UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Alocação de Atividades em um Sistema de Gerência de *Workflow*

por

RODOLFO MIRANDA DE BARROS

Dissertação submetida como requisito parcial para
a obtenção do grau de Mestre em Ciência da Computação

Prof. Dr. José Valdeni de Lima
Orientador

Porto Alegre, dezembro de 1997

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Barros, Rodolfo Miranda de

Alocação de Atividades em um Sistema de Gerência de *Workflow* por Rodolfo Miranda de Barros. - Porto Alegre: CPGCC da UFRGS, 1997.
120f.:il.

Dissertação (Mestrado) - Universidade Federal do Rio Grande do Sul, Curso de Pós Graduação em Ciência da Computação, Porto Alegre, 1997. Lima, José Valdeni de, orientador.

1. *Workflow*. 2. WFMS. 3. Alocação de atividades. 4. PERT/CPM.
I. Lima, José Valdeni de. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitora: Prof. Dra. Wrana Panizzi
Pró-Reitor de Pós-Graduação: Prof. Dr. José Carlos Ferraz Hennemann
Diretor do Instituto de Informática: Prof. Dr. Roberto Tom Price
Coordenador do CPGCC: Prof. Dr. Flávio Rech Wagner
Bibliotecária-Chefe do Instituto de Informática: Zita Prates de Oliveira

Agradecimentos

Como a maioria dos trabalhos que possuem páginas dedicadas a agradecimentos, começo escrevendo que é extremamente difícil lembrar todos os nomes de pessoas que contribuíram na realização deste trabalho. Por isso, inicialmente, agradeço a todos aqueles que, direta ou indiretamente, influenciaram na realização deste trabalho.

À CAPES, pelo suporte financeiro durante o curso.

A todos os professores do Instituto de Informática, em especial ao Prof. José Palazzo Moreira de Oliveira, coordenador do convênio UEL/UFRGS.

À Universidade Norte do Paraná (UNOPAR), pela ajuda e compreensão neste último ano.

À Universidade Estadual de Londrina (UEL) e aos colegas do Departamento de Computação, que sempre me incentivaram. Em especial aos amigos: Fábio Sakuray, Fábio Cesar Martins, Marcos Antonio Cassiolato Batista, Mario Lemes Proença Jr. e Maria Angélica O. C. Bruneto.

Aos amigos Juergen e Vera Rocholl, pela amizade construída nestes dois últimos anos.

Ao Prof. José Valdeni de Lima, pela sua orientação, confiança e por sua amizade.

Aos meus pais, por todo o amor, carinho e, principalmente, pelo exemplo de vida, que pretendo copiar e seguir pelo resto de minha vida.

A minha irmã Junia, que, da sua maneira, sempre esteve e estará do meu lado, para o que der e vier.

Finalmente, encerro esta página de agradecimentos, agradecendo a Deus, que sempre esteve presente na minha vida, ajudando-me nas horas difíceis e mostrando-me o caminho certo a seguir.

Sumário

LISTA DE ABREVIATURAS **6**

LISTA DE FIGURAS **7**

LISTA DE TABELAS **9**

RESUMO **10**

ABSTRACT **11**

1 INTRODUÇÃO **12**
1.1 A PROPOSTA DO TRABALHO 14
1.2 APRESENTAÇÃO DO TRABALHO 16
2 *WORKFLOW* **17**
2.1 INTRODUÇÃO 17
2.2 DEFINIÇÃO E TERMINOLOGIAS 18
2.3 OUTRAS DEFINIÇÕES DE *WORKFLOW* 21
2.4 FUNCIONALIDADES DE UM SISTEMA DE *WORKFLOW* 23
2.5 ARQUITETURA DE UM SISTEMA DE *WORKFLOW* 25
2.6 TIPOS DE *WORKFLOW* 27
2.6.1 Classificação segundo McCREADY 27
2.6.1.1 *Workflow* de Produção (*Production Workflow*) 28
2.6.1.2 *Workflow* Ad Hoc (*Ad Hoc Workflow*) 28
2.6.1.3 *Workflow* Administrativo (*Administrative Workflow*) 29
2.6.2 *Workflow* orientado a pessoas e *workflow* orientado a sistemas 29
2.6.3 *Workflow* Transacional 30
2.7 CLASSIFICAÇÃO DAS ATIVIDADES 30
2.7.1 Baseada na Instância do *Workflow* X *Batch* 31
2.7.2 Estruturada X Não-Estruturada 31
2.7.3 Internas X Externas 31
2.7.4 Quanto à Finalidade (Função) 32
2.8 LIMITAÇÕES DOS SISTEMAS DE GERENCIAMENTO DO *WORKFLOW* 32
2.9 IMPLICAÇÕES 33
2.9.1 Implicações Positivas 33
2.9.2 Fraquezas e Obstáculos 34
2.10 MODELOS PARA ESPECIFICAÇÃO DE *WORKFLOW* 35
2.10.1 Introdução 35
2.10.2 Modelos Baseados em Atividades 37
2.10.2.1 Modelagem por Gatilhos (*Trigger Modelling*) 37
2.10.2.2 Modelo de Barthelmes/Wainer 39
2.10.2.3 Modelo de Casati 41
2.10.2.4 Redes de Petri 46
2.10.3 Modelagem baseada em Comunicação 49

