



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

(21) **PI0708917-1 A2**

(22) Data de Depósito: 21/02/2007
(43) Data da Publicação: 14/06/2011
(RPI 2110)



(51) *Int.Cl.:*
G06F 9/44 2006.01

(54) Título: **TRATAMENTO DE FALHA ASSÍNCRONA EM PROGRAMAS CENTRALIZADOS EM PROCESSO**

(30) Prioridade Unionista: 30/03/2006 US 11/393.093

(73) Titular(es): Microsoft Corporation

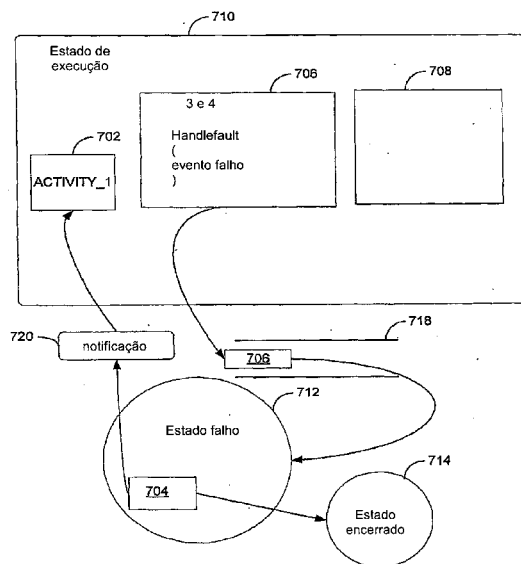
(72) Inventor(es): Akash J. Sagar, Bob Schmidt, Dharma Shukla, Karthik Raman, Mayank Mehta, Nathan Talbert

(74) Procurador(es): Nellie Anne Daniel Shores

(86) Pedido Internacional: PCT US2007004642 de 21/02/2007

(87) Publicação Internacional: WO 2007/120391 de 25/10/2007

(57) **Resumo:** TRATAMENTO DE FALHA ASSÍNCRONA EM PROGRAMAS CENTRALIZADOS EM PROCESSO É divulgado o tratamento de falha assíncrona para um fluxo de trabalho. É definido um autômato de estado para uma atividade no fluxo de trabalho. O autômato de estado inclui pelo menos um estado de execução, um estado falho e um estado encerrado, e classifica uma vida útil de execução da atividade. A atividade é definida para incluir itens de trabalho e inclui uma hierarquia de execução para os itens de trabalho. Cada item de trabalho inclui uma operação para executar uma parte da atividade. Cada item de trabalho é transicionado ao estado de execução. A operação incluída dos itens de trabalho transicionados é executada no estado de execução. Um ou mais dos itens de trabalho transicionados são identificados em resposta ao evento falho em função da hierarquia de execução e da operação incluída. O evento falho é tratado de forma assíncrona pela transição dos um ou mais itens de trabalho identificados ao estado falho durante a execução da operação incluída dos itens de trabalho transicionados restantes.





“TRATAMENTO DE FALHA ASSÍNCRONA EM PROGRAMAS CENTRALIZADOS EM PROCESSO”

ANTECEDENTES DA INVENÇÃO

Programas orientados a processo ou centralizados em processo têm evoluído para

5 habilitar o processamento de instruções complexas que modelam interações do mundo real. Programas centralizados em processo se espelham em processos do mundo e real e se espelham em interações entre entidades do mundo real. Sistemas existentes tentam mapear problemas empresariais em fluxos de trabalho de alto nível pela modelagem do problema empresarial. Entretanto, fluxos de trabalho do mundo real variam em uma multiplicidade de

10 dimensões, tais como, (a) complexidade de execução e de modelagem, (b) conhecimento da estrutura do fluxo no momento do projeto, (c) estaticamente definido ou *ad-hoc* / dinâmico, (d) facilidade da autoria e da edição do fluxo em vários pontos no seu ciclo de vida e (e) fraca ou forte associação da lógica empresarial com o processo central do fluxo de trabalho. Modelos existentes deixam de conciliar estes fatores.

15 Adicionalmente, a maior parte dos modelos de fluxo de trabalho existentes tem base tanto em abordagens com base em linguagem (por exemplo, BPEL4WS, XLANG/S e WSFL) quanto em abordagens com base em aplicação. Abordagens com base em linguagem são linguagens de fluxo de trabalho de alto nível com um conjunto fechado de construções pré-definidas que ajudam a modelar o processo do fluxo de trabalho para o

20 usuário / programador. As linguagens de fluxo de trabalho portam toda a informação semântica para o conjunto fechado de construções para habilitar o usuário a construir um modelo de fluxo de trabalho. Entretanto, as linguagens não são extensíveis pelos desenvolvedores e representam um conjunto fechado de elementos básicos que constitui o modelo do fluxo de trabalho. As linguagens são amarradas ao compilador da linguagem

25 embarcado pelo revendedor do sistema de fluxo de trabalho. Somente o revendedor do produto do sistema de fluxo de trabalho pode estender o modelo pela extensão da linguagem com um conjunto inédito de construções em uma versão futura do produto. Frequentemente, isto exige atualização do compilador associado com a linguagem. Além do mais, usualmente, as linguagens não expõem ou definem declarativamente funções ou

30 operações que podem ser usadas de forma fácil e eficiente por outros programas.

Abordagens com base em aplicação são aplicações que têm capacidades de fluxo de trabalho na aplicação para resolver um problema específico de domínio. Estas aplicações não são verdadeiramente extensíveis nem têm um modelo programável.

35 Além do mais, com as abordagens existentes, as questões de complexidade, de conhecimento prévio, de fluxos de trabalho dinâmicos, de facilidade de autoria e de força de associações com a lógica empresarial e com fluxos de trabalho centrais não são adequadamente abordadas. Não há estruturas projetistas de fluxo de trabalho extensíveis,

customizáveis e ré-hospedáveis disponíveis para construir projetistas de fluxo de trabalho visual para modelar diferentes classes de fluxos de trabalho. Sistemas existentes carecem de uma experiência de projeto de fluxo de trabalho no estilo rápido desenvolvimento de aplicação (RAD), que permite que usuários projetem graficamente o processo do fluxo de trabalho e associem a lógica empresarial em uma linguagem de programação da escolha do desenvolvedor.

Também, os processos de fluxo de trabalho lidam com preocupações de atalho ortogonal e confuso que abrangem múltiplas etapas de um modelo de processo de fluxo de trabalho. Por exemplo, embora partes dos processos de fluxo de trabalho sejam projetadas para participar de transações de longo prazo, outras partes do mesmo processo são projetadas para execução concorrente ou para acessar um recurso compartilhado. Em função das deficiências de projeto, sistemas existentes deixam de fornecer intercalação das linhas de execução que habilitam usuários a projetar execução síncrona ou intercalada das atividades. Ainda outras partes do mesmo processo de fluxo de trabalho exigem rastreamento, embora outras partes tratem de exceções em nível empresarial ou da aplicação. Há uma necessidade de aplicar certos comportamentos a uma ou mais partes de um processo do fluxo de trabalho.

Algumas abordagens da modelagem do fluxo de trabalho são impraticáveis, já que elas exigem uma completa descrição com base no fluxo de todo o processo empresarial, incluindo todas as exceções e intervenções humanas. Algumas destas abordagens fornecem funcionalidade adicional à medida que as exceções surgem, embora outras abordagens empreguem exclusivamente uma abordagem com base em restrição em vez de uma abordagem com base no fluxo para modelar um processo empresarial. Sistemas existentes implementam tanto a abordagem com base no fluxo quanto a abordagem com base em restrição. Tais sistemas são muito inflexíveis para modelar muitas situações empresariais comuns. Estes sistemas também carecem da capacidade de tratar de forma assíncrona exceções ou cancelamentos.

SUMÁRIO DA INVENÇÃO

Modalidades da invenção habilitam o tratamento de falha assíncrona ou de exceção por ter um estado falho em um autômato de estado que define a vida útil de execução de uma atividade no fluxo de trabalho. Por ter o estado falho, aspectos da invenção habilitam desenvolvedores ou programas a projetar declarativamente programas para tratamento de exceção ou de falha, de maneira tal que partes do programa ou da atividade possam estar em tratamento de falha no estado falho, embora outras partes do programa ou da atividade possam estar não afetados pela exceção ou pelo evento falho.

Modalidades alternativas da invenção habilitam a propagação ou a transmissão de uma notificação do tratamento de falha. Em uma ainda outra modalidade alternativa, tal

propagação ou transmissão da notificação pode ser suprimida ou inibida. Além do mais, uma modalidade alternativa adicional responde à entrada de um usuário para tratar operações pós-falha ou pós-exceção.

Este Sumário é fornecido para introduzir uma seleção de conceitos de uma forma simplificada que é adicionalmente descrita na seguinte Descrição Detalhada. Não pretende-se que este Sumário identifique recursos chaves ou recursos essenciais do assunto em questão reivindicado, nem pretende-se que seja usado como um auxílio na determinação do escopo do assunto em questão reivindicado.

Outros recursos ficarão, em parte, aparentes e serão, em parte, salientados a seguir.

DESCRIÇÃO RESUMIDA DOS DESENHOS

A figura 1 é um diagrama de blocos que ilustra um paradigma de programação existente.

A figura 2 é um diagrama de blocos exemplar que ilustra uma virtualização de uma estrutura de projeto de fluxo de trabalho de acordo com uma modalidade da invenção.

A figura 3 é um diagrama exemplar que ilustra um fluxo de trabalho exemplar de acordo com uma modalidade da invenção.

A figura 4 é um diagrama que ilustra um ambiente de computação exemplar de um sistema para processar atividades de fluxo de trabalho de acordo com uma modalidade da invenção.

A figura 5 é um diagrama que ilustra uma estrutura hierárquica de uma atividade do fluxo de trabalho de acordo com uma modalidade da invenção.

A figura 6 é um diagrama que ilustra um autômato de estado exemplar que descreve a vida útil da execução de uma atividade de acordo com uma modalidade da invenção.

As figuras 7A a 7E são diagramas de blocos que ilustram um tratamento assíncrono dos eventos de falha de um fluxo de trabalho de acordo com uma modalidade da invenção.

A figura 8 é um fluxograma que ilustra um método para o tratamento assíncrono de um evento falho para uma atividade de um fluxo de trabalho de acordo com uma modalidade da invenção.

A figura 9 é um diagrama de blocos que ilustra uma mídia legível por computador exemplar na qual aspectos da invenção podem ser armazenados.

O Apêndice A ilustra uma implementação exemplar da elevação declarativa de uma exceção de acordo com uma modalidade da invenção.

Caracteres de referência correspondentes indicam partes correspondentes por todos os desenhos.

DESCRIÇÃO DETALHADA

Primeiro, em relação à figura 1, um diagrama de blocos ilustra um paradigma de programação existente para projetar programas para atividades centralizadas em processo, tal como um fluxo de trabalho. Por exemplo, o diagrama mostra um modelo de virtualização de três níveis do paradigma de programa existente com um nível de um ambiente de execução gerenciada sendo o nível mais superior, e uma unidade de processamento sendo o nível mais inferior. Neste sistema de projeto de programação, mesmo no nível do ambiente de execução gerenciada, programas, especialmente programas centralizados em processo que tratam processos do fluxo de trabalho, carecem de capacidade e de eficiência para conciliar complexas interações entre os processos em um fluxo de trabalho.

Versados na técnica sabem que certas restrições estão associadas com software ou programas de aplicação de projeto. Neste exemplo, na gravação de um programa de software do sistema operacional 104, os códigos ou rotinas de programação dependem do tipo ou da configuração das unidades de processamento 102, sendo específico, ao tipo de arquitetura computacional (por exemplo, compatível com IBM®, com computadores APPLE® ou com outros sistemas), ou de outras restrições. Além do mais, tipicamente, linguagens de programação precisam identificar e utilizar precisamente estruturas de dados, tais como pilhas, árvore binária, base de linha de execução ou outras estruturas específicas de hardware, para o sistema operacional 104 funcionar apropriadamente.

