

# Sistemas de Gerência de *Workflows*: Características, Distribuição e Exceções

Luiz Antônio M. Pereira  
lpereira@inf.puc-rio.br

Marco Antonio Casanova  
casanova@inf.puc-rio.br

PUC-RioInf.MCC11/03 Março, 2003

## Abstract

Workflows can be defined as any set of tasks that have to be executed in a coordinated way, in sequence and/or in parallel, by two or more individuals or teams sharing a common goal. Workflow Management Systems - WfMS - are pieces of software designed to automate, at least, the administrative/coordination tasks associated to workflows execution. In this work we will present the main concepts and characteristics of workflow systems considering, as a first step, centralized environments. Further on in the text, due to the typical necessity of workflows to cross departmental borders within an enterprise and between enterprises, we discuss extensions of WfMS required to allow cooperative work of individuals or teams that are geographically dispersed. In this work we also discuss issues related to possible failures during workflow executions.

**Keywords:** Workflow Systems, Workflow Management Systems, Distributed Workflows, Exceptions in Workflows.

## Resumo

*Workflows*, ou fluxos de trabalho, podem ser definidos como qualquer conjunto de atividades executadas de forma coordenada, em série e/ou em paralelo, por indivíduos ou grupos com um objetivo em comum. Sistemas de Gerência de *Workflow* - SGWs - são peças de *software* que visam automatizar (pelo menos) as atividades de administração/coordenação relativas à execução de *workflows*. Neste trabalho apresentamos os principais conceitos e características dos sistemas de *workflow* buscando, em uma primeira etapa, nos ater a ambientes centralizados. Mais adiante, em função da necessidade típica dos fluxos de trabalho atravessarem as fronteiras setoriais dentro de uma mesma empresa, e até entre empresas, discutimos extensões necessárias aos SGWs para que estes permitam o trabalho cooperativo de indivíduos e equipes situadas em locais geograficamente distantes. Nesse trabalho também são discutidas questões quanto a possíveis falhas durante a execução de *workflows*.

**Palavras-chave:** Sistemas de *Workflow*, Sistemas de Gerência de *Workflows*, *Workflows* Distribuídos, Exceções em *Workflows*.

# 1 – Introdução

## 1.1 – Workflows e Sistemas de Workflow: Fundamentos

Existem inúmeras definições para *workflow* na literatura. Segundo o Modelo de Referência de *Workflow* da *Workflow Management Coalition - WfMC* [1], *workflow* é a automação de um processo de negócio, por inteiro ou por partes, durante o qual documentos, informações e atividades são passadas de um participante para outro para que estes desenvolvam ações respeitando um conjunto de regras procedimentais. *Workflow*, ou fluxo de trabalho, também pode ser definido como [5] qualquer conjunto de atividades executadas de forma coordenada, em série ou em paralelo, por dois ou mais membros de um grupo de trabalho, visando um objetivo comum.

Embora a WfMC considere a possibilidade de um *workflow* ser manualmente organizado, na prática a maioria deles o é dentro do contexto de um sistema de TI, para prover suporte computadorizado às suas automações procedimentais. As atividades podem ser executadas em seqüência ou simultaneamente, por diferentes indivíduos, ou pela combinação dos dois. Segundo Moro em [5], se somente uma pessoa executar todas as atividades, isso não caracteriza um *workflow*, na medida em que, como o próprio nome sugere, um processo é um *workflow* se os artefatos<sup>1</sup> "fluem" de um indivíduo (executor) para outro, produzidos e/ou consumidos pelas diversas atividades do processo. Os participantes de um *workflow* devem estar *colaborando* em busca de um objetivo comum, ou seja, projetos independentes não constituem um *workflow*. Além disso, *workflows* não se aplicam unicamente a processos de negócio.

Considerando as definições e conceitos comuns em toda a literatura pesquisada, entendemos que um *workflow* pode ser definido como uma coleção de atividades organizadas para realizar um processo, *quase sempre* de negócio. Essas atividades podem ser executadas por um ou mais sistemas de computador, por um ou mais agentes humanos ou de *software*, ou então por uma combinação destes. Do que consistem, a ordem de execução e as pré-condições das atividades estão definidas no *workflow*, sendo que o mesmo é capaz ainda de representar a sincronização das atividades e o fluxo de informações entre elas.

*Workflow* é um conceito intimamente relacionado à reengenharia e automação de negócios e de processos de informação em uma organização [20]. A necessidade de escritórios mais eficientes resultou no conceito de Reengenharia de Processo de Negócio e na tecnologia chamada de Sistemas ou Software de Gerência de *Workflow* (SGWs) [2][20] ou, simplesmente, Sistemas de *Workflow* [3].

SGWs são, em geral, ferramentas colaborativas [26] e provêm a automação procedimental de *workflows*, gerenciando a seqüência de atividades de trabalho e chamando ou invocando os recursos humanos e/ou eletrônicos apropriados, que são associados com as várias atividades que compõem o processo. SGWs definem, gerenciam e executam completamente *workflows*, através da execução de *software*

---

<sup>1</sup> Qualquer recurso, físico ou digital, produzido ou utilizado em uma etapa do processo.

baseada em uma representação da lógica ou modelo (dados, operações e regras) do *workflow* no computador.

Nesses sistemas as atividades são definidas e agendadas, os recursos necessários são relacionados e, na medida em que as atividades vão sendo executadas pelos respectivos executores, o sistema trata de coordenar e encaminhar automaticamente os resultados alcançados e os demais recursos necessários para os executores das atividades seguintes na seqüência.

## **1.2 – Motivação e Objetivos**

Nosso grupo de pesquisas concentra-se no estudo de infraestruturas de execução, armazenamento e recuperação de componentes de ensino a distância (objetos de aprendizado).

No contexto de EAD verifica-se a aplicabilidade das técnicas de *workflow*, na medida em que a participação coordenada e colaborativa de alunos, professores e autores deve, idealmente, existir, não só na execução de um curso, quando participam alunos e professores, como também na sua modelagem (preparação), quando participam grupos de professores/autores.

Com esse trabalho objetivamos, portanto, fundamentar nossas pesquisas em EAD através do estudo das funcionalidades de Sistemas de Gerência de Workflow e do estudo das questões de distribuição e heterogeneidade de SGWs, considerando, inclusive, a possibilidade de ocorrência de exceções durante a execução.

Em seguida, em um próximo trabalho, estaremos pesquisando objetos de aprendizado (*learning objects* ou LOs), padrões de estruturas, suas composições e operações.

Esses estudos devem nos deixar em posição de, em uma terceira etapa, propormos uma arquitetura otimizada para um ambiente de EAD baseado em *workflows*, onde os artefatos são objetos de aprendizado e os processos referem-se à aplicação e avaliação de conteúdos de aprendizado.

## **1.3 – Conteúdo do Trabalho**

No capítulo 2 apresentaremos as características gerais dos SGWs, fazendo um breve histórico, e descrevendo seus componentes (atividades, executores, rotas, documentos e regras), categorias, funcionalidades, vantagens e exemplos de SGWs de mercado.

No capítulo 3 apresentaremos as duas metodologias de modelagem de *workflows*: as baseadas em comunicação e as baseadas em atividades. Faremos, ao final, uma proposta de modelagem de *workflows* utilizando os diagramas de atividades da UML.

Os primeiros SGWs atenderam a grupos de poucos usuários em ambientes centralizados. Com a crescente aceitação dos *workflows*, a tecnologia é usada cada vez mais em grandes organizações, que geram fluxos de trabalho complexos, pesados e que são inerentemente distribuídos. Isso nos leva a pensar em estender os SGWs existentes dotando-os de funcionalidades que atendam à necessidade de distribuição, gerando SGWs distribuídos, ou SGWDs, cuja heterogeneidade deve ser considerada. Esses aspectos são tratados no capítulo 4.

É importante, também, considerarmos a possibilidade de ocorrência de exceções durante a execução de um *workflow*. Exceções podem demandar tratamentos complexos, sobretudo quando ocorrem em ambientes distribuídos. Exceções são discutidas no capítulo 5.

Finalmente, diante da possibilidade de aplicarmos as soluções já estudadas para problemas de sistemas de gerência de banco de dados distribuídos e heterogêneos (SGBDDHs) em SGWDs, no capítulo 6 discutiremos as semelhanças entre SGBDs e SGWs<sup>2</sup>; o primeiro provê *software* que permite o controle, através de um conjunto bem definido de operações e regras disponíveis, sobre *dados armazenados*, o segundo provê *software* que permite o controle, também via um conjunto bem definido de operações e regras, sobre processos definidos através dos dados que os especificam.

## **2 - Características dos Sistemas de Gerência de Workflow**

### **2.1 - Breve História dos Sistemas de Gerência de Workflow**

Os sistemas de *workflow* têm suas origens a partir das pesquisas em automação de escritórios nos idos anos 70 [3]. O principal foco destas pesquisas estava em oferecer soluções sobre como gerar, armazenar, compartilhar e *rotear* documentos em organizações, visando à diminuição da manipulação de documentos em papel.

A partir da década de 80, o principal objetivo da tecnologia de *workflow* passou a ser o de unir-se as chamadas “ilhas de trabalho e informação” individuais e personalizadas de cada membro (ou pequeno grupo de indivíduos) de uma organização, buscando a integração das *ilhas* através do roteamento do trabalho entre elas. Nessa época havia uma preocupação em como definir paradigmas e linguagens para a modelagem de processos de trabalho e em como construir arquiteturas para a implementação de sistemas capazes de melhor interpretar e executar tais processos.

Nos anos 90, a tecnologia de sistemas de *workflow* evoluiu muito rapidamente, atuando, ao mesmo tempo, como causa e consequência do também rápido crescimento das infra-estruturas de redes de computadores e dos ambientes para interação entre grupos que se formaram em torno dessas infra-estruturas. As recentes questões

---

<sup>2</sup> Sistemas de Gerência de *Workflows* têm despertado o interesse da comunidade de pesquisadores da área de bancos de dados como têm sido estudados em disciplinas relacionadas em universidades do mundo inteiro.

relacionadas ao processamento distribuído e interoperabilidade entre aplicações trouxeram novos desafios à definição e construção de arquiteturas para sistemas de *workflow* [3].

Hoje, a tecnologia de *workflow* alcança bem mais do que a redução do fluxo de documentos em papel nas organizações. Os conceitos e paradigmas de trabalho em grupo, preconizados pelas pesquisas em CSCW (*Computer Supported Collaborative Work* - Trabalho Colaborativo Suportado por Computador) e *groupware* (agendamento de conferências, correio eletrônico, vídeo conferência, por exemplo) influenciaram a definição destes sistemas como ferramentas para a coordenação do trabalho de equipes, impulsionando seu desenvolvimento. Além disso, as necessidades de interação intra-organizacionais estenderam-se para níveis inter-organizacionais (*Business-to-Business* ou B2B), agora contando com o potencial da WWW. Esse fato levou as pesquisas em *workflow* a um novo patamar voltado para a definição de *arquiteturas distribuídas* de execução de processos. O trabalho cooperativo, agora, é descentralizado, de forma a permitir:

- Que cada parte do processo de *workflow* possa ser executada no local mais apropriado, usando os recursos disponíveis desse local;
- Que cada componente ou fragmento do processo remoto possa progredir o mais independentemente possível dos outros processos com os quais é coordenado;
- Que os dados locais, o estado de execução, as ferramentas e as demais partes do processo de *workflow* possam ser manuseados remotamente, de acordo com esquemas de segurança de acesso pré-definidos.

