pode receber dados que definem o dimensionamento de uma janela de destino, uma região de uma janela de origem para desenhar em uma janela de destino como uma miniatura, a opacidade de uma miniatura, a visibilidade de uma miniatura, os efeitos especiais de uma miniatura, etc.

[0023] Um outro aspecto da presente invenção provê ainda uma miniatura dinâmica compreendendo a exportação de uma interface de

programação de aplicação para o registro de uma alça de miniatura, a manutenção de uma lista de registros de miniaturas, o desenho de uma miniatura ou janela de destino com base em uma janela de origem, em que a janela de origem pode ser uma janela de aplicação, a atualização das propriedades de uma miniatura, e o não registro da miniatura dinâmica. Além disso, a modificação de uma janela de origem pode ser alceada automaticamente de modo a realizar uma modificação correspondente em uma janela de destino.

[0024] Nota-se que várias conexões são apresentadas entre elementos na descrição a seguir. Nota-se que estas conexões em geral, a menos que especificado de outra forma, podem ser diretas ou indiretas, e que o presente relatório descritivo não pretende ser limitante a este respeito.

[0025] Os aspectos da presente invenção provêm uma interface de programação de aplicação (API), um sistema e método para o controle da imagem dos elementos de vídeo, como, por exemplo, janelas ou outra representação de janelas, tais como, miniaturas em um vídeo. Em um exemplo da presente invenção, uma versão alternativa de uma janela de aplicação é uma versão menor da janela de aplicação e é exibida no vídeo. A versão menor da janela de aplicação pode ser ainda exibida em um local no vídeo diferente da janela de aplicação original. De maneira alternativa, a janela de aplicação original pode ser minimizada de tal modo que a janela de aplicação original não seja exibida enquanto a versão menor da janela de aplicação é exibida no vídeo. Ainda, a versão menor da janela de aplicação pode ela mesma ser minimizada, por exemplo, quando um espaço adicional no vídeo é desejado. Embora a versão alternativa da janela de aplicação possa ser uma versão menor da aplicação conforme descrita, nota-se que a versão alternativa da janela de aplicação pode ser maior que a janela de aplicação original ou qualquer outro dimensionamento dependendo

das necessidades do usuário.

[0026] Por exemplo, a versão alternativa da janela de aplicação pode ser uma miniatura da janela de aplicação. Uma miniatura é tipicamente uma imagem pequena ou representação de um gráfico maior, ou uma parte de uma imagem maior, correspondente à janela de aplicação. A miniatura pode também representar uma associação entre uma janela de origem, que provê a origem ou o conteúdo da miniatura, e uma janela de destino que pode ser uma janela na qual a miniatura provê o conteúdo da janela de origem. As miniaturas oferecem a vantagem da capacidade de exibir diversas janelas de uma só vez em um formato conveniente, ao mesmo tempo minimizando o uso de espaço no vídeo. Ainda, as miniaturas podem ser usadas na seleção e na classificação de imagens e janelas de uma forma mais fácil.

[0027] Em um outro aspecto da presente invenção, a miniatura ou versão alternativa da janela de aplicação pode ser exibida no vídeo quando solicitado. Por exemplo, ao se pairar um cursor sobre um botão de barra de tarefas, pode-se invocar a exibição de uma miniatura da janela de aplicação. Em um exemplo, a miniatura é uma miniatura dinâmica e pode ser modificada conforme a janela de aplicação original se modifica. Por exemplo, as mudanças feitas à janela de aplicação se refletem na miniatura dinâmica correspondente à janela de aplicação. Tais mudanças podem se refletir na miniatura dinâmica em tempo real, por exemplo.

[0028] Em um outro aspecto da presente invenção, as miniaturas ou as miniaturas dinâmicas podem ser dispostas dentro de uma outra janela, ou janela de destino. Por exemplo, múltiplas miniaturas dinâmicas de diferentes aplicações podem ser incluídas em uma janela de destino de tal modo que a janela de destino exiba as miniaturas de cada uma das diferentes aplicações. Ainda, quaisquer miniaturas dinâmicas exibidas na janela de destino podem refletir a condição atual da

janela de aplicação correspondente naquele momento. De maneira alternativa, uma miniatura pode ser exibida em uma outra miniatura ou as miniaturas podem ser dispostas em cascata de tal modo que uma primeira miniatura possa ser exibida em uma segunda miniatura que, por sua vez, pode ser exibida em uma terceira miniatura, e assim por diante.

[0029] Em um outro aspecto da presente invenção ainda, um sistema e método são providos de modo a preparar janelas fora da tela (off-screen). A composição das janelas pode em seguida ser feita na tela (on-screen), economizando, desta maneira, os recursos do computador. Ainda, a preparação das janelas fora da tela e a composição das janelas na tela possibilitam a funcionalidade das miniaturas, conforme descrito na presente invenção.

AMBIENTE COMPUTACIONAL DE USO GERAL

[0030] A Figura 1A ilustra um exemplo de um ambiente de sistema computacional adequado 100 no qual a presente invenção pode ser implementada. O ambiente de sistema computacional 100 é apenas um exemplo de um ambiente computacional adequado e não pretende sugerir nenhuma limitação quanto ao âmbito de uso ou funcionalidade da presente invenção. Nem deve o ambiente computacional 100 ser interpretado como tendo qualquer dependência ou exigência com relação a nenhum componente ou combinação de componentes ilustrados no ambiente operacional exemplar 100.

[0031] A presente invenção é operacional com inúmeros ambientes ou configurações de sistema computacional de uso geral ou de uso especial. Exemplos de sistemas de computação, ambientes e/ou configurações bem conhecidos que podem ser adequados para uso com a presente invenção incluem, porém não se limitam a, computadores pessoais, computadores servidores, dispositivos de computação do tipo laptop ou portáteis, sistemas multiprocessadores, sistemas base-

ados em microprocessadores, equipamentos eletrônicos programáveis pelo consumidor, PC de rede, minicomputadores, computadores de grande porte, ambientes computacionais distribuídos que incluem qualquer um dos sistemas ou dispositivos acima, ou coisa do gênero.

[0032] Com referência à Figura 1A, um sistema ilustrativo para a implementação da presente invenção inclui um dispositivo computacional de uso geral na forma de um computador 110. Os componentes do computador 110 podem incluir, sem, no entanto, se limitar a, uma unidade de processamento 120, uma memória de sistema 130, e a um barramento de sistema 121 que acopla os vários componentes de sistema, incluindo a memória de sistema à unidade de processamento 120. O barramento de sistema 121 pode ser qualquer um dentre os diversos tipos de estruturas de barramento, incluindo um barramento de memória ou uma controladora de memória, um barramento periférico, ou um barramento local que usa qualquer uma dentre uma variedade de arquiteturas de barramento. À guisa de exemplo, e não de limitação, tais arquiteturas incluem o barramento de Arquitetura de Padrão Industrial (ISA), o barramento de Arquitetura de Micro Canal (MCA), o barramento de Arquitetura ISA Aperfeiçoada (EISA), o barramento local da Associação de Padrões Eletrônicos de Vídeo (VESA), ou o barramento de Interconexão de Componentes Periféricos (PCI), também conhecido como barramento Mezanino.

[0033] O computador 110 inclui tipicamente uma variedade de meio legíveis em computador. Os meios legíveis em computador incluem meios voláteis ou não voláteis, e meios removíveis ou não removíveis. À guisa de exemplo, e não de limitação, os meios legíveis em computador podem compreender meios de armazenamento em computador e meios de comunicação e incluem, porém não se limitam a, memória RAM, memória ROM, memória EEPROM, memória flash ou outra tecnologia de memória, CD-ROM, discos versáteis digitais

(DVD) ou outro armazenamento de disco ótico, cassetes magnéticos, fita magnética, armazenamento de disco magnético ou outros dispositivos de armazenamento magnéticos, ou qualquer outro meio que possa ser usado para armazenar as informações desejadas e que podem ser acessadas pelo computador 110. Os meios de comunicação tipicamente incorporam instruções legíveis em computador, estruturas de dados, módulos de programa, ou outros dados em um sinal de dados modulados, como, por exemplo, uma onda portadora ou outro mecanismo de transporte, e incluem quaisquer meios de liberação de informações. O termo "sinal de dados modulados" significa um sinal que tem uma ou mais de suas características definidas ou modificadas de tal maneira a codificar informações no sinal. À guisa de exemplo, e não de limitação, os meios de comunicação incluem meios de conexão física, como, por exemplo, uma rede conectada ou de conexão direta, ou meios sem fio, como, por exemplo, os meios acústicos, de RF, infravermelhos ou outros meios sem fio. As combinações de quaisquer dentre o acima devem ser também incluídas dentro do âmbito dos meios legíveis em computador.

[0034] A memória de sistema 130 inclui um meio de armazenamento em computador na forma de memória volátil e/ou não volátil, como, por exemplo, a memória de leitura (ROM) 131 e a memória de acesso aleatório (RAM) 132. Um sistema de entrada / saída básico 133 (BIOS), contendo as rotinas básicas que ajudam a transferir informações entre elementos dentro do computador 110, como, por exemplo, durante a inicialização, é tipicamente armazenado na memória ROM 131. A memória RAM 132 tipicamente contém dados e/ou módulos de programa que são imediatamente acessíveis à e/ou que são correntemente operados pela unidade de processamento 120. À guisa de exemplo, e não de limitação, a Figura 1 ilustra o sistema operacional 134, os programas de aplicação 135, outros módulos de programa

136, e os dados de programa 137.

[0035] O computador 110 pode incluir ainda outros meios de armazenamento em computador removíveis / não removíveis, voláteis / não voláteis. À guisa de exemplo somente, a Figura 1 ilustra uma unidade de disco rígido 140 que lê a partir de ou grava em um meio magnético não removível, não volátil, uma unidade de disco magnético 151 que lê a partir de ou grava em um disco magnético removível, não volátil 152, e uma unidade de disco ótico 155 que lê a partir de ou grava em um disco ótico removível, não volátil 156, como, por exemplo, um CD ROM ou outro meio ótico. Outros meios de armazenamento em computador removíveis / não removíveis, voláteis / não voláteis que podem ser usados no ambiente operacional exemplar incluem, porém não se limitam a, cassetes de fita magnética, cartões de memória flash, discos versáteis digitais, fita de vídeo digital, memória RAM em estado sólido, memória ROM em estado sólido, ou coisa do gênero. A unidade de disco rígido 141 é tipicamente conectada ao barramento de sistema 121 através de uma interface de memória não removível, como, por exemplo, a interface 140, e a unidade de disco magnético 151 e a unidade de disco ótico 155 são tipicamente conectadas ao barramento de sistema 121 por meio de uma interface de memória removível, como, por exemplo, a interface 150.

[0036] As unidades e seus meios de armazenamento em computador associados apresentados acima e ilustrados na Figura 1A provêm o armazenamento de instruções legíveis em computador, estruturas de dados, módulos de programa e outros dados para o computador 110. Na Figura 1A, por exemplo, a unidade de disco rígido 141 é ilustrada como o sistema operacional de armazenamento 144, os programas de aplicação 145, outros módulos de programa 146, e como os dados de programa 147. Observa-se que estes componentes podem ser iguais aos ou diferentes do sistema operacional 134, dos pro-

gramas de aplicação 135, de outros módulos de programa 136, ou dos dados de programa 137. O sistema operacional 144, os programas de aplicação 145, outros módulos de programa 146, e os dados de programa 147 recebem números diferentes no presente documento a fim de ilustrar que, no mínimo, os mesmos são cópias diferentes. Um usuário pode entrar comandos e informações no computador 20 através de dispositivos de entrada, tais como um teclado 162 ou um dispositivo de indicação 161, comumente referidos como mouse, trackball ou mesa sensível ao toque. Outros dispositivos de entrada (não mostrados) podem incluir um microfone, um joystick, uma controladora de jogos, uma antena parabólica para satélite, um leitor ótico (scanner), ou coisa do gênero. Estes e outros dispositivos de entrada são, com freqüência, conectados à unidade de processamento 120 através de uma interface de entrada de usuário 160 que é acoplada ao barramento de sistema, mas podem ser conectados por meio de outra interface e estruturas de barramento, tais como, por uma porta paralela, uma porta de jogos, ou por um barramento serial universal (USB). Um monitor 191 ou outro tipo de dispositivo de imagem é também conectado ao barramento de sistema 121 através de uma interface, como, por exemplo, uma interface de vídeo 190. Além do monitor 191, os computadores podem incluir ainda outros dispositivos de saída periféricos, tais como os altofalantes 197 e a impressora 196, os quais podem ser conectados através de uma interface periférica de saída 190.

[0037] O computador 110 pode operar em um ambiente de rede utilizando conexões lógicas a um ou mais computadores remotos, como, por exemplo, um computador remoto 180. O computador remoto 180 pode ser um computador pessoal, um servidor, um roteador, um PC de rede, um dispositivo de rede não hierárquica ou outro nó de rede comum, e tipicamente inclui muitos dos ou todos os elementos descritos acima com relação ao computador 110, embora apenas um dis-

positivo de armazenagem de memória 181 seja ilustrado na Figura 1A. As conexões lógicas ilustradas na Figura 1A incluem uma rede de área local (LAN) 171 e uma rede de área remota (WAN) 173, mas podem incluir ainda outras redes. Estes ambientes de rede são comuns em escritórios, em redes de computador empresariais, em intranets e na Internet.

[0038] Quando utilizado em um ambiente de rede LAN, o computador 110 é conectado à rede LAN 171 através de uma interface ou adaptadora de rede 170. Quando utilizado em um ambiente de rede WAN, o computador 110 tipicamente inclui um modem 172 ou outro meio para o estabelecimento de comunicações pela rede WAN 173, como, por exemplo, pela Internet. O modem 172, que pode ser interno ou externo, pode ser conectado ao barramento de sistema 121 através da interface de entrada de usuário 160, ou por outro mecanismo apropriado. Em um ambiente de rede, os módulos de programa ilustrados com relação ao computador 110, ou porções do mesmo, podem ser armazenados no dispositivo de armazenamento de memória remoto. À guisa de exemplo, e não de limitação, a Figura 1A ilustra programas de aplicação remotos 185 residindo no dispositivo de memória 181. Será apreciado que as conexões de rede mostradas são exemplares e outros meios para se estabelecer um enlace de comunicação entre os computadores podem ser utilizados.

