

2

Workflows - Conceitos e História

Este capítulo apresenta um resumo de diversos aspectos relativos aos sistemas de gerência de workflow. A Seção 2.1 apresenta uma introdução à história dos sistemas de gerência de workflow. A Seção 2.2 aborda os modelos de workflow transacionais existentes na literatura. A Seção 2.3, por sua vez, apresenta os esforços de padronização em que estão envolvidos os sistemas de gerência de workflow e o modelo de referência proposto pelo WfMC. A Seção 2.4 discute algumas questões de projeto de sistemas de gerência de workflow e, a Seção 2.5, o modelo de implementação proposto pelo WfMC. A Seção 2.6 lista algumas das limitações dos sistemas de workflow e, por fim, a Seção 2.7 apresenta a relação entre workflows e processos de negócio. A Seção 2.8 apresenta um resumo deste capítulo.

2.1

Evolução dos Sistemas de Workflow

Sistemas de gerência de workflow (ou simplesmente sistemas de workflow) são bastante antigos, mas tiveram a sua popularidade aumentada apenas nas últimas décadas, em decorrência das necessidades de negócio e organização que surgiram ao longo dos anos nas empresas.

Historicamente, sistemas de gerência de workflow têm sua origem na automação do trabalho [81]. Eles eram, no entanto, monolíticos por natureza, de modo que as políticas de negócio e acesso às informações estavam rigidamente definidas nas aplicações. Além disso, estes sistemas quase sempre se encontravam isolados, sem ligação alguma com outros sistemas de workflow ou aplicações genéricas.

A primeira geração dos sistemas de workflow compreendeu, segundo descrito em [39], aplicações monolíticas de uma área de domínio particular. A segunda geração já dividiu os sistemas em componentes diversos, mas ainda os deixou fortemente dependentes das aplicações. A terceira geração apresentou máquinas de workflow genéricas que forneciam uma infra-

estrutura robusta para workflows orientados à produção. Nesta geração, a descrição dos workflows era construída através de ferramentas gráficas e, posteriormente, era interpretada pelas máquinas de workflow, as verdadeiras responsáveis pela sua execução. Segundo os autores, uma quarta geração seria a que se está presenciando nos últimos anos, a de sistemas de workflows que oferecem uma gama de serviços.

Com a evolução natural dos sistemas, tornou-se necessário separar processos de negócio e componentes de workflow das aplicações existentes, visando aumentar a flexibilidade e o poder de manutenção destes sistemas [39]. Além disso, os sistemas passaram de “orientados a dados” para “orientados a processos”. Como descrito em [5], a descentralização das organizações e da tomada de decisões, a necessidade por informação detalhada sobre as atividades diárias, assim como a ênfase na arquitetura cliente/servidor, a relevância dos sistemas federados e a crescente disponibilidade de tecnologia para processamento distribuído indicaram o fim do processamento centralizado e monolítico.

Atualmente, sistemas de workflow têm se tornado cada vez mais populares na medida em que conseguem resolver os problemas dos processos nas organizações, abrangendo desde o campo bancário e de seguros, até departamentos governamentais, empresas de telecomunicações, de atendimento ao usuário e de gerência de documentos. Com isso, diversas empresas têm se organizado no sentido de criar consórcios para o estudo e a padronização deste tipo de sistema.

A seção seguinte apresenta um breve histórico dos modelos transacionais até o surgimento dos então conhecidos *modelos de workflow transacionais*.

2.2

Modelos de Workflow Transacionais

O termo *workflow transacional* foi introduzido em 1993 por Sheth e Rusinkiewicz [109] para enfatizar a relevância das propriedades transacionais nos workflows, necessárias para especificar critérios de coordenação, corretude, consistência de dados, confiabilidade e suporte a falhas [133, 100]. Em [132, 133] é apresentada uma discussão sobre as diferenças entre os modelos transacionais de banco de dados e as transações em ambientes de workflow. Esta seção apresenta uma visão geral da evolução dos modelos transacionais até os modelos de workflow.

Modelos de transação convencionais estão diretamente ligados aos sistemas de banco de dados aos quais se aplicam e são modelos há muito consolidados. Resumidamente, modelos transacionais convencionais são centrados nos dados manipulados através do banco de dados e possuem como objetivo garantir as propriedades ACID (*Atomicity, Consistence, Isolation and Durability*) das transações. Por exemplo, quando uma transação falha, ela é abortada e, pelo princípio da atomicidade, todo o efeito das operações já realizadas deve ser revertido.

Modelos de transação convencionais foram se tornando cada vez mais limitados diante das grandes aplicações de banco de dados que se tornaram comuns. Surgiram então os chamados modelos estendidos ou avançados (os ATM, do inglês *Advanced Transaction Models*), cujo objetivo principal era relaxar as propriedades ACID das transações, permitindo execuções mais flexíveis em termos transacionais, ou seja, transações de maior duração e com requisitos de cooperação, coordenação e corretude mais complexos. Além disso, estes modelos estendidos vieram permitir que transações planas fossem agrupadas em estruturas hierárquicas, formando árvores transacionais.

Árvores transacionais possuem duas características importantes:

1. permitem a definição da ordem de execução das transações representadas na estrutura;
2. permitem a especificação de requisitos de isolamento e atomicidade sobre unidades menores, como, por exemplo, sobre uma ou mais subtransações.

Modelar um workflow como uma transação estendida significa que as subtransações correspondem às tarefas do workflow ou a subworkflows e a estrutura de execução da transação estendida corresponde ao fluxo de controle do workflow [100].

Sagas, transações flexíveis, transações aninhadas, dentre outros, são alguns dos modelos avançados encontrados na literatura. Um breve resumo destes modelos é apresentado em [105, 133].