A flagrante tendência de utilização de estações de trabalho com poderes de computação cada vez maiores e as infra-estruturas de rede cada vez mais velozes e confiáveis, corroboram para a perpetuação do cenário acima descrito.

## **2.2 - Vantagens do Uso de Sistemas de Gerência de *Workflow***

Embora não tenhamos, ainda, uma descrição das funcionalidades tipicamente encontradas nos SGWs, podemos identificar e aplicar ao contexto de fluxos de trabalho uma série de vantagens genéricas entre processos manuais e os resultantes de suas automações. Algumas vantagens do uso de SGWs podem, com isso, ser destacadas:

- Diminuição da necessidade circulação de documentos em papel;
- Possibilidade de acesso remoto por parte dos participantes do grupo;
- Simplificação das atividades de arquivamento e recuperação de informações;
- Rapidez na pesquisa de informações armazenadas;
- As informações dos responsáveis por cada atividade do processo são mantidas sempre (automaticamente) atualizadas;

- Conhecimento do status do processo a cada instante, possibilitando saber-se quais os participantes estão atuando, quais são os próximos a atuar, e quando, etc.;
- O SGW *coordena* a execução das atividades automaticamente com o uso de agendas e trocas de mensagens eletrônicas com os participantes.

Ainda, de [17]:

- Eficiência melhorada - A automação de muitos processos de negócios resulta na eliminação de muitos passos desnecessários;
- Melhor controle do processo - Um melhor gerenciamento de processos de negócios é atingido por meio da padronização dos métodos de trabalho e da disponibilidade de registros para auditoria;
- Melhor atendimento ao cliente - Consistência nos processos levam à uma maior previsibilidade nos níveis de resposta aos clientes;
- Flexibilidade - O controle dos processos via software permite o re-projeto em linha com as necessidades de mudança no negócio, e
- Melhoria no processo de negócio - O foco nos processos de negócio levam à obtenção de processos mais eficientes e simples.

## **2.3 - Componentes dos Workflows**

Os componentes fundamentais de um processo ou fluxo de trabalho são as atividades. Atividades, por sua vez, pressupõem que existam atores que as executam (*interpretando* seus papéis), os documentos produzidos e consumidos durante a execução das atividades, rotas que definem os destinos de cada fluxo de informação e regras que devem ser respeitadas durante a execução das atividades. Damos, a seguir, uma breve definição de cada um desses componentes.

### **2.3.1 - Atividades**

Uma atividade em um fluxo de trabalho corresponde a uma etapa a ser executada dentro de um processo. É necessário que os SGWs ofereçam recursos para a definição dos dados das atividades como, por exemplo, o nome da atividade, seus objetivos, instruções a serem seguidas durante sua execução e os dados ou documentos necessários para sua realização.

### **2.3.2 - Executores: Indivíduos, papéis, atores, grupos e agentes**

Cada atividade deve ter pelo menos um *executor* responsável por sua realização. Esse executor deve ser devidamente cadastrado como usuário do SGW. A associação do

executor às atividades sob sua responsabilidade pode ser realizada no momento da definição do fluxo de trabalho ou (semi) automaticamente pelo SGW, caso ele conheça os perfis dos executores. Executores podem ser indivíduos – usuários do SGW - específicos (referenciados por seus nomes) ou, idealmente, em função da típica rotatividade de pessoal nas organizações, devem ser referenciados por seus *papéis*. Um papel, representado por um *ator* (classe de indivíduos), caracteriza um conjunto de atributos (perfil) e um conjunto de responsabilidades que os indivíduos que os desempenham devem possuir. Indivíduos devem ser associados a papéis e estes, então, devem ser associados às atividades. Um mesmo indivíduo pode representar vários papéis; um mesmo papel pode ser representado por vários indivíduos.

SGWs podem permitir, ainda, que sejam definidos grupos de indivíduos responsáveis pela execução de atividades em processos. Dentro de um grupo, cada indivíduo pode ainda ter um papel definido.

Atividades podem ainda ser executadas por *agentes* de software (ou agentes computacionais), aplicações ou dispositivos quaisquer.

### **2.3.3 - Rotas**

A definição de um fluxo de trabalho compreende, também, a explicitação do encadeamento de atividades do processo (*roteamento*). Esse encadeamento ocorre, de um modo geral, segundo um grafo orientado, onde atividades (representadas pelos nós do grafo) podem ser executadas em paralelo, segundo uma ordenação parcial, e/ou seqüencialmente, quando o grafo orientado de execução de atividades particularizar-se em uma seqüência única de atividades. Expressões de guarda podem ainda ser aplicadas às arestas dos grafos para adicionar-se condições de execução de parte das atividades. A figura 2.1 ilustra.

### **2.3.4 - Documentos e Formulários**

Um item importante na definição de um fluxo de trabalho é a especificação de quais informações irão fluir durante a execução dos processos. Os documentos e demais artefatos manipulados ao longo da execução dos processos poderão ser armazenados em *pacotes de trabalho*. Documentos podem ser consultados, alterados, armazenados ou retirados dos pacotes de trabalho pelos executores das diversas atividades, conforme regras e permissões definidos *a priori*. Formulários também podem ser roteados ao longo da execução de um fluxo de trabalho, onde cada executor é responsável pelo preenchimento/atualização de um conjunto de informações.

### **2.3.5 - Regras**

A execução de qualquer atividade em um ambiente organizado pressupõe a aceitação e obediência a regras pré-estabelecidas. Quando a execução de atividades dever ser feita de forma coordenada, envolvendo mais de um executor, as regras desempenham uma função vital para a consecução do objetivo final de um processo.

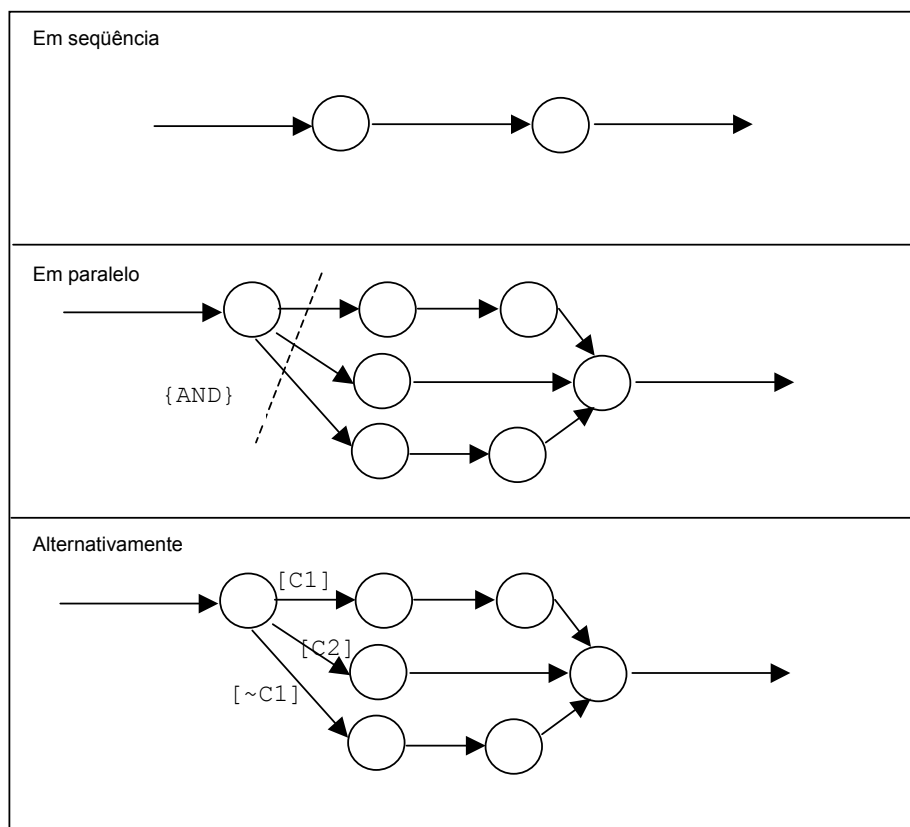


Figura 2.1 - Tipos de rotas em um fluxo de trabalho

Regras dizem respeito a restrições e diretrizes impostas por um *negócio* e/ou pela cultura de uma organização. Regras definem quais informações irão transitar pelo fluxo e sob quais condições, ou seja, são atributos que definem de que forma os dados que trafegam no fluxo de trabalho devem ser processados, roteados e controlados pelo sistema de *workflow* [6]. Regras devem ser informadas ao SGW durante a modelagem do processo.

## 2.4 - Funcionalidades de um Sistema de Gerência de Workflow

Um SGW corresponde a um conjunto de ferramentas que auxiliam o projeto e implementam esquemas de definição de fluxos de trabalho, sua instanciação e execução controlada e a coordenação e integração das diferentes ferramentas disponíveis dentro de um mesmo fluxo de trabalho.

No nível conceitual, os sistemas SGWs devem suportar aplicações em três áreas funcionais:

- Funções de tempo de construção (*build-time functions*): preocupam-se com a definição e modelagem do processo de *workflow* e suas atividades constituintes;
- Funções de controle em tempo de execução (*run-time control functions*): preocupam-se com gerenciamento de processos de *workflow* em um dado



ambiente operacional e colocação das várias atividades nas seqüências apropriadas para serem manuseadas como partes de cada processo;

- Funções de interação em tempo de execução (*run-time interaction functions*): tratam da gerência da interação dos SGWs com usuários humanos e outros sistemas de TI.

A figura 2.2 abaixo ilustra as relações entre as principais funções (e respectivas áreas funcionais) dos SGWs.

Em linhas gerais, após ter-se conduzido a etapa de análise do processo de negócio, passa-se à definição do processo de trabalho, que significa traduzir do mundo real para uma formalização computacional processável através do uso de uma ou mais técnicas de análise e modelagem [3]. O resultado da definição é um modelo ou representação do processo a ser executado. Uma vez definido, um processo pode ser executado através da interpretação de sua definição pelo sistema de *workflow*. Esta interpretação compreende o roteamento das atividades definidas aos atores designados para sua execução. Cada ator ou participante do processo, por sua vez, necessita interagir com o sistema, não só para realizar as atividades a ele designadas, como também para tomar conhecimento de sua necessidade de participação no processo.