[0039] Será apreciado que as conexões de rede são exemplares e outros meios para se estabelecer um enlace de comunicação entre os computadores podem ser usados. A existência de quaisquer dentre os vários protocolos bem conhecidos, tais como, o protocolo TCP/IP, a Ethernet, o protocolo FTP, o protocolo HTTP, ou coisa do gênero, é presumida, e o sistema pode ser operado em uma configuração cliente-servidor de modo a permitir a um usuário recuperar páginas da rede mundial (web) a partir de um servidor baseado na rede mundial.

Quaisquer dentre os diversos navegadores da rede mundial (web browsers) podem ser usados para exibir e manipular dados nas páginas da rede mundial.

[0040] Uma interface de programação (ou mais simplesmente "interface") pode ser visualizada como qualquer mecanismo, processo, ou protocolo de modo a permitir que um ou mais segmentos de código se comunique com ou acesse a funcionalidade provida por um ou mais outros segmentos de código. De maneira alternativa, uma interface de programação pode ser visualizada como um ou mais mecanismos, métodos, chamadas de função, módulos, objetos, etc. de um componente de um sistema capaz de um acoplamento comunicativo com um ou mais mecanismos, métodos, chamadas de função, módulos, etc. de outros componentes. O termo "segmento de código" acima pretende incluir uma ou mais instruções ou linhas de código, e inclui, por exemplo, módulos de código, objetos, sub-rotinas, funções, e assim por diante, independente da terminologia aplicada ou se os segmentos de código são compilados separadamente, ou se os segmentos de código são providos como uma fonte, intermediário, ou código de objeto, se os segmentos de código são utilizados em um sistema ou processo de tempo de execução, ou se os mesmos se localizam na mesma ou em máquinas diferentes ou distribuídos através de múltiplas máquinas, ou se a funcionalidade representada pelos segmentos de código são implementados totalmente em um software, totalmente em um hardware, ou em uma combinação de hardware e software.

[0041] Em termos nocionais, uma interface de programação pode ser visualizada genericamente, conforme mostrado na Figura 1B ou Figura 1C. A Figura 1B ilustra uma interface Interface1 como um conduto através do qual o primeiro e o segundo segmentos se comunicam. A Figura 1C ilustra uma interface que compreende os objetos de interface 11 e 12 (os quais podem ou não fazer parte do primeiro e do

segundo segmentos de código), os quais permitem que o primeiro e o segundo segmentos de código de um sistema se comuniquem através de um meio M. De acordo com a Figura 1C, pode-se considerar os objetos de interface 11 e 12 como interfaces separadas do mesmo sistema, além de se pode considerar também que os objetos 11 e 12 mais o meio M compreendem a interface. Embora as Figuras 1B e 1C mostrem um fluxo bidirecional e interfaces em cada lado do fluxo, certas implementações podem ter apenas um fluxo de informação em uma direção (ou nenhum fluxo de informação, conforme descrito abaixo) ou pode ter apenas um objeto de interface em um lado. A guisa de exemplo, e não de limitação, termos tais como interface de programação de aplicação (API), ponto de entrada, método, função, sub-rotina, chamada de procedimento remota, e interface de modelo de objeto componente (COM), são abrangidos dentro da definição da interface de programação.

[0042] Aspectos de tal interface de programação podem incluir o método por meio do qual o primeiro segmento de código transmite informação (quando "informação" é usada em seu sentido mais amplo e inclui dados, comandos, solicitações, etc.) para o segundo segmento de código; o método por meio do qual o segundo segmento de código recebe as informações; e a estrutura, seqüência, sintaxe, organização, esquema, sincronização, e conteúdo da informação. A este respeito, o próprio meio de transporte subjacente pode ser insignificante para a operação da interface, que o meio seja de uma conexão física ou sem fio, ou uma combinação de ambas, contanto que a informação seja transportada da maneira definida pela interface. Em determinadas situações, a informação não pode ser passada em uma ou ambas as direções no sentido convencional, uma vez que a transferência de informação pode ser feita através de um outro mecanismo (por exemplo, a informação colocada em um armazenador temporário (buffer), arqui-

vo, etc., separada do fluxo de informação entre os segmentos de código) ou inexistente, quando, por exemplo, um segmento de código simplesmente acessa a funcionalidade executada por um segundo segmento de código. Quaisquer ou todos estes aspectos podem ser importantes em uma dada situação, por exemplo, dependendo se os segmentos de código fazem parte de um sistema em uma configuração frouxamente acoplada ou em uma configuração estritamente acoplada, e, deste modo, esta lista deve ser considerada ilustrativa e não limitante.

[0043] Esta noção de uma interface de programação é conhecida aos versados na técnica e é clara a partir da descrição detalhada acima da presente invenção. existem, no entanto, outras maneiras de se implementar uma interface de programação, e, a menos que expressamente excluído, estas também pretendem estar abrangidas pelas reivindicações apresentadas ao final do presente relatório descritivo. Estas outras maneiras parecem mais sofisticadas ou complexas que a visão simplista das Figuras 1B e 1C, mas, no entanto, as mesmas realizam uma função similar de modo a atingir o mesmo resultado geral. A seguir, serão brevemente descritas algumas modalidades alternativas de uma interface de programação.

FATORAÇÃO

[0044] Uma comunicação a partir de um segmento de código para outro pode ser feita indiretamente por meio da quebra da comunicação em múltiplas comunicações discretas. Isto é ilustrado esquematicamente nas Figuras 1D e 1E. Conforme mostrado, algumas interfaces podem ser descritas em termos de conjuntos divisíveis de funcionalidade. Sendo assim, a funcionalidade da interface das Figuras 1B e 1C pode ser fatorada de modo a chegar ao mesmo resultado, exatamente como uma pessoa pode matematicamente ter 24, ou 2 vezes 2 vezes 3 vezes 2. Por conseguinte, conforme ilustrado na Figura 1D, a função

provida pela interface Interface1 pode ser subdividida de modo a converter as comunicações da interface em múltiplas interfaces Interface1A, Interface1B, Interface1C, etc., e ao mesmo tempo chegar ao mesmo resultado. Conforme ilustrado na Figura 1E, a função provida pela interface I1 pode ser subdividida em múltiplas interfaces I1a, I1b, I1c, etc. e ao mesmo tempo chegar ao mesmo resultado. De maneira similar, a interface I2 do segundo segmento de código que recebe informação a partir do primeiro segmento de código pode ser fatorada em múltiplas interfaces I2a, I2b, I2c, etc. Ao se fatorar, o número de interfaces incluídas no primeiro segmento de código não precisa corresponder ao número de interfaces incluídas no segundo segmento de código. Em ambos os casos das Figuras 1D e 1E, o espírito funcional das interfaces Interface1 e I1 permanece igual aos das Figuras 1B e 1C, respectivamente. A fatoração das interfaces pode também seguir propriedades associativas, comutativas, ou outras propriedades matemáticas, de tal modo que a fatoração pode se tornar difícil de reconhecer. Por exemplo, a ordenação das operações pode ser insignificante, e, consequentemente, uma função realizada por uma interface pode ser executada mesmo antes de se obter a interface, por meio de um outro pedaço de código ou interface, ou executada por meio de um componente separado do sistema. Além disso, uma pessoa com habilidade simples nas técnicas de programação poderá apreciar que existem várias maneiras de se fazer diferentes chamadas de função diferentes que chegam ao mesmo resultado.

REDEFINIÇÃO

[0045] Em alguns casos, pode ser possível ignorar, adicionar, ou redefinir determinados aspectos (por exemplo, parâmetros) de uma interface de programação enquanto ainda se obtém o resultado pretendido. Isto é ilustrado nas Figuras 1F e 1G. Por exemplo, pressupondo que interface Interface1 da Figura 1b inclui um Quadrado de

chamada de função (entrada, precisão, e saída), uma chamada que inclui três parâmetros, entrada, precisão e saída, e que é emitida a partir do Primeiro Segmento de Código para o Segundo Segmento de Código. Quando a precisão do parâmetro intermediário não é um problema em um dado cenário, conforme mostrado na Figura 1F, a mesma só pode ser ignorada e ainda substituída por um parâmetro sem significado (nesta situação). Pode-se ainda adicionar um parâmetro adicional sem problema. Em ambos os eventos, a funcionalidade do quadrado pode ser obtida, contanto que a saída retorne depois de a entrada ser elevada ao quadrado pelo segundo segmento de código. A precisão pode ser exatamente um parâmetro significativo para alguma porção a jusante ou outra porção do sistema computacional. No entanto, quando se reconhece que a precisão não é necessária para o estreito propósito de se calcular o quadrado, a mesma pode ser substituída ou ignorada. Por exemplo, ao invés de se passar um valor de precisão válido, um valor sem sentido, como, por exemplo, uma data de nascimento, pode ser passado sem afetar de maneira adversa o resultado. De maneira similar, conforme mostrado na Figura 1G, a interface I1 é substituída pela interface I1', redefinida de modo a ignorar ou adicionar parâmetros à interface. A interface I2 pode, de maneira similar, ser redefinida como a interface I2', redefinida de modo a ignorar parâmetros desnecessários, ou parâmetros que podem ser processados em outro momento. A questão neste caso é que, em algumas situações, uma interface de programação pode incluir aspectos, tais como, parâmetros, que não são necessários para algum fim, e, portanto, podem ser ignorados ou redefinidos, ou processados em outro momento para outros fins.

CODIFICAÇÃO ALINHADA

[0046] Pode ser ainda exeqüível unir algumas ou todas as funcionalidades de dois módulos de código separados de tal modo que a

"interface" entre os mesmos mude a sua forma. Por exemplo, a funcionalidade das Figuras 1B e 1C pode ser convertida à funcionalidade das Figuras 1H e 1I, respectivamente. Na Figura 1H, o Primeiro e o Segundo Segmentos de Código anteriores da Figura 1B são unidos em um módulo contendo ambos os segmentos. Neste caso, os segmentos de código podem ainda se comunicar entre si, mas a interface pode ser adaptada a uma forma que seja mais adequada para o módulo único. Sendo assim, por exemplo, as instruções formais de Chamar e Retornar podem não mais ser necessárias, porém um processamento ou resposta similar de acordo com a interface Interface1 pode ainda se encontrar em efeito. De maneira similar, mostrada na Figura 1I, parte ou toda a interface I2 da Figura 1C pode ser escrita alinhada com a interface I1 de modo a formar a interface I1". Conforme ilustrado Conforme ilustrado, a interface I2 é dividida na nas interfaces I2a e I2b, e a porção de interface I2a é codificada alinhada com a interface I1 de modo a formar a interface I1". Para um exemplo concreto, considere-se que a interface I1 da Figura 1C realiza uma função de chamada quadrada (entrada, saída), que é recebida pela interface I2, que depois de processamento, o valor passado com a entrada (para calcular o quadrado de uma entrada) pelo segundo segmento de código, passa novamente o resultado quadrado com a saída. Neste caso, o processamento realizado pelo segundo segmento de código (entrada elevada ao quadrado) pode ser feito pelo primeiro segmento de código sem uma chamada para a interface.

DIVÓRCIO

[0047] Uma comunicação de um segmento de código para outro pode ser feita indiretamente quebrando a comunicação em múltiplas comunicações discretas. Isto é ilustrado esquematicamente nas Figuras 1J a 1K. Conforme mostrado na Figura 1J, um ou mais pedaços de código (Interfaces de Divórcio, uma vez que as mesmas divorciam a

funcionalidade e/ou as funções de interface da interface original) são providos de modo a converter as comunicações da primeira interface, Interface1, a fim de conformar as mesmas a uma interface diferente, neste caso às interfaces Interface2A, Interface2B e Interface2C. Isto pode ser feito, por exemplo, quando existe uma base instalada de aplicações desenhadas para se comunicar com, por exemplo, um sistema operacional de acordo com um protocolo de Interface1, mas em seguida o sistema operacional é alterado para usar uma interface diferente, neste caso as interfaces Interface1A, a Interface2B e a Interface2C. A questão é que a interface original usada pelo Segundo Segmento de Código é alterada a tal modo que a mesma se torna incompatível com a interface usada pelo Primeiro Segmento de Código, e, assim, uma intermediária é usada a fim de tornar a antiga e a nova interfaces compatíveis. De maneira similar, conforme mostrado na Figura 1K, um terceiro segmento de código pode ser introduzido com uma interface de divórcio DI1 de modo a receber as comunicações da interface I1, e com a interface de divórcio DI2 de modo a transmitir a funcionalidade de interface para, por exemplo, as interfaces I2a e I2b, redesenhadas para funcionar com a interface DI2, porém provendo o mesmo resultado funcional. De maneira similar, as interfaces KI1 e DI2 podem funcionar juntas de modo a transmitir a funcionalidade das interfaces I1 e I2 da Figura 1C para um sistema operacional, ao mesmo provendo o mesmo resultado funcional ou um resultado funcional similar.