Ao lidar com complexos processos de fluxo de trabalho, aplicações existentes usam um conceito de um ambiente de execução gerenciada 106 (por exemplo, um ambiente de tempo de execução em que programas podem compartilhar funções ou classes orientadas a objetos comuns) nos quais programas escritos em uma linguagem de programação podem chamar funções em outros programas escritos em uma linguagem de programação diferente. Em um ambiente de execução como este, estes programas em diferentes linguagens de programação são compilados em uma linguagem intermediária de maneira tal que o ambiente de execução gerenciada 106 possa expor parâmetros, argumentos ou esquemas ou funções a diferentes programas, para que os programas possam interagir uns com os outros.

Embora este ambiente de execução 106 crie um ambiente de comunicação comum entre programas, o ambiente de execução 106 inclui várias rígidas exigências que podem não ser adequadas para tratar a complexidade e a capacidade dos programas centralizados em processo. Por exemplo, o ambiente de execução 106 exige que programas sejam confirmados em um formato de arquivo específico. O ambiente de execução 106 também exige que funções ou operações nos programas usem um conjunto fixo de funções ou uma classe de funções definidos pelo ambiente de execução 106.

Modalidades da invenção construídas em uma fundação ou estrutura extensível 202 da figura 2 superam as deficiências do modelo de programação existente. Pela

permissão de que programas sejam escritos em qualquer linguagem de programação e compostos em qualquer formato de arquivo, aspectos da invenção habilitam desenvolvedores de programa a projetar programas com funções específicas sem comprometer suas funcionalidades e especificidades. Pela definição das atividades, tais como tarefas ou processos do fluxo de trabalho, como a classe base a ser executada na estrutura do fluxo de trabalho, desenvolvedores podem construir de forma fácil e eficiente códigos de operação específicos de domínio (por exemplo, ambientes de execução específicos, tais como programas na indústria da saúde, na indústria financeira ou congêneres) (doravante “op-code”) sem se unir ao rígido, embutido em código, inflexível e fixo conjunto de funções ou de classes de atividades no ambiente de execução existente. Além do mais, a fundação do fluxo de trabalho que incorpora aspectos da invenção é um tempo de execução com base em continuação disposto em camadas no topo de todas as estruturas existentes (por exemplo, tanto um ambiente de execução gerenciada, um ambiente de sistema operacional quanto um nível de unidade de processamento de hardware).

Aspectos da invenção liberam a restrição de definir atividades em um formato de arquivo em particular pela habilitação de projetos de fluxo de trabalho em qualquer maneira ou representação (por exemplo, um fluxograma, um diagrama, uma descrição numerada ou congêneres), contanto que as atividades no fluxo de trabalho possam ser construídas a partir da representação dos projetos do fluxo de trabalho.

Além do mais, a estrutura ou fundação do fluxo de trabalho pode tratar falha ou exceção elevada de um nível inferior (por exemplo, OS) ou funções de elevação de exceção escritas em outros formatos (por exemplo, linguagem intermediária).

A figura 3 ilustra uma vista simplista de um fluxo de trabalho 300 de acordo com uma modalidade da invenção. Por exemplo, o fluxo de trabalho 300 pode ser um fluxo de trabalho para processar uma ordem de compra, e este fluxo de trabalho 300 de ordem de compra pode incluir processos ou atividades tais como receber uma ordem de compra, transmitir confirmação a um cliente, aprovar a ordem de compra por um gerente ou congêneres.

O fluxo de trabalho 300 pode começar a partir de um ponto de início 302. Por exemplo, o ponto de início 302 para o fluxo de trabalho da ordem de compra pode ser receber uma ordem de um cliente. O fluxo de trabalho 300 também pode incluir uma declaração condicional 304 (tais como uma “declaração SE” ou uma “declaração ENQUANTO”), e ele pode ser subdividido em declarações condicionais adicionais 306 e 308. O fluxo de trabalho 300 também pode incluir uma estrutura paralela 310, que inclui adicionalmente uma ou mais seqüências ou atividades 312. Por exemplo, a estrutura paralela 310 inclui atividades, tais como verificar o inventário e atualizar os transportadores

disponíveis, que podem ser processadas em paralelo. No exemplo mostrado, atividades, tais como “Transmitir Correio Eletrônico” e “Receber Aprovação”, podem ser processadas em paralelo. Em “deixar atividades aqui” 316, um usuário pode adicionar ou complementar adicionalmente mais atividades no fluxo de trabalho 300. Para completar o fluxo de trabalho 5 300, os processos ou atividades concluirão em uma etapa ou ponto de conclusão 314.

Em uma modalidade, as atividades podem ser hierarquicamente arranjadas em uma estrutura de árvore (veja figura 5) 500 ou em outras seqüências de execução. Por exemplo, uma atividade pode ser uma atividade composta na qual a atividade inclui mais de um item de trabalho associado a ela. Em uma outra modalidade, uma coleção de atividades 10 pode ser uma atividade composta. Um método ou operação de atividade pode estar em um nó raiz 502 com dois nós filhos ou folhas 504 e 506. Os métodos ou operações da atividade nos nós filhos 504 e 506 (por exemplo, work item_1 e work item_2, respectivamente) podem ser executados de acordo com a estrutura hierárquica. Além do mais, os nós filhos 504 e 506 também podem incluir outros nós filhos com respectivos itens de trabalho a ser 15 executados.

Em uma outra modalidade, atividades incluem um ou mais dos tipos seguintes: uma atividade simples, atividade de recipiente e atividade raiz. Nesta modalidade, há uma atividade raiz no modelo, e nenhuma ou qualquer quantidade de atividades simples ou de atividades de recipiente no interior da atividade raiz. Uma atividade de recipiente pode incluir 20 atividades simples ou de recipiente. Todo o processo de fluxo de trabalho pode ser usado como uma atividade para construir processos de fluxo de trabalho de ordem superior. Adicionalmente, uma atividade pode ser interrompível ou não interrompível. Uma atividade composta não interrompível não inclui atividades interrompíveis. Uma atividade não interrompível carece de serviços que podem fazer com que a atividade seja bloqueada.

Além do mais, na execução de atividades e dos itens de trabalho incluídos nas atividades, a estrutura de fluxo de trabalho ou um contexto ou ambiente de execução definem um escopo ou o limite para cada um dos itens de trabalho. Este escopo ou limite inclui e expõe informação (por exemplo, na forma de dados, metadados ou congêneres), tais como os dados ou recursos compartilhados a ser acessados pelos itens de trabalho, 30 propriedades associadas, rotinas de tratamento, restrições e interações entre agentes autônomos. Também, cada atividade pode ser configurada por um código de usuário em qualquer linguagem de programação. Por exemplo, o código de usuário pode representar lógica empresarial ou de aplicação ou regras escritas em um domínio ou ambiente de execução específico. Cada atividade pode suportar ganchos de pré-interceptação e ganchos 35 de pós-interceptação na execução no código do usuário. Cada atividade tem semânticas e comportamento de execução em tempo de execução associado (por exemplo, gerenciamento de estado, transações, tratamento de evento e tratamento de exceção).

Atividades podem compartilhar estados ou recursos com outras atividades. Além do mais, atividades podem ser atividades primitivas ou agrupadas em uma atividade composta. Uma atividade primitiva ou básica não tem subestrutura (por exemplo, atividades filhas) e, assim, é um nó folha em uma estrutura de árvore. Uma atividade composta contém subestrutura (por exemplo, ela é o pai de uma ou mais atividades filhas).

A figura 4 é um diagrama que ilustra um sistema 400 para processar atividades de fluxo de trabalho de acordo com uma modalidade da invenção. O sistema 400 inclui um processador 402, que pode ser uma unidade de processamento ou uma coleção de unidades de processamento. O sistema 400 também inclui um armazenamento ou área de memória 404 para armazenar dados acessíveis pelo processador 402. Em uma modalidade, o sistema 400 pode ser um computador com um ou mais processadores ou unidades de processamento (por exemplo, processador 402) e uma memória de sistema (por exemplo, área de memória 404) com outros componentes para acoplar vários componentes de sistema, incluindo a memória do sistema, no processador 402.

Em um exemplo, a área de memória 404 pode incluir mídia legível por computador, seja mídia volátil, não volátil, removível ou não removível, implementada em qualquer método ou tecnologia para armazenamento de informação, tais como instruções legíveis por computador, estruturas de dados, módulos de programa ou outros dados. Por exemplo, mídia de armazenamento no computador inclui RAM, ROM, EEPROM, memória flash ou outra tecnologia de memória, CD-ROM, discos versáteis digitais (DVD) ou outro armazenamento em disco ótico, cassetes magnéticos, fita magnética, armazenamento em disco magnético ou outro dispositivo de armazenamento magnético, ou qualquer outra mídia que pode ser usada para armazenar a informação desejada e que pode ser acessada pelo sistema 400. A memória 404 também pode incluir mídia de comunicação que incorpora instruções legíveis por computador, estruturas de dados, módulos de programa ou outros dados em um sinal de dados modulado, tais como uma onda portadora ou outro mecanismo de transporte, e inclui qualquer mídia de distribuição de informação. Versados na técnica estão familiarizados com o sinal de dados modulado, que tem uma ou mais de suas características ajustadas ou modificadas de uma maneira tal para codificar informação no sinal. Mídia com fios, tais como rede com fios ou conexão direta com fios, e mídia sem fios, tais como acústica, RF, infravermelho, e outras mídias sem fios, são exemplos de mídia de comunicação. Combinações de qualquer um dos expostos também estão incluídas no escopo da mídia legível por computador.

Por exemplo, a área de memória 404 armazena uma pluralidade de atividades 406 para processamento em um fluxo de trabalho (por exemplo, o fluxo de trabalho 300). Cada uma da pluralidade de atividades 406 inclui um ou mais itens de trabalho, e os itens de trabalho podem ser organizados em uma estrutura hierárquica, tal como uma estrutura de

árvore (veja figura 5). No processamento da pluralidade de atividades 406, o processador 402 acessa ou executa um agendador 408, que é configurado para ajustar um conjunto organizado de atividades.

Por exemplo, o processador 408 acessa os itens de trabalho na pluralidade de atividades 406 por meio de um componente ou de um conjunto de instruções legíveis por computador, tal como o agendador 408, para enfileirar ou armazenar os itens de trabalho 422 em uma fila 410. Um expedidor 412, acessível pelo processador 402, expede os itens de trabalho 422 para execução. Por exemplo, um item de trabalho 422-1 pode incluir um método de atividade ou uma operação de atividade 424, rotina ou uma coleção de códigos para realizar uma função de “solicitar entrada de um usuário”. Um ou mais outros métodos de atividade, operações de atividade, rotinas ou códigos podem ser incluídos em cada um dos itens de trabalho 422 sem fugir do escopo da invenção.

Uma vez que os itens de trabalho 422 são expedidos pelo expedidor 412, o processador 402 executa cada um dos métodos 424 nos itens de trabalho 422 em 414. No exemplo do item de trabalho 422-1, o processador 402 pode cuidar para que um usuário, por meio de uma interface de usuário (UI), insira a informação ou dados solicitados. Em uma outra modalidade, o processador 402 pode conectar ou acessar uma fonte de dados externa para solicitar entrada do usuário. Mediante a conclusão do método da atividade ou da operação da atividade 424, o processador 402 conclui a execução dos itens de trabalho 422 em 416. Em uma modalidade, o processador 402 passiva o estado de execução dos itens de trabalho em 418 em um armazenamento de dados 420.

Em uma outra modalidade, o processador 402 executa os itens de trabalho 422 de acordo com um autômato de estado, tal como o autômato mostrado na figura 6, que é um diagrama que ilustra um autômato de estado exemplar 600 que descreve os estados de processamento dos itens de trabalho associados com uma atividade de acordo com uma modalidade da invenção. Em uma modalidade, o autômato de estado 600 define uma vida útil de execução de uma atividade. Em um exemplo, o autômato de estado 600 pode incluir um estado inicializado, um estado de execução e um estado encerrado (da forma mostrada na figura 4). Em uma outra modalidade, o autômato de estado 600 inclui um estado inicializado 602, um estado de execução 604, um estado de cancelamento 606, um estado falho 608, um estado de compensação 610 e um estado encerrado 612.