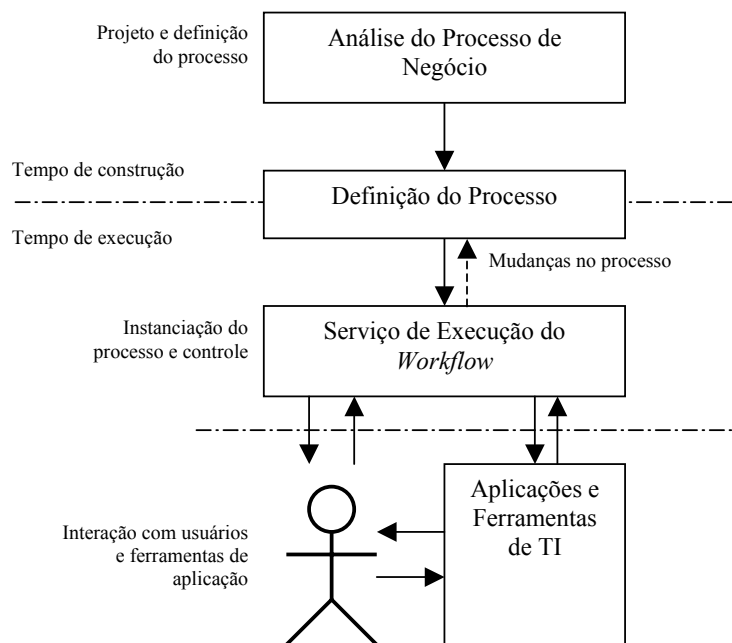


Figura 2.2 - Categorias das Funcionalidades Típicas de Sistemas de Gerência de *Workflows*

Uma apresentação um pouco mais detalhada dessas funcionalidades, presentes na maioria dos sistemas de *workflow* disponíveis no mercado [3][5], é apresentada a seguir.

## 2.4.1 - Modelagem de fluxos de trabalho

A descrição (ou modelagem) dos processos deve conter todos os dados necessários sobre as atividades a serem executadas e/ou coordenadas pelo sistema de *workflow*. Além dos dados sobre as atividades que compõem os processos (usuários encarregados, artefatos utilizados e produzidos nas atividades) deve-se relacionar as operações que devem ser executadas por cada usuário e as regras, ou condições, que devem ser respeitadas durante a execução das atividades. Em outras palavras, a definição de um processo a ser automatizado em um sistema de *workflow* deve conter informações completas sobre quem tem quais responsabilidades, através de quais operações essas responsabilidades devem/podem ser exercidas, as seqüências em que as operações devem ser executadas, em que momento (e respectivas tolerâncias de tempo) e quais os caminhos que levam e trazem os pacotes de informação necessários para a execução de cada atividade.

## 2.4.2 - Execução de fluxos de trabalho

Fluxos de trabalho são executados através de uma ou mais instâncias de execução das atividades que os compõem. Essas atividades são definidas durante a modelagem do processo no SGW e podem ser representadas graficamente através de diagramas de atividades da UML (vide capítulo 3).

O SGW deve acompanhar e coordenar a execução de cada uma das instâncias de execução das atividades que estão ativas, seguindo o fluxo definido no modelo do processo e encaminhando cada atividade para o(s) executor(es) correspondente(s), compondo sua(s) *lista(s) de trabalho*. As atividades são executadas nos ambientes de trabalho de cada ator/agente do processo através de aplicações ou ferramentas específicas para a atividade em questão.

## 2.4.3 - Controle de Interações

Para a execução das atividades, os usuários interagem com o SGW selecionando e executando atividades que constam de suas respectivas listas de trabalho. A realização de uma determinada atividade envolve a manipulação dos documentos estipulados no fluxo de trabalho para análise de informações, tomada de decisões ou preenchimento de dados. A finalização de uma atividade por um executor repõe o controle de volta ao fluxo, permitindo que o SGW prossiga na busca da conclusão do processo. Isso é feito através do "disparo" de novas atividades, considerando os resultados obtidos até então.

## 2.4.4 - Acompanhamento

Recursos para o acompanhamento da execução de atividades devem ser idealmente providos pelo SGW. Um mecanismo interessante para a visualização do *status* de execução é o próprio *mapa do processo*, onde são apresentadas as atividades já realizadas, as atividades em execução e as atividades a serem ainda executadas, assim como de quem foram/são as respectivas responsabilidades. Recursos, como estatísticas

de quantidades de insumos requeridos na execução de atividades específicas, podem também estar disponíveis através da manutenção de uma base de dados que reflita a eficiência e a eficácia dos processos atualmente desempenhados pela organização.

#### **2.4.5 - Administração**

As atividades de administração dos processos devem ser executadas por usuários com atribuições específicas para tal. Funções para suspensão, cancelamento de instâncias de execução e alteração de prioridades são exemplos dessas atividades.

#### **2.4.6 - Outras funcionalidades**

As funcionalidades apresentadas nos itens anteriores são as chamadas funcionalidades básicas [3] e devem ser idealmente providas por qualquer SGW. Sistemas comerciais, no entanto, implementam outras funcionalidades que julgamos igualmente essenciais, das quais citamos as abaixo.

- Definição de pesos para grupos: Possibilidade de definição de pesos distintos para cada integrante de um grupo, de forma que uma atividade possa ser designada a um membro do grupo baseada nos pesos definidos;
- Agente de correio: Possibilidade de execução de ações baseadas no recebimento de mensagens;
- Auditoria automática: Possibilidade de manutenção, em um sistema de gerência de documentos, de versões dos documentos/formulários relativos a cada passo do processo;
- *Log* das atividades: Possibilidade de registro de cada atividade executada durante os processos, assim como disponibilidade de funções para execução de consultas e elaboração de estatísticas diversas baseadas nos registros feitos;
- Sub-processos: Possibilidade de definição de sub-processos de um processo;
- Visão de carga de trabalho: Possibilidade de determinação da carga de trabalho de cada usuário, permitindo que o administrador do processo determine quantas e de quais tipos são as atividades pendentes para um usuário;
- Grupos dinâmicos: Possibilidade de definição de grupos responsáveis por uma atividade no momento de sua execução, ao invés de no momento da modelagem do processo, apenas.

Apresentações mais detalhadas dessas e outras funcionalidades podem ser encontradas em [3], [7] e [8].

## 2.5 - Categorias de (Sistemas de) Workflows

Temos dado destaque até então a *workflows* que visam a automação de processos de negócio. *Workflows* desse tipo, em geral, envolvem políticas e práticas bastante bem definidas pelas corporações que os aplicam. Esses *workflows* são chamados de *workflows de transação* ou de *produção* e suas execuções permitem pouca ou nenhuma variação entre instâncias de execução entre as práticas das empresas. São, portanto, caracterizados pelo alto grau de estruturação e dão ênfase aos processos envolvidos na execução das atividades.

Cabe mencionar também outras duas categorias de *workflows*, que categorizam, em consequência, os correspondentes SGWs [5][15]:

- *administrativo ou de roteamento de documentos*: baseado, primariamente, nas capacidades dos sistemas de *e-mail* e suas extensões. Esse tipo de *workflow* trata tarefas administrativas de rotina e, em geral, estabelece apenas fluxos de informação e faz o roteamento de documentos. Exemplos de *workflows* administrativos são aprovação de despesas de viagem e aprovação de empréstimos;
- *ad-hoc*: Existem muitas tarefas e atividades nas corporações que são mais orientadas a projeto, envolvendo objetivos e produtos cujos passos e dinâmica entre usuários são muito difíceis de serem definidos em detalhes e com algum grau de previsibilidade. Usam processos e procedimentos onde se enquadram as ferramentas de *groupware* e onde não existe uma estrutura pré-definida para o processo, ou esta estrutura pode ser modificada em tempo de execução. Embora, também nesse caso, existam *deadlines* e objetivos (produtos) bem definidos, as responsabilidades individuais podem ser alteradas. *Workflows* ad-hoc tendem a envolver executores mais criativos e de mais alto grau de conhecimento. As atividades, vez por outra, prescindem do uso dos SGWs. Um elemento tipicamente importante na execução desse tipo de *workflow* é o mecanismo de comunicação entre executores. Exemplos de aplicações: criação de documentos, desenvolvimento de software e campanha de marketing para lançamento de produto.

Já Georgakopoulos em [20] caracteriza *workflows* conforme abaixo (transcrição de [9]):

- *Workflows orientados para pessoas*, que envolve pessoas na realização das tarefas. O suporte do sistema é fornecido para facilitar a colaboração e a coordenação entre as pessoas, mas são elas próprias que têm, em última análise, a responsabilidade pela coerência das ações.
- *Workflows orientados para sistemas* são aqueles que consistem em tarefas com uso intenso de computação e tarefas especializadas que podem ser executadas por um computador. Nesse caso, o suporte do sistema é substancial e envolve controle da concorrência e recuperação, execução automática de tarefas, notificação etc.

- *Workflows transacionais* variam entre *workflows* orientados para pessoas e *workflows* orientados para sistemas, e tomam emprestadas características de ambos. Eles envolvem "execução coordenada de várias tarefas que (a) podem envolver pessoas, (b) exigem acesso a sistemas HAD (heterogêneos, autônomos e/ou distribuídos) e (c) admitem o uso seletivo de propriedades transacionais (isto é, propriedades ACID) para tarefas individuais ou *workflows* inteiros".

SGWs podem, também, incorporar apenas parte ou mesclar as funcionalidades mencionadas anteriormente, com vistas às suas aplicações em áreas bastante específicas. *Workflows* científicos, por exemplo, lidam, em geral, com processamento massivo ou que consomem e/ou produzem volumes grandes de dados. Podem executar processos bem definidos *a priori*, mas podem necessitar de mudanças em tempo de execução, em função de resultados intermediários. Nesse caso, também, o roteamento desses resultados deve ser definido com critério, podendo ser alterado em tempo de execução, em função do volume dos dados. Da mesma forma, em Ensino à Distância (EAD), os tamanhos dos artefatos manipulados podem ser tais que condicionem, em tempo de execução, um grau de liberdade maior para uma seqüência de processos (i.e., execução colaborativa de uma lição ou curso) que tipicamente é definido *à priori*.

## 2.6 – SGWs Web-Enabled e Web-Based

SGWs que fornecem interfaces *Web* são chamados *Web-enabled*, enquanto que *Web-based* são chamados os SGWs em cujas tecnologias está presente a tecnologia *Web*, através de interfaces, protocolos de comunicação e recursos típicos, tanto a nível de aplicações clientes quanto de servidoras.

## 3 - Modelagem de Workflows

### 3.1 - Metodologias de Modelagem de Workflow

Existem, basicamente, duas classes de metodologias de modelagem de *workflow*: as baseadas em comunicação e as baseadas em atividades [20]. Nos dois itens a seguir faremos uma apresentação breve dessas duas metodologias. No item 3.2 apresentaremos os diagramas de atividades da UML como formas de modelar *workflows* através de metodologias baseadas em atividades.

#### 3.1.1 - Metodologias de Modelagem Baseadas em Comunicação

As metodologias baseadas em comunicação consideram que o objetivo da (re) engenharia de um processo de negócio é o de melhorar a satisfação do cliente. Cada ação no *workflow* é reduzida a um ciclo envolvendo um cliente e um executor e composto por quatro fases, que são: preparação, negociação, execução e aceitação. Na preparação, um cliente solicita a execução e um executor se oferece para executar uma ação. Na negociação, cliente e executor concordam com a ação a ser executada e

definem os parâmetros de satisfação. A ação é executada na fase de execução e, na aceitação, o nível de satisfação do cliente é definido e reportado. Nessas metodologias são definidas relações clientes-provedores, onde os provedores assumem as responsabilidades pelas execuções de ações e cada um deles, possivelmente, repassa a um ou mais outros provedores as responsabilidades de execução de parte ou do total de suas próprias responsabilidades, gerando, com isso, outros sub-ciclos. Nesses casos, o provedor em uma relação cliente-provedor passa a ser o cliente em outra(s).