O primeiro grande passo na evolução dos modelos transacionais básicos foi exatamente a extensão da estrutura das transações plana para uma estrutura hierárquica, de múltiplos níveis, cujo modelo foi chamado de *Modelo de Transações Aninhadas*. Neste modelo, uma transação-filha apenas pode começar depois que sua transação-pai tiver começado e cada transação-pai apenas pode terminar depois que todas as suas

transações-filhas forem terminadas. Ainda, se uma transação-pai é abortada, todas as suas transações-filhas também o são. Assim, o Modelo de Transações Aninhadas fornece isolamento completo, porque as transações-filhas são protegidas pelas transações-pai e, também, permite aumento da modularidade, tratamento de falhas em um nível mais baixo e um grau de concorrência entre as subtransações mais alto do que nas transações tradicionais. Uma extensão a este modelo, o *Modelo de Transações Aninhadas Abertas*, relaxa a propriedade de isolamento, tornando os resultados das subtransações terminadas visíveis a outras transações concorrentes aninhadas.

O modelo de **Sagas** [47] relaxa a propriedade de isolamento através do conceito de subtransações compensatórias. De acordo com este modelo, uma transação de longa duração é dividida em subtransações atômicas e a cada subtransação existe associada uma subtransação compensatória. Neste modelo, as transações são linearmente especificadas (ordem seqüencial), e quando uma subtransação falha, as subtransações executadas são compensadas em ordem reversa. No modelo de *Sagas Aninhadas*, a propriedade de atomicidade é relaxada permitindo que subtransações aninhadas que falharam sejam compensadas e a execução continue, método este conhecido por *forward recovery*.

No modelo de *Transações Flexíveis*, caminhos alternativos são fornecidos para que sejam seguidos caso a transação falhe. Uma transação flexível termina se as suas subtransações principais ou seus caminhos alternativos terminam. A definição destes caminhos alternativos está relacionada com a definição de possíveis estados de terminação para a transação. Pelo fato de transações flexíveis possuírem muitas das características de um modelo de workflow, elas foram provavelmente as primeiras transações avançadas a serem testadas em aplicações de workflow, segundo [133].

No entanto, apesar de terem sido diversos os modelos de transação avançados criados, eles não puderam de fato ser aplicados aos sistemas de workflow, segundo [109, 108, 105, 51, 132, 133, 39], pois:

- apesar de possuírem objetivos semelhantes, sistemas de workflow apresentam requisitos muito mais complexos dos que os grandes sistemas de banco de dados para os quais os modelos de transação avançados foram projetados;
- os modelos de transação avançados tornaram-se cada vez mais específicos às aplicações para as quais foram sendo projetados;

- o funcionamento de sistemas de banco de dados e de sistemas de workflow é bastante distinto. Por exemplo, em sistemas de workflow, tarefas podem não ser automáticas e o balanceamento de carga é possível através da análise das listas de trabalho (conjunto de atividades a executar) dos usuários, enquanto que em sistemas de banco de dados tarefas são sempre automáticas (operações realizadas sobre os dados) e o balanceamento de carga não é possível;
- workflows são atividades horizontais espalhadas entre diversos agentes/usuários e centradas no processo, de modo oposto aos bancos de dados, nos quais as operações são mais verticais, ou seja, cada usuário realiza diversas operações sobre os dados;
- a recuperação em caso de falhas em sistemas de workflow deve permitir que a execução continue. Além disso, essa recuperação é necessária não apenas no nível do próprio workflow, como também no nível de tarefa. Em sistemas de banco de dados, a unidade de recuperação é a própria transação;
- em sistemas de workflow nem sempre é viável e semanticamente correto desfazer o efeito de toda a transação que representa a instância de workflow em execução, em caso de ocorrência de falhas. Em sistemas de banco de dados, toda a transação é desfeita, segundo a propriedade de atomicidade;
- sistemas de workflow podem não fornecer todas as facilidades descritas pelos modelos de transação avançados;
- a execução de workflows representa processos bastante longos, que podem durar até dias, o que normalmente não acontece com as transações convencionais de sistemas de banco de dados.

A não aplicabilidade dos modelos avançados aos sistemas de workflow impulsionou o surgimento dos *modelos de workflow transacionais*. Workflows transacionais podem ser vistos como workflows suportados por um modelo avançado de transação que define critérios de corretude e confiabilidade, conforme descrito em [51]. Dentre os critérios, podem ser citados aqueles relativos ao controle de concorrência sobre itens de dados vinculados a diversas atividades ou subworkflows distintos, à recuperação em caso de falhas e a coordenação das atividades envolvidas em um determinado workflow.

Em [4] os autores argumentam que os modelos de workflow transacionais são um superconjunto dos modelos de transação avançados

ou estendidos e mostram, por exemplo, como os modelos de sagas e de transações flexíveis podem ser implementados através de modelos de workflows.

2.3

Modelo de Referência do WfMC

Em agosto de 1993 foi criado o mais importante dos consórcios ligados aos estudos de sistemas de gerência de workflow: o WfMC, do inglês *Workflow Management Coalition*. Ele, em conjunto com a WARIA (*Workflow And Reengineering International Association*), e com outras organizações não diretamente ligadas ao estudo de sistemas de workflow (como o OMG, do inglês *Object Management Group*, e o W3C, o maior consórcio de estudo e produção de tecnologias para a Web), procura definir padrões, conceitos e interfaces aplicados aos sistemas de workflow. WARIA é também uma organização sem fins lucrativos, cuja missão é estudar o que está acontecendo na intersecção entre re-engenharia de processos, gerência de workflows e comércio eletrônico.

De acordo com o WfMC, *workflow* é a automação ou facilitação computadorizada de um processo de negócio, no todo ou em parte. Também conforme o WfMC, um *sistema de gerência de workflow* é um sistema que completamente define, gerencia e executa workflows, cuja ordem de execução é dirigida por uma representação em computador de sua lógica [59].