Por exemplo, o autômato de estado 600 descreve um fluxo de processo de execução dos itens de trabalho (por exemplo, itens de trabalho 422) em uma atividade de fluxo de trabalho. O item de trabalho 422-1, da forma ilustrada na figura 4, é inicializado primeiro quando ele está enfileirado na fila 410. A seguir, o item de trabalho 422-1 é retirado ou removido da fila até o expedidor 412 antes de ser executado no estado de execução (por exemplo, estado de execução 604, na figura 6). Dependendo dos parâmetros ou condições

durante a execução do item de trabalho 422-1, o item de trabalho 422-1 pode prosseguir até o estado de cancelamento 606 (por exemplo, o estado de cancelamento 426 na figura 4) ou até o estado falho 608. Em uma modalidade, o item de trabalho 422-1 pode prosseguir do estado de cancelamento 606 até o estado falho 608. Em uma modalidade alternativa, o estado de compensação 610 descreve um conjunto de operações ou funções a ser realizado quando o defeito ou exceção tiver ocorrido.

Por exemplo, suponha que uma exceção ocorra durante a execução de um item de trabalho (por exemplo, item de trabalho 422-1), tal como a falta de um parâmetro para uma função. O sistema 400 transiciona o item de trabalho 422-1 até o estado falho 608. Fazendo isto, o sistema 400 também realiza operações da coleta de lixo (por exemplo, pela remoção da parte previamente executada das operações do cache ou da memória, pelo reinício dos valores de parâmetro, ou congêneres) no estado de compensação 610 antes de transicionar o item de trabalho 422-1 ao estado encerrado 612. Por exemplo, itens de trabalho no estado de compensação 610 podem disparar operações, tal como a recuperação de dados que eram previamente usados para executar outros itens de trabalho. O estado encerrado 612 indica que a execução da atividade (por exemplo, atividade 500 na figura 5) foi concluída.

Em uma modalidade, o autômato de estado 600 estabelece relacionamento entre itens de trabalho em uma atividade composta. Por exemplo, uma das regras de relacionamento pode incluir que, antes de transicionar métodos ou itens de trabalho no nó raiz da árvore de atividade ao estado encerrado 612, todos os itens de trabalho nos nós filhos devem estar no estado inicializado 602 ou no estado encerrado 612. Uma outra regra pode exigir que, a fim de transicionar os itens de trabalho no nó filho da árvore de atividade ao estado de execução 604, o item de trabalho no nó raiz já deve estar no estado de execução 604.

Em uma outra modalidade, um ou mais estados adicionais podem ser definidos no autômato de estado 600 sem fugir do escopo das modalidades da invenção.

A seguir, em relação às figuras 7A e 7E, diagramas de blocos ilustram o tratamento assíncrono dos eventos de falha em um fluxo de trabalho de acordo com uma modalidade da invenção. Com propósitos de simplificação somente, e sem limitações, a figura 7A mostra uma atividade composta 702 que inclui três itens de trabalho filhos organizados em uma estrutura de árvore: transaction_1 704, transaction_2 706 e transaction_3 708. Da forma ilustrada, a atividade raiz 702 inclui um método para “escrever texto na tela”. Métodos ou operações da atividade para os itens de trabalho expostos também incluem o seguinte:

transaction_1 704:

```
{ INSERIR TEXTO ("1 E 2");  
  HANDLEFAULT();  
}
```

```

transaction_2 706:
{ INSERIR TEXTO ("3 E 4");
  HANDLEFAULT();
}
5 transaction_3 708:
{ INSERIR TEXTO ("5 E 6");
  PAUSAR 180 SEGUNDOS;
  INSERIR TEXTO("FIM");
}

```

10 Na figura 7B, o transaction_1 704, o transaction_2 706 e o transaction_3 708 são transicionados ao estado de execução 710. Da forma ilustrada, o transaction_1 704 executa as operações incluídas pela inserção de textos ("1 e 2") na tela (por exemplo, uma interface de utilidade 428) até um usuário 430.

Durante o estado de execução 710, ocorreram um evento falho 722 ou uma
 15 exceção. O evento falho 722 pode incluir uma notificação de advertência para dados ausentes, uma falha de execução, um acesso impreciso a um armazenamento de dados, ou congêneres. Neste exemplo, o transaction_1 704 inclui uma função handleFault() 716 para tratar o evento falho 722. Em uma modalidade, a função handleFault() 716 é similar a uma função de "captura" para o tratamento de falha em outros ambientes de execução, tais como
 20 um sistema operacional ou um ambiente de execução gerenciada. Como tal, a propagação ou expedição da falha até a função handleFault() 716 ou até a rotina de tratamento de "captura" é assíncrona.

Mediante a ocorrência do evento falho 722, o transaction_1 704 transiciona a um estado falho 712, e o transaction_1 704 é transicionado a um estado encerrado 714. Em
 25 uma modalidade, em resposta ao evento falho 722, a função handleFault() 716 é chamada e é colocada em uma fila (não mostrada) para processamento.

Com este protocolo bem definido para a propagação e o tratamento de exceção, modalidades alternativas podem tratar múltiplas exceções, e múltiplas exceções podem ser
 30 agendadas, embora a propagação das exceções possa ser intercalada com a execução normal do programa.

Na figura 7C, o transaction_2 706 e a transação 708 estão no estado de execução 710. Similar à execução do transaction_1 704, o transaction_2 706 executa as operações incluídas. Neste exemplo, os textos ("3 e 4") são inseridos na tela. Além do mais, o transaction_2 706 também inclui uma função handleFault() similar como a função
 35 handleFault() 716 no transaction_1 704 para tratar o evento falho 722.

Em uma modalidade alternativa, a função handleFault() 716 pode propagar ou transmitir uma notificação 720 aos itens de trabalho restantes no estado de execução 710

em função da hierarquia de execução ou da estrutura hierárquica de execução da atividade. Por exemplo, enquanto o transaction_1 704 estiver no estado falho 712, a função handleFault() 716 pode propagar a notificação 720 (por exemplo, uma função de “arremesso”) para que a função handleFault() da Activity_1 702 pai possa trata-la como se a notificação 720 fosse um evento falho ou uma exceção. Em uma modalidade, uma atividade filha pode limitar o alvo da função de arremesso até seu pai na árvore de atividade. Em uma outra modalidade, o tratamento de exceção pode estar altamente associado ou amarrado com a estrutura tipo árvore das atividades.

Pelo estabelecimento do estado falho 712 para tratar eventos falhos, modalidades da invenção habilitam tratamento de falha assíncrona ou tratamento de exceção, e os itens de trabalho ou atividades restantes no estado de execução 710 continuam a ser executados. Além do mais, uma outra modalidade alternativa habilita o agendamento do tratamento dos eventos falhos. Por exemplo, mediante a resposta à notificação 720, o transaction_2 706 pode ser colocado em uma fila do agendador 718 antes de ser transicionado ao estado falho 712. Em uma outra modalidade, a notificação 720 pode ser suprimida de maneira tal que outros itens de trabalho ou atividades no estado de execução 710 continuem a ser executados. Em uma modalidade, o transaction_1 704 transiciona até um estado encerrado 714 depois de propagar ou transmitir a notificação 720. Em uma ainda outra modalidade, a propagação e o tratamento de falha sobrevivem aos ciclos de passivação e abarcam através deles.

Na figura 7D, o transaction_3 708 está sendo executado no estado de execução 710. Por exemplo, as operações incluídas do transaction_3 708 inserem os textos (“5 e 6”) e pausa por 180 segundos antes de inserir o texto (“FIM”) na tela. Entretanto, as operações incluídas não incluem funções para o tratamento de falha. Como tal, mediante a conclusão das operações incluídas, o transaction_3 708 é transicionado ao estado encerrado 714 da figura 7E. Além do mais, o transaction_2 706 também transiciona até o estado encerrado 714 depois de ser retirado da fila do agendador 718 até o estado falho 712.

Sem limitações, o Apêndice A ilustra uma implementação exemplar da elevação declarativa de uma exceção de acordo com uma modalidade da invenção. Em uma modalidade, programadores ou desenvolvedores podem projetar uma rotina de tratamento de falha para tratar um tipo em particular de eventos de falha ou de exceções. Em uma ainda outra modalidade, itens de trabalho ou atividades no fluxo de trabalho podem não incluir uma função ou podem ser incapazes de tratar eventos de falha. Nesta modalidade, o ambiente de execução do fluxo de trabalho trata os eventos de falha. Em uma ainda outra modalidade, uma ou mais operações pós-tratamento de falha podem ser fornecidas, por meio da UI 428, ao usuário 430 da figura 4.

Embora as figuras 7A a 7E ilustrem instantâneos do estado de execução ou de

partes do autômato de estado seqüencialmente (por exemplo, transações são executadas seqüencialmente), itens de trabalho no estado de execução podem ser processados simultaneamente ou de forma substancialmente simultânea sem fugir do escopo da invenção.

5 A figura 8 é um fluxograma que ilustra um método para tratar de forma assíncrona um evento falho para uma atividade de um fluxo de trabalho de acordo com uma modalidade da invenção. Por exemplo, o método ilustrado na figura 8 pode ser representado como instruções executáveis por computador a ser armazenadas em uma mídia legível por computador mostrada na figura 9. Por exemplo, a máquina de estado 902 define um
10 autômato de estado (por exemplo, autômato de estado 600) para uma atividade em 902, e o autômato de estado inclui pelo menos um estado de execução, um estado falho e um estado encerrado. Um componente de atividade 904 define a atividade a incluir uma pluralidade de itens de trabalho em 804. A atividade definida tem uma hierarquia de execução ou uma seqüência de execução (por exemplo, uma estrutura de árvore) para a pluralidade de itens
15 de trabalho. Cada um dos itens de trabalho inclui uma operação para executar uma parte da atividade.

Um componente agendador 906 transiciona cada um dos itens de trabalho até o estado de execução em 806. Um componente de execução 908 executa a operação incluída dos itens de trabalho transicionados no estado de execução em 808. Em 801, um
20 componente de identificação 901 identifica um ou mais dos itens de trabalho transicionados em resposta ao evento falho com base na hierarquia de execução e na operação incluída. Em 812, um processo para tratamento de falha 912 trata de forma assíncrona o evento falho invocando uma operação de tratamento de falha (por exemplo, a função `handleFault()` 716) nos um ou mais itens de trabalho identificados para transicionar os um ou mais itens de
25 trabalho identificados até o estado falho, executando a operação incluída dos itens de trabalho transicionados restantes não identificados em resposta ao evento falho pelo componente de identificação. Em uma modalidade, o processo para tratamento de falha 912 trata de forma assíncrona o evento falho pela transição dos um ou mais itens de trabalho identificados até o estado falho. Em uma ainda outra modalidade, o processo para
30 tratamento de falha 912 trata de forma assíncrona o evento falho enfileirando os um ou mais itens de trabalho identificados em uma fila do agendador (por exemplo, fila do agendador 718).

Em uma modalidade alternativa, a mídia legível por computador 900 inclui adicionalmente um componente de propagação de falha 914 para transmitir uma notificação
35 dos um ou mais itens de trabalho identificados até os itens de trabalho transicionados restantes em função da hierarquia de execução da atividade. A notificação 700 indica que os um ou mais itens de trabalho identificados estão no estado falho. Em uma modalidade

adicional, a mídia legível por computador 900 inclui adicionalmente um componente de transição 916 para transicionar os itens de trabalho transicionados restantes do estado de execução até o estado falho em resposta à notificação transmitida.

5 A mídia legível por computador também inclui um componente de compensação 918 para recuperar ou compensar dados associados com a atividade em função do tratamento assíncrono do evento de falha em uma ainda outra modalidade alternativa. Um componente de inibição também pode ser parte da mídia legível por computador 900 para suprimir a transmissão da notificação aos itens de trabalho transicionados restantes.

10 Embora descritas em conjunto com um ambiente de sistema de computação exemplar, tal como o sistema 400 da figura 4, modalidades da invenção são operacionais com inúmeros outros ambientes ou configurações de sistema de computação de uso geral ou de uso especial. Não pretende-se que o ambiente do sistema de computação sugira nenhuma limitação ao escopo de uso ou à funcionalidade de nenhum aspecto da invenção. Além do mais, o ambiente do sistema de computação não deve ser interpretado com
15 nenhuma dependência ou exigência relacionadas a nenhum dos componentes ilustrados no ambiente operacional exemplar ou de suas combinações. Exemplos dos sistemas, ambientes e/ou configurações de computação bem conhecidos que podem ser adequados para uso com aspectos da invenção incluem, mas sem limitações, computadores pessoais, computadores servidores, dispositivos de mão ou portáteis, sistemas multiprocessadores,
20 sistemas com base em microprocessador, conversores de sinal de frequência, dispositivos eletrônicos programáveis pelo cliente, telefones celulares, PCs em rede, minicomputadores, computadores de grande porte, ambientes de computação distribuída que incluem qualquer um dos sistemas ou dispositivos expostos, e congêneres.