Esse tipo de modelagem estabelece árvores de delegações de responsabilidades e de execução de tarefas, onde a composição dos resultados (das atividades representadas pelas folhas das árvores) provê o objetivo do *workflow*.

As especificações de *workflows* que usam essa metodologia não indicam quais atividades podem ser executadas em paralelo ou se existem ações condicionais ou alternativas.

Os diagramas de interação da UML (seqüências e colaboração) poderiam ser utilizados nessas metodologias.

### **3.1.2 - Metodologias de Modelagem Baseadas em Atividades**

Metodologias baseadas em atividades objetivam a modelagem do trabalho, e não dos compromissos assumidos entre humanos, conforme nas metodologias baseadas em comunicação, tratadas anteriormente.

As metodologias baseadas em atividades caracterizam-se pela definição da coordenação entre atividades, eventualmente considerando pré e pós-condições para suas execuções. As atividades, entendidas como processos de transformação quaisquer, podem ser modeladas através de composições de sub-atividades a serem executadas ordenadamente (parcialmente ou em seqüência), em paralelo ou de forma recursiva.

Existem vários modelos de especificação de *workflows* baseados em atividades, dentre eles o de Casati/Ceri, o de gatilhos, o de redes de Petri e o modelo proposto por Barthelmess e Wainer em [21]. O autor propõe o uso da UML, através de seus diagramas - especialmente o de atividades, para o caso de *workflows* baseados em atividades - e linguagem de especificação de restrições de objetos (OCL - *Object Constraint Language*), para a modelagem de *workflows*.

Os diagramas de atividades (DA) da UML podem ser utilizados como uma WGDL (*Workflow Graphic Definition Language*), pois provêm elementos de modelagem de processos suficientes para emprego nas metodologias de modelagem baseadas em atividades. Diagramas de atividades permitem definir-se condições de execução de atividades, iterações, paralelismo, fluxos de artefatos e atribuição de responsabilidade a executores.

### 3.2 - Modelagem de Workflows Usando os Diagramas de Atividades da UML

A figura 2.1 ilustra alguns dos possíveis relacionamentos entre atividades de um processo. Abaixo transcrevemos de [18] os possíveis relacionamentos entre atividades, necessários para a modelagem de *workflows* através de metodologias baseadas em atividades.

- i. Roteamento seqüencial (*Sequential Routing*), onde atividades são executadas em seqüência, correspondendo a um único fio de execução (*thread*);
- ii. Roteamento paralelo (*Parallel Routing*), onde duas ou mais instâncias de atividades estão sendo executadas em um mesmo tempo, correspondendo a dois ou mais fios de execução no SGW;
- iii. Iteração, que corresponde à execução repetitiva de uma ou mais atividades até que uma dada condição seja alcançada;
- iv. A divisão-E, necessária para o estabelecimento do roteamento paralelo;
- v. A junção-E, que define ponto de sincronismo após a execução de atividades em paralelo;
- vi. A divisão-OU, necessária para introdução de condicionalidades na execução de rotas (ramos) alternativas, e
- vii. A junção-OU, que marca a re-convergência de rotas alternativas de *workflows*.

Segundo Barthelmess e Wainer em [21], além desses relacionamentos, também precisamos dispor de elementos para a associação de papéis a executores.

A Linguagem Unificada de Modelagem (*Unified Modeling Language* - UML), através de Diagramas de Atividades, provê representação gráfica capaz de modelar processos usando todos os relacionamentos apresentados acima. O uso dos diagramas de atividades da UML apresenta vantagens, dentre elas [19][23]:

- É padrão e incorpora as “melhores práticas” utilizadas nas empresas de produção de software;
- Provê uma linguagem que permite o entendimento e utilização por humanos e por máquinas. Esse aspecto é importante, pois facilita a implementação dos modelos;
- Contempla as necessidades de modelagem de negócios e sistemas de pequenos e simples a grandes e complexos.

A figura 3.1 ilustra o emprego de diagramas de atividades na modelagem de *workflows*. A figura 3.1(a) são ilustradas seqüências de atividades, além de iteração, representada pelo "\*" no canto superior direito da caixa da Atividade 1. A figura 3.1(b)

são ilustrados o ponto de *fork*, para a execução concorrente das Atividades 1 e 2, e o ponto de sincronismo, que garante que a Atividade 3 só será executada após o término das Atividades 1 e 2. A figura 3.1 (c) ilustra desvios condicionais e suas respectivas *expressões de guarda* (condições).

A figura 3.2 ilustra a modelagem de um *workflow* para processamento de um pedido de compra hipotético. As *swimlanes* ("raias de natação") enriquecem o modelo, permitindo uma fácil identificação dos papéis e das responsabilidades dos executores.



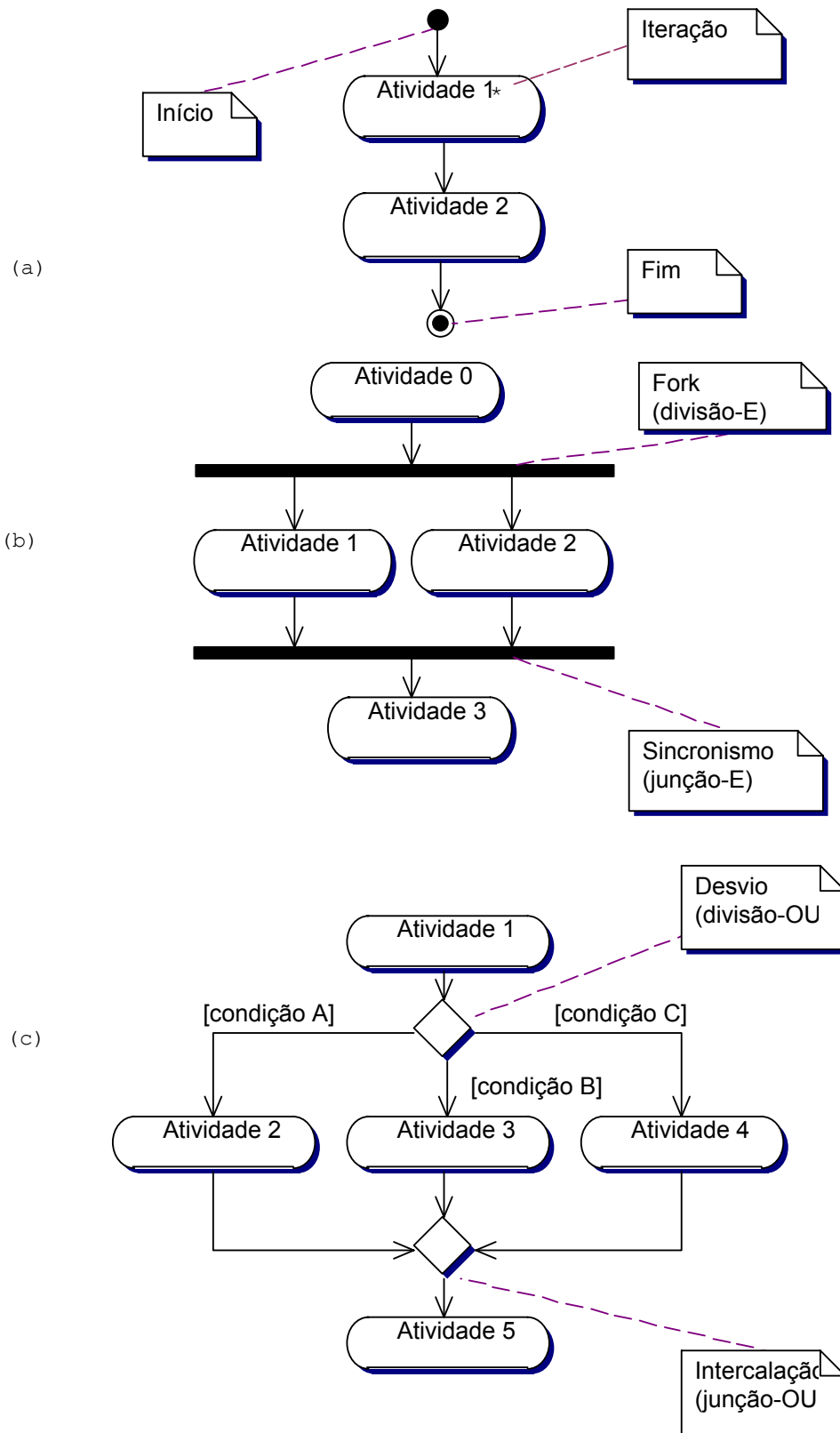


Figura 3.1 - Diagramas de Atividades da UML modelando (a) roteamento seqüencial, (b) divisão-E e (c) divisão-OU

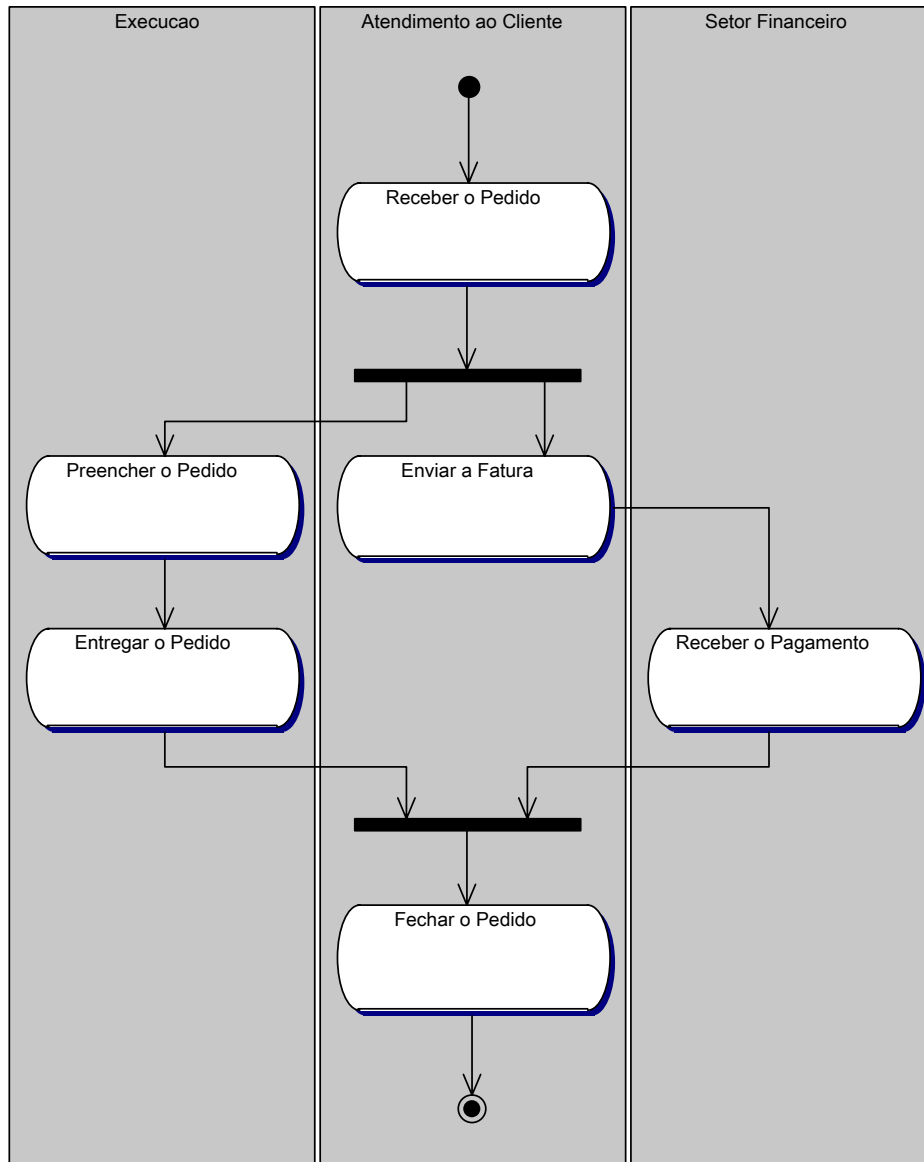


Figura 3.2 - Uso de Diagramas de Atividade da UML com raias de natação, para a visualização de responsabilidades e atores [23].