Em [118] são apontadas 4 idéias básicas relativas aos sistemas de workflow:

- uma organização pode ser entendida como “realizando” processos;
- processos são descritos em uma forma que pode ser entendida ou interpretada por computadores;
- a informação a ser manipulada durante a execução de um processo pode ser compartilhada entre diversas instâncias de processo em execução;
- o sistema de workflow pode ser usado para coordenação do trabalho em uma organização descentralizada.

Visto deste modo, um sistema de gerência de workflow é um sistema que conecta usuários trabalhando em um processo em diferentes localidades e ambientes em diferentes partes de uma organização.

Uma das contribuições mais importantes do consórcio WfMC foi a padronização de diversos conceitos relativos a workflows. A Figura 2.1 apresenta estes principais conceitos e seus relacionamentos. O documento [130] contém uma relação dos conceitos aplicados a sistemas de workflow e uma explicação de como o WfMC trata cada um deles.

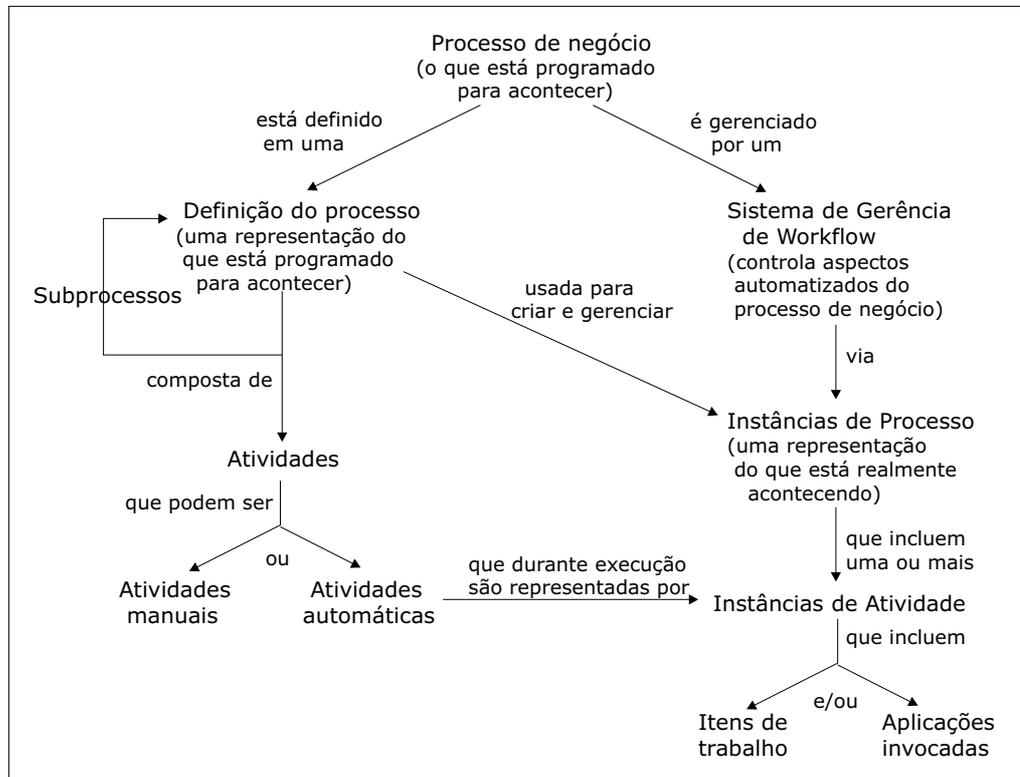


Figura 2.1: Conceitos relativos a sistemas de workflow padronizados pelo WfMC [59].

De acordo com a Figura 2.1, um processo de negócio é definido por uma definição de processo e é composto por um conjunto de atividades, manuais ou automáticas, ou por outras definições de processos. Um processo é gerenciado pelo sistema de gerência de workflows. Assim como atividades automáticas, processos são representados em tempo de execução por instâncias. Instâncias de processo são compostas pelas instâncias das atividades que dele fazem parte. Estas atividades podem incluir itens de trabalho ou aplicações invocadas.

Cada tarefa em um processo pode ser definida através de um modelo abstrato, que é basicamente uma máquina de estados cujo comportamento é dado pelas transições definidas entre eles. A estrutura de uma tarefa, no entanto, pode ser diretamente afetada pelas características do usuário que vai executá-la. O comportamento transacional do usuário, ou seja, o modo como ele executa as transações que envolvem as tarefas do processo,

determina o diagrama de estados da tarefa que nele executa. Tarefas podem estabelecer comunicação entre elas através das variáveis locais ao workflow. O fluxo de dados entre as subtransações (e portanto o seu relacionamento) é determinado através da associação de parâmetros de entrada e de saída às tarefas.

A Figura 2.2 ilustra as principais funções de um sistema de gerência de workflow. Após a definição do processo, ele é instanciado e executado pelo serviço de execução de workflows. A interação de usuários com o sistema durante a execução de instâncias ocorre de duas formas: (1) diretamente, sem nenhum intermediário, ou (2) através de ferramentas e aplicações que efetuem algum tipo de trabalho na execução.