25 Modalidades da invenção podem ser descritas no contexto geral das instruções executáveis por computador, tais como módulos de programa, executadas por um ou mais computadores ou outros dispositivos. No geral, módulos de programa incluem, mas sem limitações, rotinas, programas, objetos, componentes e estruturas de dados que realizam tarefas em particular ou que implementam tipo de dados abstratos em particular. Aspectos da invenção também podem ser realizados em ambientes de computação distribuída em
30 que tarefas são realizadas pelos dispositivos de processamento remotos que são ligados por meio de uma rede de comunicações. Em um ambiente de computação distribuída, módulos de programa podem ficar localizados em mídia de armazenamento no computador tanto local quanto remota, incluindo dispositivo de armazenamento de memória.

35 Em operação, o sistema 400 executa instruções executáveis por computador, tais como aquelas ilustradas nas figuras, tal como a figura 8, para implementar aspectos da invenção.

A ordem de execução ou de realização das operações nas modalidades da

invenção aqui ilustradas e descritas não é essencial, a menos que de outra forma especificado. Isto é, as operações podem ser realizadas em qualquer ordem, a menos que de outra forma especificado, e modalidades da invenção podem incluir operações adicionais ou menos operações do que aquelas aqui divulgadas. Por exemplo, percebe-se que
5 executar ou realizar uma operação em particular antes, durante ou depois de uma outra operação está no escopo dos aspectos da invenção.

Modalidades da invenção podem ser implementadas com instruções executáveis por computador. As instruções executáveis por computador podem ser organizadas em um ou mais componentes ou módulos executáveis por computador. Aspectos da invenção
10 podem ser implementados com qualquer número e organização de tais componentes ou módulos. Por exemplo, aspectos da invenção não são limitados às instruções executáveis por computador específicas ou aos componentes ou módulos específicos ilustrados nas figuras e aqui descritos. Outras modalidades da invenção podem incluir diferentes instruções ou componentes executáveis por computador com mais ou menos funcionalidade do que
15 aqui ilustrado e descrito.

Durante a introdução dos elementos dos aspectos da invenção ou das suas modalidades, pretende-se que os artigos “um”, “uma”, “o”, “a”, “dito” e “dita” signifiquem que há um ou mais dos elementos. Pretende-se que os termos “compreendendo”, “incluindo” e “tendo” sejam inclusivos e signifiquem que pode haver elementos adicionais diferentes dos
20 elementos listados.

Tendo sido descritos aspectos da invenção com detalhes, ficará aparente que modificações e variações são possíveis sem fugir do escopo dos aspectos da invenção definido nas reivindicações anexas. Já que várias mudanças podem ser feitas nas construções, produtos e métodos expostos sem fugir do escopo dos aspectos da invenção,
25 pretende-se que todo o assunto contido na descrição exposta e mostrado nos desenhos anexos seja interpretado como ilustrativo e não em um sentido limitante.

APÉNDICE A

[0066] <myActivities:Sequence x:Name="myWorkflow"

x:Class="myApp.myWorkflow"

xmlns:myActivities="http://schemas.com/myActivities"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/workflow">

<myActivities:WriteLine Text="One"/>

<myActivities:WriteLine Text="Two"/>

<ThrowActivity FaultType="{x:Type InvalidOperationException}" />

<myActivities:WriteLine Text="Unreachable code"/>

<FaultHandlersActivity>

<FaultHandlerActivity FaultType="{x:Type
InvalidOperationException}">

<myActivities:WriteLine Text="Three"/>

</FaultHandlerActivity>

<FaultHandlerActivity FaultType="{x:Type
AppDomainUnloadedException}">

<myActivities:WriteLine Text="Four"/>

</FaultHandlerActivity>

</FaultHandlersActivity>

</myActivities:Sequence>

<myActivities:Sequence x:Name="myWorkflow" x:Class="myApp.myWorkflow"

xmlns:myActivities="http://schemas.com/myActivities"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/workflow">

```
<myActivities:WriteLine Text="Hello World"/>  
<ThrowActivity FaultType="{x:Type InvalidOperationException}" />  
<myActivities:WriteLine Text="Unreachable Code"/>  
</myActivities:Sequence>
```

REIVINDICAÇÕES

1. Método para tratamento assíncrono de um evento de falha (722) para uma atividade de um fluxo de trabalho, **CARACTERIZADO** pelo fato de que o dito método compreende:

5 definir um autômato de estado (600) para uma atividade (702), o dito autômato de estado (600) incluindo pelo menos um estado de execução (604), um estado falho (608) e um estado encerrado (612), o dito autômato de estado (600) classificando uma vida útil de execução da atividade (702);

10 definir a atividade (702) a incluir a pluralidade de itens de trabalho (422), a dita atividade (702) definida com uma hierarquia de execução (500) para a pluralidade de itens de trabalho (422), cada um dos itens de trabalho (422) incluindo uma operação para executar uma parte da atividade (702);

 transicionar cada um dos itens de trabalho (422) até o estado de execução (604);

15 executar a operação incluída dos itens de trabalho transicionados (422) no estado de execução (604);

 identificar um ou mais dos itens de trabalho transicionados em resposta ao evento falho (722) em função da hierarquia de execução (500) e da operação incluída; e

20 tratar de forma assíncrona o evento falho (722) pela transição dos um ou mais itens de trabalho identificados (422) ao estado falho (608) durante a execução da operação incluída dos itens de trabalho transicionados restantes não identificados em resposta ao evento falho (608).

2. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o tratamento assíncrono do evento falho (722) compreende invocar uma operação de tratamento de falha nos um ou mais itens de trabalho identificados (422).

25 3. Método, de acordo com a reivindicação 2, **CARACTERIZADO** pelo fato de que invocar a operação de tratamento de falha compreende enfileirar os um ou mais itens de trabalho identificados em uma fila do agendador (718).

30 4. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que compreende adicionalmente propagar uma notificação (720) dos um ou mais itens de trabalho (422) identificados aos itens de trabalho (422) transicionados restantes em função da hierarquia de execução (500) da atividade (702), a dita notificação (720) indicando que os um ou mais itens de trabalho (422) identificados estão no estado falho (608).

35 5. Método, de acordo com a reivindicação 4, **CARACTERIZADO** pelo fato de que compreende adicionalmente transicionar os itens de trabalho (422) transicionados restantes do estado de execução (604) ao estado falho (608) em resposta à notificação propagada (720).

6. Método, de acordo com a reivindicação 4, **CARACTERIZADO** pelo fato de que

compreende adicionalmente suprimir a propagação da notificação (720) aos itens de trabalho (422) transicionados restantes.

7. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que compreende adicionalmente fornecer operações a um usuário para tratamento pós-falha em resposta aos um ou mais itens de trabalho (422) identificados que são transicionados ao estado falho (608).

8. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que uma ou mais mídias legíveis por computador têm instruções executáveis por computador para realizar o método da reivindicação 1.

9. Sistema (400) para o tratamento assíncrono de eventos falhos (722) em um fluxo de trabalho, **CARACTERIZADO** pelo fato de que o dito sistema (400) compreende:

um armazenamento (404) para armazenar dados associados com itens de trabalho de uma atividade (406) no fluxo de trabalho, a dita atividade (406) com uma hierarquia de execução (500) para os itens de trabalho (422), cada um dos itens de trabalho (422) incluindo uma operação para executar uma parte da atividade (406);

um processador (402) configurado para executar instruções executáveis por computador para:

definir um autômato de estado (600) para a atividade (406), o dito autômato de estado (600) incluindo pelo menos um estado de execução (604), um estado falho (608) e um estado encerrado (612), o dito autômato de estado (600) classificando uma vida útil de execução da atividade (406);

transicionar cada um dos itens de trabalho (422) ao estado de execução (604);

executar a operação incluída dos itens de trabalho (422) transicionados no estado de execução (604);

identificar um ou mais dos itens de trabalho (422) transicionados em resposta a um evento falho (608) com base na hierarquia de execução (500) e na operação incluída; e

tratar de forma assíncrona o evento falho (722) pela transição dos um ou mais itens de trabalho (422) identificados ao estado falho (608) durante a execução da operação incluída dos itens de trabalho (422) transicionados restantes não identificados em resposta ao evento falho (722).

10. Sistema (400), de acordo com a reivindicação 9, **CARACTERIZADO** pelo fato de que o processador (402) é configurado para tratar de forma assíncrona o evento falho (722) usando um ou mais do seguinte método: invocar uma operação de tratamento de falha nos um ou mais itens de trabalho (422) identificados, e enfileirar os um ou mais itens de trabalho (422) identificados em uma fila do agendador (718).

11. Sistema (600), de acordo com a reivindicação 9, **CARACTERIZADO** pelo fato de que compreende adicionalmente um componente para propagar uma notificação (720)

dos um ou mais itens de trabalho identificados aos itens de trabalho (422) transicionados restantes em função da hierarquia de execução (500) da atividade (406), a dita notificação (720) indicando que os um ou mais itens de trabalho (422) identificados estão no estado falho (608).

5 12. Sistema (400), de acordo com a reivindicação 11, **CARACTERIZADO** pelo fato de que compreende adicionalmente um componente para transicionar os itens de trabalho transicionados restantes do estado de execução (604) ao estado falho (608) em resposta à notificação propagada (720).

10 13. Sistema (400), de acordo com a reivindicação 9, **CARACTERIZADO** pelo fato de que compreende adicionalmente um componente para compensar a recuperação de dados associados com a atividade do armazenamento (404) em função do tratamento assíncrono do evento falho (722) pelo processador (402).

15 14. Sistema (400), de acordo com a reivindicação 9, **CARACTERIZADO** pelo fato de que compreende adicionalmente um componente para suprimir a propagação da notificação (720) aos itens de trabalho (422) transicionados restantes.

15 15. Mídia legível por computador (900) com componentes executáveis por computador para tratamento assíncrono de um evento falho (722) em um fluxo de trabalho, **CARACTERIZADA** pelo fato de que os ditos componentes executáveis por computador compreendem:

20 uma máquina de estado (902) para definir um autômato de estado (600) para uma atividade (406), o dito autômato de estado (600) incluindo pelo menos um estado de execução (604), um estado falho (608) e um estado encerrado (612), o dito autômato de estado (600) classificando um ciclo de vida de execução da atividade (406);

25 um componente de atividade (904) para definir a atividade (406) para incluir uma pluralidade de itens de trabalho (422), a dita atividade (406) definida com uma hierarquia de execução (500) para a pluralidade de itens de trabalho (422), cada um dos itens de trabalho (422) incluindo uma operação para executar uma parte da atividade (406);

 um componente agendador (906) para transicionar cada um dos itens de trabalho (422) ao estado de execução (604);

30 um componente de execução (908) para executar a operação incluída dos itens de trabalho (422) transicionados no estado de execução (604);

 um componente de identificação (910) para identificar um ou mais dos itens de trabalho transicionados em resposta ao evento falho (722) com base na hierarquia de execução (500) e na operação incluída; e

35 um processo para o tratamento de falha (912) para o tratamento assíncrono do evento falho (722) invocando uma operação de tratamento de falha nos um ou mais itens de trabalho (422) identificados para transicionar os um ou mais itens de trabalho (422)

identificados ao estado falho (608) durante a execução da operação incluída dos itens de trabalho (422) transicionados restantes não identificados em resposta ao evento falho (722) pelo componente de identificação (910).

16. Mídia legível por computador (900), de acordo com a reivindicação 15,
5 **CARACTERIZADA** pelo fato de que o processo para o tratamento de falha (912) enfileira os um ou mais itens de trabalho (422) identificados em uma fila do agendador (718) para transicionar os um ou mais itens de trabalho identificados ao estado falho (608).

17. Mídia legível por computador (900), de acordo com a reivindicação 15,
10 **CARACTERIZADA** pelo fato de que compreende adicionalmente um componente de propagação de falha (914) para transmitir uma notificação (720) dos um ou mais itens de trabalho (422) identificados aos itens de trabalho (422) transicionados restantes em função da hierarquia de execução (500) da atividade (406), a dita notificação (720) indicando que os um ou mais itens de trabalho (422) identificados estão no estado falho (608).