## 4 - Sistemas de Gerência de Workflows Distribuídos e Heterogêneos

### 4.1 - A Necessidade de Distribuição

Um processo de *workflow* pode envolver milhares de fragmentos de processos, variando de atividades de automação simples à coordenação entre processos através de diferentes níveis de abstração.

As diversas atividades que compõem os processos se enquadram em níveis individual, de grupo, de empresa ou inter-organizacionais. Essas atividades e suas associações aos executores são fortemente influenciadas pelos recursos que são oferecidos localmente (ferramentas e mecanismos diversos para manusear os diversos artefatos), pelo contexto de trabalho e pela cultura local. Atividades diferentes são, por conta dessas influências, das imposições políticas e administrativas (restrições/regras) e também de custos, executadas de forma descentralizada. Intui-se, também, que a distribuição de um processo pode levar a uma relação benefício/custo ótima, considerando a utilização dos recursos também distribuídos. O trabalho descentralizado sugere que os módulos componentes dos SGWs sejam processados de forma distribuída, cooperativa e complementar.

Colaborando ainda para esse cenário, a já mencionada crescente aceitação dos SGWs nas organizações vem resultando em suas aplicações em processos mais complexos e pesados, contando com grande número de executores, possivelmente situados em localizações geográficas distantes. A tendência crescente de integração de fluxos de trabalhos entre sítios distintos de uma mesma organização e entre organizações distintas (aplicações B2B, por exemplo) indica a necessidade de que os SGWs, até então restritos aos ambientes de uma mesma organização, possam ser agora integrados em parte ou no total, mantendo ou não suas autonomias. Com isso, além da distribuição, deve-se considerar a possível heterogeneidade dos SGDs.

A distribuição de atividades e informação entre os participantes é, como vimos no capítulo 2, uma das funcionalidades ditas básicas de um SGW. A função de distribuição pode operar em uma variedade de níveis - de grupos de usuários em um mesmo setor até a operação inter-organizacional - e pode contar com uma variedade de mecanismos de comunicação como correio eletrônico e/ou tecnologia de objetos distribuídos. A figura 4.1 abaixo ilustra uma possível instância de *workflow* onde é mostrado o aspecto de distribuição.

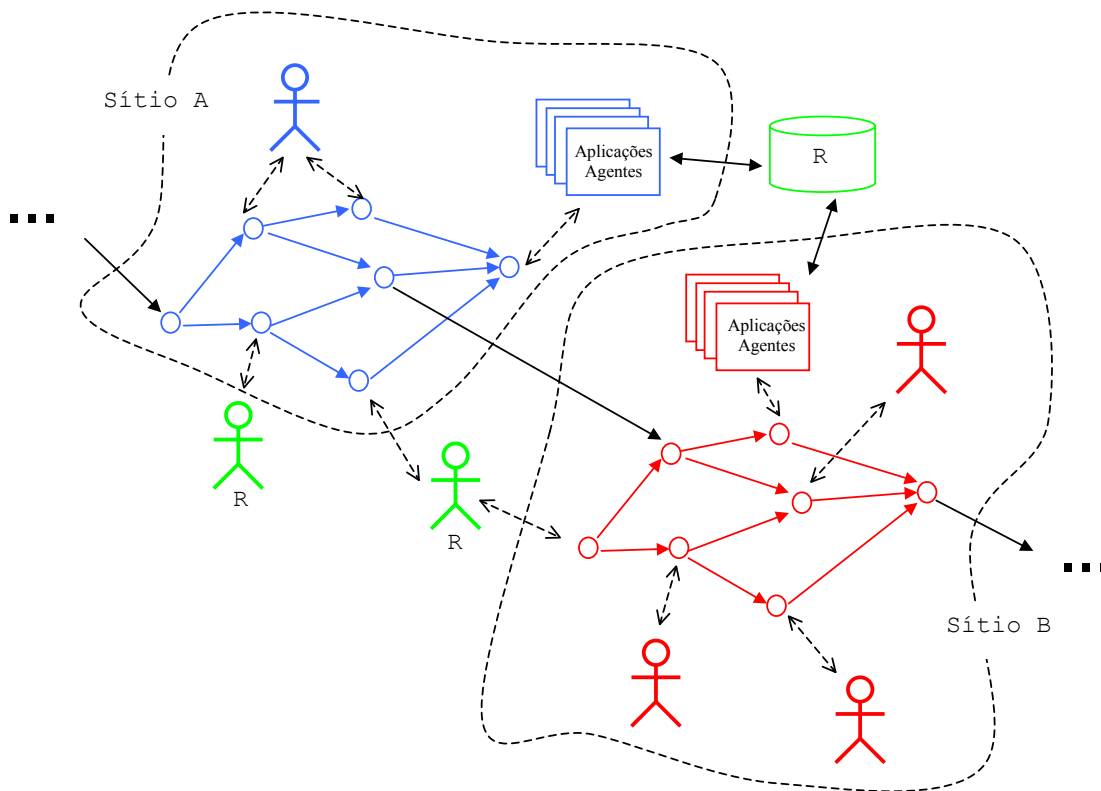


Figura 4.1 - Distribuição e compartilhamento de *workflows*

Na figura acima, os dois serviços de *workflow*, localizados em sítios diferentes (A e B) são autônomos e possuem usuários locais e remotos (estes marcados com “R”). As aplicações locais a cada sítio podem compartilhar bases de dados, que podem ser remotas. Através de um mecanismo padronizado de comunicação, uma atividade executada em um dos sítios pode gerar artefatos que serão usados em atividade(s) de outro(s) sítio(s). Os sítios representados podem pertencer a uma mesma organização ou a organizações distintas. Os SGWs podem ser, ainda, produtos comerciais distintos. Em outras palavras, o fluxo do trabalho pode envolver a transferência de atividades entre diferentes vendedores de produtos de *workflow* para habilitar partes diferentes do processo de negócio a ser desempenhado em plataformas ou sub-redes diferentes e usando produtos específicos para cada estágio dos processos.

## 4.2 - Considerações Sobre a Infraestrutura Necessária

As necessidades em termos de infra-estrutura classificam-se em dois tipos: armazenamento e recuperação de dados e comunicação.

É necessário, primeiramente, que os diversos artefatos e informações (estados) sobre as atividades sendo executadas em um *workflow* sejam mantidas em um ou mais repositórios. A maioria dos SGWs é baseada na arquitetura cliente-servidor na qual, mesmo sendo as atividades individuais executadas em nós distintos, artefatos e estados são guardados em um banco de dados centralizado em um dos servidores [26]. Obviamente a alternativa de distribuição desses dados pode ser considerada. Prós e contras de cada uma dessas alternativas são discutidos no item 4.3.

Além de infra-estrutura de armazenamento é fundamental dispor-se de uma infra-estrutura de comunicação rápida, confiável e que suporte o estabelecimento de mecanismos seguros (do ponto de vista de segurança da informação), que tornem transparentes as distâncias físicas entre os diversos participantes de um processo.

Infra-estruturas de comunicação são úteis para o atendimento de necessidades em dois níveis:

- Para que sejam feitas atualizações, idealmente em tempo real, da informação para controle de execução dos SGWs (estados de execuções, por exemplo). Dizem, portanto, respeito à infra-estrutura necessária para execução dos SGWs .
- Necessidades (óbvias) de comunicação entre executores, como trocas de mensagens e artefatos. Essas necessidades, muitas das vezes, não são atendidas diretamente pelos SGWs, pois demandam recursos que, em geral, já estão presentes em qualquer plataforma operacional com infra-estrutura de comunicação<sup>3</sup>. Nesse nível encontram-se, também, as necessidades de troca de artefatos supridas por outros mecanismos nativos das plataformas operacionais modernas, como NFS e compartilhamento em rede de unidades de disco.

### 4.3 - Processos Distribuídos

Muitos esforços em pesquisas têm sido desenvolvidos no sentido de investigar-se as técnicas de partição de sistemas de *workflow* em componentes e de distribuição desses componentes entre os executores. A execução distribuída de *workflows* necessita, entretanto, ser mais estudada [2].

Como vimos, *workflows* podem ser especificados, gráfica ou textualmente, durante a fase de modelagem. No final da modelagem essas especificações, sejam elas gráficas ou textuais, precisam ser convertidas em conjuntos de componentes computáveis semanticamente equivalentes que são, então, distribuídos para e pelos servidores de *workflows*, conforme também definido na modelagem ou de forma automatizada, objetivando um custo global de execução menor e/ou o atendimento de possíveis restrições.

A desvantagem dessa abordagem é sua rigidez, já que suas adaptações à dinâmica do ambiente, tal como a entrada de um novo servidor no ar, não são facilmente conseguidas.

Uma abordagem diferente, adotada no *Altavista Works*, provê uma flexibilidade maior em tempo de execução. Isso foi conseguido ao se converter o sistema original, centralizado em um SGW distribuído (SGWD) [2].

---

<sup>3</sup> Alguns SGWs, como o Lotus Notes, implementam, por sobre a infra-estrutura básica de comunicação provida pelos ambientes operacionais, mecanismos de comunicação mais apropriados para *workgroups*, como agentes de correio que provêm emissão automática de comprovantes de recebimento de mensagens, por exemplo.

Os requisitos gerais para a execução distribuída de um *workflow*, apresentados adiante, devem considerar como os dados de tempo de construção são armazenados e acessados, como deve ser controlada a execução distribuída e, também, como são distribuídas as atividades.