2.10.3.1 O Modelo *Action Workflow* 49
2.11 WORKFLOW MANAGEMENT COALITION SPECIFICATION (WFMC) 50
3 ALOCAÇÃO DE ATIVIDADES 54
3.1 INTRODUÇÃO 54
3.2 PAPEL X MODELOS DE ESPECIFICAÇÃO DE *WORKFLOW* 56
3.3 OS ESTADOS DE UMA ATIVIDADE SEGUNDO [CAS95A][CAS96A] E O PROCESSO DE ALOCAÇÃO 56
3.4 OS TIPOS DE ALOCAÇÃO ENCONTRADOS NA BIBLIOGRAFIA 57
3.5 ALOCAÇÃO DE ATIVIDADES X TIPOS DE *WORKFLOW* 59
3.6 ALOCAÇÃO DE ATIVIDADES E ALGUNS PRODUTOS COMERCIAIS 60
3.7 OS PROBLEMAS EM TORNO DA ALOCAÇÃO DAS ATIVIDADES 62
3.7.1 Situações que causam atrasos e interferem no processo de alocação das atividades 63
3.7.2 Situações que ocasionam um reorganização na alocação das atividades 64
3.8 FATORES QUE INTERFEREM NA ALOCAÇÃO DE ATIVIDADES 66
3.8.1 Fatores ligados às pessoas 67
3.8.2 Fatores ligados à atividade 68
3.8.2.1 Fatores ligados à atividade/ subgrupo execução 69
3.8.2.2 Fatores ligados à atividade/ subgrupo alocação 71
3.9 COMO OS FATORES LIGADOS ÀS PESSOAS INTERFEREM NA ALOCAÇÃO 74
3.10 COMO OS FATORES LIGADOS À ATIVIDADE INTERFEREM NA ALOCAÇÃO 75
3.11 ONDE FICAM ARMAZENADAS AS INFORMAÇÕES NECESSÁRIAS AO PROCESSO DE ALOCAÇÃO DE ATIVIDADES 78
4 PROPOSTA PARA A ALOCAÇÃO DE ATIVIDADES 80
4.1 ALOCAÇÃO DE ATIVIDADES "*TOPDOWN*" E "COMPLETA" 80
4.2 OS ESTADOS DE UMA ATIVIDADE SEGUNDO A ALOCAÇÃO "*TOPDOWN*" E "COMPLETA 83
4.3 ALTERANDO O FLUXO DE ATIVIDADES 85
4.3.1 Um exemplo da alteração do fluxo 87
4.4 UTILIZANDO REDES PERT/CPM 89
4.4.1 Método de Caminho Crítico (CPM) 91
4.4.2 Primeiras e Últimas Datas das Atividades 92
4.4.3 *Workflow* X Redes CPM 93
4.4.4 Como distribuir a folga entre as atividades que não fazem parte do caminho crítico? 96
4.4.5 O que fazer com as atividades alocadas que não serão executadas? Como tratar as atividades que somente será conhecido o número de execução da mesma em tempo de execução? 97
4.5 TRATANDO ATIVIDADES QUE PODEM SER EXECUTADAS EM LOCAIS DIFERENTES 99
4.6 O MODELO DE ESTADOS E A AGENDA DO ATOR 101
4.7 SITUAÇÕES IMPOSTAS PELO GERENTE AS QUAIS DEVEM SER TRATADAS PELO WFMS 102
4.8 O MODELO PROPOSTO 103
4.8.1 Considerações Iniciais 103
4.8.2 O Modelo 105
4.9 FUNCIONALIDADES DO PROCESSO DE ALOCAÇÃO DE ATIVIDADES 106
5 O PROTÓTIPO 109
5.1 INTRODUÇÃO 109
5.2 O CENÁRIO DO PROTÓTIPO 109
5.3 OS GRUPOS DE FUNÇÕES 110
5.4 A ARQUITETURA DO PROTÓTIPO 112
6 CONCLUSÕES E TRABALHOS FUTUROS 114

BIBLIOGRAFIA 116

Lista de Abreviaturas

API:	*Application Program Interface*
CPM:	*Critical Path Method*
FIFO:	*First In First Out*
HAD:	Heterogêneo, Assíncrono e Distribuído
PDI:	Primeira Data de Início
PDT:	Primeira Data de Término
PERT:	*Project Evaluation and Review Technique*
SQL:	*Structured Query Language*
UDI:	Última Data de Início
UDT:	Última Data de Término
WfMC:	*Workflow Management Coalition*
WFMS:	*Workflow Management Systems*
WIDE:	*Workflow on Intelligent Distributed database Environment*
WPDL:	*Workflow Process Definition Language*
WSS:	*Workflow Support System*

Lista de Figuras

FIGURA 2.1	-	Fluxo de atividades de uma das etapas da montagem de um carro	19
FIGURA 2.2	-	Exemplo de possíveis papéis e seus respectivos participantes (atores)	20
FIGURA 2.3	-	Lista de trabalho dos Atores	21
FIGURA 2.4	-	Modelo Entidade-Relacionamento	22
FIGURA 2.5	-	Arquitetura genérica de um sistema de *Workflow* [JOO95a]	25
FIGURA 2.6	-	Exemplo de um *workflow* de Produção (requisição de seguro) [GEO95]	28
FIGURA 2.7	-	Exemplo de um *workflow* Ad hoc (revisão de trabalhos por avaliadores) [GEO95]	29
FIGURA 2.8	-	Exemplo de um *workflow* administrativo (revisão de trabalhos por avaliadores) [GEO95]	29
FIGURA 2.9	-	Fases do desenvolvimento de sistemas de *workflow*	36
FIGURA 2.10	-	*Workflow* Documentação Ortodôntica a nível de usuário leigo	38
FIGURA 2.11	-	*Workflow* Documentação Ortodôntica Avançado	39
FIGURA 2.12	-	Modelo de *Triggers* X Proposta apresentada em [BAR95]	41
FIGURA 2.13	-	Símbolos usados na Linguagem de Especificação	44
FIGURA 2.14	-	*Workflow* da matrícula de um aluno no estágio	45
FIGURA 2.15	-	Estrutura de uma tarefa	47
FIGURA 2.16	-	Rede de Petri para seleção de artigos	48
FIGURA 2.17	-	Estrutura do laço de *Workflow* do modelo *Action Workflow*	50
FIGURA 2.18	-	Esquema do modelo de interoperabilidade da WfMC	51
FIGURA 2.19	-	Meta-modelo proposto pela WfMC	52
FIGURA 3.1	-	Influência do Processo de Alocação	55
FIGURA 3.2	-	Diagrama de estados de uma atividade [CAS95a][CAS96a]	57
FIGURA 3.3	-	Exemplo da agenda de quatro atores e suas respectivas atividades	63
FIGURA 3.4	-	Fatores que interferem na alocação de atividades	66
FIGURA 3.5	-	Possíveis Combinações do Subgrupo Alocação	73
FIGURA 3.6	-	Possíveis estados de um ator	75
FIGURA 3.7	-	Atraso no início da execução das atividades 4,5 e 6 provocando por um atraso na execução da atividade 3	76
FIGURA 4.1	-	Estados de uma atividade na visão do ator	85
FIGURA 4.2	-	Cenário do processo de alocação	86
FIGURA 4.3	-	Fluxo normal de atividades	88
FIGURA 4.4	-	Vários fluxos de atividades	89
FIGURA 4.5	-	Exemplo de uma rede com cinco atividades	91
FIGURA 4.6	-	Situações contempladas pelas redes PERT/CPM	92
FIGURA 4.7	-	Exemplo para calcular as primeiras e últimas datas	95
FIGURA 4.8	-	Cancelando atividades que não serão executadas	99
FIGURA 4.9	-	Cancelando atividades a partir da atividade predecessora	99
FIGURA 4.10	-	Atividades sendo executadas em locais remotos	101
FIGURA 4.11	-	Exemplo da agenda de um ator	102
FIGURA 4.12	-	Exemplo de uma configuração para o exemplo da Figura 4.11	103