18. Mídia legível por computador (900), de acordo com a reivindicação 17,
15 **CARACTERIZADA** pelo fato de que compreende adicionalmente um componente de transição (916) para transicionar os itens de trabalho transicionados restantes do estado de execução (604) ao estado falho (608) em resposta à notificação transmitida (720).

19. Mídia legível por computador (900), de acordo com a reivindicação 15,
20 **CARACTERIZADA** pelo fato de que compreende adicionalmente um componente de compensação (918) para recuperar dados associados com a atividade (406) em função do tratamento assíncrono do evento falho (608).

20. Mídia legível por computador (900), de acordo com a reivindicação 15,
25 **CARACTERIZADA** pelo fato de que compreende adicionalmente um componente de inibição (920) para suprimir a transmissão da notificação (720) aos itens de trabalho transicionados restantes.

FIG. 1

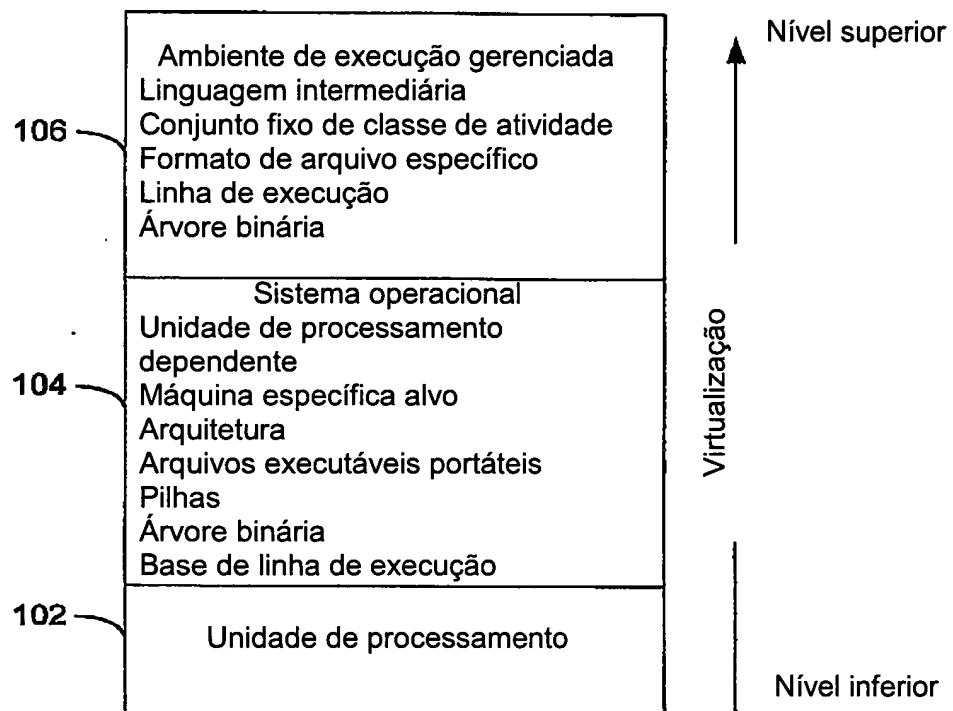


FIG. 2

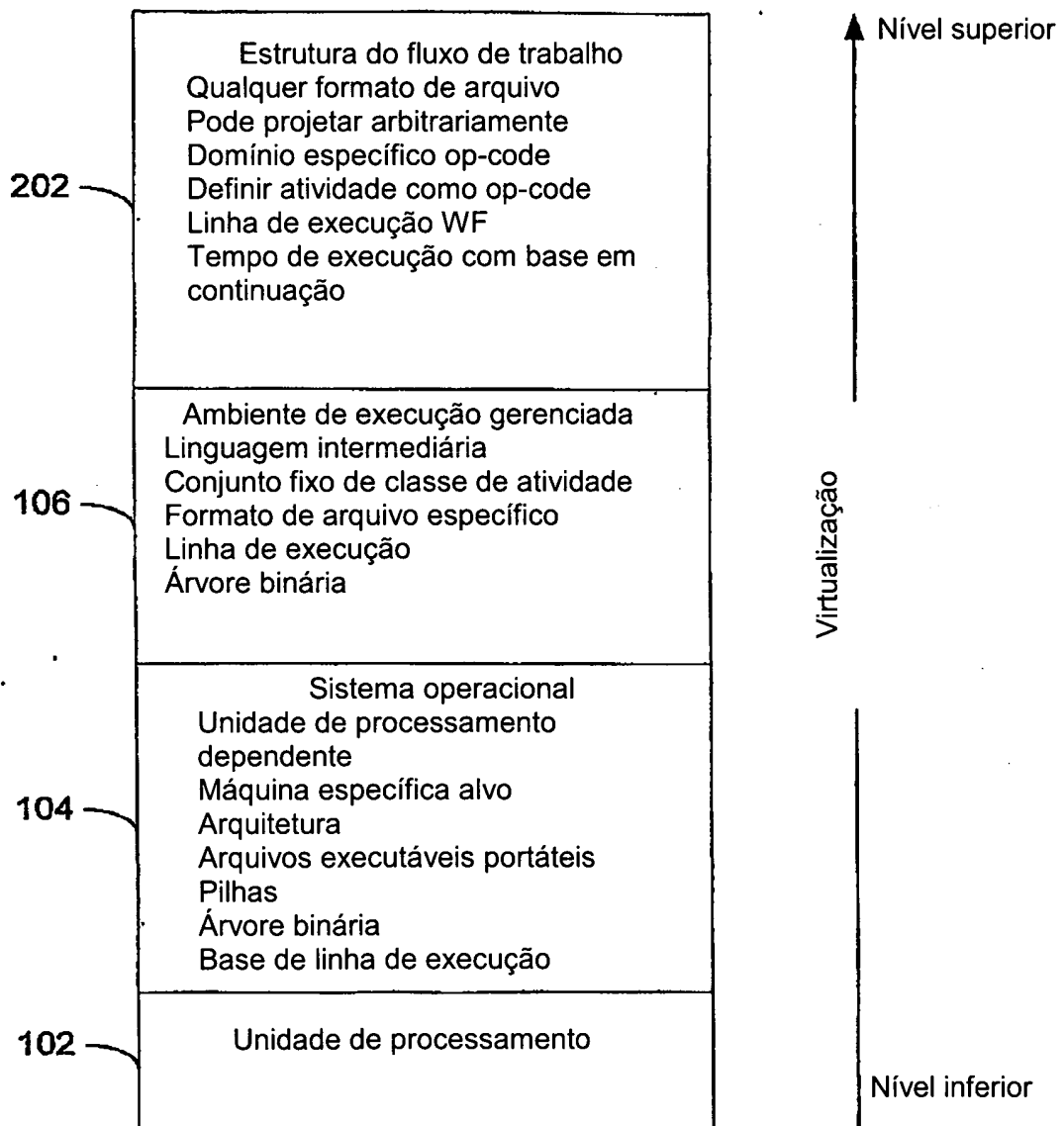


FIG. 3

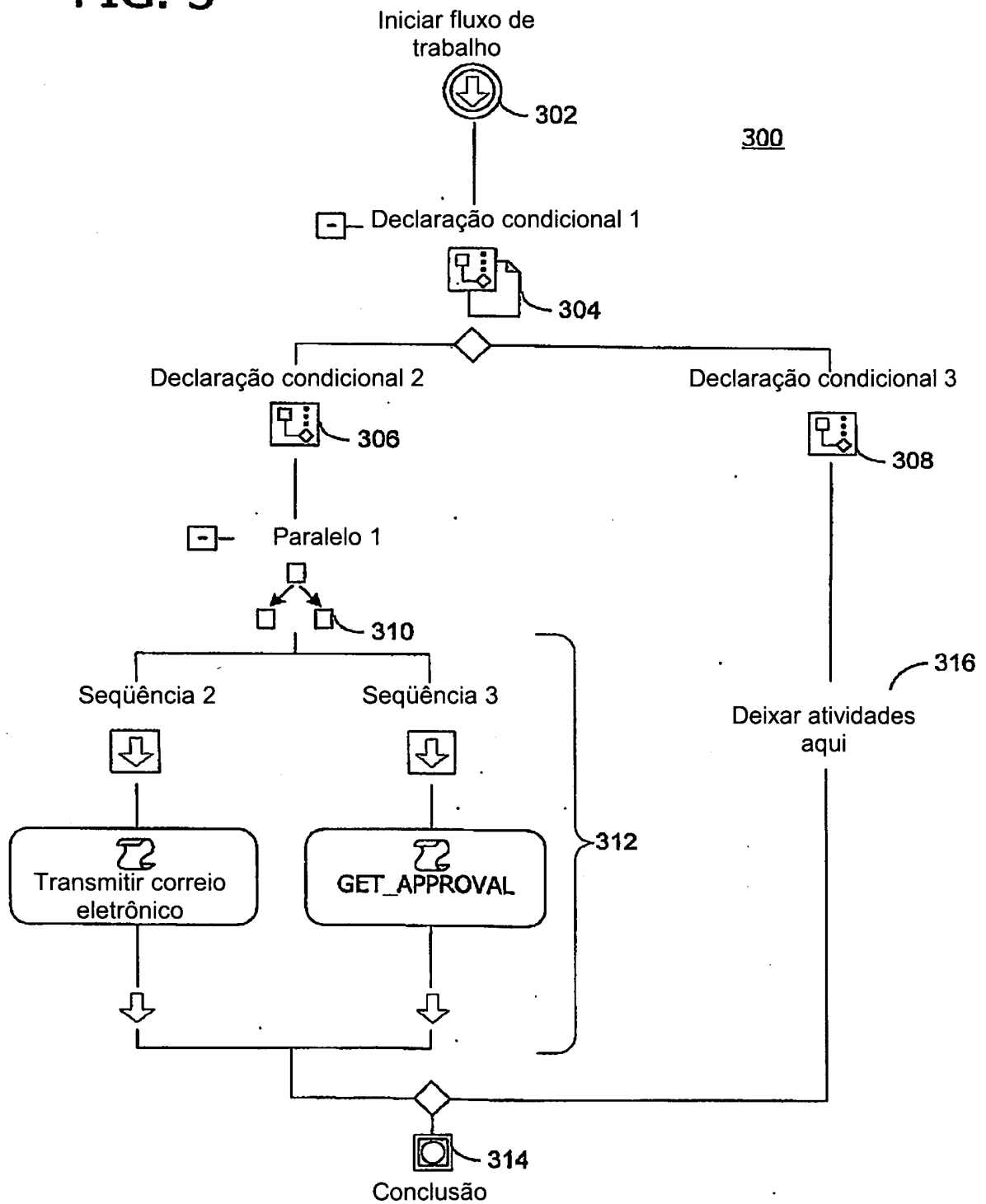


FIG. 4

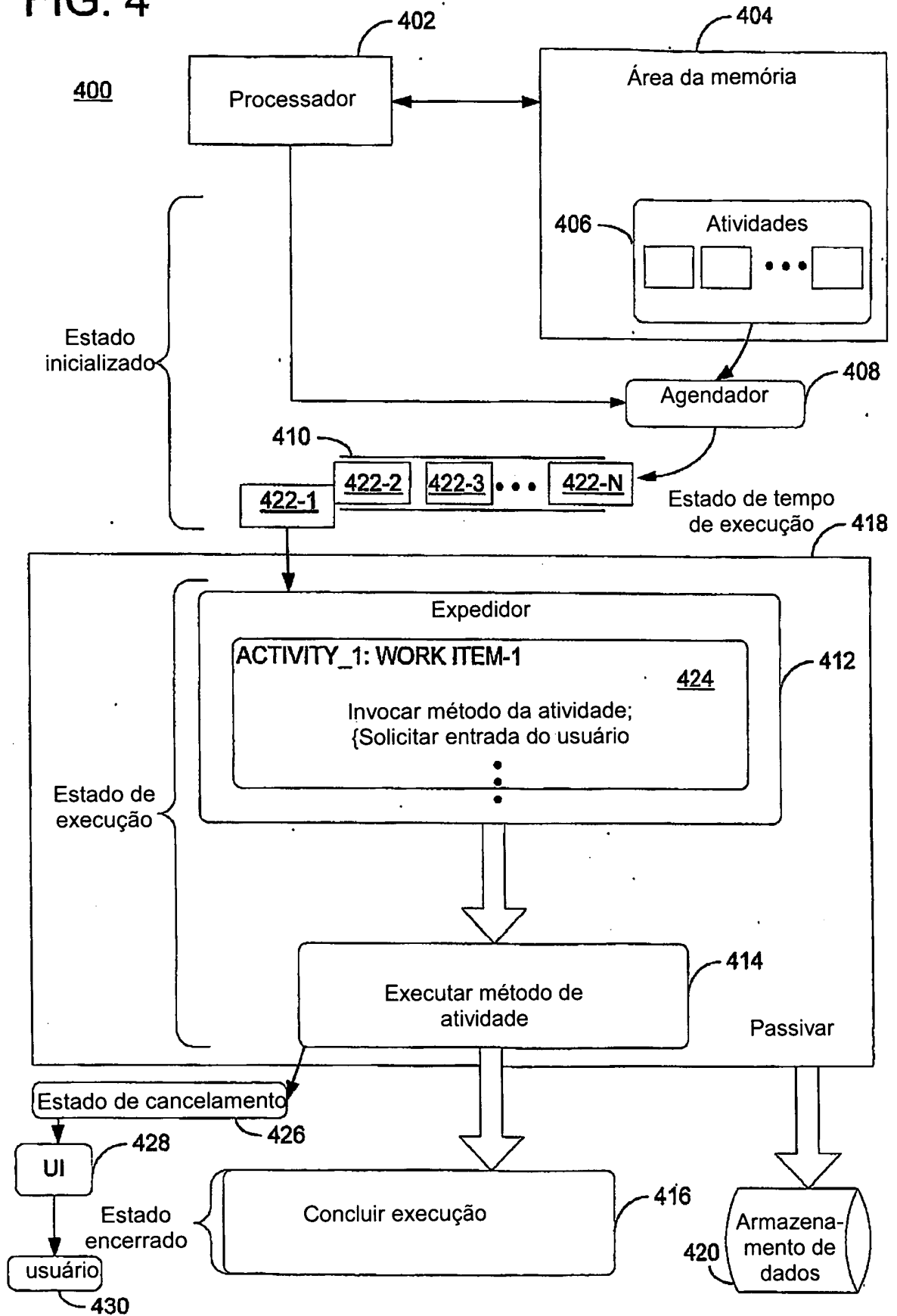
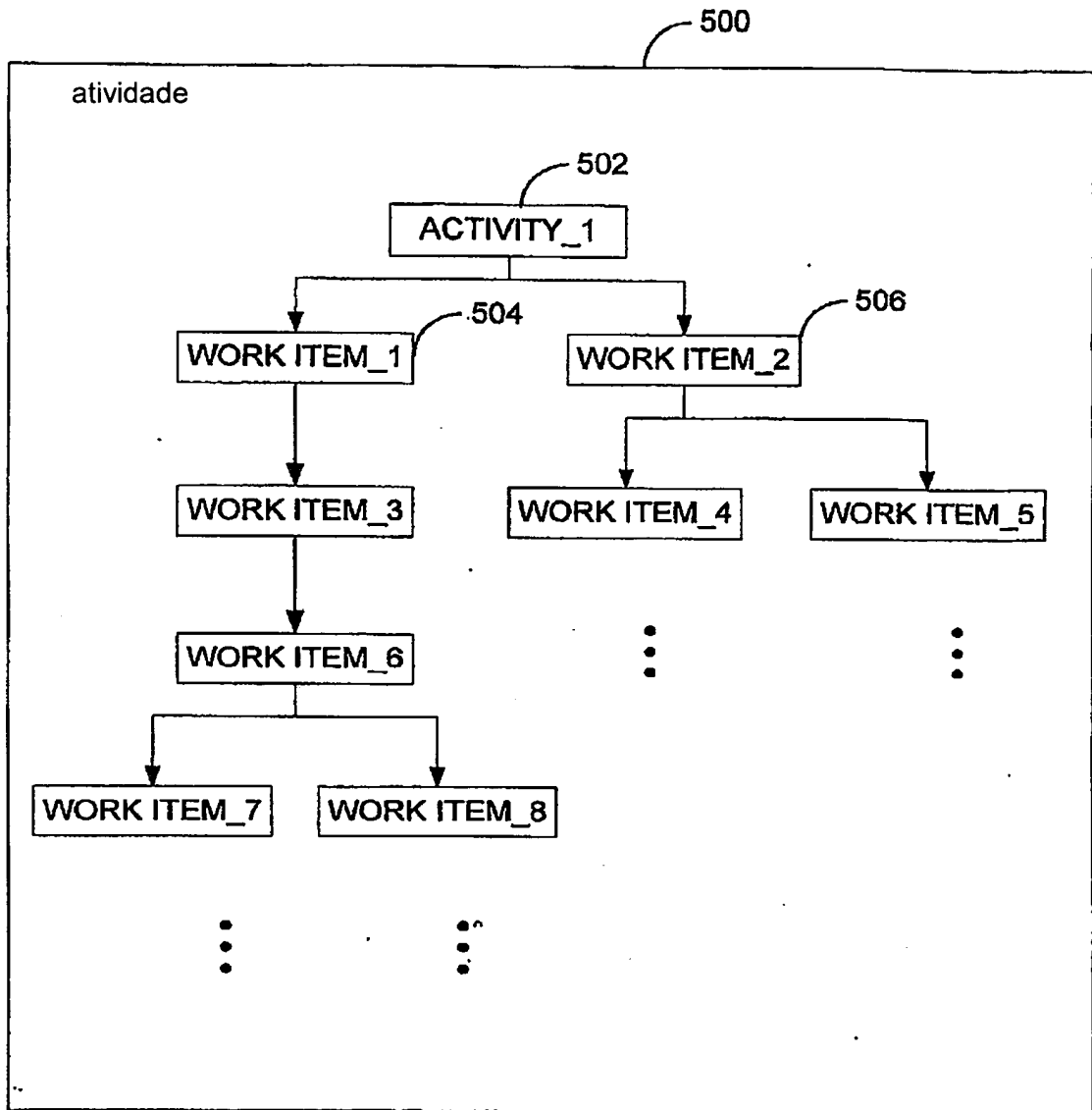