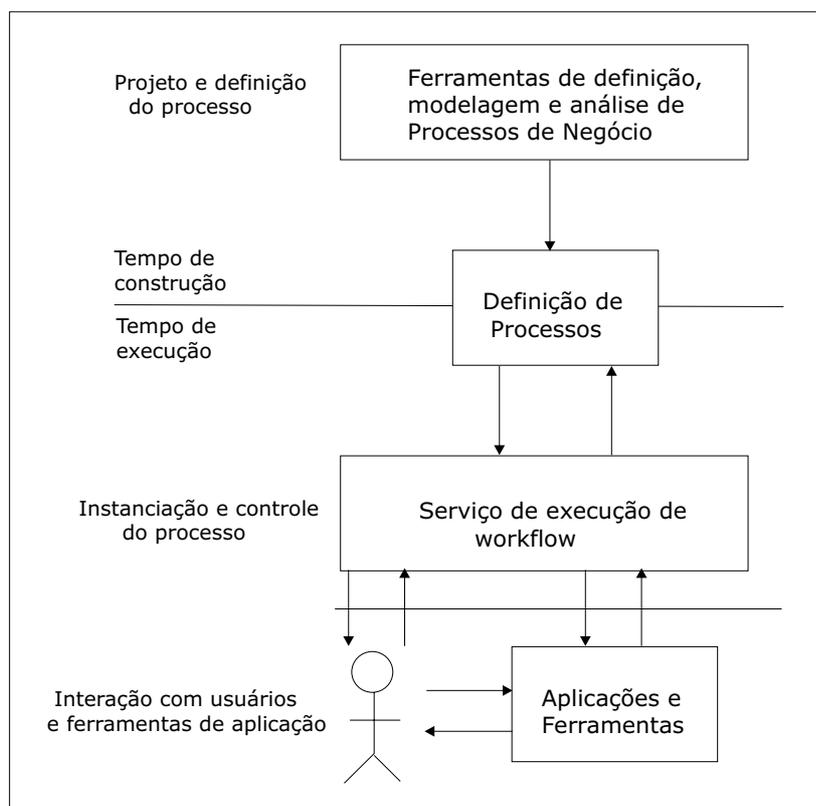


Figura 2.2: Características básicas dos sistemas de workflow e o relacionamento entre suas principais funções [59].

A distribuição de cada parte de um processo durante a sua execução é apresentada na Figura 2.3. Cada parte do processo pode ser executada em uma máquina particular dispersa na rede, podendo ou não necessitar da interação com o usuário. Algumas tarefas apenas fazem chamada direta a uma aplicação ou acesso a um banco de dados.

Os conceitos pertinentes aos sistemas de gerência de workflow foram utilizados posteriormente pelo WfMC para a criação do modelo de referência para sistemas de workflow [59], apresentado na Figura 2.4. Este modelo

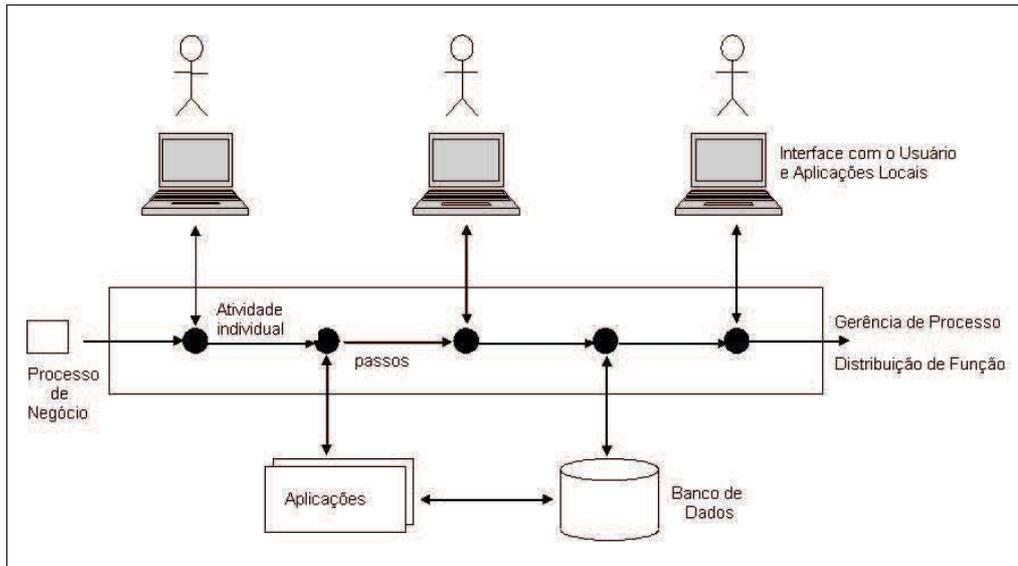


Figura 2.3: Distribuição de cada parte de um processo durante a sua execução [59].

não fornece detalhes de implementação. Ele é apenas uma abstração, que através de 5 interfaces permite que diferentes sistemas de workflow sejam interoperáveis.

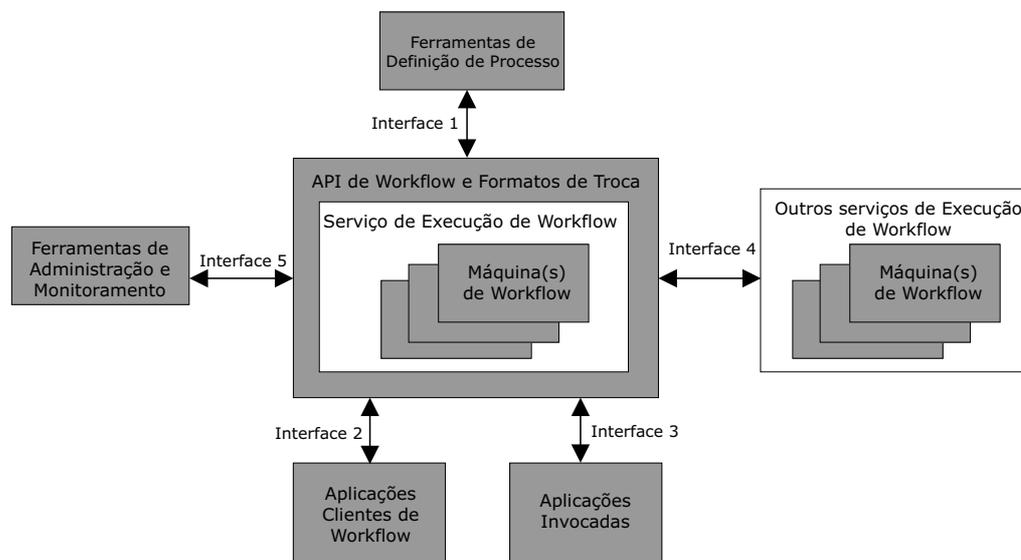


Figura 2.4: Modelo de referência do WfMC [59].