FIGURA 4.13	-	Diagrama Entidade-Relacionamento do modelo de alocação proposto	105
FIGURA 4.14	-	Modelo de alocação proposto pelo trabalho	106
FIGURA 5.1	-	Cenário de alocação do protótipo	110
FIGURA 5.2	-	Arquitetura do protótipo	114

Lista de Tabelas

TABELA 3.1	-	Comparação entre os modelos de especificação em relação ao conceito de papel	56
TABELA 3.2	-	Comparação entre produtos comerciais	62
TABELA 3.3	-	Informações referentes ao histórico do ator na execução das atividades	67
TABELA 3.4	-	Características e habilidades do ator	68
TABELA 3.5	-	Características e habilidades requeridas pela atividade	70
TABELA 3.6	-	Agenda dos Atores	75
TABELA 3.7	-	Início/Término e tempo de execução das atividades de um ator	77
TABELA 4.1	-	Comparação Instância X Tempo de Execução X Atraso para um fluxo de atividades	89
TABELA 4.2	-	Comparação Instância X Tempo de Execução X Atraso para seis fluxos de atividades	90
TABELA 4.3	-	Duração das Atividades	93
TABELA 4.4	-	Duração dos Caminhos	93
TABELA 4.5	-	Cálculo das datas da Figura 4.5	94
TABELA 4.6	-	Atividades e suas respectivas durações para o exemplo da Figura 4.7	96
TABELA 4.7	-	Vias e os respectivos tempos	96
TABELA 4.8	-	Resultado do cálculo das PDI, UDI, PDT, UDT e Folga para o exemplo da Figura 4.7	96
TABELA 4.9	-	Tempo de execução da atividade 2 de acordo com a origem da atividade	102

Resumo

A tecnologia de *workflow* encontra-se atualmente ainda em fase inicial, porém, apresentando um grande crescimento nos últimos anos. Negócios em todo mundo estão usando o suporte automatizado do *workflow* para tornar suas ilhas de automação administrativas em eficiente e efetivos instrumentos com conseqüências comerciais importantes.

Esta tecnologia facilita, através de softwares e metodologias, a modelagem do processo de negócio como uma especificação de *workflow*, a reengenharia de processos especificados e, finalmente, a automação do processo de negócio.

Porém, não existe uma padronização para esta tecnologia. Uma atenção especial está sendo dada por parte dos pesquisadores no intuito de padronizar conceitos que sejam aceitos pela comunidade em geral, ou seja, usuários, desenvolvedores e os próprios pesquisadores.

Este trabalho tem como finalidade apresentar e discutir esta tecnologia, mais especificamente sobre uma de suas funcionalidades, a alocação de atividades. A alocação de atividades é responsável pela distribuição das atividades às pessoas aptas por sua execução.

Os principais conceitos envolvidos na tecnologia são citados e comentados. O problema da alocação é apresentado, discutido e soluções são propostas. Finalmente, é implementado um protótipo para avaliar e validar as idéias apresentadas.

Palavras-chave: *Workflow*, WFMS, Alocação de Atividades, PERT/CPM.

TITLE: "ACTIVITIES ALLOCATION IN A WORKFLOW MANAGEMENT SYSTEM"

ABSTRACT

The workflow technology is in an initial phase, however, it has been growing in the last years. Business all over the world are using automated support of workflow to turn their islands of automated administrations into an efficient and effective instrument with important commercial consequences.

This technology makes easy the business process modeling as a workflow specification, the process specification reengineering and, last, the business process automation, by the use of software and methodologies.

However, a standard for this technology does not exist. A special attention has been giving by the academic community to obtain a standard, which will be accepted by the users, the developers and the academic comunity.

This work has the goal of presenting and discussing this techonology, more specifically about the problem of activities allocation in a workflow system. The activities allocation is responsible for the distribution of an activity to the right person.

The main concepts related to workflow are presented and discussed. The allocation problem is presented, discussed and solutions are suggested. A prototype is implemented to evaluate the presented ideas.

Keywords: *Workflow,* WFMS, Allocation Activities, PERT/CPM.

1 Introdução

O mundo está passando por uma fase acelerada de mudanças. Cada vez mais a tão falada globalização e suas conseqüências do ponto de vista de aumento de concorrência tem exigido uma maior capacidade de gerenciar todas as atividades de uma empresa. Segundo [HAM94], entre os fatores econômicos, encontram-se grandes transformações que vêm afetando a economia mundial nos últimos anos: o aumento exacerbado da concorrência - de local para global - e a mudança de comportamento de grande parte dos clientes - de reativo para ativo -, entre outros.

No entanto, estas mudanças são apenas parte da rápida evolução social, tecnológica e sobretudo mental que a humanidade vem experimentando. Do ponto de vista empresarial, estas mudanças têm trazido ameaça à sobrevivência das empresas em todo mundo pelos mais variados motivos:

- seu produto perdeu utilidade pelo lançamento de um outro produto melhor e mais barato;
- seus custos estão incontroláveis e conseqüentemente não têm preço para competir no mercado interno e externo;
- certos países importadores de seu produto passam a impor exigências normativas difíceis de serem alcançadas com seu atual processo;
- seu concorrente possui maneiras mais eficientes de se aproximarem dos clientes do que as suas;

Estes motivos caracterizam a era em que vivem as empresas. Empresas até então aparentemente inexpugnáveis podem, devido às rápidas mudanças, ter sua sobrevivência ameaçada.

Devido a essas rápidas mudanças, as empresas estão trabalhando em função de reduzir seus custos, e desenvolver rapidamente novos produtos, antes que os mesmos sejam lançados no mercado já com tecnologia obsoleta. Segundo [HAM94], é difícil encontrar uma empresa cuja gerência não declare, ao menos publicamente, que deseja uma organização suficientemente flexível para se ajustar rapidamente às condições mutantes do mercado, suficientemente enxuta para derrotar o preço de qualquer concorrente, suficientemente inovadora para manter-se tecnologicamente atualizada em seus produtos e serviços e suficientemente dedicada para fornecer o máximo de qualidade e de atendimento aos clientes.