#### 4.3.1 - Dados de Tempo de Construção (Modelagem)

É uma pré-condição básica que, em SGWDs, os dados de tempo de construção (dados da modelagem ou, em inglês, *built-time data*) sejam acessíveis por todos os participantes ou executores. Durante a associação dinâmica de papéis a executores, por exemplo, o SGW seleciona a unidade de execução mais apropriada para um certo papel. Para ser capaz de selecionar um candidato não local, o WMS deve "conhecer" os dados de tempo de construção para um dado *workflow*. A questão passa ser, então, como obter a visão global dos dados de tempo de construção?

Uma possibilidade é prover-se acesso transparente aos dados centralizados, através dos serviços de acesso compartilhado a dados providos pela própria rede (compartilhamento de diretórios). Como a consulta aos dados de tempo de construção, durante a execução de um *workflow*, é feita intensamente, o tráfego gerado em rede seria, com isso, muito grande. Além disso, essa abordagem possui outras desvantagens bem conhecidas: pequena *escalabilidade* e baixa confiabilidade devida à concentração dos dados em um único sítio.

Na medida em que os dados de tempo de construção, em geral, não são volumosos e são pouco alterados durante a execução de um *workflow*, uma solução mais interessante seria a replicação desses dados por todos os participantes. Além do *Altavista Works*, outro SGWDs que adota essa solução é o *Exotica* [2]

#### 4.3.2 - Controle da Execução Distribuída

A execução distribuída de um *workflow* necessita de algum controle centralizado, que depende do tipo de distribuição usado (vide seção adiante). O mínimo de controle que podemos esperar é a sincronização no tempo dos participantes do *workflow*. Com isso, os relógios dos servidores de *workflow* só podem divergir dentro de uma dada tolerância e um cuidado especial com fusos horários e suas mudanças deve ser tomado.

#### 4.3.3 - Tipos de Execuções Distribuídas

As formas de se executar *workflows* distribuídos categorizam-se, basicamente, segundo dois tipos: (1) o servidor que iniciou o *workflow*, denominado de *home-server*, controla, de forma centralizada, a execução remota das atividades ou (2) os *workflows* *migram* de um servidor para o outro, levando com eles seus estados de execução.

No primeiro caso, o *home-server* permanece responsável por sua execução até que o processo chegue ao final. As diversas atividades podem ser executadas de forma distribuída e o resultado da execução de qualquer atividade deve ser sempre reportado ao *home-server*.

Como diferentes *workflows* podem ser criados e iniciados em diversos servidores, essa abordagem é considerada *parcialmente centralizada* e suas principais vantagens são as facilidades de implementação e de monitoramento.

No segundo caso (migração) a instância de execução de um *workflow* leva consigo seu estado a medida em que sua execução passa de um servidor a outro, não sendo necessário, então, um controle de execução centralizado. Em contrapartida, o espaço necessário para o armazenamento do estado de execução, que migra de servidor para servidor através da rede, pode assumir um tamanho considerável. Esse tráfego pode ser reduzido se distribuirmos, antecipadamente, cópias dos dados de tempo de construção e/ou das partes invariantes dos dados. A alternativa de migração tem a vantagem óbvia de que, tanto o código quanto os dados de controle estão localizados em um mesmo sítio. A grande desvantagem dessa alternativa é a dificuldade de monitoramento já que, para tal, é necessário que se trate toda história de um processo como parte do estado de execução do *workflow* ou pela escolha de um servidor específico na rede para recepção e armazenamento da mesma (recaindo em caso semelhante ao anterior), com os prós e contras já discutidos dessas alternativas.

Exceções provocadas por falhas dos servidores e/ou na infra-estrutura de comunicação podem ser tratadas, no primeiro caso, da seguinte forma: ao iniciar um *workflow*, o *home server* mantém uma cópia atualizada do estado do *workflow* e de seus mecanismos de controle para um outro servidor da rede (com escolha idealmente baseada em redundância de caminhos de rede) que assumirá o controle, caso o *home server* original falhe. Caso isso venha a ocorrer, o servidor escolhido assume o controle de execução do *workflow*, imediatamente escolhendo outro servidor para sua contingência. No segundo caso, o servidor que passa o controle a outro mantém o estado do *workflow* armazenado até que este servidor reporte o final de suas tarefas. Caso isso não ocorra, o primeiro pode tomar a decisão de escolher outro, fazendo nova tentativa.

As exceções devidas a falhas podem ser recursivas (falhas no tratamento de falhas), tornando a solução bastante complexa. Esse problema é idêntico ao tratamento de falhas em transações distribuídas em SGHDDs. Trataremos mais do assunto no capítulo 5.

#### **4.4 - A Questão da Heterogeneidade**

A tecnologia de *workflows* é usada, cada vez mais, em grandes organizações, na medida em que estas operam com fluxos de trabalho complexos, pesados e, quase invariavelmente, descentralizados. Como consequência, essas organizações necessitam automatizar o gerenciamento desses fluxos de trabalho.

Como já mencionamos, a necessidade de interações intra-organizacionais atravessou as fronteiras dessas organizações, alcançando níveis inter-organizacionais. Essas relações, chamadas de *Business-to-Business* ou B2B, agora contam com o potencial da WWW.

Esses dois aspectos nos levam a pensar em estender os SGWs existentes dotando-os de funcionalidades que atendam à necessidade de distribuição, gerando, como também vimos, SGWs distribuídos, ou SGWDs, onde a possibilidade de heterogeneidade dos sistemas participantes deve ser considerada.

Caso a arquitetura do SGW consista de uma camada de aplicação que opera sobre um SGBD (convencional ou não) que armazena os estados dos processos e os artefatos, o problema pode se reduzir ao problema de integração de dados gerenciados por sistemas heterogêneos, com diversas soluções propostas e outras tantas em estudo.

Caso os dados dos *workflows* sejam gerenciados por SGWs com arquiteturas monolíticas e proprietárias, podemos, ainda sim, integrá-los usando envoltórias (*wrappers*), desde que haja aberturas em suas interfaces.

A WfMC, através da publicação do Modelo de Referência de *Workflow* [1], identificou e agrupou em cinco módulos os componentes de um sistema de *workflow* genérico, além das interfaces (APIs e formatos de troca) entre os mesmos, descrevendo-os conceitualmente. O objetivo é que SGWs de fabricantes diferentes interoperem. Os cinco componentes identificados dão origem a 5 interfaces, conforme mostra a figura 4.2 abaixo. O componente central corresponde à máquina de execução do *workflow*, composto de um ou mais *motores*. Os demais componentes provêm ferramentas para definição dos processos, para administração e monitoramento da execução, aplicações clientes e aplicações externas invocáveis pela máquina de execução. Essas cinco interfaces agrupadas foram denominadas de WAPI (*Workflow APIs and Interchange Formats*).

Essa arquitetura elimina o problema de heterogeneidade de SGWs, na medida em que os motores dos diferentes SGWs são os únicos componentes que se comunicam diretamente, o que fazem de forma padronizada. Com isso, também, a heterogeneidade dos gerenciadores de dados não precisa ser considerada, já que os acessos aos dados de um SGW é feito, exclusivamente, através do(s) motor(es) do SGW.

Alonso et al em [29], entretanto, defente a abordagem contrária: a gerência de dados deve ser implementada como um conjunto de bancos de dados replicados fracamente sincronizados, a serem usados como repositório distribuído, comum a todos os sítios que participam da execução de um processo de negócio. A vantagem dessa abordagem, é que a carga da manipulação de dados não mais ficaria sobre o motor do *workflow*, aumentando, com isso, sua capacidade como uma ferramenta de coordenação e permitindo que o gerenciador de dados pudesse suportar funcionalidades adicionais que seriam difíceis de serem incorporadas no motor do SGW.



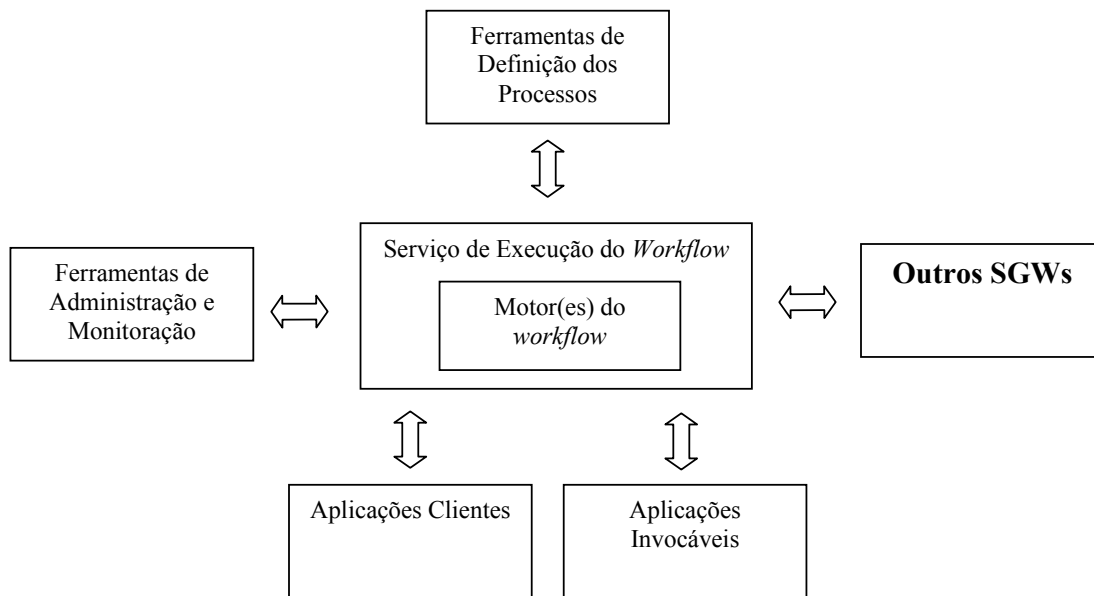


Figura 4.2 - Modelo de Referência de *Workflow* - Componentes e Interface [1]

## 5 - Exceções na Execução de Workflows

### 5.1 - Introdução

Tratamentos especiais, sob medida, são necessários para enfrentar situações que venham a ocorrer, durante a execução de um *workflow*, situações essas que não foram previstas durante a fase de modelagem [24]. *Exceções* são distanciamentos que ocorrem entre o efetivo estado de execução de um *workflow* e a situação normal, definida no projeto do *workflow* [22]. Podem resultar da falha de um executor (não completando a tarefa no tempo esperado ou completando de forma errada) ou da ocorrência de um evento não previsto no ambiente.

Embora exceções possam ser, também, descritas sob o ponto de vista organizacional, interessa-nos abordá-las enfocando suas repercussões na execução do SGW.

Esforços em pesquisas nessa área têm focado no uso de métodos transacionais para garantir a integridade de dados em *workflows* estruturados (ou transacionais), especialmente quando atividades concorrentes devem compartilhar dados. Outras pesquisas buscam modelos mais robustos e flexíveis de detecção e recuperação de falhas para *workflows* onde o domínio é dinâmico [25].

Nesse capítulo discutiremos a questão das exceções que podem ocorrer durante execução de *workflows*.