A Interface 1 do modelo de referência é uma interface padrão entre ferramentas de definição de processos e a própria máquina de workflow, para a importação e a exportação destas definições. Em dezembro de 2002, o WfMC anunciou a liberação da XPDL (*XML Process Definition Language*), a linguagem padrão para esta interface. Note que a máquina de workflow é apenas um dos componentes do sistema de gestão de workflow.

A definição do processo se divide, segundo o WfMC, entre o tempo de construção ou projeto e o tempo de execução. Esta divisão da etapa de definição do processo representa atualmente um dos desafios mais importantes para um sistema de workflow: permitir modelagens e, conseqüentemente, execuções mais flexíveis.

A Interface 2 define um padrão para a máquina de workflow gerenciar itens de trabalho, os quais são apresentados aos usuários (executores de tarefas). Ela facilita a integração da aplicação cliente com diferentes sistemas de workflow. Através desta interface, o usuário consegue selecionar tarefas para execução, recuperar detalhes da tarefa a ser cumprida e, por exemplo, invocar aplicações necessárias para a realização das tarefas.

Para essa interface foram definidas operações dentro do escopo da WAPI (*Workflow API and Interchange Formats*) [128]. A WAPI pode ser considerada um conjunto de construtores através dos quais os serviços do sistema de workflow podem ser acessados e os quais regulam as interações entre a máquina de workflow e os outros componentes do sistema. Para esta interface, são disponibilizadas operações de conexão e desconexão entre máquinas participantes, funções relativas ao estado do processo e comandos de manipulação das listas de trabalho de cada usuário, dentre outras operações.

A Interface 3 permite que a máquina de workflow invoque uma variedade de aplicações que devem ser executadas para o cumprimento de determinadas tarefas. Para dar suporte a esta interface, foram definidas algumas operações dentro da WAPI, dentre as quais podem ser citadas as operações de iniciar, suspender e abortar tarefas.

A Interface 4, por sua vez, define modelos de interoperabilidade, que determinam como ocorre a cooperação entre entidades (máquinas de workflow) distintas. Em [131], o WfMC apresenta uma especificação em XML para esta interface, conhecida como Wf-XML. Ela modela os requisitos de transferência de dados entre diferentes máquinas de workflow em XML, permitindo que o estado do processo e os resultados obtidos sejam trocados entre as múltiplas máquinas. Atualmente esta especificação encontra-se na versão 2.0.

Por fim, a Interface 5 define funções de monitoramento e controle da execução de um processo, através da especificação de um modelo comum para auditar dados. Em [128], o WfMC apresenta os métodos de acesso da WAPI projetados para estas funções de gerência, tais como delegação e suspensão de privilégios a usuários e grupos de usuários, terminação de instâncias de processos, dentre outras funções. Em [129] é apresentada uma

especificação de como criar um arquivo de log de um sistema de gerência de workflow, ou seja, de como auditar dados deste sistema.

Vale a pena ressaltar que as interfaces do modelo de referência do WfMC fazem uma separação clara entre uma visão abstrata e uma visão concreta, ou seja, as interfaces foram primeiramente definidas em termos abstratos e, posteriormente, foram definidos *bindings* concretos para as mesmas, de acordo com as tecnologias disponíveis. Como a tecnologia tem evoluído com o tempo, um grande número de diferentes especificações para interfaces - os chamados *bindings* concretos - tem sido definido. Cada especificação é feita de acordo com a tecnologia apropriada, mas sempre aderindo ao mesmo modelo abstrato.

2.4

Questões de Projeto em Sistemas de Workflow

Em [82, 6] são apresentadas algumas das questões mais importantes relativas ao projeto de sistemas de workflow. Dentre elas, podemos citar: arquitetura, recuperação em caso de falhas e atividades compensatórias, disponibilidade de dados e controle de concorrência. A maioria destas questões continua sendo alvo de muitos estudos porque ainda não foram devidamente resolvidas em grande parte dos sistemas atuais. Em [66] também pode ser encontrado um resumo sobre questões relativas à correteude em sistemas de workflow.

2.4.1

Arquitetura

Há basicamente três tipos de arquitetura para os sistemas de workflow [105]:

- arquitetura centralizada: nesta arquitetura, existe um único coordenador que é o responsável por distribuir todas as tarefas dos workflows concorrentes para execução;
- arquitetura parcialmente distribuída: nesta arquitetura, existe exatamente um coordenador para cada instância de workflow em execução;
- arquitetura completamente distribuída: nesta arquitetura, não existe um coordenador geral, mas cada máquina participante do sistema possui a tarefa de se comunicar com as demais para que as

dependências entre as tarefas sejam garantidas e para a satisfação de outros requisitos do sistema.

A arquitetura deve ser definida conforme a necessidade do sistema. No caso desta tese, a arquitetura definida no Capítulo 8 é distribuída.

2.4.2

Recuperação em Caso de Falhas

A recuperação de sistemas de workflow em caso de falhas é uma das questões mais discutidas atualmente. O importante é que estes sistemas garantam que a execução nunca seja interrompida, independentemente da falha ocorrida.

Há três categorias de erros possíveis associados a sistemas de gerência de workflow [133]:

- erros de infra-estrutura: erros de comunicação ou de hardware;
- erros de sistema: provenientes de falha no próprio software do sistema de workflow; e
- erros de aplicações e usuários: decorrentes da execução de uma determinada tarefa por parte dos mesmos.

Se o erro é de sistema e ocorre no banco de dados onde estão contidas as informações dos workflows, o mecanismo de replicação, se presente, pode resolver o problema (veja comentários na Seção 2.4.3). Neste caso, quando os dados necessários para a execução da tarefa estiverem locais ao dispositivo em que ela está sendo executada, a execução não é interrompida. No entanto, se o acesso for remoto, não será possível continuar a execução da tarefa.