A complexidade dos mecanismos do setor produtivo, aliado aos fatores anteriormente citados, fazem com que as empresas busquem cada vez mais metodologias e ferramentas que facilitem o seu trabalho e as mantenham competitivas, ágeis e rápidas nas operações de seus negócios, frente a este novo cenário.

A metodologia "*just in time*", empregada por várias empresas em busca de minimizar a complexidade do trabalho e aumentar a satisfação dos clientes, é um

exemplo da busca das empresas em manterem-se competitivas perante um mercado cada vez mais globalizado. A compra de uma automóvel por um cliente junto à *FIAT* no programa *FIAT on line* é um exemplo prático desta metodologia. Através de uma programação prévia de recursos (materiais e humanos), a empresa entrega na data combinada o automóvel desejado pelo cliente. O fluxo de trabalho pré-determinado, a alocação das atividades aos respectivos executores, bem como os insumos já estocados para a montagem do automóvel, fazem com que a empresa (*FIAT*) produza o automóvel em tempo hábil para entregá-lo ao cliente. Desta forma, a mesma satisfaz os desejos do cliente, e mantém-se competitiva.

Como já enfatizado, rapidez para encontrar soluções frente a problemas inesperados, flexibilidade em alterar planos e prioridades, rapidez em adaptar-se às novas realidades do mercado, rapidez e qualidade no atendimento aos clientes, são características que ditam as empresas que se sobressaem neste novo cenário. Porém, para alcançar tais características, faz-se necessário que a empresa repense a forma de conduzir os seus negócios. O fluxo de trabalho empregado para conduzir seus negócios é de suma importância para o sucesso de uma organização.

Para sobreviver neste novo cenário, as empresas também contam com um forte aliado, a tecnologia. Seu poder rompedor [HAM94], a sua capacidade de transportar as regras que limitam a condução do trabalho, tornam-na fundamental para as empresas em busca da vantagem competitiva. Um exemplo pode ser visto na evolução tecnológica dos sistemas computacionais, tanto em capacidade de processamento como na comunicação de dados, que se apresentam como uma excelente opção para dar suporte às empresas em busca da competitividade.

Dentro deste contexto, encontra-se em fase de grande desenvolvimento a tecnologia de *workflow*. Negócios em todo mundo estão usando o suporte automatizado do *workflow* para tornar suas ilhas de automação administrativas em eficiente e efetivos instrumentos com conseqüências comerciais importantes [JOO94a].

Esta tecnologia vem de encontro com os anseios das empresas, uma vez que a mesma facilita, através de *softwares* e metodologias, a modelagem do processo de negócio como uma especificação de *workflow*, a reengenharia dos processos especificados e, finalmente, a automação do processo de negócio. As ferramentas de gerência de *workflow* possibilitam que as empresas alterem seu ritmo de trabalho e, quando usados de maneira efetiva, aumentem a velocidade, flexibilidade e eficiência do processo de negócio.

Segundo [GEO95], uma vez definido o processo de negócio, este pode ser redefinido em busca de melhorá-lo ou adaptá-lo a uma nova realidade. Razões para esta redefinição incluem: aumentar a satisfação do cliente, melhorar a eficiência na administração dos negócios, aumentar a qualidade dos produtos, reduzir custos, permitir mudanças rápidas de estratégias frente a um mercado em franca evolução, etc., o que comprova a importância da tecnologia de *workflow* para as empresas em busca da competitividade.

Apesar desta tecnologia ser emergente, apresentando vários produtos comerciais, o mercado atual de workflow, até mesmo por estar evoluindo tão rapidamente, apresenta-se um tanto imaturo e carente de definições claras [JOO95a]. Por exemplo,

não há um consenso claro, entre os desenvolvedores de soluções de *workflow*, sequer sobre o que *é workflow* e sobre quais recursos um sistema de *workflow* deve oferecer. Outro ponto em discussão envolve a integração entre sistemas de *workflow* de fabricantes diferentes. Nesta área, busca-se encontrar formas que permitam aos diversos sistemas de *workflow* trabalhar em conjunto na gerência de processos. Este requisito é fundamental quando deseja-se que o sistema de *workflow* abranja não só departamentos isolados, mas sim integre toda a organização - e, possivelmente, várias delas - na execução de um processo. Assim, seria possível uma comunicação transparente entre estes softwares, resultando em um ambiente de trabalho realmente integrado. Como poderá ser visto no decorrer deste trabalho, já existe um órgão, o *Workflow Management Coalition* (WfMC), trabalhando ativamente, propondo definições e interfaces comuns para a integração entre sistemas de *workflow* de fabricantes diferentes [WMC96b].

1.1 A Proposta do Trabalho

De acordo com o *Workflow Management Coalition* [WFC96d], *workflow* pode ser compreendido como a automação total ou parcial de um fluxo de atividades, onde documentos e informações são passadas entre os executores destas referidas atividades, visando a execução das mesmas.

Este fluxo de atividades pode implicar em atividades seqüenciais, paralelas e opcionais.

Dependendo da organização na qual o *workflow* está sendo executado, para cada atividade pode existir um ou mais participantes aptos por sua execução. No momento da especificação do *workflow*, nem sempre é possível ou desejável se definir previamente o responsável pela execução. Por esse motivo, surge o conceito de papel. Papel pode ser definido como um conjunto de participantes que possuem um mesmo leque de características (habilidades) que os tornam aptos a executarem a atividade relacionada ao papel. Desta forma, ao definir-se um *workflow*, ao invés de associar um participante à atividade, associa-se um papel.

Como um *workflow* pode ser executado mais de uma vez e ao mesmo tempo, ou seja, podem existir várias instâncias de um *workflow* sendo executadas num mesmo intervalo de tempo e, para cada atividade tem-se um responsável (participante) por sua execução, conclui-se que existe a possibilidade de um participante ser responsável por mais de uma atividade, porém, nunca ao mesmo tempo.

Neste contexto, para que a execução das instâncias de um *workflow* tenham sucesso, torna-se importante a política adotada pelo processo de alocação para definir quem será o responsável pela execução da atividade. Esta política deve-se basear em vários fatores, entre eles destacam-se a agenda dos participantes, o tempo de execução da atividade, a data e horário de início e término da atividade, as habilidades desejadas, a prioridade da instância, o histórico da performance do ator, a carga de trabalho dos atores, eventuais intervalos que o ator ficará impossibilitado de trabalhar, entre outras. Neste trabalho, será priorizado a agenda dos participantes, o tempo de execução da atividade, a data e horário de início e término da atividades e a prioridade da instância.