FIG. 5



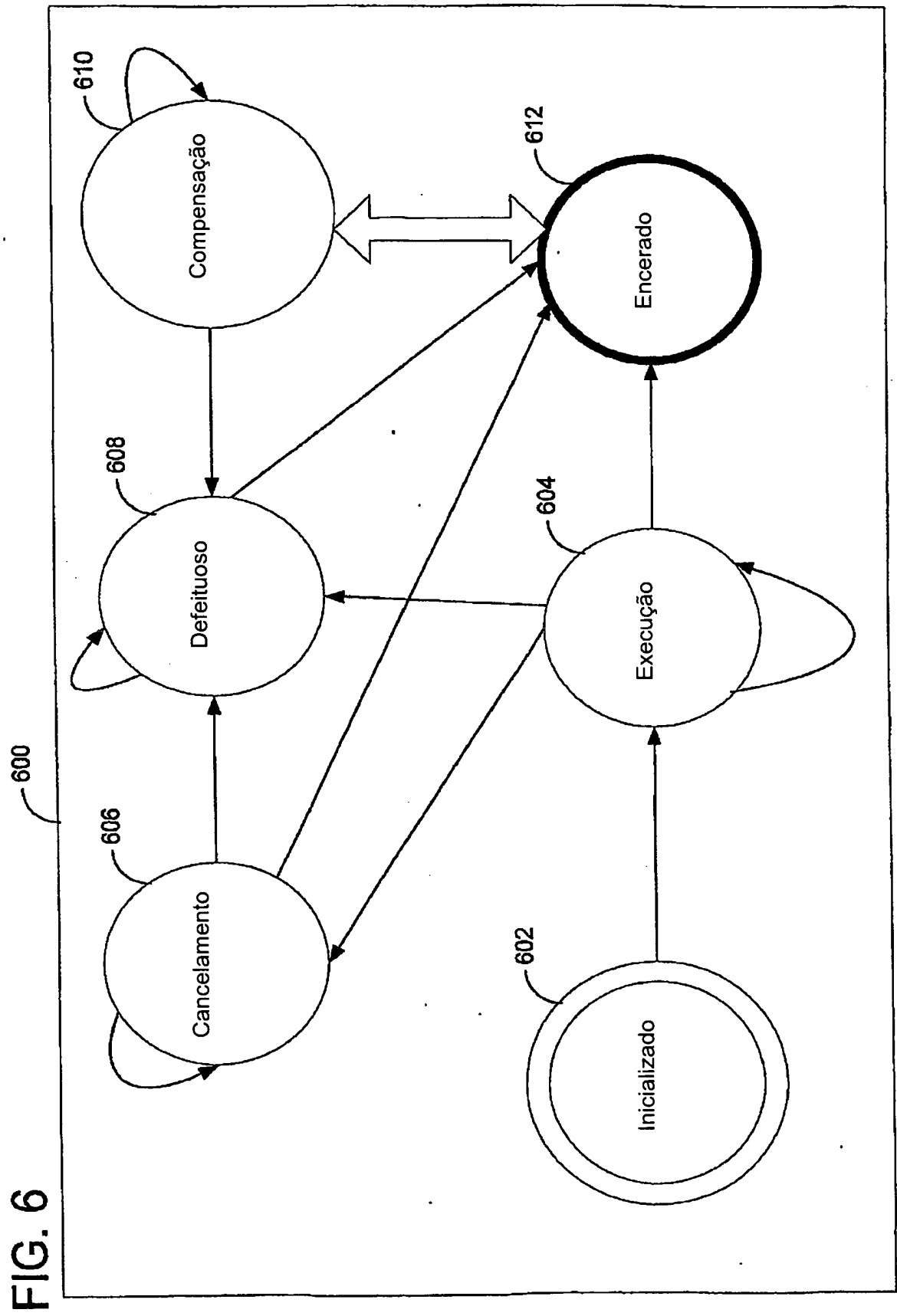


FIG. 7A

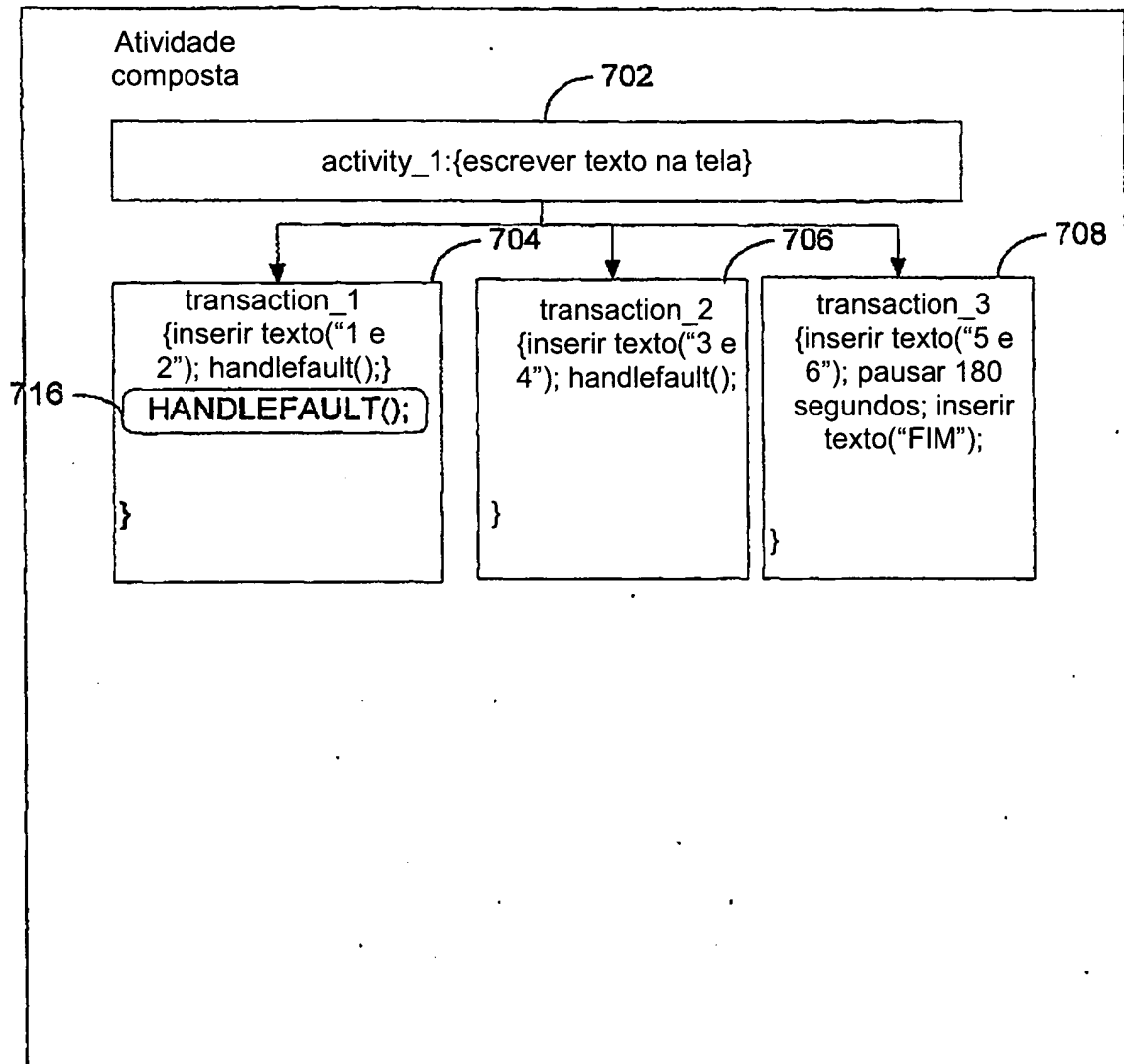


FIG. 7B

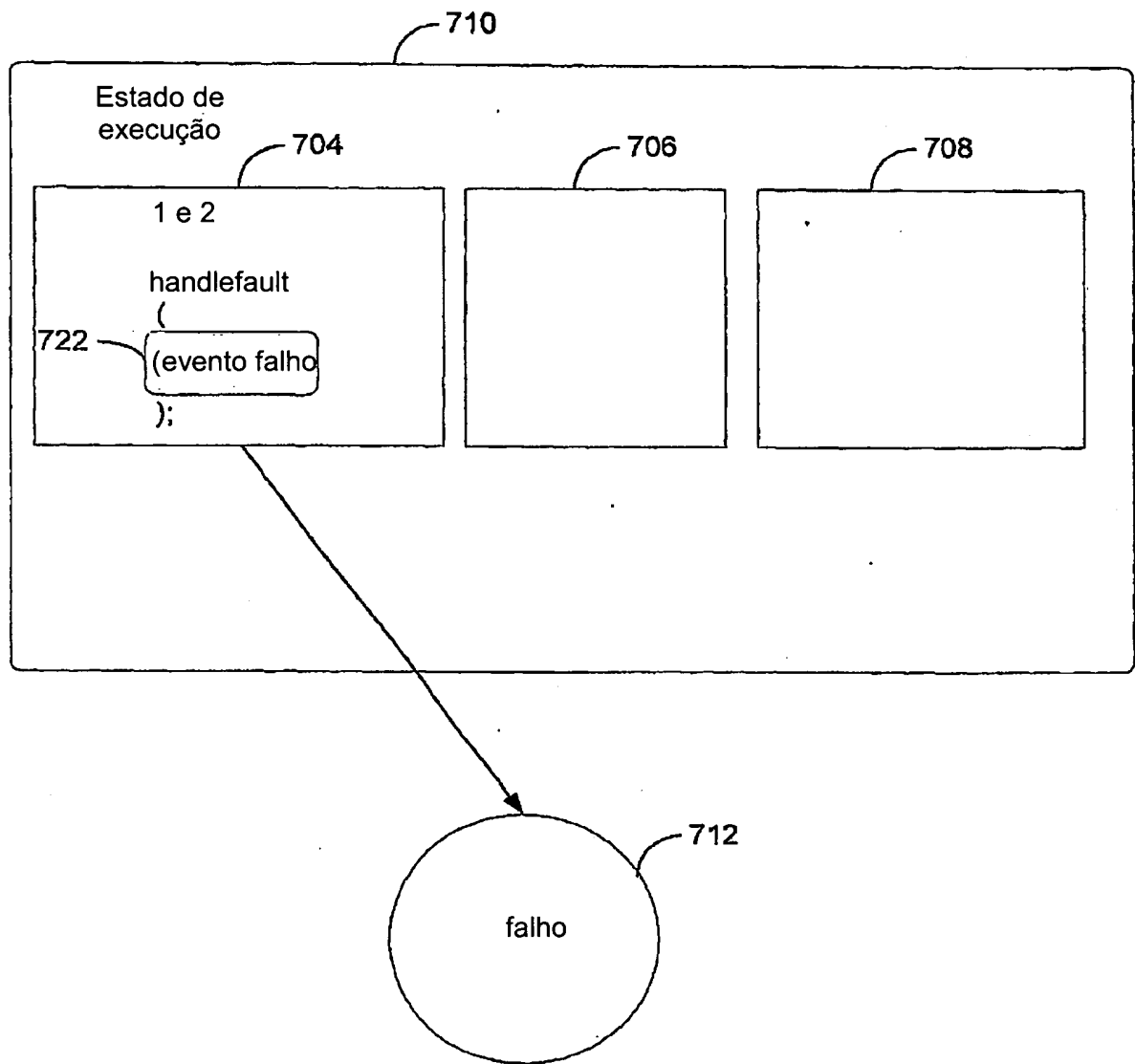