Se o erro, por outro lado, ocorre no sistema de gerência de workflow propriamente dito, as operações sob responsabilidade do sistema devem ser executadas em partes que ainda estão funcionando corretamente. Para que isto seja possível, devem ser permitidas múltiplas conexões entre os componentes do sistema, de modo que a falha em uma conexão não altere o funcionamento das demais.

No contexto de recuperação em caso de falhas ou erros, algumas alternativas podem ser utilizadas com o intuito de evitar que a execução seja interrompida. Por exemplo, quando uma tarefa não puder ser executada, diversas tentativas podem ser feitas até que se consiga executá-la (como no caso da falta de recursos) ou tarefas alternativas podem ser executadas (quando um certo número de tentativas falhou) para evitar que a tarefa

falhe. Se uma tarefa alternativa for executada no lugar da que tinha falhado, considera-se que a falha foi resolvida. No entanto, esta abordagem de utilização de tarefas alternativas ou novas tentativas não ocorre em grande parte dos sistemas de workflow.

O mecanismo de tratamento de exceção proposto nesta tese faz justamente o trabalho de evitar que a execução de uma instância de workflow (ou processo, como será chamado) seja interrompida quando não se tem os recursos necessários para a execução de um subworkflow, não se conhece todas as informações necessárias para executá-lo, ocorre conflito ou quando a instância é abortada.

Para que a recuperação seja possível no nível de tarefas e não apenas no nível de workflows, é necessário que toda a informação de execução do workflow, inclusive aquelas relativas a cada tarefa individual, sejam armazenadas de modo persistente.

Se a execução de uma instância de workflow é de fato abortada, ações compensatórias devem ser executadas pela máquina de execução para desfazer os efeitos semânticos das operações, sempre que possível. Estas ações não levam, necessariamente, o banco de dados que controla a execução do workflow para o estado válido anterior à falha, mas sim o leva para um estado válido e semanticamente próximo. Isso porque muitas vezes os efeitos transacionais das tarefas de um workflow não podem ser desfeitos.

Existem basicamente duas maneiras possíveis de se recuperar uma instância de workflow que falhou: através de *backward recovery* e através de *forward recovery*. Na abordagem de *backward recovery* a instância de workflow é retornada, através de ações compensatórias, para o estado consistente que existia antes da transação que representa a instância de workflow abortada (que falhou). Na abordagem de *forward recovery* a instância que falhou volta para um estado válido e continua a sua execução a partir do ponto em que havia falhado.

De acordo com o mecanismo que será apresentado nesta tese, uma instância de workflow pode se recuperar de uma falha através de workflows compensatórios, conforme a abordagem *forward recovery*. No entanto, assim como ocorre tradicionalmente nas transações de banco de dados convencionais, não é o caso que para todo workflow existe um workflow compensatório.

Mecanismos de *check points* também podem ser úteis e tornar o processo de recuperação mais eficiente, já que os dados são salvos de tempos em tempos.

Na verdade, os mecanismos de recuperação são sempre intimamente

ligados à infra-estrutura do sistema, à sua arquitetura, à natureza dos executores (pessoas, aplicações, protocolos), aos tipos de tarefas (tarefas do sistema ou do usuário, tarefas transacionais) e da natureza da própria aplicação de workflow. Dessa forma, um único mecanismo de recuperação de falhas pode não ser capaz de resolver problemas de vários sistemas semanticamente distintos.

2.4.3

Disponibilidade de Dados

A disponibilidade de dados também é um fator essencial em sistemas de gerência de workflow. Na maioria dos casos, alta disponibilidade de dados é alcançada por meio de mecanismos de replicação, onde os dados são replicados nos dispositivos de cada membro do grupo em cooperação.

Além de replicação, também são necessários mecanismos que garantam que, mesmo diante da reconfiguração dinâmica do sistema, por exemplo, da incorporação ou da exclusão de servidores, o sistema permaneça acessível.

Este requisito é bastante importante e requer mecanismos específicos, conforme apontado acima. Porém, aborda questões ortogonais ao foco desta tese e, portanto, não será tratado no texto.

2.4.4

Controle de Concorrência

A questão de concorrência em sistemas de workflow também é de fundamental importância, porque na maioria das vezes vários usuários têm acesso concorrente ao mesmo dado durante a execução. Algumas abordagens para o controle de concorrência são apresentadas em [51].

Na abordagem baseada em *check-in* e *check-out*, cada item de dado sobre o qual é feito o *check-out* é marcado no sistema, de modo que este tem controle de quem está utilizando qual item. No momento do *check-in*, este item é atualizado, para que sempre os usuários em cooperação trabalhem com a versão mais atualizada possível do dado. Observe que esta abordagem é diferente da abordagem de bloqueio de um item de dado, já que na abordagem de *check-in* e *check-out* um mesmo item pode estar sendo utilizado concorrentemente por vários usuários.

A abordagem baseada em versões é basicamente uma abordagem otimista, que permite que diversos usuários tenham acesso ao mesmo item de dado, mas criando novas versões para este item a cada nova atualização.

Nesta abordagem é necessária a intervenção humana ou algum algoritmo de seleção de valor, que determine, em caso de valores conflitantes, qual deve ser o valor correto para o item de dado para o qual foram criadas diferentes versões.

Na abordagem de execução proposta nesta tese, utilizamos uma política otimista. A máquina de execução recupera os valores necessários à execução de uma determinada instância, sem bloqueá-los. No entanto, o comportamento transacional de uma instância, em adição ao teste de consistência, conforme explicado no Capítulo 7, garante a correta terminação das instâncias em execução.

2.5

Modelo de Implementação do WfMC

Em [59] o WfMC define um modelo de implementação geral para sistemas de workflow, ilustrado na Figura 2.5, baseado no modelo de referência apresentado na Figura 2.4.