Sendo assim, quanto melhor for o processo de alocação de atividades em um sistema de gerência de *workflow*, maior será a qualidade do resultado do *workflow* como um todo. Caso ocorra algum problema neste processo de alocação, pode-se ter um atraso na execução do *workflow* ou algum outro tipo de problema, ocasionado, conseqüentemente, problemas para a empresa.

Para uma melhor compreensão desses problemas, serão citadas duas situações:

- uma escolha aleatória pode alocar uma pessoa que já esteja sobrecarregada de atividades, podendo, desta forma, atrasar a execução da atividade, e, conseqüentemente, atrasar a execução do *workflow*, visto que o mesmo, provavelmente, não dará conta de executar o trabalho no intervalo de tempo esperado;

- uma dada atividade foi alocada para uma pessoa que não possuía todas as habilidades desejadas para sua execução. Pode-se ter um resultado não desejado, ou seja, na execução da atividade, pelo fato do executor não possuir as habilidades necessárias, o produto resultante da atividade não estará de acordo com o preestabelecido. Caso exista a necessidade do mesmo ser refeito para se obter o resultado esperado, um atraso pode vir a ocorrer, prejudicando o cumprimento de prazos que foram por ventura assumidos.

Como o processo de alocação interfere na execução do *workflow* e, conseqüentemente, nas atividades da empresa, este processo deve buscar não só a alocação da pessoa certa, como também oferecer funcionalidades que possibilitem a otimização dos recursos e dos tempos, para que a empresa possa atingir seus objetivos da melhor maneira possível.

Desta forma, o presente trabalho visa identificar quais os fatores relevantes que interferem na alocação, bem como a forma que os mesmos interferem e, principalmente, propor funcionalidades que um sistema de gerência de workflow deve oferecer através do seu processo de alocação de atividades, para facilitar e ajudar a empresa na alocação eficiente de seus recursos. Numa análise superficial, estas funcionalidades deveriam possibilitar ao responsável pela gerência dos *workflows:*

- a identificação com antecedência de possíveis gargalos devido à falta de recursos (pessoal, tempo, etc.);

- a determinação de prazos para o término de atividades e instâncias de fluxo de trabalho;

- a estipulação de novos prazos para as instâncias em prol dos objetivos e prioridades da empresa;

- a realocação de atividades como medida corretiva em caso de identificação de gargalos e/ou impossibilidade de cumprimento de prazos.

Um sistema de workflow capaz de executar estas funcionalidades será de grande valia para o avanço da organização em relação ao controle do seu processo de trabalho, bem como coerente com a globalização e o conseqüente aumento de competição.

1.2 Apresentação do Trabalho

O texto do trabalho será estruturado de forma a apresentar uma introdução à tecnologia de *workflow*, seguindo-se da problemática da alocação de atividades e das propostas sugeridas para a mesma.

De acordo com esta estrutura, o capítulo 2, irá apresentar detalhes da tecnologia de *workflow*. Conceitos, definições, especificação dos processos, enfim, este capítulo apresentará o estado da arte desta tecnologia.

No capítulo 3 serão apresentados os problemas envolvidos na alocação de atividades e os fatores que interferem neste processo.

As funcionalidades de alocação que um sistema deve oferecer propostas neste trabalho, bem como as características e informações que o mesmo deve conter e oferecer para que as mesmas sejam implementadas, serão apresentadas no capítulo 4. Este capítulo também apresenta uma proposta para o processo de alocação.

No capítulo 5, serão apresentadas a arquitetura e características de implementação de cada uma das funcionalidade previstas, baseadas no processo de alocação proposto. Para finalizar, as conclusões e os trabalhos futuros serão apresentados no capítulo 6.

2 *Workflow*

Este capítulo oferece uma introdução à tecnologia de *workflow*. Os conceitos fundamentais relacionados à esta tecnologia são apresentados e detalhados.

2.1 Introdução

O conceito de *workflow* está envolvido com a noção de processo advinda da manufatura e do escritório [GEO95]. Tal noção de processo está relacionada com a busca da eficiência das atividades concentrado-as em rotinas. As atividades do trabalho são separadas em tarefas bem definidas, papéis, regras e procedimentos.

Segundo [MED92], o processo dentro de uma organização pode ser categorizado de três formas:

a) processo material: reunir componentes físicos e produzir um produto físico;

b) processo de informação: relacionadas com tarefas automatizadas ou parcialmente automatizadas;

c) processo de negócio: estão centradas nas descrições das atividades da organização, implementadas como processo material e/ou processo de informação.

Uma vez capturado o processo de negócio, este pode ser melhorado e adaptado às novas realidades em busca de eficiência e produtividade.

Segundo [GEO95], *Workflow* é um conceito relacionado intimamente com a reengenharia. Um *workflow* deve descrever as tarefas de um processo de negócio em um nível necessário para se entender, melhorar e reprojetar o processo.

W*orkflow* também pode ser visto como um tipo de *groupware* relativamente simples. *Groupware* é uma coleção de ferramentas que facilitam a cooperação dentro de uma organização.

Uma parte significante do trabalho de uma pessoa ocorre mais em grupo do que em um contexto individual, o que torna o suporte ao trabalho em grupo um assunto importante. Coordenação, comunicação e cooperação são os pontos fundamentais da atividade em grupo em busca de uma sinergia de um sistema. A automação do *workflow* é parte desta disciplina, devido ao fato de que o propósito da automação do *workflow* é coordenar atividades em grupo [DUI94]. Para tanto, ele deve suportar processos distribuídos e interações assíncronas. Assíncrono significa que os membros do grupo podem participar do processo em diferentes momentos. Já o conceito de distribuído significa que um dado processo pode ser feito em diversos lugares.

De acordo com estas noções e conceitos, a tecnologia de *workflow* é uma poderosa ferramenta para as empresas em busca da competitividade. A possibilidade de se descrever as tarefas de um processo, com o intuito de estudá-lo e melhorá-lo, é de

suma importância para a otimização do mesmo e, conseqüentemente, para a empresa como um todo. A coordenação das atividades por parte do *workflow* faz com que o empresário tenha em suas mãos o controle da empresa, ou seja, informações sobre o que está sendo feito, quanto tempo falta para acabar um processo, enfim, informações úteis para que o mesmo possa tomar decisões rápidas e ágeis no intuito de competir neste mercado globalizado e dinâmico, já descrito no capítulo 1.