REESCRITA

[0048] Uma outra variante ainda possível é dinamicamente reescrever o código a fim de substituir a funcionalidade da interface por qualquer outra coisa, mas que obtenha o mesmo resultado geral. Por exemplo, pode haver um sistema no qual um segmento de código apresentado em uma linguagem intermediária (por exemplo, a Microsoft IL, a Java ByteCode, etc.) é provido para um compilador ou inter-

pretador Just-in-Time em um ambiente de execução (tal como o provido pela estrutura .Net, pelo ambiente de tempo de execução Java, ou outros ambientes do tipo tempo de execução). O compilador JIT pode ser escrito de modo a converter dinamicamente as comunicações do Primeiro Segmento de Código para o Segundo Segmento de Código, isto é, conformar os mesmos a uma interface diferente conforme possa ser requerido pelo Segundo Segmento de Código (quer o original ou um Segundo Segmento de Código diferente). Isto é ilustrado nas Figuras 1L e 1M. Como se pode ver na Figura 1L, esta abordagem é similar ao cenário de Divórcio descrito acima. Isto pode ser feito, por exemplo, quando uma base instalada de aplicações é desenhada para se comunicar com um sistema operacional de acordo com um protocolo de Interface1, mas, em seguida, o sistema operacional é alterado para usar uma interface diferente. O compilador JIT pode ser usado para conformar as comunicações instantâneas das aplicações instaladas na base à nova interface do sistema operacional. Conforme ilustrado na Figura 1M, esta abordagem de se reescrever dinamicamente as interfaces pode ser aplicada a uma fatoração dinâmica, ou para, de outra forma, alterar também as interfaces.

[0049] Observa-se ainda que os cenários acima descritos para a obtenção do mesmo resultado e de um resultado similar ao da interface através de modalidades alternativas podem também ser combinados de várias maneiras, serialmente ou em paralelo, ou com outro código interveniente. Deste modo, as modalidades alternativas apresentadas acima não são mutuamente exclusivas e podem ser misturadas, conjugadas e combinadas de modo a produzir os mesmos cenários ou equivalentes aos cenários genéricos apresentados nas Figuras 1B e 1C. Nota-se ainda que, de acordo com a maioria das construções de programação, existem outras formas similares de se obter a mesma funcionalidade ou uma funcionalidade similar de uma interface que po-

de não estar descrita no presente documento, mas, que, no entanto, se encontra representada pelo espírito e âmbito da presente invenção, ou seja, observa-se que a interface é pelo menos parcialmente representada em termos de funcionalidade por, e com resultados vantajosos possibilitados por, uma interface que fundamenta o valor de uma interface.

MINIATURAS

[0050] As Figuras 4A e 4B demonstram um exemplo de uma aplicação da presente invenção na provisão de miniaturas de janelas de aplicação. No exemplo da Figura 4A, é ilustrada uma barra de tarefas 400 contendo botões de barra de tarefas. Cada um dos botões de barra de tarefas corresponde a uma aplicação ativa. O botão de barra de tarefas 402 corresponde a um programa de aplicação de jogo Paciência executado correntemente em um computador. Da mesma forma, o botão de barra de tarefas 403 corresponde a uma janela de programa de aplicação de processamento de texto, o botão de barra de tarefas 405 corresponde a uma janela de programa de aplicação de tocador de mídia, e o botão de barra de tarefas 406 corresponde a uma janela de programa de aplicação de Calculadora. Uma miniatura de jogo Paciência 401 é provida para a janela de programa de aplicação de jogo Paciência e uma miniatura de Tocador de Mídia 404 é provida para o programa de aplicação de tocador de Mídia. As miniaturas correspondentes podem ser invocadas de diversas maneiras. Em um exemplo não limitante, as miniaturas podem ser invocadas ao se pairar um cursor sobre o botão de barra de tarefas correspondente. Por exemplo, a miniatura de jogo Paciência 401 pode ser invocada ao se pairar o cursor sobre o botão de barra de tarefas do jogo Paciência 402.

[0051] Ainda ilustradas na Figura 4A, as miniaturas dinâmicas podem prover um conteúdo atualizado ou "vivo" correspondente ao programa de aplicação. Por exemplo, a aplicação de Tocador de Mídia

pode estar passando um vídeo. Quando o vídeo passa na janela de aplicação de Tocador de Mídia (não mostrada), o vídeo também passa na miniatura de Tocador de Mídia 404. Quando a própria janela de aplicação de tocador de mídia está minimizada, a miniatura de tocador de mídia 404 pode exibir a última cena na janela de aplicação de tocador de mídia antes de minimizar. De maneira alternativa, a miniatura de tocador de mídia 404 pode continuar a exibir o vídeo e continua a passar a aplicação de tocador de mídia mesmo que a própria janela de aplicação de tocador de mídia esteja minimizada.

[0052] A Figura 4B ilustra um outro exemplo de miniaturas da presente invenção. Uma barra de tarefas 401 é ilustrada contendo um botão de barra de tarefas de jogo Paciência 422 correspondente a uma janela de aplicação de jogo Paciência (não mostrada), um botão de barra de tarefas de processamento de texto 412 correspondente a uma janela de aplicação de processamento de texto (não mostrada), um botão de barra de tarefas de tocador de mídia 413 correspondente a uma janela de aplicação de tocador de mídia (não mostrada), e um botão de barra de tarefas de Calculadora 414 correspondente a uma janela de aplicação de calculadora (não mostrada). Neste exemplo, é provida uma miniatura de Calculadora 415. A miniatura de Calculadora 415 exibe os conteúdos correntes e atualizados da janela de aplicação de Calculadora. Em uma modalidade da presente invenção, a miniatura de Calculadora 415 pode ainda atualizar a janela de aplicação. Por exemplo, um cálculo pode ser feito através da miniatura de Calculadora 415 no sentido de obter um resultado matemático. Este resultado pode ser exibido na miniatura de Calculadora 415 assim como na janela de aplicação de Calculadora (não mostrada). Neste exemplo, a chave "=" é selecionada na miniatura de Calculadora 415 e um resultado é exibido na miniatura de Calculadora (isto é, "4"). Este resultado pode também ser refletido na janela de aplicação de Calculadora (não mos-

trada) mesmo que o cálculo seja feito através da miniatura de Calculadora 415.

[0053] A Figura 5 ilustra uma outra aplicação de uma miniatura da presente invenção na qual as miniaturas são exibidas em um menu de ALT-TAB. De acordo com a Figura 5, um vídeo 500 exibe janelas de aplicação. Neste caso, uma porção de uma janela de aplicação de Calculadora 502 e uma porção de uma janela de aplicação de jogo Paciência 501 estão visíveis no vídeo 500. Cada uma das aplicações tem um botão de barra de tarefas correspondente na barra de tarefas. A aplicação de Calculadora tem um botão de barra de tarefas de Calculadora 503 correspondente e a aplicação de jogo Paciência tem um botão de barra de tarefas de jogo Paciência 504 correspondente. Uma aplicação de Tocador de Mídia se encontra igualmente ativa (não mostrada) e um botão de barra de tarefas de Tocador de mídia 506 correspondente está também presente na barra de tarefas.

[0054] No exemplo ilustrado na Figura 5, um menu ALT-TAB 505 é provido e exibe miniaturas dinâmicas de aplicações ativas. Neste exemplo, cada uma das miniaturas dinâmicas exibe o conteúdo “vivo” (isto é, atualizado) de uma janela de aplicação correspondente. O conteúdo de miniatura dinâmica pode ser atualizado em tempo real. Uma vez que as janelas de aplicação são modificadas ou atualizadas, a miniatura dinâmica correspondente pode também ser atualizada em conformidade. A atualização da miniatura dinâmica pode também ser obtida em tempo real. Neste exemplo, o menu ALT-TAB 505 contém uma miniatura dinâmica de cada uma das aplicações ativas, quer a janela de aplicação da aplicação correspondente esteja visível no vídeo ou não. Uma miniatura de Calculadora 507, uma miniatura de Tocador de mídia 508, e uma miniatura de jogo Paciência 509 são ilustradas no menu ALT-TAB 505 com a miniatura de jogo Paciência 509 sendo selecionada. Cada uma das miniaturas dinâmicas pode prover um conte-

údo “vivo”. Por exemplo, um vídeo pode ser exibido na miniatura de Tocador de Mídia 508 em tempo real.

[0055] As Figuras 2A e 2B são diagramas que ilustram os componentes de uma vídeo de amostra. A área de trabalho é representada na Figura 2A em um gráfico de cena ou árvore de vídeos como o elemento 200 e contém o papel de parede da área de trabalho 201 e duas aplicações, a Calculadora 202 e o jogo Paciência 203. Além disso, a área de trabalho contém ainda uma barra de tarefas 208. Neste método de representação, os quadros em volta das janelas e a área de cliente de cada janela são todos parte do gráfico de cena. O movimento de janela torna-se valores de transformação mutáveis na árvore de vídeos. Cada uma das aplicações neste exemplo contém um quadro de janela 204, 205, respectivamente, e o conteúdo de janela 206, 207, respectivamente. A barra de tarefas 208 contém o conteúdo de janela 209. Na Figuras 2A, os elementos no vídeo são renderizados da esquerda para a direita. Sendo assim, o papel de parede 201 da área de trabalho é renderizado atrás da janela de aplicação da Calculadora 203 que é renderizada atrás da janela de aplicação do jogo Paciência 202.

[0056] A Figura 2B ilustra o vídeo conforme representado na Figura 2A. O vídeo contém o papel de parede 220 no fundo. A janela de aplicação de Calculadora 222 é exibida na frente do papel de parede da área de trabalho e a janela de aplicação do jogo Paciência 221 é exibida na frente da janela de aplicação da Calculadora 222 e do papel de parede da área de trabalho 220. Conforme mostrado na Figura 2B, cada uma das janelas de aplicação pode ser representada pelos botões de barra de tarefas na barra de tarefas 225. Neste exemplo, a aplicação de Calculadora é representada pelo botão de barra de tarefas 223 na barra de tarefas 225, e o programa de aplicação de jogo Paciência é representado pelo botão de barra de tarefas 224 na barra

de tarefas 225. Cada uma das janelas de aplicação de Calculadora 222 e de aplicação de jogo Paciência 221 contém um quadro de janela e um conteúdo de janela conforme exibidos no Papel de parede de Área de trabalho 220.

[0057] Qualquer sub-nó da árvore conforme ilustrado na Figura 2A pode ser usado como o conteúdo de um outro nó da árvore. Sendo assim, os conteúdos de uma parte da árvore podem ser visualizados em outro local em qualquer ponto da árvore. Existem muitos métodos para se obter uma versão de um nó na árvore em qualquer outro local na árvore. Como um exemplo não limitante, o Visual Brush pode ser usado. O "Visual Brush" é uma técnica usada para se referir a uma parte existente do gráfico de cena, mas para renderizar o mesmo com o conjunto "corrente" de contexto gráfico, como, por exemplo, posicionamento, ajuste de escala, rotação, opacidade, efeitos, etc. Isto faz com uma porção da árvore renderizada em um local seja renderizada em qualquer outro lugar. Por exemplo, caso se deseje uma miniatura dinâmica da janela de aplicação de Calculadora, uma interface de programação apropriada (por exemplo, uma Interface de Programação de Aplicação ou API) pode ser calculada de modo a prover uma alça HWND de Host de Miniatura ou alça de janela. Este exemplo é demonstrado nas Figuras 3A e 3B.

[0058] A Figura 3A é um gráfico de cena ou diagrama de árvore de vídeos ilustrando um exemplo de componentes de um vídeo da presente invenção. A área de trabalho é representada na Figura 3A como o elemento ou nó 300 e contém o papel de parede de área de trabalho 301 e duas aplicações. A calculadora 302 e o jogo Paciência 303. Cada uma das aplicações neste exemplo contém um quadro de janela 305, 306, respectivamente, e o conteúdo de janela 307, 308, respectivamente. A área de trabalho neste exemplo contém ainda uma barra de tarefas 307.

[0059] Neste exemplo, uma miniatura da janela de aplicação de Calculadora é provida por meio da chamada de uma interface API apropriada. Conforme ilustrado na Figura 3A, uma alça de janela HWND de Host de Miniatura 304 é criada por uma aplicação e pode registrar uma miniatura de uma aplicação, tal como a aplicação de Calculadora. Por exemplo, na Figura 3A, a barra de tarefas 307 contém uma alça HWND de Host de Miniatura 304 e um conteúdo de janela 310. A alça HWND de Host de Miniatura 304 contém uma miniatura de Calculadora 311 e pode anexar uma miniatura da janela de aplicação ao host de miniatura. A seta pontilhada na Figura 3A ilustra que a miniatura de Calculadora 311 da alça HWND de Host de Miniatura 304 aponta para trás ou redesenha a Calculadora a partir da janela de aplicação de Calculadora. Como um resultado, uma miniatura "viva" ou dinâmica é provida, apresentando o conteúdo atualizado da janela de aplicação correspondente (neste caso, a janela de aplicação de Calculadora) ao invés de visualizações simplesmente estáticas do conteúdo de janela de aplicação.

[0060] De acordo com o exemplo da Figura 2A, os elementos do vídeo representado na Figura 3A são renderizados da esquerda para a direita. Sendo assim, o papel de parede 301 da área de trabalho é renderizado atrás da janela de aplicação da Calculadora 302 que é renderizada atrás da janela de aplicação do jogo Paciência 303 e da barra de tarefas 208. A Figura 3B demonstra o vídeo correspondente à Figura 3A. De acordo com a Figura 3B, o vídeo contém o papel de parede 320 da área de trabalho no fundo. A janela de aplicação de Calculadora 322 é exibida na frente do papel de parede 320 da área de trabalho, e a janela de aplicação do jogo Paciência 321 é exibida na frente da janela de aplicação da Calculadora 322 e do papel de parede 320 da área de trabalho. A barra de tarefas 326 é igualmente exibida. Cada uma das aplicação é associada a um botão de barra de tarefas

correspondente na barra de tarefas 326. Neste exemplo, a aplicação de Calculadora é associada ao botão de Barra de tarefas 324, e a aplicação de jogo Paciência é associada ao botão de barra de tarefas 325. Além disso, a Figura 3B ilustra que uma miniatura é provida associada à aplicação de Calculadora. O programa de aplicação de Calculadora e a janela 322 são associadas ao botão de barra de tarefas 324 na barra de tarefas 326, e o programa de aplicação de jogo Paciência e a janela 321 são associados ao botão de barra de tarefas 325.