## 5.2 - Classificação de Exceções

Segundo Barthelmess [22], as exceções podem ser classificadas como a seguir:

- *Exceções de Infraestrutura*: nessa classe enquadram-se falhas originadas por problema(s) no SGW, por erro de *software*, por falha na rede de comunicação, pelo fato de estações, SGBDs ou impressoras estarem *offline*, etc. Exceções de infraestrutura devem ser tratadas pelo próprio SGW, na medida em que dizem respeito, exclusivamente, ao seu próprio funcionamento (e não ao *workflow*, propriamente dito).
- *Exceções de informação*: ocorrem quando os requisitos (que podem ter mudado) não correspondem ao que foi modelado.
- *Exceções de Dados*: exceções de dados ocorrem por erros ou ausência de dados, tipicamente identificados durante a execução das atividades onde os mesmos são usados. Essas exceções são, em geral, tratadas através de nova(s) execução(ões) da(s) atividade(s) que os produziram.
- *Exceções de Sinal*: são produzidas por informação assíncrona externa que, de alguma forma, altera o fluxo normal (conforme modelado) de trabalho. Seus tratamentos são diversos, dependendo o *workflow*. Pode ser necessário que sejam simplesmente interrompidas algumas atividades em execução e/ou que sejam desconsiderados os efeitos de atividades que, porventura, já tenham terminado e/ou que sejam executadas outras atividades não necessárias para o fluxo normal de trabalho, ou mesmo que sejam executados *workflows* inteiros afim de reverterem os efeitos causados pela exceção (*workflows* ou transações compensatórias), etc.

## 5.3 - Atomicidade de Falha em Workflows

Baseado na semântica de um *workflow*, o projetista do *workflow* pode especificar os requisitos de *atomicidade de falha* desse *workflow* [10]. Caso a noção tradicional de *atomicidade de falha* seja aplicada, a falha em qualquer tarefa resulta na falha de todo o *workflow*. Essa noção pode, entretanto, ser relaxada, na medida em que, em alguns casos, a execução do *workflow* pode sobreviver à falha de execução de uma ou outra tarefa. Esse conceito pode ser implementado através da definição de um conjunto de atividades consideradas vitais que, se falharem, causam a falha de todo o *workflow* [9]. É possível, também, definir-se tarefas de contingência, que são invocadas se as tarefas principais equivalentes falharem (exceções tratáveis). Cabe, portanto, ao projetista do *workflow* a definição do critério de falhas aceitáveis e das possíveis tarefas de contingência, cabendo ao SGW a permanente verificação se o critério de falhas definido é atendido e a instanciação, quando necessária, das tarefas de contingência.

Os possíveis estados de término de um *workflow* que atendem ao critério de falhas aceitável são chamados de *estados de término aceitáveis*. Qualquer outro estado de término é chamado de *estado de término não aceitável*, nos quais o critério de atomicidade de falha estabelecido foi violado.

Um estado de término aceitável possui ainda dois sub-estados: consolidado (*committed*) ou abortado. Um *estado de término aceitável consolidado* é aquele em que o *workflow* terminou atingindo seus objetivos originais. Caso contrário, o estado de término é chamado de *estado de término aceitável abortado*, cujos efeitos indesejáveis da execução parcial devem ser desfeitos, conforme o critério de atomicidade de falha estabelecido.

Diante de um estado de término não aceitável, o SGW deve procurar trazer o *workflow* de volta a um estado aceitável.

## 5.4 - Recuperação de Workflows Diante de Falhas

O objetivo da recuperação diante de falhas na gerência de *workflows* é garantir a atomicidade de falha definida em tempo de modelagem [10]. Os procedimentos de recuperação devem garantir que, se uma falha ocorrer durante a execução do *workflow*, o mesmo deverá ser levado a um estado de término aceitável (abortado ou consolidado) para, daí prosseguir, como se a falha não tivesse ocorrido. A recuperação do contexto do ambiente de execução necessita que sejam recuperadas as informações de estado do executor e de estado de execução de cada tarefa no momento anterior à falha, o que implica que essas informações sejam armazenadas usando-se um mecanismo de armazenamento estável. O mesmo cuidado deve ser tido no armazenamento das filas de mensagens efetivamente transmitidas e recebidas pelos diversos executores até o momento da falha, de forma a evitar-se que, na recuperação, qualquer mensagem seja enviada duas vezes ou que seja esquecida.

Embora os requisitos transacionais ACID usuais sejam muito fortes ou mesmo não implementáveis em aplicações de *workflow*, esses devem satisfazer a um conjunto limitado de propriedades transacionais que garantem que um processo não é deixado em um estado inconsistente [10].

Essa abordagem do problema de tratamento de falhas, discutida em [10] sob o título "*Workflows Transacionais*", baseia-se, como vimos, em pontos de checagem (*checkpoints*) e em reversão (*rollback*). A mesma permite o tratamento adequado de falhas para as formas mais estruturadas de *workflows*, enquanto que para outras, onde o domínio é dinâmico (onde há mudanças contínuas e imprevisíveis do negócio), ou onde a re-execução, descarte de resultados ou reversão de execução (*undo*) de etapas é impossível ou extremamente custoso, são necessários métodos mais flexíveis de recuperação de falhas. Nesses casos é imprescindível a habilidade de adaptar-se processos, em tempo de execução, em resposta a falhas ou mudanças no negócio, onde técnicas de Inteligência Artificial (IA) podem ser aplicadas [25].

## 5.5 - Execução Offline (com Desconexão)

Nas duas definições que apresentamos para *workflows* estão presentes os conceitos de coordenação e colaboração. Coordenação não só pressupõe uma certa ordem de execução das etapas (colaboração) como, no caso da maioria dos processos (pelo menos os de negócio), é necessário que essa ordem de execução das atividades gere, para cada etapa,

uma correspondência com pontos de início e fim em uma escala de tempo. Consideradas as possíveis tolerâncias de tempo, o SGW deve dispor de mecanismos de alerta/contingência quando uma colaboração não é realizada no tempo previsto, seja por falha no executor, seja por quebra (voluntária ou não) no *link* de comunicação entre este e os demais executores. Operação com desconexão, por uma razão ou por outra, é identificada como uma das *muitas* maneiras de utilização dos computadores no futuro [30], sendo *óbvio* que se deva tentar combiná-la com *workflows*.

Do ponto de vista do SGW, os dois casos anteriores são idênticos - a colaboração não foi realizada - e a solução do problema depende *do negócio*.

Em determinados casos a desconexão pode não ser crítica, por exemplo nas situações em que os SGWs possam estar "preocupados" apenas com o de término de uma atividade (horário e seus resultados), necessitando de *sincronização* apenas no final desta, e quando a atividade não compartilha estado com alguma outra sendo executada em outro sítio. Nesses casos, a execução da atividade pode se dar, desde que os artefatos necessários e o ambiente de execução estejam disponíveis no nó que esteja operando sem conexão [26]. Quando tratamos de execução distribuída de *workflows* (item 4.3.3), mencionamos que, no caso de migração, a instância de execução de um *workflow* leva consigo seu estado e os demais artefatos (dados), a medida em que sua execução passa de um servidor a outro. Essa alternativa (que não a ampla replicação pura e simples de controle e dados) é bastante interessante no caso de possibilidade ou necessidade de desconexão, já que cada nó de execução torna-se auto-suficiente em termos de dados e processos.

Nossas pesquisas estão voltadas para Ensino à Distância. Nesse contexto entendemos que a aplicação de *workflows* é bastante proveitosa, já que, além da coordenação das atividades que constituem os módulos, é consenso, modernamente, que a interação dos diversos executores - professores e alunos - deva se dar de forma colaborativa. Atividades como leitura e avaliação individual podem (às vezes devem) ser executadas estando os alunos desconectados do ambiente. Antes de executá-las os alunos se conectam ao ambiente, recebem todo o material necessário incluindo os módulos executores, se desconectam, executam as tarefas e, ao final, se conectam novamente, informando ao SGW os resultados de suas atividades.

## 6 – SGWs e SGBDs

Há quem considere que SGWs são aplicações que "rodam" sobre sistemas de gerência de bancos de dados (SGBDs) [13]. Há quem considere que essa abordagem simplifica o projeto de um SGW, mas que a mesma não contempla a natureza distribuída de aplicações de *workflow*, impondo sérias limitações em termos de escalabilidade e confiabilidade [29]. Há ainda [11] quem procure identificar semelhanças entre SGBDs e SGWs, buscando aplicar as já tão bem consolidadas soluções dadas para os problemas encontrados em SGBDs nos problemas encontrados em SGWs (vide capítulos 4 e 5).

Nesse capítulo investigaremos essa questão, procurando relacionar algumas das diversas abordagens encontradas na literatura considerando duas perspectivas: estrutura de armazenamento de dados e funcionalidade.

No capítulo 2 apresentamos as características dos sistemas de gerência de *workflows*, abordando, no item 2.4, as funcionalidades tipicamente disponíveis nestes sistemas. Para que possamos identificar as semelhanças entre SGBDs e SGWs devemos, também, relacionar as características e funcionalidades dos SGBDs, o que faremos, de forma resumida, a seguir.

Um SGBD é uma coleção de arquivos relacionados e um conjunto de programas que permitem que os usuários dos dados os acessem [9]. SGBDs permitem [9][10]:

- Que os usuários dos dados tenham visões abstratas dos mesmos;
- Integrar dados operacionais de um empreendimento e proporcionar acesso centralizado, e portanto controlado, a esses dados;
- Reduzir a redundância, inconsistência e dificuldade de acesso de dados;
- Prover isolamento entre dados e *aplicações* que os acessam;
- Manter a integridade dos dados com respeito a regras estabelecidas;
- Garantir a atomicidade de transações;
- Eliminar a possibilidade de anomalias advindas de acessos concorrentes;
- Prover acesso aos dados com segurança;
- Prover mecanismos de recuperação diante de falhas de hardware e de software.

Levando-se em consideração essas características e funcionalidades, podemos definir SGBDs como sendo conjuntos de componentes de software que provêm persistência, recuperação, compartilhamento, integridade e segurança de dados de forma transparente aos seus usuários.

Passando a considerar, também, que os dados estão geograficamente dispersos, necessitamos incorporar aspectos provenientes da área Redes de Computadores, na medida em que a tecnologia de sistemas de bancos de dados distribuídos (SGBDDs) é a união das tecnologias de redes e de bancos de dados.

"Sistemas de gerência de bancos de dados distribuídos estendem as facilidades usuais de gerência de dados de tal forma que o armazenamento de um banco de dados possa ser dividido ao longo dos nós de uma rede de comunicação de dados, sem que com isto os usuários percam uma visão integrada do banco" [12].

A idéia de utilizar-se SGBDDs é atrativa sob muitos aspectos, dentre os quais citamos:

- SGBDDs permitem que cada setor de uma organização geograficamente dispersa mantenha controle de seus próprios dados;
- SGBDDs oferecem compartilhamento a nível global no uso destes dados;
- SGBDDs podem diminuir os custos de comunicações, pois o projeto de distribuição considera, dentre outros aspectos, que os dados devem estar onde são mais usados;
- SGBDDs facilitam a escalabilidade do sistema como um todo, notadamente quando os comparamos aos sistemas centralizados baseados em equipamentos de grande porte;
- SGBDDs aumentam a confiabilidade através da replicação das partes críticas do banco em mais de um nó.