2.2 Definição e Terminologias

Workflow, segundo definição em [WMC96d], é "a automação total ou parcial de um processo de negócio, durante a qual documentos, informações e tarefas são passadas entre os participantes do processo".

Já o conceito de processo, segundo [WMC96d], é dado como "um conjunto coordenado de atividades (seqüenciais ou paralelas) que são interligadas com o objetivo de alcançar uma meta comum". Como exemplos de processos, tem-se:

- processamento de matrícula de alunos em uma universidade;
- processamento de pedido de compra de material de expediente;
- processamento de uma ordem de serviço em uma oficina mecânica;
- processamento de ordem de produção de um produto em uma empresa;
- processamento de requisições de seguros por uma seguradora;
- processamento de avaliações de empréstimos por instituições financeiras;
- processamento de faturas bancárias e de cartão de crédito pelos bancos.

Segundo [WMC96d], atividade é "uma descrição de um fragmento de trabalho que contribui para o cumprimento de um processo".

Como pode-se ver, a idéia básica de um *workflow* é a quebra do processo em partes menores, sendo estas denominadas atividades.

Um exemplo de um fluxo de atividades visando uma etapa de montagem de um carro pode ser visto na Figura 2.1. Os quadrados representam as atividades, as setas dirigidas que chegam ao quadrado representam as atividades antecessoras e as setas que saem do quadrado representam as atividades sucessoras. Isto significa que uma atividade só pode ser executada se suas atividades antecessoras foram todas realizadas.

O fluxo de atividades da Figura 2.1 é composto de cinco atividades. As atividades 3 e 4 são seqüenciais. A atividade 2 é paralela as atividades 3/4. A atividade 1 é antecessora das atividades 2 e 3, isso vale dizer que as atividades 2 e 3 só terão seu início após o término da atividade 1. Já a atividade 5, tem como antecessoras as atividades 2 e 4, o que implica em dizer que ela só terá seu início após o término das atividades 2 e 4.

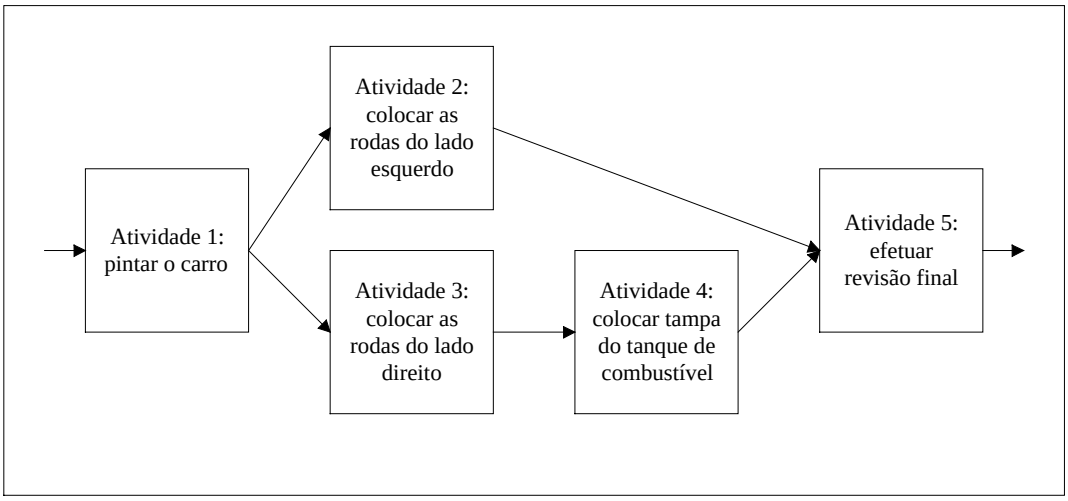


FIGURA 2.1 - Fluxo de atividades de uma das etapas da montagem de um carro

Para tornar possível a coordenação da execução das atividades, é necessário respeitar não apenas a ordem planejada de execução das mesmas como, também, o cumprimento das dependências e pré-condições entre elas. Ainda, de acordo com [WMC96d], sistemas de gerência de *workflow* (WFMS) são "sistemas que permitem definição, criação e gerência da execução de *workflows* através do uso de software, executado em um ou mais motores de *workflow* (*workflow* engine), o qual é capaz de interpretar a definição do processo, interagir com os participantes do *workflow* e, quando necessário, invocar ferramentas e aplicações de sistemas de informação".

Quando um *workflow* é disparado, ou seja, inicializado, é criada uma instância do processo pelo WFMS. Da mesma forma, são criadas instâncias das atividades que compõem o processo.

Um conceito importante relacionado ao *workflow* diz respeito ao participante. Um participante é responsável pela execução parcial ou total de uma determinada atividade. Um participante pode ser um humano (mecânico, secretária, gerente), como também pode ser um software (um software para testar algum componente de um carro, um software para envio e recebimento de fax, um software para atendimento automático de ligações), como também pode ser uma combinação desses. Ator [JOO94a], entidades processadoras [CAS95a] e agentes [CAS96a], são outros termos associados ao responsável pela execução da atividade.

Segundo [CAS96a], na execução de um *workflow* tem-se os seguintes tipos de agentes:

- executor do *Workflow*: é o agente que inicia (dispara) o *workflow*. Em geral, o *workflow* possui uma lista de possíveis executores, ou seja, pessoas aptas a iniciá-lo;

- responsável: é o agente que tem responsabilidade sobre o *workflow*. Pode ser tanto o executor, como qualquer outra pessoa pertencente à organização;

- executor da Atividade: é o responsável pela execução da atividade.

Como visto anteriormente, dependendo da organização na qual o *workflow* está sendo executado, para cada atividade pode existir um ou mais participantes aptos por sua execução. Nem sempre é possível ou desejável definir previamente o responsável pela execução de uma atividade. Sendo assim, utiliza-se o conceito de papel, que nada mais é do que um conjunto de participantes (atores) que possuem um mesmo leque de características (habilidades) que os tornam aptos a executarem a atividade relacionada ao papel. Desta forma, ao definir-se um *workflow*, ao invés de associar um participante (pessoa) à atividade, associa-se um papel. Outro ponto a ser destacado, é o fato de um mesmo participante (pessoa) poder executar mais de um papel (Figura 2.2).

Desta forma, torna-se importante para o WFMS, conhecer a estrutura hierárquica da organização na qual o *workflow* está sendo executado, para indicar quais os participantes que representam aquele papel [BAR95].