[0061] A Figura 3B ilustra ainda uma miniatura dinâmica 323 da janela de aplicação de Calculadora 322. A miniatura 323 é provida pelo nó de Miniatura de Calculadora 311 da alça HWND de Host de Miniatura 304 que aponta para trás ou redesenha o quadro de janela 305 e o conteúdo de janela 308 da aplicação de Calculadora 302, conforme ilustrado na Figura 3A. O resultado é uma miniatura dinâmica 323 da janela de aplicação de Calculadora 322 que se encontra "viva", ou seja, a miniatura é modificada conforme modificações correspondentes são feitas à própria janela de aplicação. Conforme ilustrado na Figura 3B, o valor "243" calculado na janela de aplicação de Calculadora 322 também se reflete na miniatura correspondente 323.

[0062] Em um aspecto da presente invenção, as miniaturas dinâmicas são obtidas através das propriedades de registro, de não registro e de atualização das miniaturas. Estas propriedades de registro, não registro e atualização ou modificação de miniaturas podem ser obtidas através das interfaces de programação de aplicação (API) no Gerenciador de Janela de Área de trabalho (DWM). Três interfaces API exemplares são descritas de modo a obter as miniaturas da presente invenção.

Exemplo 1 de Interface API - Registro da miniatura

[0063] Uma miniatura pode ser gerada a partir de uma janela, por exemplo, de uma janela de aplicação. A janela de aplicação, neste ca-

so, é janela de origem, uma vez que a janela de aplicação provê a base para o conteúdo da miniatura. Em um exemplo, a miniatura é uma versão miniaturizada da janela de aplicação e contém os conteúdos completos da janela de aplicação. De maneira alternativa, a miniatura pode conter apenas uma porção selecionada da janela de origem.

[0064] O conteúdo da miniatura a partir da janela de origem é renderizado para uma segunda área. Esta segunda área é desenhada como a janela de destino que acomoda o conteúdo da miniatura com base na janela de origem. A janela de destino pode formar a base da própria miniatura. Por exemplo, um quadro de janela pequena em um local diferente da janela de aplicação, uma área designada no vídeo, ou um menu instantâneo podem ser usados como a janela de destino a fim de exibir a miniatura com base na janela de aplicação.

[0065] Quando uma miniatura é registrada, pode ser feita uma operação que estabelece a relação entre a janela de origem e a janela de destino. Uma alça de janela (HWND) pode ser especificada, designando a janela de destino. A janela de destino especificada pode servir como o alvo da renderização de miniatura e é possuída pelo processo que chama ou registra a miniatura. Quando o processo que registra a miniatura também possui a janela de destino, alterações indesejadas na aplicação por parte de outras aplicações podem ser evitadas. De maneira similar, uma alça HWND para a janela de origem pode também ser especificada.

[0066] O dimensionamento de um mapa de bits de cache da miniatura pode também ser especificado. Por exemplo, um dimensionamento mínimo pode ser indicado de tal modo que o sistema seja informado do último mapa de bits conhecido da janela de origem. Desta maneira, quando a janela de origem é minimizada em seguida e não se torna visível no vídeo, um mapa de bits da janela fica, no entanto, retido e a miniatura fica ainda disponível. O dimensionamento pode

variar com base nas necessidades da aplicação que chama. Se houver múltiplos registros da mesma janela de origem, um processo poderá determinar que dimensionamento de miniatura se aplicará com base nas necessidades presentes. Por exemplo, no caso de múltiplos registros de vários dimensionamentos de miniatura, o dimensionamento alvo poderá ser usado.

[0067] A interface API pode conter ainda uma alça representando o registro da miniatura. Esta alça pode ser única para o processo que emite a chamada, de tal modo que a miniatura possa ser em seguida não registrada apenas a partir do processo com o qual a mesma foi registrada.

[0068] Um exemplo de uma interface API para o registro de uma miniatura é como segue:

```
Typedef          HANDLE
HTHUMBNAIL;

HRESULT
DwmRegisterThumbnail (
HWND                              dstWindow,
HWND                              srcWindow,
SIZE                              *1pMinimizedSize,
OUT HTHUMBNAIL                    *1phThumbnail
);
```

[0069] No qual dstWindow representa a janela de destino, srcWindow representa a janela de origem, lpMinimizedSize representa o dimensionamento no qual o sistema faz cache no último mapa de bits conhecido da janela de origem (srcWindow), uma vez que o srcWindow é minimizado e o lphThumbnail é um valor retornado representando a alça única que representa o registro feito na chamada.

Exemplo 2 de Interface API - Modificação ou Definição de Propriedades de Miniatura

[0070] Após o registro de uma miniatura, o conteúdo da janela de origem ou porção da mesma é colocado como a miniatura na janela de destino. Por exemplo, quando chamada, uma atualização de propriedades de miniatura, a interface API poderá modificar ou atualizar as propriedades da renderização da miniatura em uma janela de destino. Por exemplo, "Atualizar Propriedades de Miniatura" poderá ser usada para definir propriedades de miniaturas, tais como visibilidade, forma, retângulos, etc. Em uma Atualizar propriedades de miniatura, a interface API poderá indicar a alça de uma miniatura a ser atualizada. As atualizações de mínimos incluem qualquer modificação da miniatura, incluindo, porém não se limitando ao, posicionamento da miniatura, dimensionamento da miniatura, porção da janela de origem a ser usada para a miniatura, porção da janela de destino a ser usada para a miniatura, opacidade ou transparência da miniatura, visibilidade da miniatura, efeitos especiais da miniatura (por exemplo, girar, distorcer, fragmentar, etc.), etc.

[0071] Em um outro exemplo, as miniaturas podem ser atualizadas ou modificadas a partir dos processos nos quais as mesmas são registradas. Em um exemplo, uma alça de miniatura é repetidamente chamada para uma janela de destino de tal modo que a miniatura possa ser movimentada em volta da janela de destino. Neste caso, a miniatura pode ser reposicionada dentro da janela de destino. Por exemplo, em uma janela ALT-TAB, as miniaturas podem ser rearranjadas em uma ordem variada ao chamar novamente a miniatura para a janela de destino.

[0072] Podem ser usados parâmetros na modificação ou na definição de propriedades de miniaturas. Existem muitos exemplos de parâmetro para a modificação ou definição de propriedades de miniatura, como, por exemplo, mas sem se limitar às, sinalizações. Através do uso dos parâmetros, a estrutura de interface API pode ser mais facil-

mente modificada, tornando a interface API mais flexível para futuras atualizações. Os parâmetros podem designar as propriedades da miniatura a serem usadas e podem elas próprias serem definidas por meio de constantes específicas. Por exemplo, os parâmetros podem definir ou corresponder a um parâmetro de janela de origem que define uma área ou porção de uma janela de origem a ser usada como uma miniatura em uma janela de destino, quando se deseja usar menos que toda uma janela de origem para a miniatura. Neste exemplo, apenas a porção especificada da janela de origem conforme indicada pelo parâmetro da janela de origem é usada para a miniatura na janela de destino. Em um exemplo, a porção da janela de origem a ser usada como a miniatura na janela de destino é especificada como coordenadas do vídeo. Por exemplo, se a porção desejada é um retângulo dentro da janela de origem, a interface API poderá especificar um parâmetro para a indicação das coordenadas de um retângulo intersectado com a janela de origem a fim de obter o retângulo final a ser usado como a miniatura. Como um exemplo, as coordenadas (0, 0) podem ser definidas como o canto esquerdo superior da área de janela. Um parâmetro que define e que corresponde ao parâmetro de janela de origem desejado pode ser definido a fim de indicar que apenas uma porção específica da janela de origem deve ser usada como a miniatura, a porção especificada da janela de origem que é definida em termos das coordenadas providas. Embora existam muitas maneiras pelas quais se pode especificar as coordenadas e a porção desejada da janela de origem, um exemplo pode incluir o uso da porção da janela de origem definida pela interseção do retângulo provido com a presente janela de origem intersectada.

[0073] Ainda, a porção da janela de destino a ser usada para a miniatura pode ser especificada através de um parâmetro que define e corresponde a um parâmetro de janela de destino. Neste caso, quando

não se deseja usar toda a janela de destino na miniatura, mas sim apenas uma porção da janela de destino deve ser usada na miniatura, a área da janela de destino na qual a miniatura pode ser renderizada pode ser especificada por meio do parâmetro da janela de destino. Se este for o caso, uma abordagem seria renderizar apenas dentro da área designada desejada conforme definida pelo parâmetro da janela de destino (isto é, o recorte da imagem quando a renderização ocorre fora da área designada). Ainda, um método para se determinar a localização da porção da janela de destino na qual a miniatura pode ser conduzida é determinar a interseção da porção da janela de destino a ser usada e a própria janela de destino. Isto impede a renderização fora de uma área designada. Além disso, um parâmetro que define ou corresponde a um parâmetro de janela de destino pode ser usado para designar uma área dentro da área designada a ser usada como uma miniatura a partir da janela de origem. Neste exemplo, se um parâmetro de janela de destino não é definido, toda a janela designada poderá ser usada para o conteúdo de miniatura.

[0074] Qualquer número de propriedades de miniatura pode ser especificado na interface API. Por exemplo, a opacidade ou a transparência da miniatura pode ser especificada. Em um aspecto deste exemplo da presente invenção, um parâmetro pode ser atribuído para designar a opacidade (ou transparência) da miniatura, por exemplo, 225 pode ser atribuído como totalmente opaco, enquanto 0 pode ser atribuído como totalmente transparente. Deste modo, se um parâmetro que define ou corresponde a um parâmetro de opacidade for definido, um valor numérico atribuído designado para opacidade da miniatura poderá ser aplicado. Como um resultado, a miniatura pode exibir graus variados de opacidade conforme desejado por um usuário.

[0075] Como um exemplo adicional, a miniatura pode ser visível ou invisível, conforme necessário. Um parâmetro que define ou cor-

responde a um parâmetro de visibilidade pode ser provido. O parâmetro de visibilidade pode ser um valor Booleano, de tal modo que, quando o parâmetro de visibilidade é verdadeiro (TRUE), a miniatura fica visível, enquanto a miniatura é invisível se o parâmetro de visibilidade for falso (FALSE). A presente invenção não se limita a nenhuma modificação em particular da miniatura, uma vez que qualquer modificação da miniatura está dentro do âmbito da presente invenção.

[0076] Um exemplo de uma interface API para atualizar ou modificar uma miniatura é:

```
DwmUpdateThumbnailProperties, como segue:
HRESULT
DwmUpdateThumbnailProperties (
        HTHUMBNAIL hThumbnail,
        DWM_THUMBNAIL_PROPERTIES
*ptnProperties
);
#define DWM_TNP_RECTDESTINATION 0X00000001
#define DWM_TNP_RECTSOURCE 0X00000002
#define DWM_TNP_OPACITY 0X00000004
#define DWM_TNP_VISIBILITY 0X00000008

Typedef struct_DWM_THUMBNAIL_PROPERTIES
{
        DWORD           dwFlags;
        RECT            rcDestination;
        RECT            rcSource;
        BYTE            opacity;
        BOOL            fVisible;
} dwm_THUMBNAIL_PROPERTIES;
```

[0077] em que dwFlags definem os parâmetros ou sinalizações correspondentes aos parâmetros que especificam as propriedades subseqüentes a serem usadas. Neste exemplo, o parâmetro rcSource

é um parâmetro de janela de origem que especifica a região da janela de origem (srcWindow) a usar como a miniatura, rcDestination é um parâmetro de janela de destino que especifica a janela de destino (dstWindow) para a qual a miniatura é renderizada, opacity é um parâmetro de opacidade que controla a opacidade da miniatura (por exemplo, definido de 0 a 255), e fVisible é um parâmetro de visibilidade que especifica a miniatura como invisível ou visível baseado no valor Booleano.

Exemplo 3 de Interface API - Não Registro da Miniatura

[0078] Uma associação entre as janelas de origem e de destino é removida quando não mais necessária. Por exemplo, quando uma janela não mais se encontra ativa, a associação entre a janela e a miniatura pode ser feita ao não registrar a associação. Isto resulta em recursos livres. Além disso, as miniaturas podem ser ainda liberadas quando o processo de registro é retirado. Um exemplo de uma interface API para o não registro de uma miniatura é o gerenciador DwmUnregisterThumbnail, como segue:

```
HRESULT
DwmUnregisterThumbnail(
        HTHUMBNAIL hThumbnail
);
```

[0079] A Figura 6 é um fluxograma que ilustra um exemplo de um processo para a provisão de miniaturas, incluindo as miniaturas "vivas" ou dinâmicas, da presente invenção. Na etapa 601, uma interface API é provida por meio da qual uma aplicação pode solicitar uma primeira janela de nível mais alto ou uma janela de origem a fim de exibir uma miniatura que pode também se localizar em uma seção de uma segunda janela de nível mais alto ou janela de destino. Deste modo, a primeira janela de nível mais alto ou janela de origem pode ser renderizada como o objeto mais superior dentro da janela de destino. Um

exemplo não limitante desta implementação é uma janela ALT-TAB na qual uma miniatura correspondente a uma janela de origem é provida dentro da janela ALT-TAB (isto é, a janela de destino).

[0080] Na etapa 602, é feita uma chamada de Registro para o gerenciador DWM, que pode prover a alça de miniatura. Pode haver ainda múltiplas chamadas na mesma janela de origem ou na mesma janela de destino. No exemplo ALT-TAB, múltiplas chamadas na mesma janela de origem ou na mesma janela de destino resultam no redisposição das miniaturas dentro da janela ALT-TAB, assim como na provisão de uma representação maior e/ou estilizada de uma miniatura de uma aplicação quando a miniatura é selecionada. Em um outro exemplo, a ordem em Z da miniatura pode ser alterada por meio de chamadas de registro subseqüentes. Quando se deseja uma modificação da ordem em z da miniatura com relação a outros elementos de vídeo, uma nova miniatura pode ser registrada com uma outra chamada de registro de miniatura. Depois de as propriedades serem definidas para a nova miniatura que correspondem à miniatura antiga, a miniatura antiga pode ser não registrada. Sendo assim, neste exemplo, a ordem em z da miniatura pode ser alterada por meio de múltiplos registros da miniatura.