Re-visitando a relação das características e funcionalidades dos SGWs (e SGWDs), podemos observar que, além dos serviços tipicamente oferecidos pelos SGBDs e SGBDDs convencionais necessitamos, para a gerência automatizada de *workflows*, de:

- Um mecanismo que armazene os estados (igualmente dados) dos processos.
- Capacidade de armazenamento e recuperação de artefatos diversos, de não estruturados a completamente estruturados, e.g., documentos, código executável, imagens, vídeo, som, arquivos de dados, etc.
- Capacidade de gerenciar a execução de processos, escalonando-os, verificando pré-condições estabelecidas para execução de processos, disparando avisos aos executores, etc.

Em [13] encontramos:

"Os sistemas de Gerência de *Workflows* atuais usam um Sistema de Gerência de Bancos de Dados para armazenar descrições das atividades e implementar toda a funcionalidade do *workflow* em módulos que *rodam* sobre o SGBD."

Em outras palavras, enquanto bancos de dados garantem o armazenamento seguro e fácil acesso a grandes conjuntos de dados, sistemas de gerência de *workflows* objetivam o suporte básico para o fluxo de informação nos mesmos ambientes em que os bancos de dados são usados [28].

É importante ressaltar-se que não há um consenso a respeito dos SGWs rodarem sobre SGBDs de mercado. Muitos autores falam em utilizar-se *extensões* de SGBDs, enquanto outros como em [2], defendem a alternativa de se dispor de uma infraestrutura específica de gerência de dados de *workflows*.

Verificamos que sistemas de gerência de bancos de dados orientados a objetos (SGBDOOs) atendem a boa parte das necessidades de SGWs. Estes possuem, também, um caráter ativo, pois não é necessário que um usuário sempre acione um comando explicitamente para que o SGW execute ou interrompa uma operação; muitas ações do sistema devem ser executadas quando o módulo gerenciador *percebe* determinadas condições. Essa *percepção* pode ser implementada no próprio SGBD (nesse caso dito ativo) ou através de *software* em uma camada externa que verifica constantemente os estados dos processos.

SGWs podem, dessa forma, ser entendidos como uma camada de aplicação que opera sobre uma camada de software que trata do acesso aos dados (esta última sendo o software de um SGBD convencional). Podem, também, compor-se de uma camada de software menor que opera sobre um SGBD não convencional (OO, Objeto-Relacional, Relacional-Estendido, etc.). Podem, também, ser implementados através de uma estrutura monolítica composta por camadas conceituais distintas, organizadas exatamente como no modelo de arquitetura ANSI/SPARC de SGBDs.

Embora seja um fato bem conhecido que os requisitos para um sistema de *workflow* em termos de escalabilidade e confiabilidade excedem aos requisitos das

tecnologias de banco de dados e de processamento de transações [27], essas tecnologias estão bastante mais adiantadas que as tecnologias usadas nos motores dos SGWs.

Um aspecto importante quanto à funcionalidade e que ressalta uma grande diferença entre SGBDs e SGWs é que nestes, ao contrário do que ocorre naqueles, o sistema pode não dispor de todos os elementos para abortar uma transação [28]. Em SGWs, muitas vezes, é necessário que um agente humano participe do tratamento de uma exceção, na medida em que estas podem variar de problemas de *hardware/software* a mudanças no negócio.

## 7 - Conclusões e Trabalhos Futuros

Neste trabalho apresentamos os principais conceitos e características dos sistemas de *workflow* buscando, em uma primeira etapa, nos ater a ambientes centralizados. Mais adiante, em função na necessidade dos fluxos de trabalho atravessarem as fronteiras setoriais (departamentos, filiais, etc., em uma mesma empresa) e mesmo corporativas (B2B), verificamos a necessidade de estender os SGWs para que permitissem o trabalho cooperativo de equipes situadas em locais geograficamente distantes. Consideramos, para os casos mais gerais, a possibilidade de autonomia local de uso, feita de forma integrada com os demais sítios e de forma transparente sob o ponto de vista dos usuários. Essa autonomia com integração é facilitada, hoje em dia, também pelo advento da tecnologia de *Web Services*, que se encontra em voga.

Tais requisitos, aliados às soluções tipicamente adotadas em SGWDs comerciais, levaram alguns autores a procurarem estabelecer uma correlação desses com SGBDDs, objetivando soluções comuns para problemas identificados como comuns. Essas correlações, entretanto, são pouco úteis, na medida em que os problemas que podem ocorrer durante a execução de fluxos de trabalho pouco estruturados são inúmeros, dificilmente previsíveis, quase invariavelmente carecem de soluções de negócio e que, por essa razão, muitas vezes não são considerados na fase de modelagem.

Não ficou clara a inadequação de SGBDs comerciais e suas extensões para persistência dos artefatos tipicamente manipulados pelos SGWs, embora, como afirmamos anteriormente, haja na bibliografia consultada várias referências a essas limitações e inadequações por parte dos SGBDs comerciais. Essas limitações não foram justificadas de forma convincente ou sequer explicitadas. Cabe, portanto, que uma experimentação seja feita futuramente. Nesse sentido, iniciamos no TecBD um projeto de implementação do protótipo de um gerenciador de conteúdo de objetos acadêmicos (PSs, PDFs, PPTs, códigos fontes, etc.) cujo módulo inicial deverá estar concluído até o final desse ano. Mais adiante deveremos, também, implementar parte do ambiente que está sendo concebido para o projeto PGL (*Partnership in Global Learning*), que fará o gerenciamento distribuído de objetos de aprendizado e de fluxos de controle na modelagem e execução de cursos à distância. Ambos os projetos usarão, dentre outros recursos, o DB2 da IBM e suas extensões.

Esse trabalho foi elaborado com o propósito de permitir a familiarização do autor com os conceitos e técnicas usados em sistemas de *workflows*, assim como servir como uma abordagem inicial do assunto com vistas ao levantamento dos principais requisitos de persistência dos artefatos e ambiente de controle dos SGWs.

## Referências Bibliográficas

- [1] *Workflow* Management Coalition. The *Workflow* Reference Model, document number TC00-1003, 1995. <http://www.wfmc.org/>.
- [2] Böszörményi, L.; Groiss, H.; Eisner, R., Adding Distribution to a *Workflow* Management System.
- [3] Araujo, R. M.; Borges, M. R. S., Sistemas de *Workflow*, XX Jornada de Atualização em Informática, Congresso da SBC - 2001.
- [4] Allen, R., *Workflow: An Introduction*, white-paper baixado de <http://www.wfmc.org/>, em novembro/2001
- [5] Moro, M. M., "*Workflow*" em [www.inf.ufrgs.br/~mirella/workflow/work.html](http://www.inf.ufrgs.br/~mirella/workflow/work.html), acesso em dezembro/2001
- [6] CRUZ, T., "*Workflow: A Tecnologia que vai revolucionar processos*", 2 ed., 2000, Ed. Atlas, São Paulo.
- [7] ULTIMUS, 1998, 40 Essential Features of *Workflow* Software You Will Not Find in Lotus Notes, <http://www.ultimus-workflow.de/einwf.htm>, acesso em dezembro/2001
- [8] ULTIMUS, 1988, Even the Best Form Software Lacks Essential *Workflow* Features, <http://www.ultimus-workflow.de/einwf.htm>, acesso em dezembro/2001
- [9] Özsu, T., Valduriez, P., Princípio de Sistemas de Bancos de Dados Distribuídos, tradução da 2a. edição americana, 2001, Editora Campus.
- [10] Silberschatz, A., Korth, H., Sudarshan, S., Database System Concepts, 3rd. Edition, 1997, McGraw-Hill.
- [11] Michirefe, M. S. C., Em Busca de Um Sistema de Gerência de *Workflow* Baseado na Tecnologia de Sistemas de Gerência de Bancos de Dados, Dissertação de Mestrado, Departamento de Informática, PUC-Rio, dezembro de 1998.
- [12] Casanova, M. A., Moura, A. V., Princípios de Sistemas de Gerência de Bancos de Dados Distribuídos, edição eletrônica revisada, 1999.



- [13] Ailamaki, A., Ioannidis, Y. E., Livny, M., Scientific *Workflow* Management by Database Management, SSDBM, 1998.
- [14] Oliveira, T. C.; Mathias Filho; I., Lucena, C. J. P., A Framework Based Approach to *Workflow* Software Development. Monografias da Ciência da Computação – PUC/Rio – 2001
- [15] Khoshafian, S., Buckiewicz, M., Introduction to Groupware, *Workflow*, and Workgroup Computing, Wiley, 1995.
- [16] Ailamaki, A., Ioannidis, Y. E., Livny, M., Scientific *Workflow* Management by Database Management, 1998.
- [17] E-*workflow* - the *workflow* portal. <http://www.e-workflow.org/>, acesso em 11/7/2002.
- [18] *Workflow* Management Coalition - Terminology & Glossary, document number WFMC-TC-1011, fevereiro de 1999.
- [19] OMG Unified Modeling Language Specification, Version 1.4, setembro de 2001.
- [20] Georgakopoulos, D., Amit, S., An Overview of *Workflow* Management: From Process Modeling to *Workflow* Automation Infrastructure. Distributed and Parallel Databases, no. 3, 119-152, 1995.
- [21] Barthelmess, P., Wainer, J., *Workflow* Modeling, CYTED-RITOS International Workshop on Groupware, Lisboa, Portugal, pp 1-13, setembro de 1995.
- [22] Barthelmess, P., Wainer, J., *Workflow* Systems: a few definitions and few suggestions. Proceedings of the 1995 Conference on Supporting Group Work, Milpitas, CA, USA, agosto de 1995.
- [23] Fowler, M., Scott, K., UML Essencial, Bookman, 2a. edição, 2000.
- [24] Barthelmess, P., Sistemas de *Workflow*: Análise da Área e Proposta de Modelo, dissertação de mestrado apresentada ao Instituto de Computação da UNICAMP, 1996.
- [25] Myers, K. L., Berry, P. M., At the boundary of *workflow* and AI, AAAI-99 Workshop on Agent-Based Systems in The Business Context, 1999.
- [26] Alonso, G, et al, Exotica/FMDC: Handling Disconnected Clients in a *Workflow* Management System, 3rd International Conference on Cooperative Information Systems, 1995.
- [27] Alonso, G. Schek, H.. Database Technology in *Workflow* Environments, Informatik/Informatique, Zeitschrift der schweiz. Informatikorganisationen (Journal of the Swiss Computer Society), 1996.

- [28] Alonso, G., The Role of Database Technology in Workflow Management Systems. Panel Position Paper, Proceedings of the First IFCIS International Conference on Cooperative Information Systems (CoopIS'96), 1996
- [29] Alonso, G., Reinwald, B., Mohan, C., Distributed Data Management in Workflow Environments, Proceedings of the 7th International Workshop on Research Issues in Data Engineering (RIDE'97), 1997
- [30] Alonso, G., Agrawal, D., Abbadi, A., Mohan, C., Functionality and Limitations of Current Workflow Management Systems, 1997.