PAPEL			
Programador	**Analista Jr**	**Analista Pleno**	**Projetista**
Rodolfo	Mário	Marcos	Fábio
Angélica	Rodolfo	Vítor	Marcos
Bruno	Bruno	Rafael	Miguel
Junia	Lourdes	Augusto	Lourdes

Atores com mais de uma papel

FIGURA 2.2 - Exemplo de possíveis papéis e seus respectivos participantes (atores)

A Figura 2.2 traz exemplos de atores que desempenham mais de um papel. Como pode-se ver, o ator **Bruno,** pode assumir o papel de **Programador** como, também, o papel de **Analista Jr**.

Cada atividade a ser processada por um participante do *workflow* leva o nome de item de trabalho [WMC96d]. Os itens de trabalho a serem executados por um participante formam sua lista de trabalho [WMC96d] (Figura 2.3).

A Figura 2.3 mostra as definições de dois *workflows*, as instâncias dos mesmos, e a lista de trabalho dos atores responsáveis pela execução das instâncias das atividades. Como pode-se ver, o ator 1 é o responsável pela atividades C1 de uma instância de um *workflow*, e também pela atividade F2 de uma instância de outro *workflow*.

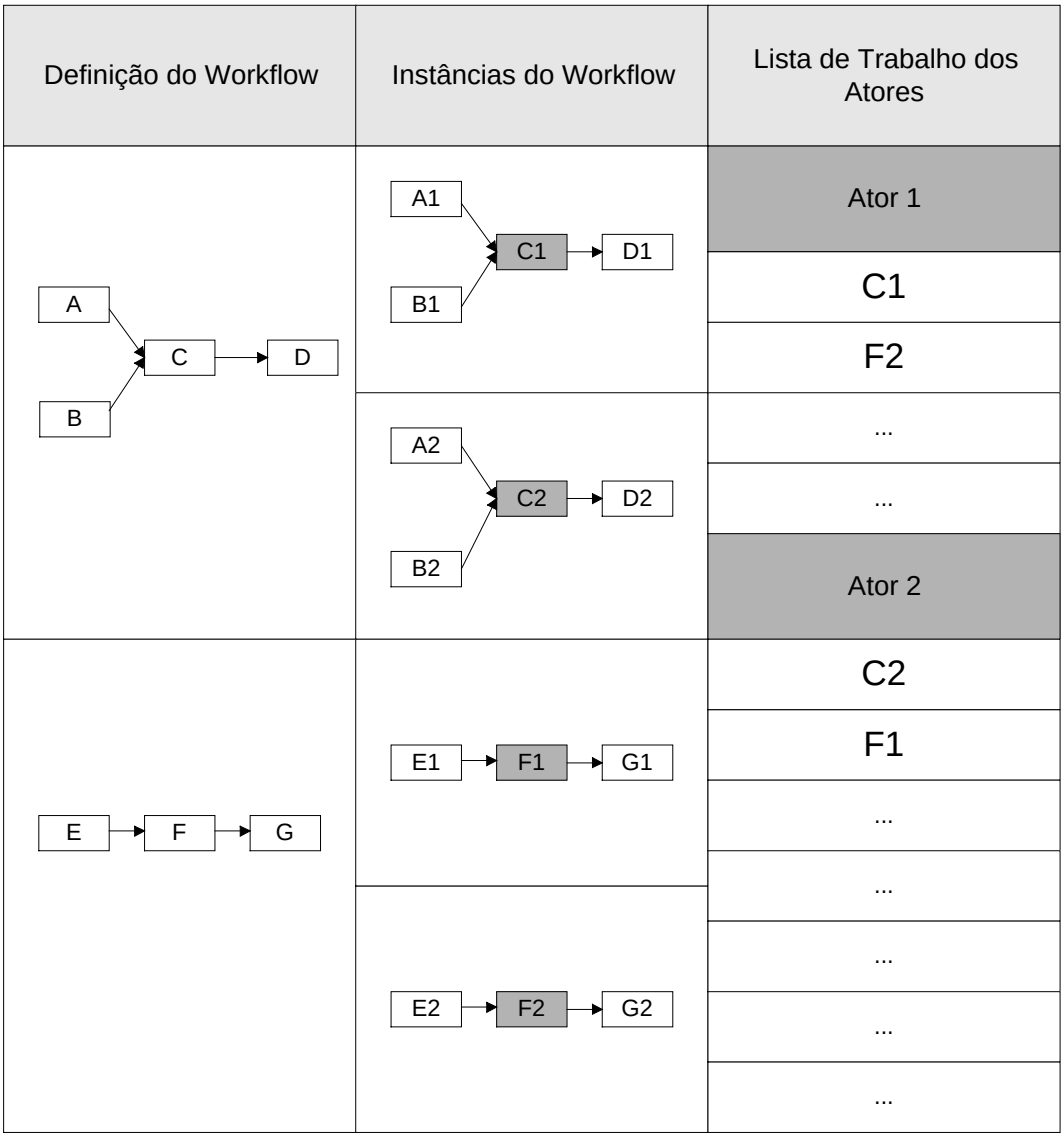


FIGURA 2.3 - Lista de trabalho dos Atores

2.3 Outras Definições de *Workflow*

Com o intuito de se analisar outras definições de *workflow*, este item apresenta uma série de definições, onde cada uma descreve a visão do seu autor desta tecnologia.

- Em [JOO94a], *workflow* é definido em termos de três elementos atômicos: evento, ator e objeto.
    - **evento**: é alguma coisa que ocorre, alguma coisa que acontece;
    - **ator**: é alguém que atua;
    - **objeto**: alguma coisa que é ou que seja capaz de ser vista, tocada ou sentida (ex.: uma carta, um produto, um serviço);

- **atividade**: é o conjunto de eventos que ocorrem sob a responsabilidade de um ator. Esta definição permite que uma atividade seja executada por várias pessoas, ao ponto que um ator é o responsável. Uma atividade é executada se os eventos da atividade ocorrem. Os atores podem ser pessoas ou máquinas (automação);

- ***trigger***: um evento e dispara uma atividade a se a ocorrência de e causa a execução de a. Porém, pode-se também ter um ator ou uma outra atividade disparando uma atividade;

- **processo**: é um conjunto de atividades que compartilham um propósito comum. A distinção entre um processo e uma atividade é motivada pela diferença de responsabilidade. Uma atividade tem um ator como responsável pela sua execução. Um processo pode envolver diferentes responsabilidades aos atores;

- ***workflow***: é um sistema cujos elementos são atividades, referindo-se uma com as outras através de *triggers*, e disparado por eventos externos;

- **sistemas de *workflow***: contém um *workflow*, todos atores, todas estruturas, e o significado envolvendo o *workflow*.