FIG. 7C

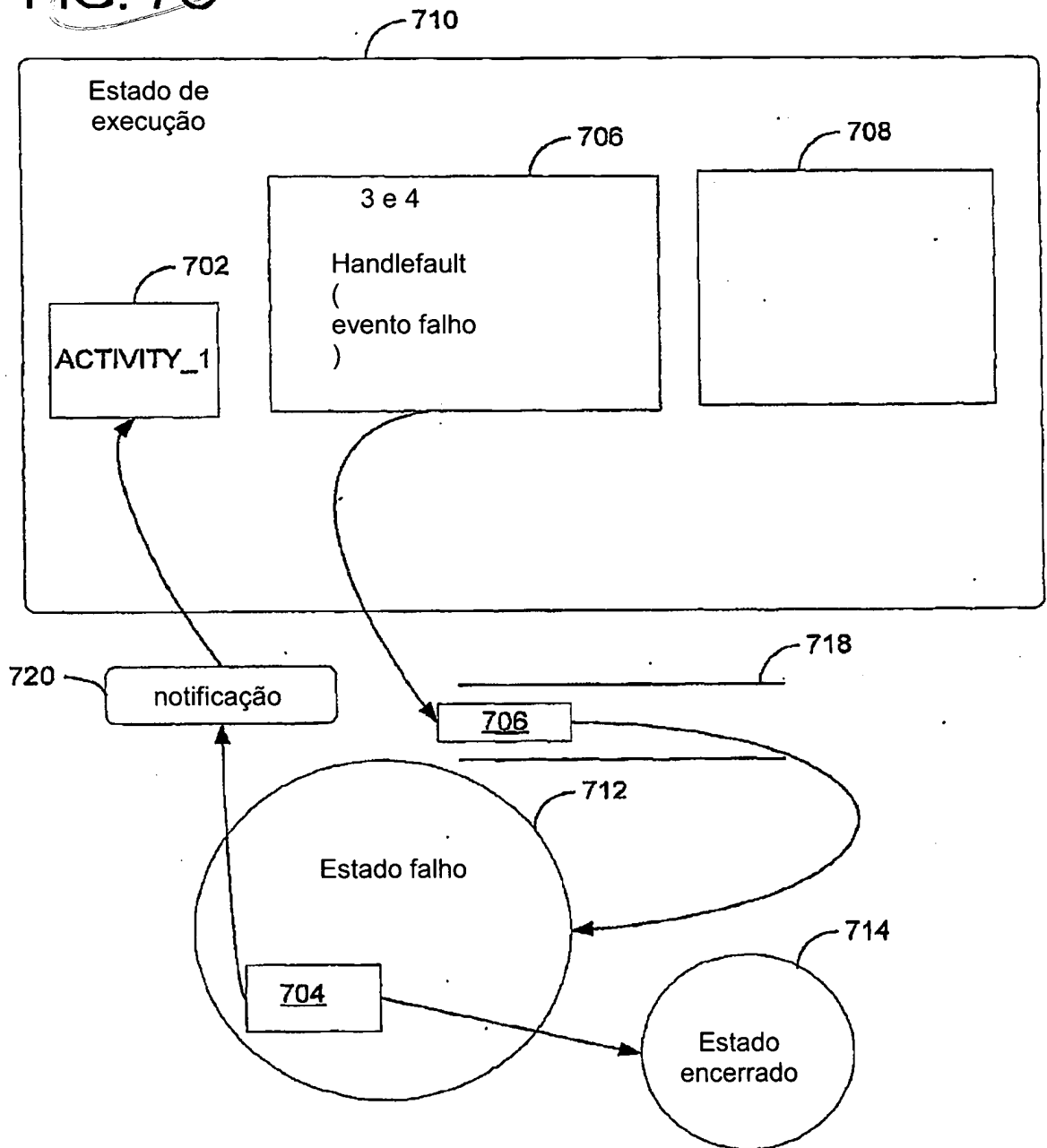


FIG. 7D

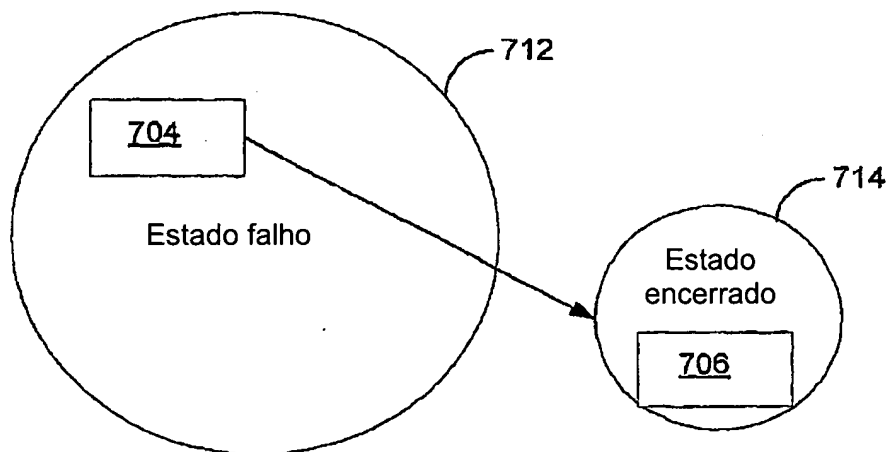
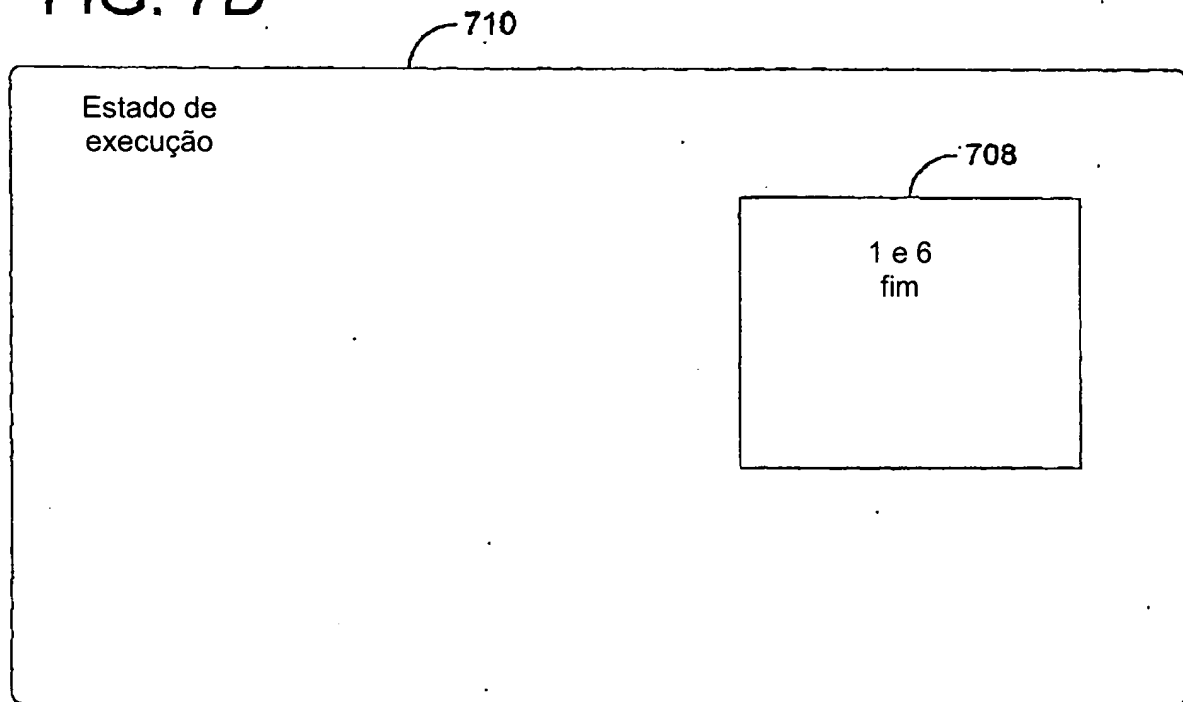