Basicamente, há três tipos de componentes:

- os componentes de software, que são a ferramenta de definição de processo, a própria máquina de execução, o programa de interface com o usuário e o programa que manipula a lista dos itens que o usuário tem para executar (chamados pelo WfMC de *itens de trabalho*);
- as aplicações ou dados externos ao sistema de workflow, mas que podem ser invocados ou utilizados por ele durante a execução de um processo;
- dados de controle do sistema. Dentre estes dados, podemos citar a própria definição do processo e os dados relevantes para a execução do workflow. Além disso, o sistema mantém controle sobre a lista de tarefas de cada usuário, sobre os dados produzidos a partir das aplicações invocadas ou dos bancos de dados acessados e sobre os dados relativos ao próprio estado de execução do processo.

A partir desta arquitetura, é possível construir sistemas de workflow interoperáveis, qualquer que seja a tecnologia escolhida. Observe que esta arquitetura define detalhes não presentes no modelo de referência, tais como aqueles relativos às listas de trabalho manipuladas via interface de usuário.

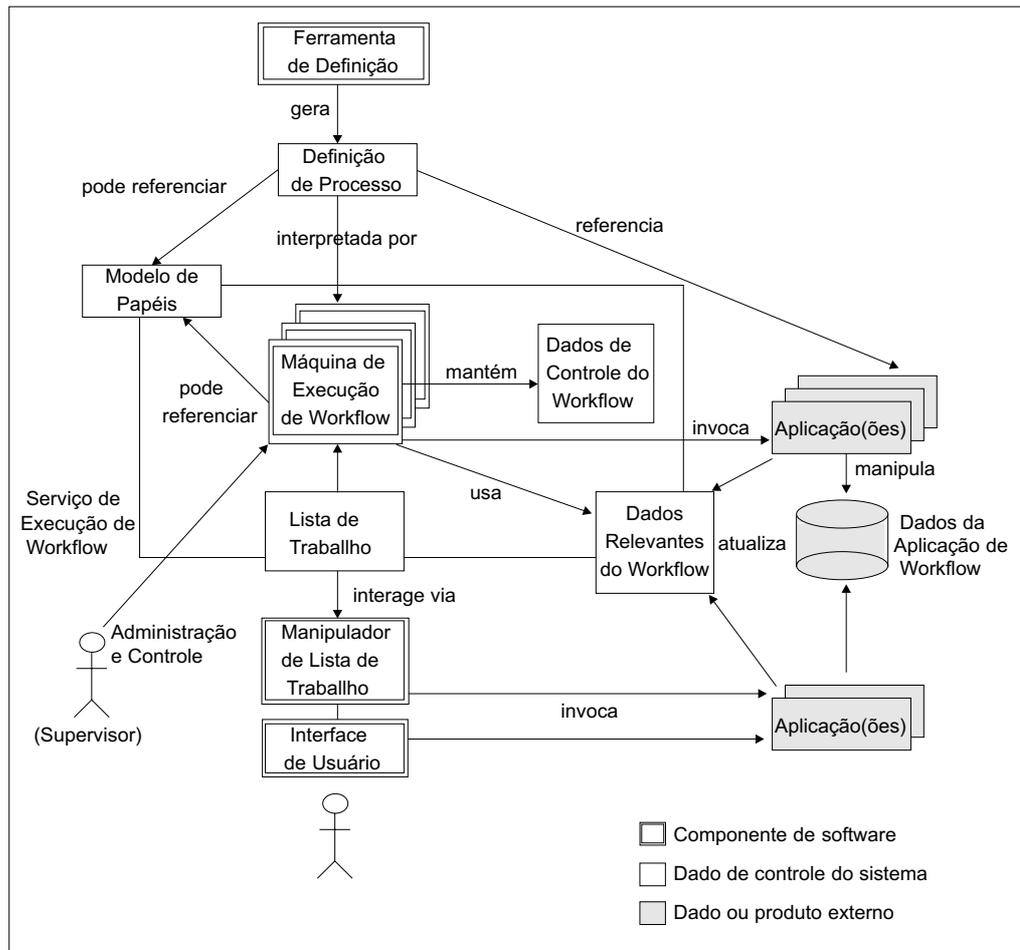


Figura 2.5: Modelo de implementação de sistemas de gestão de workflow, definido pelo WfMC [59].

2.6 Limitações dos Sistemas de Workflow

Apesar dos diversos esforços de padronização, os sistemas de workflow sempre foram bastante criticados quanto à falta de infra-estrutura adequada para atender às necessidades das organizações, à falta de flexibilidade para o atendimento de tarefas complexas, à falta de maturidade com relação à tecnologia existente, dentre outros fatores [39].

Dentre as diversas limitações geralmente citadas [6] [5], pode-se apontar:

- interoperabilidade: sistemas de workflow incorporam interpretações distintas sobretudo dos processos de negócio, tornando praticamente impossível federar sistemas diferentes, mesmo diante de esforços de padronização advindos de organizações tais como o WfMC;
- escalabilidade: a maioria dos sistemas foi originalmente planejada para dar suporte a um pequeno número de usuários em cooperação. O

aumento da demanda em face à popularização destes sistemas muitas vezes aponta a necessidade de re-projeto;

- disponibilidade de dados e arquitetura centralizada: grande parte dos sistemas possui um único ponto de falha, que é normalmente o próprio banco de dados, o qual muitas vezes não possui mecanismo de backup. Além disso, sistemas de workflow normalmente são construídos com uma única máquina de execução, acessada por todos os usuários de todos os workflows correntemente em execução. Estes dois itens do sistema passam a ser seu principal gargalo;
- corretude: muitas vezes ameaçada pela ausência de restrições que gerenciem a cooperação entre sistemas distintos. Como normalmente são vários os sistemas em cooperação, a corretude está diretamente vinculada ao controle de concorrência, em especial;
- independência de linguagem: sistemas de workflow geralmente apresentam-se bastante dependentes da linguagem de especificação do workflow, conseguindo, deste modo, apenas executar workflows descritos em uma linguagem específica.