[0081] De maneira alternativa, pode haver múltiplas janelas de origem para a mesma janela de destino. Neste exemplo, as chamadas de múltiplas janelas de origem para uma janela de destino resultam na provisão de múltiplas miniaturas diferentes dentro da janela de destino.

[0082] Em ainda outro exemplo, as miniaturas dinâmicas podem ser exibidas dentro de outras miniaturas dinâmicas. Por exemplo, uma primeira miniatura dinâmica com uma ordem em z maior pode ser colocada em uma miniatura dinâmica maior de uma aplicação com uma ordem em z menor. Neste exemplo, as miniaturas podem ser embutidas dentro de outras miniaturas. Outros efeitos podem ser aplicados

às ambas miniaturas ou a quaisquer miniaturas embutidas. De maneira alternativa, as miniaturas de qualquer ordem em z podem ser em cascata dentro de outra miniatura de qualquer ordem em z. por exemplo, pelo menos uma miniatura (independente da ordem em z) pode ser colocada em uma miniatura dinâmica maior de uma aplicação de qualquer ordem em z. Da mesma forma, a miniatura dinâmica maior pode ser colocada em uma outra miniatura dinâmica de qualquer ordem em z. Sendo assim, as miniaturas podem ser em cascata independente da ordem em z.

[0083] Na etapa 603, o gerenciador DWM registra o par processo / ID para a miniatura. O gerenciador DWM pode ainda manter uma lista global de registros de miniatura nos quais cada aplicação fica emparelhada com uma alça de miniatura correspondente (HWND). A lista global pode ser mantida de diversas maneiras dependendo das necessidades do usuário. Por exemplo, uma maneira conveniente de se manter a lista global é indexar as miniaturas por meio da janela de destino (dstWindow) de tal modo que qualquer janela de destino desejada possa ser acessada de maneira eficiente. Ainda, as miniaturas de qualquer janela de destino em particular podem ser ainda organizadas em uma ordem lógica, como, por exemplo, em ordem de sobreposição ou baseada em uma ordem em z.

[0084] Em um outro exemplo, o gerenciador DWM pode "fotografar" os conteúdos de uma janela (isto é, salvar os conteúdos correntes da janela no momento) antes de a janela ser minimizada. O gerenciador DWM fotografa os conteúdos da janela, por exemplo, em um mapa de bits de um dimensionamento em particular. O dimensionamento pode ser determinado pela interface API de registro e é um dimensionamento apropriado para o mapeamento de miniaturas. Depois de a janela ser minimizada, os conteúdos da janela já estão preservados de tal modo que uma miniatura correspondente à janela de aplicação

possa ainda estar disponível. De maneira alternativa, as janelas minimizadas podem ser temporariamente "estacionadas" fora de tela ao invés de propriamente minimizadas. Desta maneira, as aplicações podem continuar a enviar informações e dados a fim de atualizar as miniaturas. Sendo assim, quando ocorre um registro de miniatura depois de uma janela de aplicação ser "minimizada", as miniaturas podem ainda ser registradas e atualizadas como se a janela não tivesse sido minimizada. De maneira alternativa, as miniaturas podem ser substituídas por ícones ou barras de título (caption bars) caso o registro de miniatura não seja exeqüível.

[0085] Na etapa 604, as atualizações das propriedades de miniatura são determinadas ou implementadas. Ao modificar ou atualizar as propriedades de miniatura, o gerenciador DWM pode, em conseqüência, modificar a lista global. Por exemplo, a aplicação de gerenciador DWM pode adicionar miniaturas ao seu próprio gráfico (por exemplo, a Figura 2A ou 3A) no qual o nó de janela de nível mais alto apropriado que representa a janela alvo é redesenhado a partir da janela de origem (isto é, a sreWindow) baseado em instruções de desenho adicionais. Quaisquer outras alterações específicas ou efeitos específicos à miniatura podem também ser incorporados, por exemplo, ajuste de escala, posicionamento, opacidade, invisibilidade ou efeitos especiais.

[0086] Na etapa 605, a miniatura é modificada baseada nas instruções recebidas. Por exemplo, uma interface API de UpdateThumbnail pode modificar a visibilidade do nó de miniatura, caso o parâmetro fVisible for verdadeiro (TRUE) ou a interface API UpdateThumbnail pode aumentar (ou diminuir) a opacidade da miniatura. Depois que todas as modificações desejadas à miniatura são realizadas, o processo do gerenciador DWM determina se o não registro da miniatura é desejado. Este processo de determinar se o não registro de uma miniatura é desejado pode ser feito separadamente do processo UpdateThumb-

nail. Neste caso, o não registro pode ocorrer devido a uma chamada de interface API de UnregisterThumbnail (não mostrada). Na etapa 606, é determinado se uma interface API de não registro foi recebida pelo processo do gerenciador DWM. Se a interface API de não registro foi recebida, o emparelhamento do processo e da ID de miniatura correspondente é removido (por exemplo, da lista global dos registros de miniatura mantidos no processo do gerenciador DWM). Na etapa 607, o fluxo de instruções de janela de nível mais alto é re-emitido como o resultado de uma miniatura não registrada. Em conseqüência, a miniatura é removida.

[0087] Em um exemplo, um cursor paira sobre um botão da barra de tarefas resultante no vídeo da miniatura da aplicação correspondente. Quando o cursor paira sobre o botão da barra de tarefas, o processo cria uma nova alça de janela (HWND) de nível mais alto que agrupa uma miniatura. O processo chama uma interface API de rego que usa a alça de janela (HWND) da aplicação como a origem do conteúdo para a miniatura (isto é, a srcWindow). A nova alça HWND de nível mais alto que agrupa a miniatura é a janela de destino (a dstWindow). O conteúdo da janela de origem, ou uma porção desejada da mesma, é em seguida colocado na janela de destino a fim de exibir a miniatura. A miniatura pode conter todo o conteúdo de janela de origem ou pode conter apenas uma porção predeterminada ou selecionada, conforme apresentado acima. A interface API Atualizar Miniatura (Update Thumbnail) pode modificar ou movimentar a miniatura. Por exemplo, a miniatura pode ser reposicionada com relação ao ícone correspondente, ou efeitos especiais podem ser aplicados à miniatura (por exemplo, transparência, invisibilidade, distorção, etc.). Quando o cursor se movimenta para fora do botão da barra de tarefas, a alça HWND de nível mais alto que agrupa a miniatura pode ser não registrada e, desta forma, a miniatura fica oculta. De maneira alternativa, a

miniatura pode ser atualizada para invisível de tal modo que a miniatura seja ainda registrada e possa ser invocada em seguida. Em um método alternativo, cada janela após criação pode ser atribuída a uma miniatura criada para a mesma com a barra de tarefas. Neste exemplo, o código da barra de tarefas pode ser usado para manipular a visibilidade das miniaturas e o retângulo de destino para colocação da miniatura, por exemplo.

[0088] Entenda-se que os aspectos da presente invenção podem assumir muitas formas e modalidades. As modalidades mostradas no presente documento pretendem ilustrar, ao invés de limitar a presente invenção, sendo apreciado que podem ser feitas variações sem se afastar do espírito ou âmbito da presente invenção. Embora tenham sido mostradas e descritas modalidades ilustrativas da presente invenção, uma ampla faixa de modificações, mudanças e substituições é pleiteada na apresentação acima e, em algumas instâncias, determinadas características da presente invenção podem ser empregadas sem um correspondente uso de outras características. Por conseguinte, é apropriado que as reivindicações em apenso sejam construídas de uma forma ampla e consistente com o âmbito da presente invenção.

REIVINDICAÇÕES

1. Método implementado em computador para exibir uma janela (221,222,321,322,501,502) em um dispositivo de vídeo compreendendo as etapas de:

exibir uma janela de origem (221,222,321,322,501,502), a janela de origem compreendendo um primeiro conteúdo capaz de ser modificado;

minimizar a janela de origem (221,222,321,322,501,502);

exibir uma janela de destino (505) na mesma tela,

caracterizado pelo fato de que pelo menos uma porção da janela de destino (505) contém uma miniatura (323,401,404,415,507,508,509), a miniatura incluindo um segundo conteúdo correspondendo a pelo menos uma porção do conteúdo da janela de origem (221,222,321,322,501,502);

automaticamente exibir, na miniatura (323,401,404,415,507,508, 509), um ultimo conteúdo exibido pela janela de origem (221,222,321,322, 501,502) antes da minimização; e

modificar o segundo conteúdo da miniatura (323,401,404,415,507,508,509) a qualquer momento que o primeiro conteúdo da janela de origem (221,222,321,322,501,502) se altera;

receber um primeiro parâmetro para definir pelo menos uma porção da janela de origem (221,222,321,322,501,502), em que a modificação compreende determinar pelo menos uma porção da janela de origem para conduzir para a miniatura (323,401,404,415,507,508,509) com base no primeiro parâmetro; e

receber um segundo parâmetro para definir pelo menos uma porção da janela de destino (505), em que a modificação compreende ainda determinar pelo menos uma porção da janela de destino (505) para renderizar a miniatura (323,401,404,415,507,508,509) com base no segundo parâmetro;

em que a modificação da janela de origem (221,222,321,322, 501,502) é exibida na miniatura (323,401,404,415,507,508,509).

2. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** a miniatura (323,401,404,415,507,508,509) é modificada em tempo real como o primeiro conteúdo da janela de origem (221,222,321,322, 501,502) é modificado.

3. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** modificar o segundo conteúdo da miniatura (323,401,404,415, 507,508,509) compreende ainda modificar a visibilidade da miniatura.

4. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** a etapa de modificar o segundo conteúdo da miniatura (323,401,404,415,507,508,509) ocorre automaticamente sempre que o primeiro conteúdo da janela de origem (221,222,321,322,501,502) muda.

5. Sistema para exibir uma janela (221,222,321,322,501,502) em um dispositivo de exibição **caracterizado pelo fato de que** compreende:

uma tela para exibir uma janela de origem (221,222,321,322, 501,502) compreendendo conteúdo que é capaz de ser modificado e uma janela de destino (505) exibida na tela;

um gerenciador para registrar uma miniatura (323,401,404,415, 507,508,509), a miniatura estando na janela de destino (505) e correspondendo a pelo menos uma parte da janela de origem, o registro compreendendo:

receber uma alça de janela para a janela de origem;

estabelecer uma associação entre a janela de origem e a miniatura de modo que pelo menos uma porção do conteúdo da janela de origem seja conduzida para dentro da miniatura;

receber um primeiro parâmetro para definir pelo menos uma porção da janela de origem, em que modificar compreende determinar pelo menos uma porção da janela de origem para conduzir para dentro da miniatura com base no primeiro parâmetro; e

receber um segundo parâmetro para definir pelo menos uma porção da janela de destino, em que modificar compreende ainda determinar pelo menos uma porção da janela de destino para renderizar a miniatura com base no segundo parâmetro; e

recebendo uma alça única representando a associação entre a janela de origem e a miniatura;

um processador; e

uma memória que armazena instruções executáveis do computador que, quando executadas pelo processador, fazem com que o sistema execute um método para modificar o segundo conteúdo da miniatura sempre que o conteúdo da janela de origem muda, o método compreendendo as etapas de:

determinar pelo menos uma porção da janela de origem para conduzir para dentro da miniatura com base em um primeiro parâmetro para definir pelo menos uma porção da janela de origem; e

determinando a miniatura para renderizar com base em um segundo parâmetro para definir pelo menos uma porção da janela de destino para renderizar a miniatura.

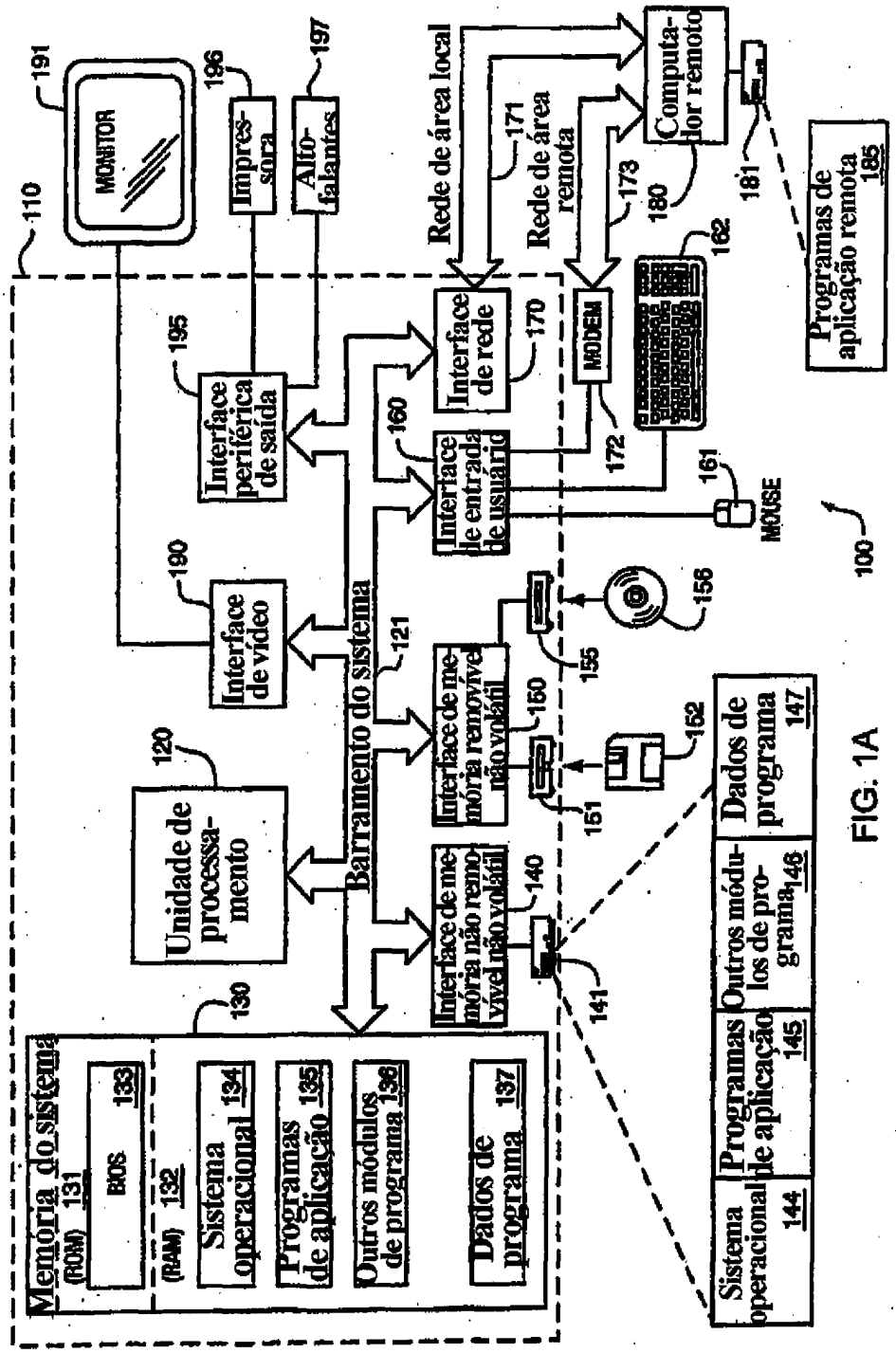
Memória do sistema (ROM) 131
BIOS 133
(RAM) 132
Sistema operacional 134
Programas de aplicação 135
Outros módulos de programa 136
Dados de programa 137
Unidade de processamento 120
130
Barramento do sistema 121
Interface de vídeo 190
Interface periférica de saída 195
Interface de memória não removível não volátil 140
Interface de memória removível não volátil 150
Interface de entrada de usuário 160
Interface de rede 170
MODEM 172
MONITOR 191
Impressora 196
Alto-falantes 197
MOUSE 161
Rede de área local 171
Rede de área remota 173
Computador remoto 180
181
Programas de aplicação remota 185
182
100
110
141
151
155
156
162
Sistema operacional 144
Programas de aplicação 145
Outros módulos de programa 146
Dados de programa 147