FIG. 7E

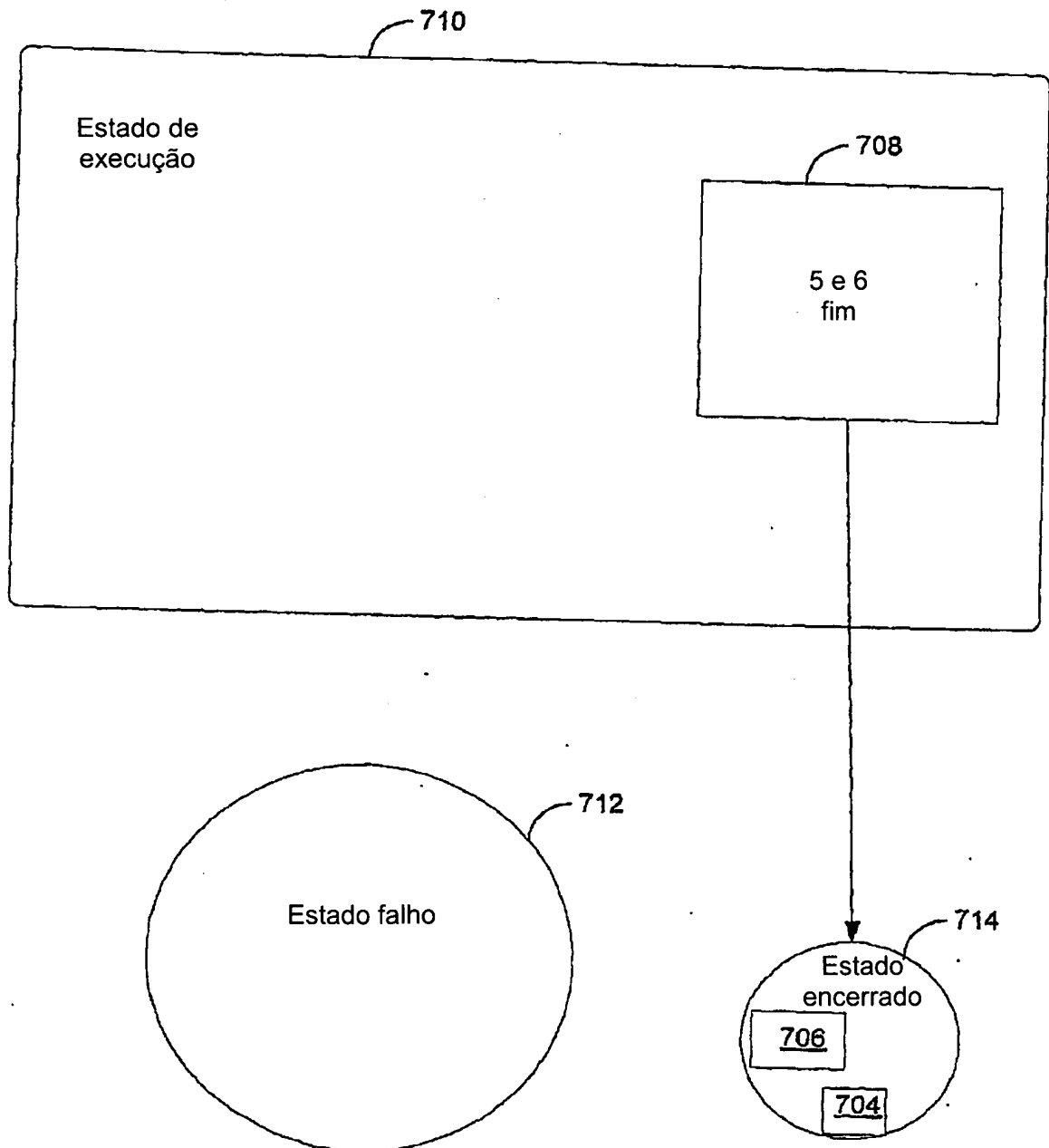


FIG. 8

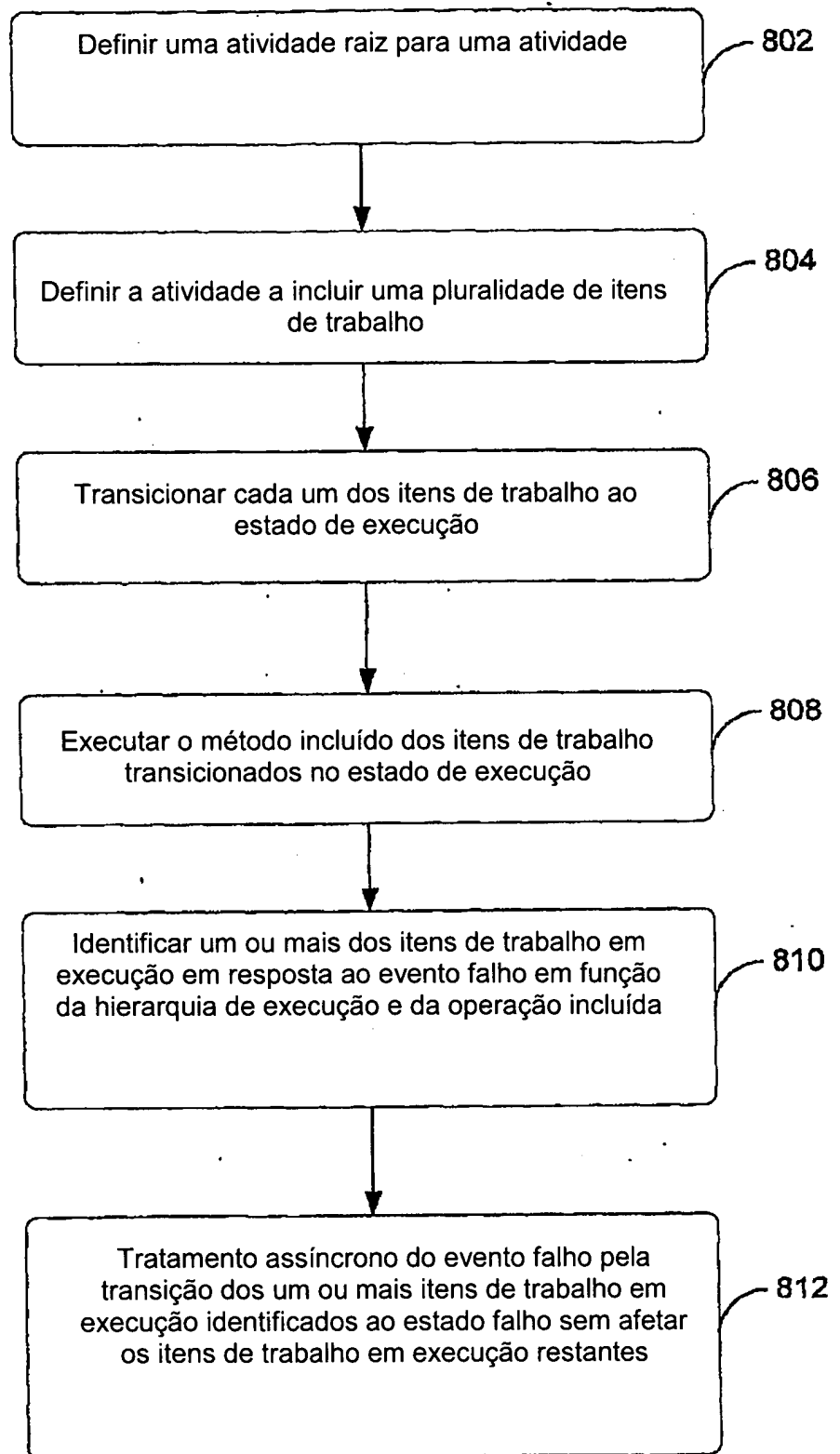
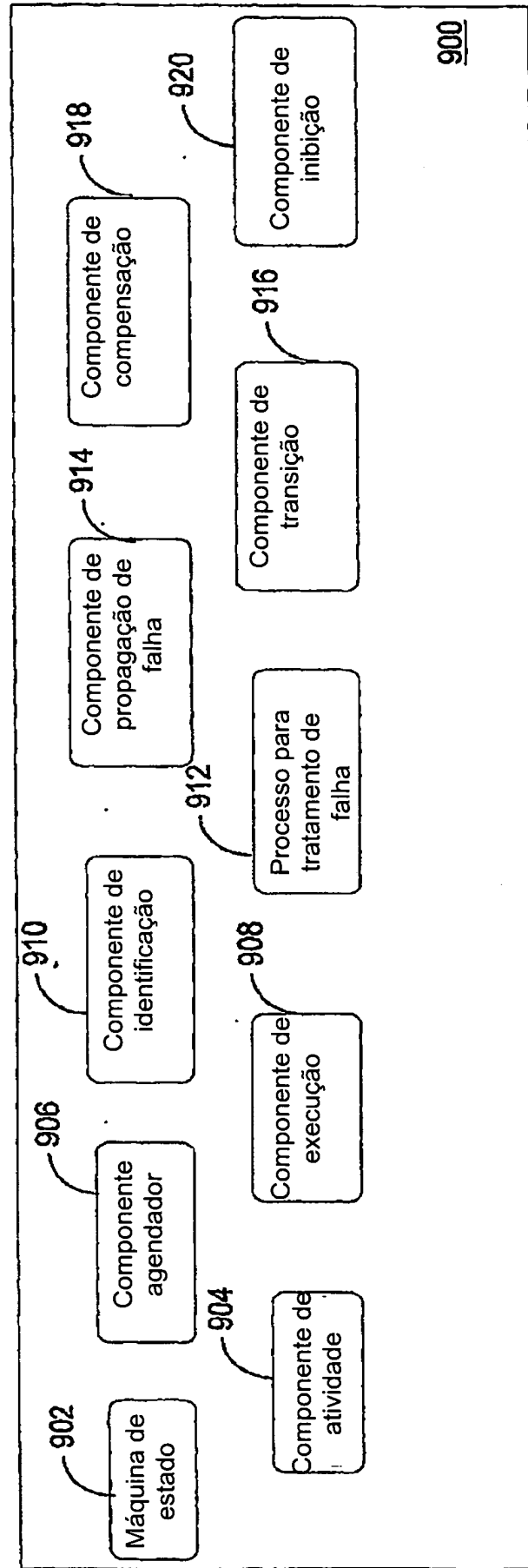


FIG. 9



RESUMO

“TRATAMENTO DE FALHA ASSÍNCRONA EM PROGRAMAS CENTRALIZADOS EM PROCESSO”

É divulgado o tratamento de falha assíncrona para um fluxo de trabalho. É definido

5 um autômato de estado para uma atividade no fluxo de trabalho. O autômato de estado inclui pelo menos um estado de execução, um estado falho e um estado encerrado, e classifica uma vida útil de execução da atividade. A atividade é definida para incluir itens de trabalho e inclui uma hierarquia de execução para os itens de trabalho. Cada item de trabalho inclui uma operação para executar uma parte da atividade. Cada item de trabalho é

10 transicionado ao estado de execução. A operação incluída dos itens de trabalho transicionados é executada no estado de execução. Um ou mais dos itens de trabalho transicionados são identificados em resposta ao evento falho em função da hierarquia de execução e da operação incluída. O evento falho é tratado de forma assíncrona pela transição dos um ou mais itens de trabalho identificados ao estado falho durante a execução

15 da operação incluída dos itens de trabalho transicionados restantes.