Existe ainda outra gama de limitações que tem sido cada vez mais estudada e que determina o foco desta tese de doutorado. Trata-se do conjunto de limitações relativas à execução dos workflows. De fato, a execução de um workflow tende a ser bastante rígida se comparada à sua descrição estática. Ou seja, normalmente a execução segue fielmente a ordem descrita na definição estática do workflow. Neste sentido, se, por exemplo, a informação do valor de uma variável relacionada a um subworkflow não é conhecida no momento em que a máquina de execução deve executá-lo, a execução do workflow maior é interrompida. Um dos objetivos desta tese é oferecer à máquina de execução informação adicional para que em situações como esta a execução possa proceder, como será apresentado nos capítulos seguintes.

Na arquitetura proposta nesta tese, a corretude é inerente ao comportamento transacional dos processos executados da máquina de execução e a escalabilidade é alcançada, na medida em que se prevê, na arquitetura, a inclusão de novas máquinas de execução (chamadas de gerentes de processo).

2.7

Workflow e Gerência de Processos de Negócio

O consórcio WfMC sempre reconheceu um workflow como a interpretação de um modelo de processo, gerência de tarefas manuais e automáticas e de regras de negócio e procedimentos de tratamento de exceções [99]. Este conceito está sendo cada vez mais utilizado na medida em que cresce a cooperação entre o WfMC e o BPMI (*The Business Process Management Initiative*).

Em [101] é apresentada uma discussão geral a respeito do conceito *Gerência de Processos de Negócio*, do inglês *Business Process Management* (BPM). Softwares de negócio tais como SAP, Oracle e vários softwares de mercado para gestão eletrônica de documentos, há muito já dão apoio a processos de negócio. No entanto, nas décadas passadas estes processos eram rígidos, difíceis de manter, caros para serem configurados e, pior do que isso, frequentemente obsoletos para o momento em que eram definidos. Segundo uma visão mais recente destes modelos, os problemas que os processos se destinam a resolver devem de fato ser resolvidos através de métodos mais baratos, rápidos e efetivos. Para isto, estes problemas são vistos como um conjunto de processos integrados e bem-definidos. Em [117] a gerência de processos de negócio é considerada uma extensão dos sistemas de gerência de workflow, não tratando de uma revolução ou de uma nova geração de ferramentas.

A Web trouxe, ao longo destes últimos anos, uma nova infra-estrutura para a construção de sistemas, com base nos conceitos de serviços na Web [68], conteúdo de informação estruturada em XML e escalabilidade [60]. No entanto, uma das características fundamentais requerida pelos processos de negócio é habilidade de dar suporte à gerência flexível de alterações dinâmicas no negócio. Isso quer dizer que é importante permitir que processos de negócio sejam modificados em tempo de execução, com o intuito de possibilitar a adequação do processo diante de desvios ou exceções não previstas, por exemplo, ou mesmo diante de alterações agora necessárias. Esta característica de flexibilidade sugere:

- *late bindings*: introduz flexibilidade ao ambiente de execução, permitindo que ligações dinâmicas sejam feitas em tempo de execução, em face de uma modelagem estática menos rígida;
- máquina de regras: facilita a avaliação de expressões complexas, independentemente da especificação do processo. Esta máquina realiza

- a leitura das regras de negócio definidas e faz a análise das mesmas, garantindo a correteza da execução de um processo;
- processos adaptativos: facilita alterações dinâmicas em tempo de execução, permitindo alterações na estrutura do processo de negócio enquanto ele está sendo executado.

Vale lembrar que estas necessidades relativas aos processos não foram descobertas apenas junto com o conceito de processo de negócio, mas sim se tornaram mais significantes à medida em que estes alcançaram um lugar de destaque entre as organizações.

Esta tese está focada em aspectos bastante semelhantes a estes. A idéia de flexibilização proposta através do conceito de tratamento de exceções em diversos níveis, no entanto, não parte do pressuposto de que alterações na estrutura do workflow são necessárias em tempo de execução, mas sim de que são necessárias algumas modificações durante a execução para que esta não seja interrompida. As modificações propostas, no entanto, são definidas de acordo com informações semânticas adicionais à descrição dos workflows. A idéia de *late binding* também é empregada.

O mecanismo de tratamento de exceção para a flexibilização empregado nesta tese permite, portanto, que a execução de uma instância de workflow continue mesmo diante de informação incompleta e também quando os recursos necessários à execução não estão disponíveis. Além disso, o mecanismo permite que a completa modelagem de um workflow seja adiada para o tempo de execução, possibilitando que a definição do workflow contenha referência a subworkflows ou recursos abstratos, ou seja, a subworkflows ou recursos não diretamente utilizáveis.

2.8 Resumo

Sistemas de gerência de workflow são bastante antigos, mas tiveram a sua popularidade aumentada nas últimas décadas, em decorrência das necessidades de negócio e organização que surgiram ao longo dos anos nas empresas. Este aumento de popularidade provocou um maior empenho pelo desenvolvimento e pelo avanço de sistemas gerência de workflow. Em particular, o WfMC vem trabalhando no sentido de tornar os sistemas de workflow capaz vez mais difundidos e de criar padrões para o seu desenvolvimento.

O foco deste trabalho é, partindo de sistemas já existentes ou mesmo pensando em novos sistemas a serem construídos, propor um mecanismo que torne flexível a execução de workflows, permitindo que:

- a execução de uma instância de workflow continue mesmo diante de informação incompleta ou quando os recursos necessários à execução não estão disponíveis;
- a completa modelagem de um workflow seja adiada para o tempo de execução, possibilitando que a definição do workflow contenha referência a subworkflows ou recursos abstratos.