FIG. 1A

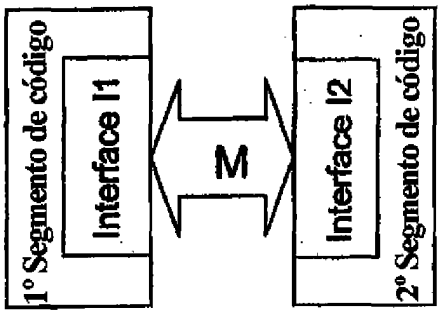

FIG. 1C

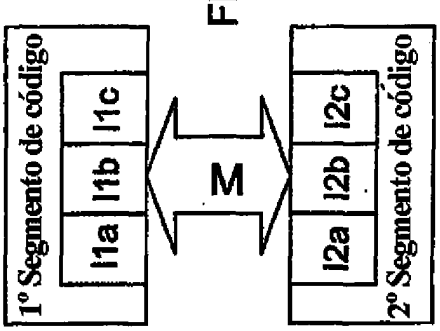

FIG. 1E

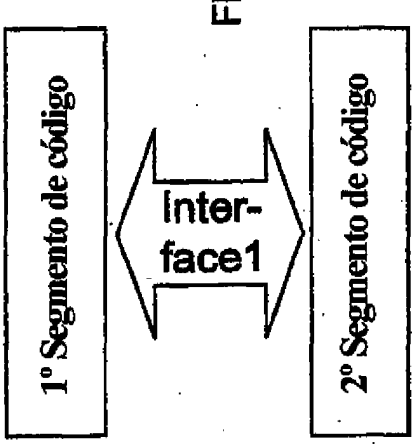

FIG. 1B

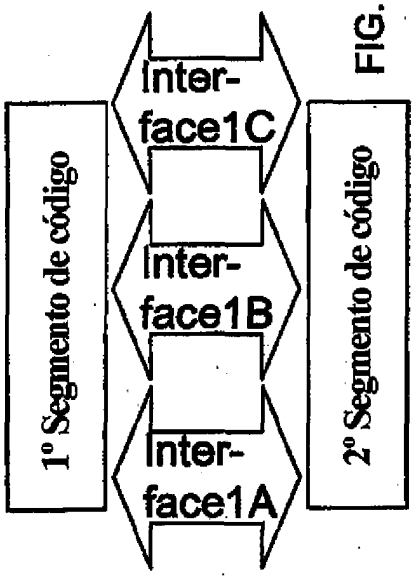

FIG. 1D

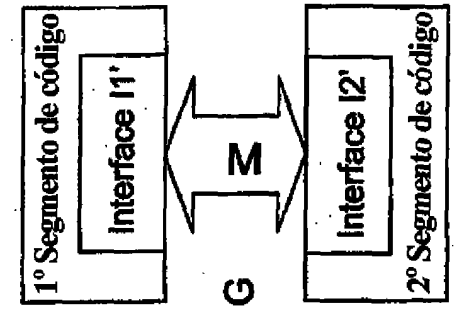

FIG. 1G

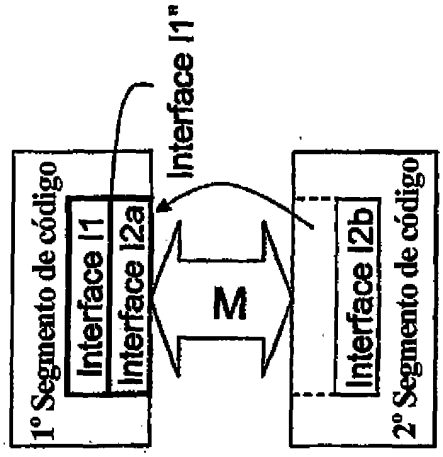

FIG. 1I

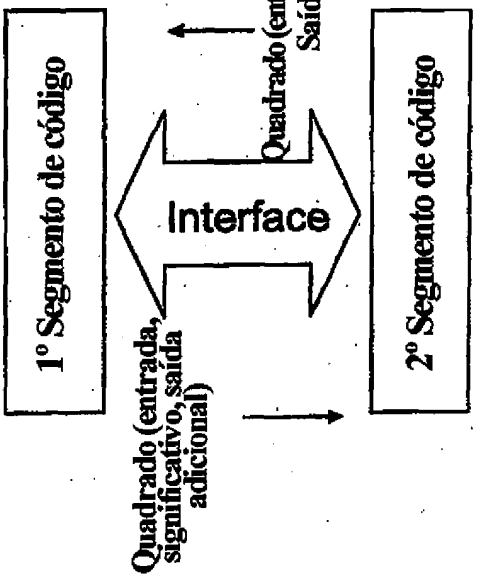

FIG. 1F

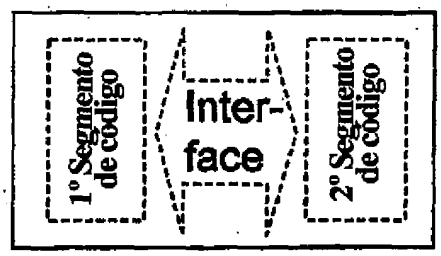

FIG. 1H

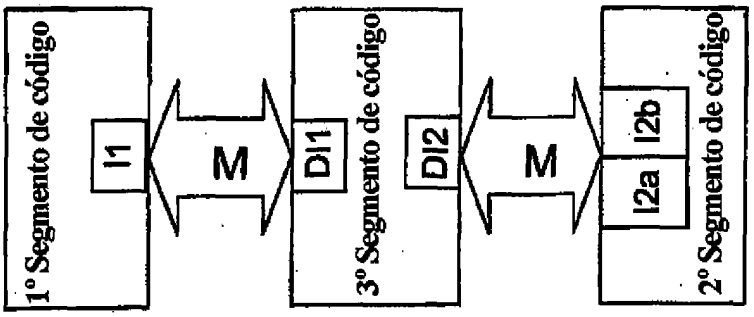
1º Segmento de código
I1
M
DI1
3º Segmento de código
DI2
M
I2b
I2a
2º Segmento de código

FIG. 1K

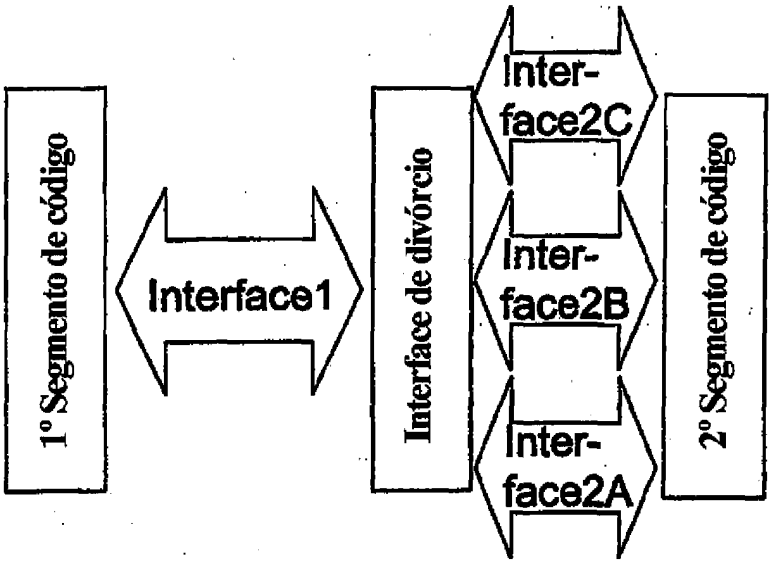
1º Segmento de código
Interface1
Interface de divórcio
Inter-face2C
Inter-face2B
Inter-face2A
2º Segmento de código

FIG. 1J

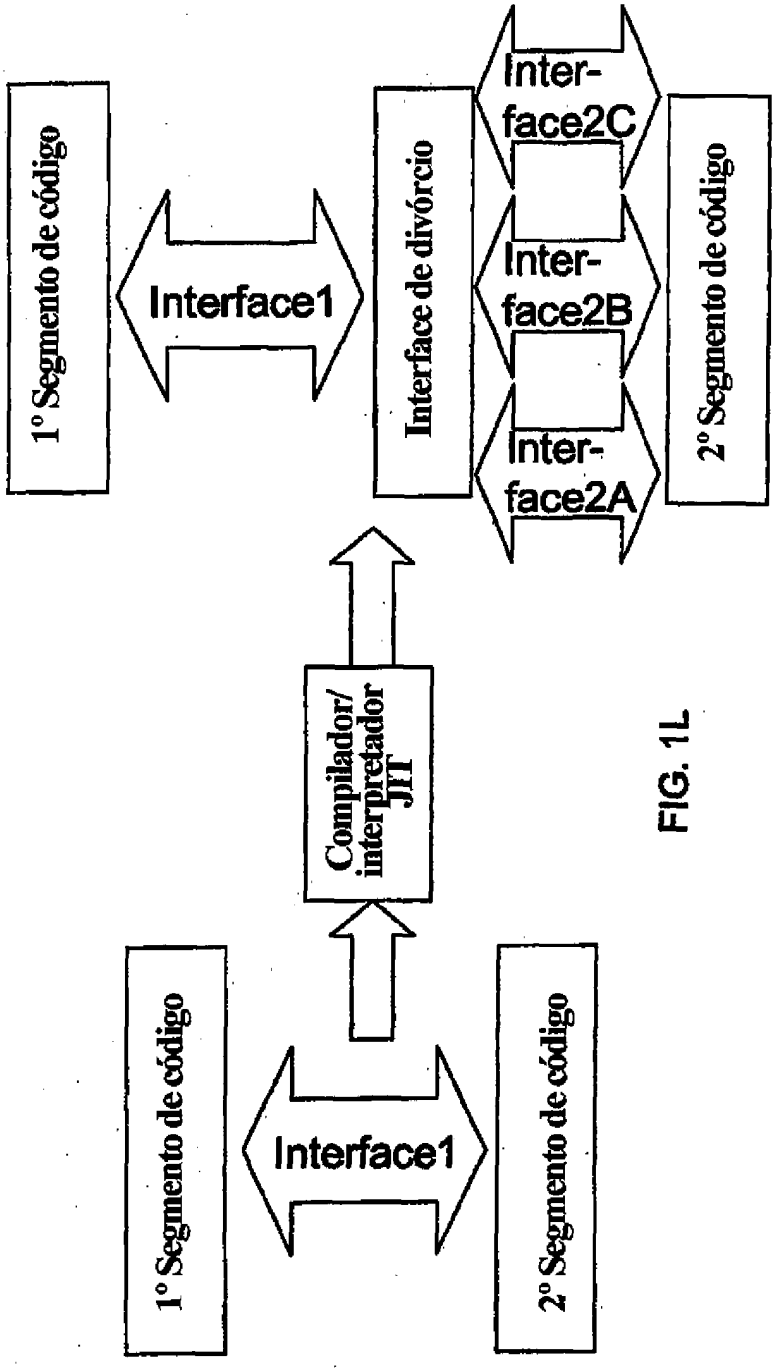
1° Segmento de código
Interface1
Interface de divórcio
Inter-face2C
Inter-face2B
Inter-face2A
2° Segmento de código
Compilador/ interpretador JIT
1° Segmento de código
Interface1
2° Segmento de código

FIG. 1L

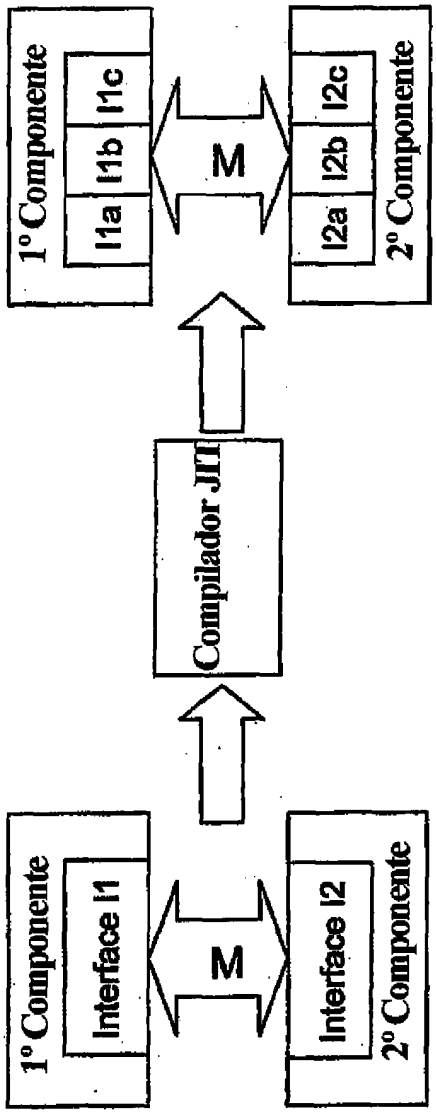
1° Componente
I1a
I1b
I1c
M
I2a
I2b
I2c
2° Componente
Compilador JIT
1° Componente
Interface I1
M
Interface I2
2° Componente

FIG. 1M

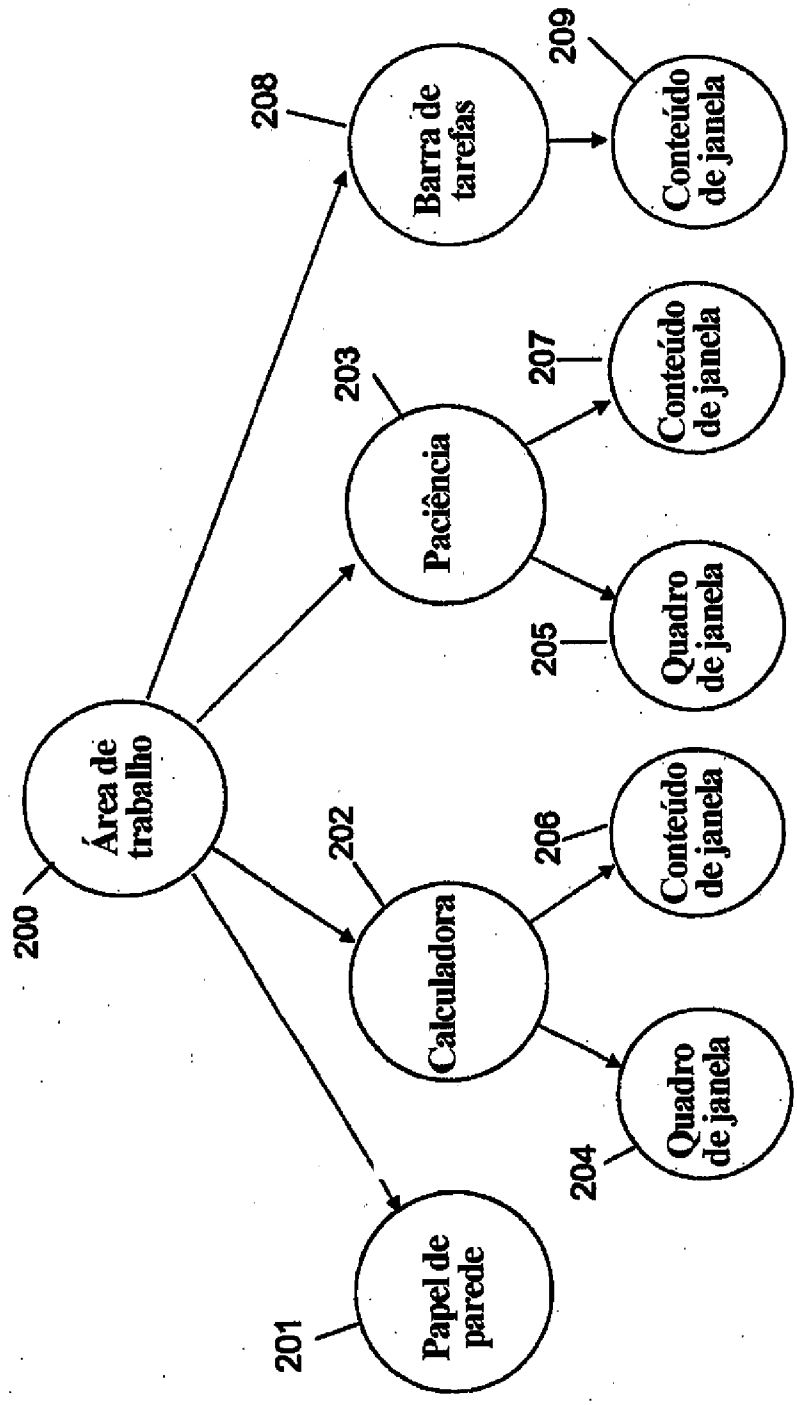
Área de trabalho
200
Papel de parede
201
Calculadora
202
Paciência
203
Quadro de janela
204
Quadro de janela
205
Conteúdo de janela
206
Conteúdo de janela
207
Barra de tarefas
208
Conteúdo de janela
209

FIG. 2A

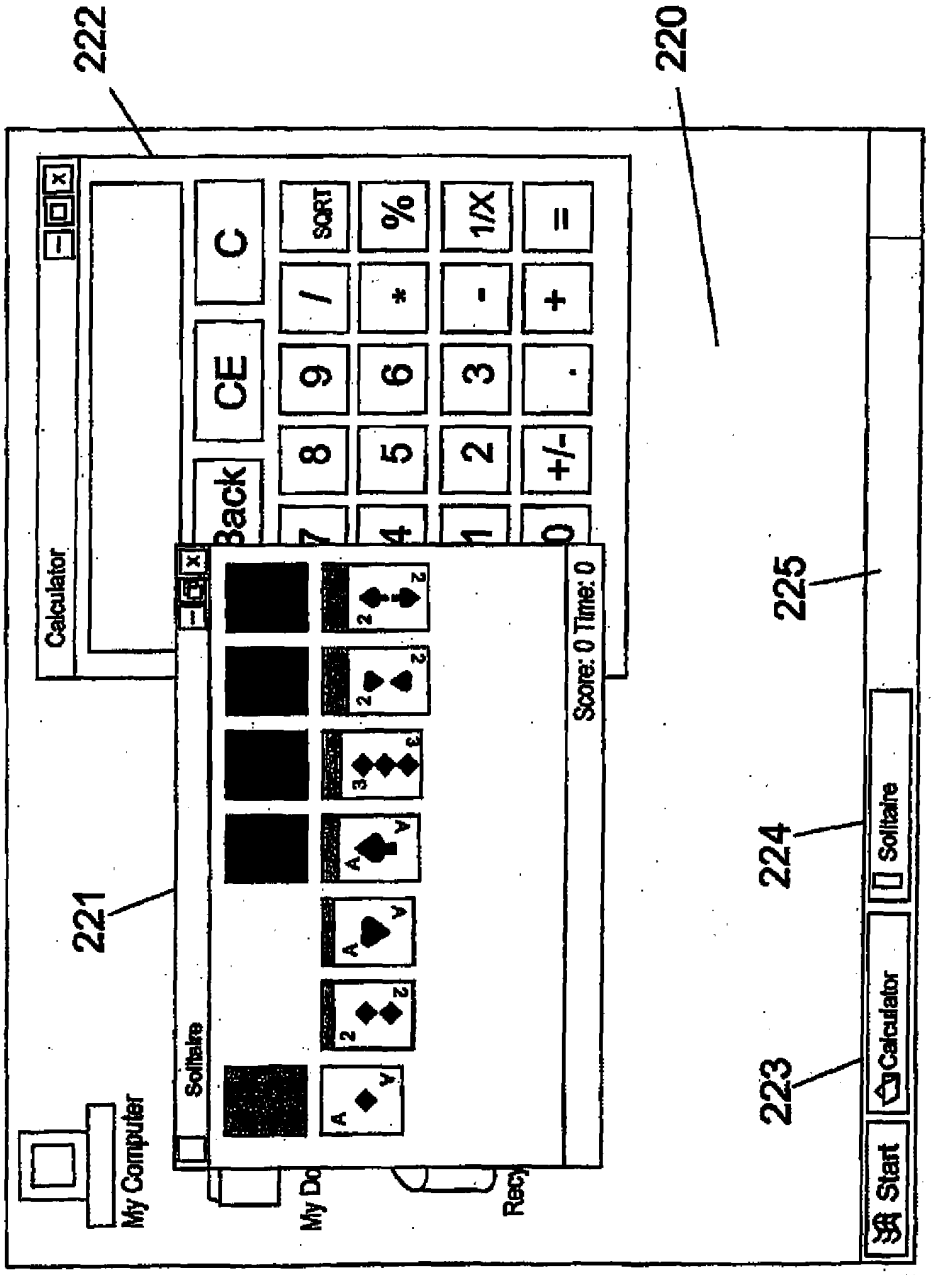
222
220
225
224
223
221
Calculator
C
CE
Back
SQRT
%
1/X
=
/
*
-
+
9
6
3
8
5
2
+/-
Solitaire
Score: 0 Time: 0
My Computer
Start
Calculator
Solitaire

Fig. 2B

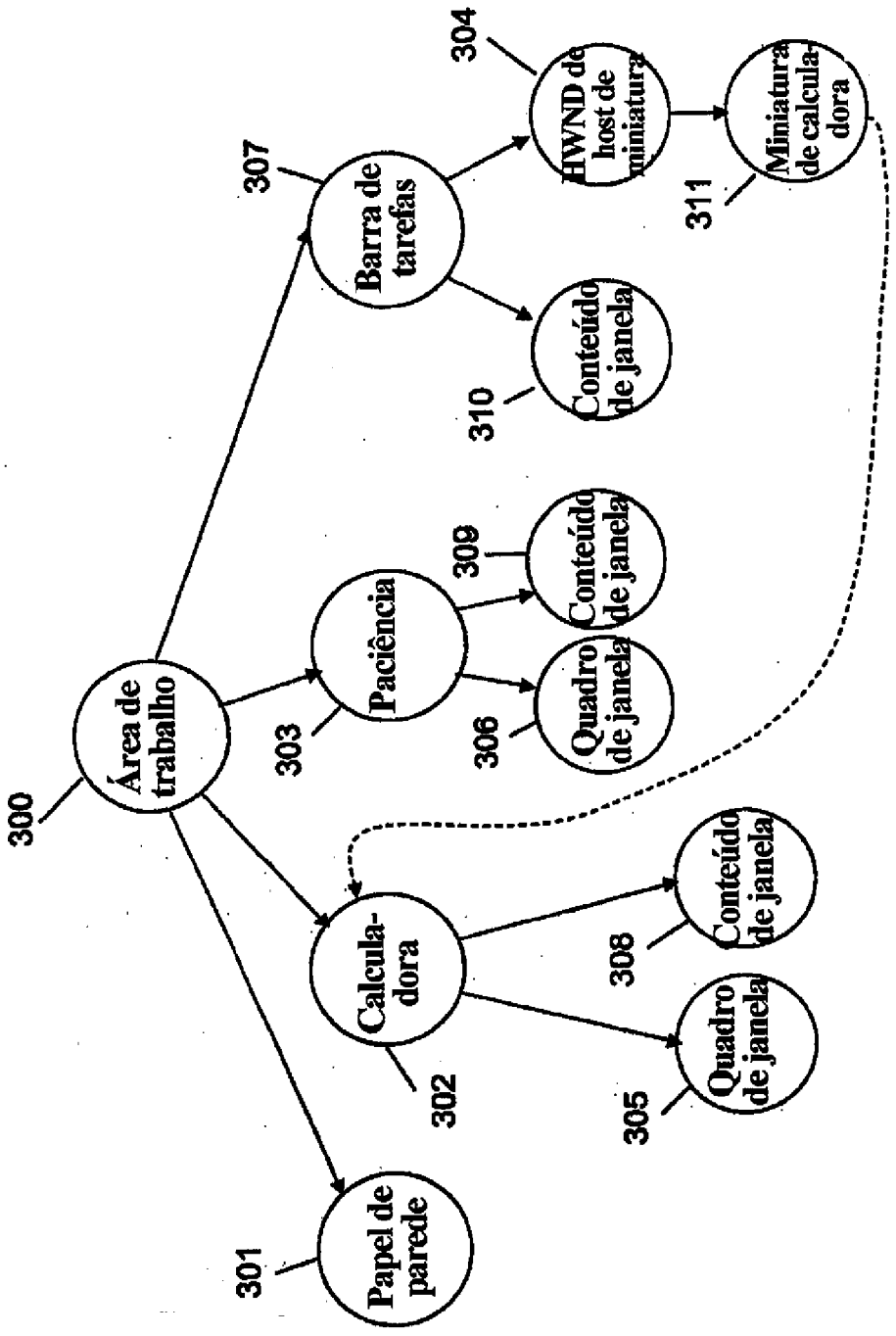
300
Área de trabalho
301
Papel de parede
302
Calcula-dora
303
Paciência
304
HWND de host de miniatura
305
Quadro de janela
306
Quadro de janela
307
Barra de tarefas
308
Conteúdo de janela
309
Conteúdo de janela
310
Conteúdo de janela
311
Miniatura de calcula-dora

FIG. 3A

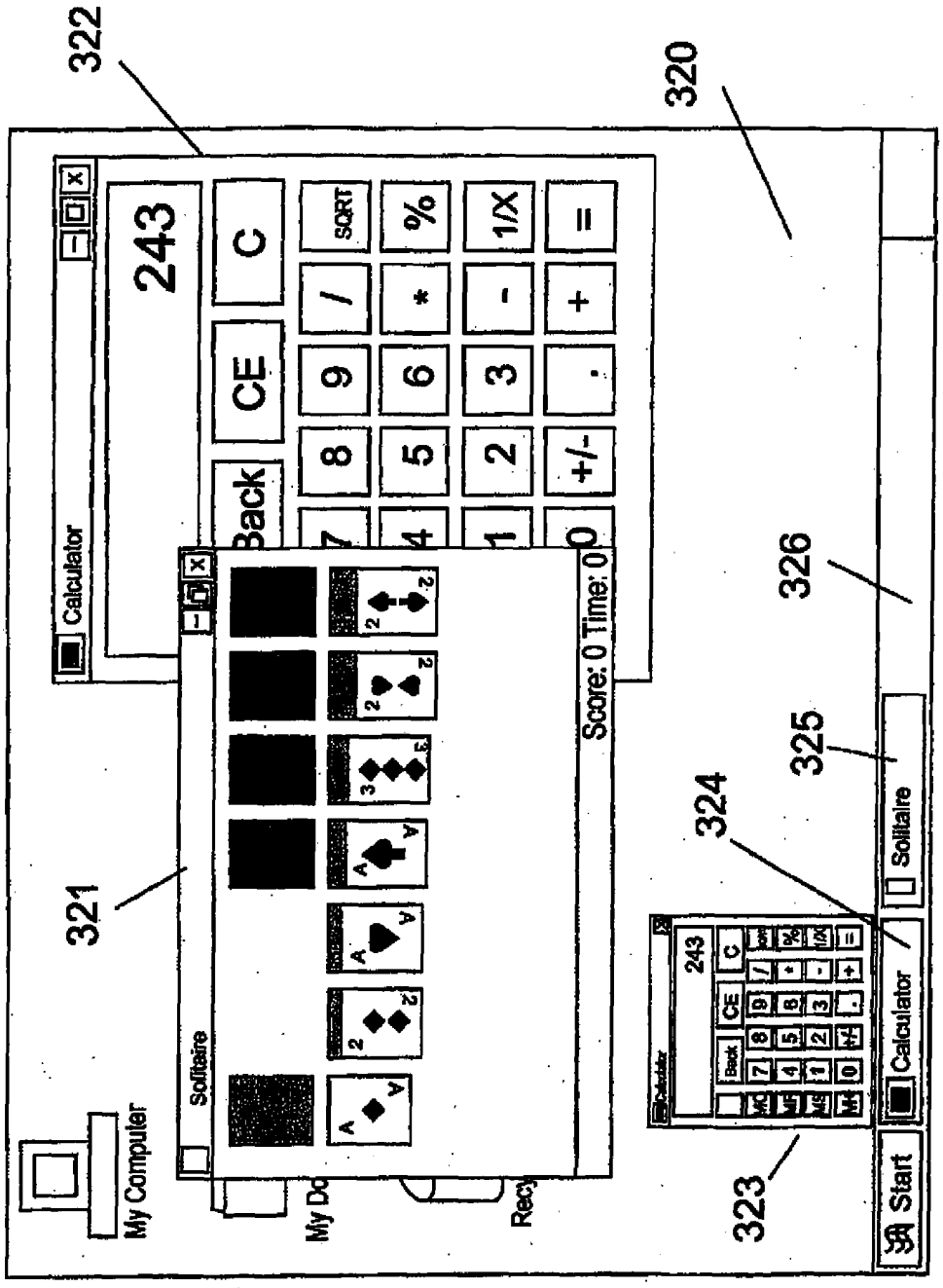
322
320
243
C
CE
Back
SQRT
%
1/X
=
/
*
-
+
9
6
3
.
8
5
2
+/-
0
Calculator
Solitaire
Score: 0 Time: 0
My Computer
Start
321
324
325
326
323

Fig. 3B

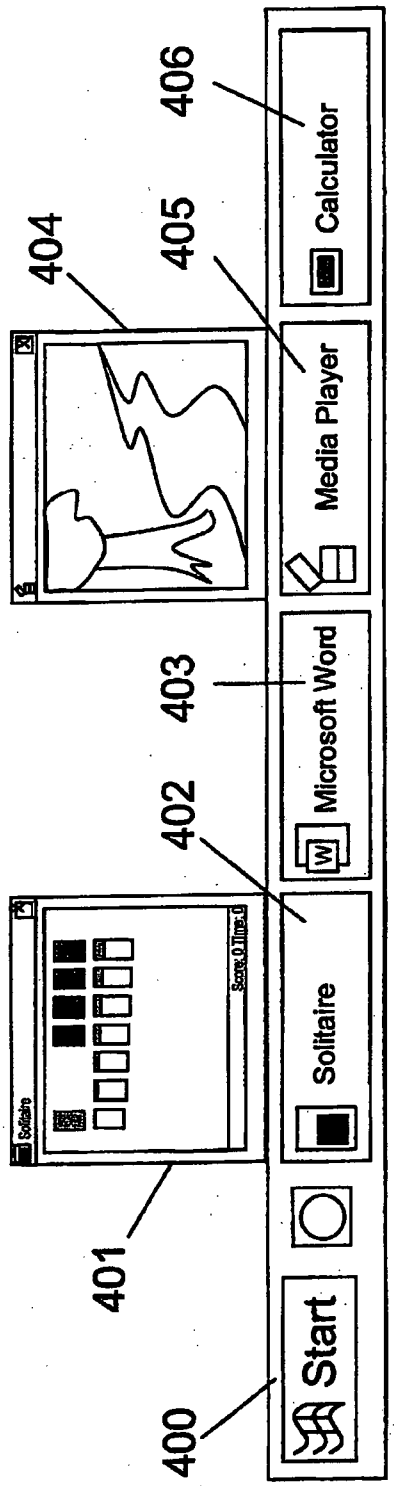
Start
Solitaire
Microsoft Word
Media Player
Calculator
400
401
402
403
404
405
406

Fig. 4A

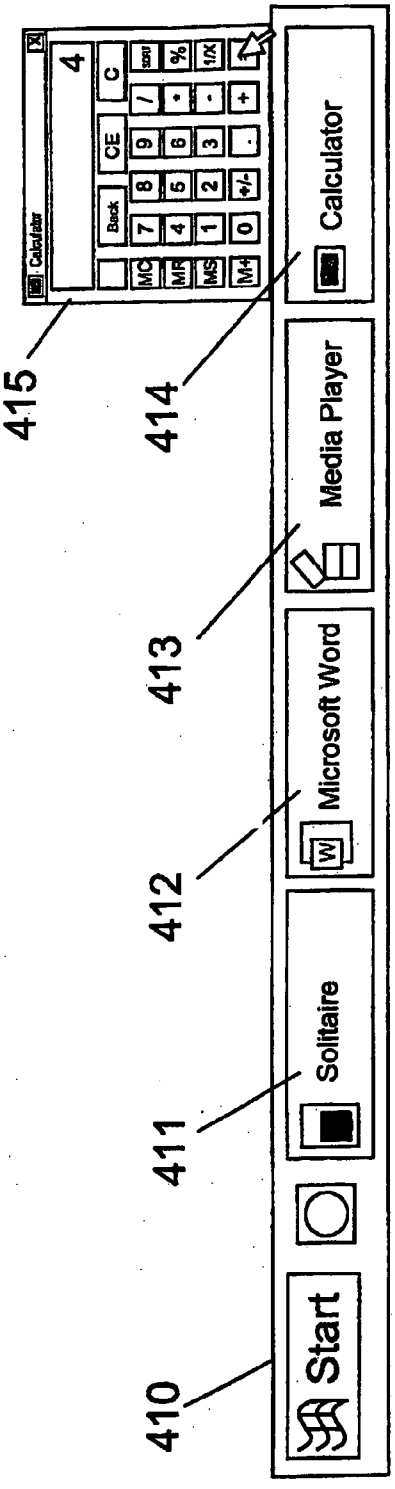
Start
Solitaire
Microsoft Word
Media Player
Calculator
410
411
412
413
414
415

Fig. 4B

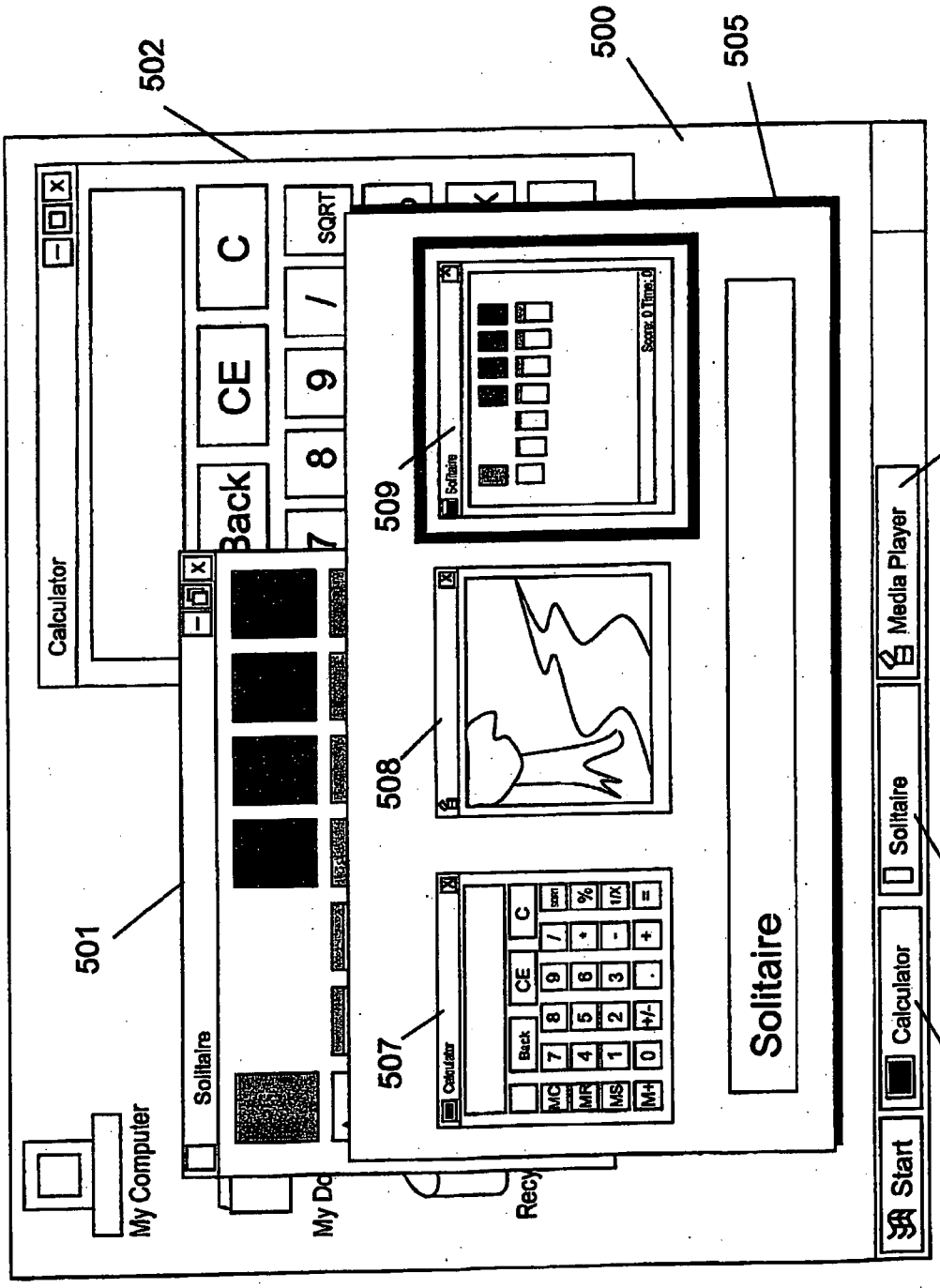
500
501
502
503
504
505
506
507
508
509
Calculator
Solitaire
Media Player
Start
My Computer
Back
CE
C
SQRT
Solitaire

Fig. 5

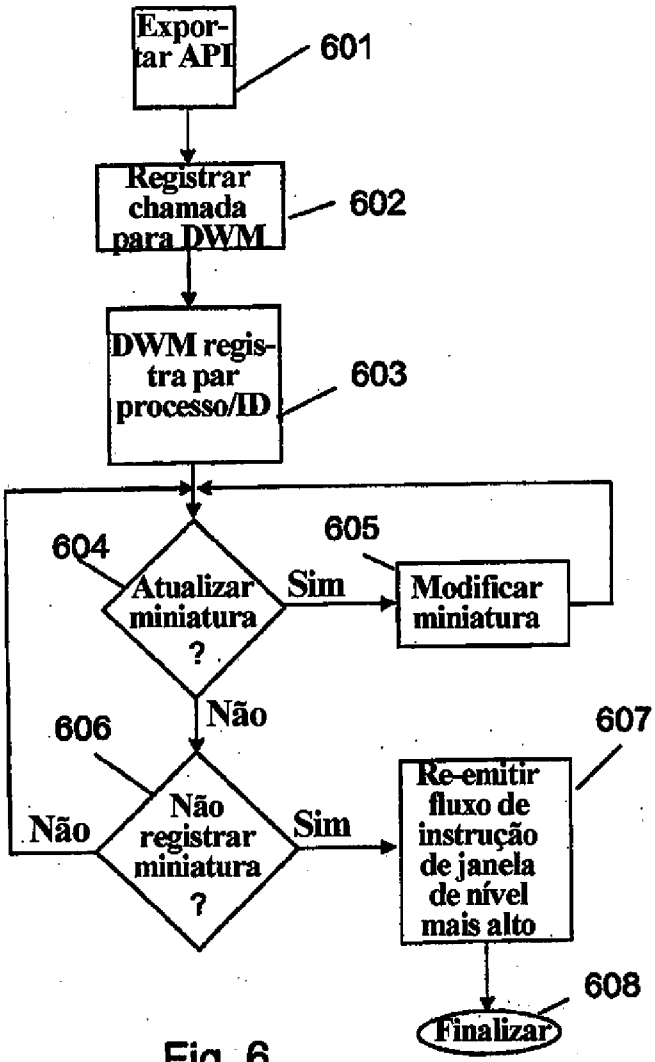
Expor-
tar API
601
Registrar
chamada
para DWM
602
DWM regis-
tra par
processo/ID
603
604
Atualizar
miniatura
?
Sim
605
Modificar
miniatura
Não
606
Não
Não
registrar
miniatura
?
Sim
607
Re-emitir
fluxo de
instrução
de janela
de nível
mais alto
608
Finalizar

Fig. 6