Em [JOO94a] é apresentado um diagrama (Figura 2.4) entidade-relacionamento [CHE90] que representa a relação entre as noções de evento, objeto, atividade, processo e ator, já descritas anteriormente.

Este diagrama é útil para se entender a composição de um *workflow*, e, principalmente, o relacionamento entre os elementos que o compõe.

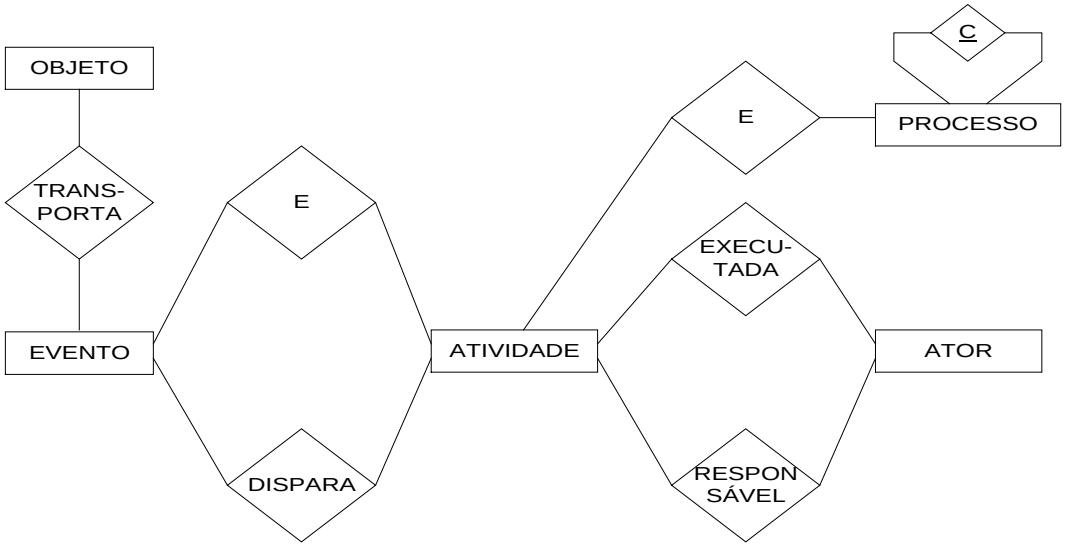


FIGURA 2.4 - Modelo Entidade-Relacionamento

Analisando o diagrama da Figura 2.4, verifica-se:

- um ator é responsável por uma atividade;

- uma atividade é executada por um ator;
- um evento carrega (traz consigo) objetos;
- uma atividade é "*triggered"* (disparada) por um evento;
- um evento pertence a uma atividade;
- uma atividade pertence a um processo;
- um processo é composto de um ou mais processos.

- em [ALO95], um modelo de *workflow* é definido como sendo um gráfico direcionado acíclico cujos nodos representam passos de execução e os fios (flechas) representam o fluxo de controle e de dados entre os diferentes passos;

- já em [CAS95a], *workflow* é definido como atividades que envolvem a execução coordenada de várias tarefas executadas por diferentes entidades processadoras. Uma tarefa define algum trabalho para ser feito por uma pessoa, por um sistema de software ou por ambos;

- segundo [AAL95a], um sistema gerenciador de *workflow* é composto por ferramentas de software que suportam e controlam tarefas administrativas em grandes organizações. Um *workflow* é um conjunto parcialmente ordenado de *jobs*, sendo o sistema gerenciador de *workflow* um sistema de computação que gerencia *workflows*;

- em [BUL92], *workflow* é definido como um sistema que ajuda organizações na execução, especificação, monitoração e coordenação dos itens do fluxo de trabalho dentro de um ambiente de escritório distribuído.

Como pode-se verificar, as definições de *workflow* são dadas sob escopos e pontos de vista diferentes. Algumas relacionam *workflow* com automação, outras a deixam de lado. Porém, todas associam *workflow* a execução de passos de maneira ordenada (totalmente ou parcialmente).

2.4 Funcionalidades de um Sistema de *Workflow*

Uma revisão rápida dos vários *workflows* (produtos comerciais) revela uma considerável disparidade no que os mesmo oferecem aos clientes. Levando-se em consideração este fato, em [DUI94][KHO95], é apresentado um conjunto de funcionalidades que um sistema de gerência de *workflow* deve oferecer, são elas:

- **roteamento de trabalho (*routing*):** possibilita a predefinição da seqüência em que as atividades serão executadas. Tipicamente, o participante do *workflow* recebe um item de trabalho e, quando termina o processamento deste, a atividade seguinte pode ser iniciada. Se as atividades trabalharem com um mesmo conjunto de documentos ou formulários, por exemplo, estes podem ser igualmente transferidos automaticamente e eletronicamente para a próxima atividade. O roteamento pode ser seqüencial ou baseado numa decisão. Neste segundo caso, significa dizer que um determinado teste ou regra será aplicado para determinar qual (ou quais) das atividades subseqüentes na definição do processo será executada. Ainda tem-se duas

opções, a primeira, denominada roteamento baseado em resposta, roteia o trabalho baseado na seleção, pelo usuário, de uma determinada ação. A segunda efetua o roteamento baseado em regras;

- **monitoramento e controle**: funções que fornecem informações sobre o *workflow*, isto é, acompanhar uma determinada instância de *workflow* e descobrir seu *status* atual de processamento, o responsável atual por sua execução, e quanto tempo ela está esperando na atividade atual. Isto ajuda a prevenir que determinados itens de trabalho fiquem parados em filas indefinidamente, principalmente, quando as atividades são escolhidas pelos participantes para serem executadas pelos mesmos. Estas informações devem ser fornecidas em tempo de execução;

- **notificação**: o trabalho coordenado também significa que o sistema notifique os usuários ou a outro *workflow* sobre tarefas e estouro de tempo. Estouro de tempo ocorre quando:
    - a atividade possui um determinado intervalo de tempo para ser executada e no término deste intervalo ela ainda não foi finalizada (ou mesmo iniciada);
    - a atividade possui uma data e horário pré-determinados de término e nesta data e horário ela ainda não foi finalizada (ou mesmo iniciada).

- **distribuição do trabalho (alocação de atividades):** quando um processo de um *workflow* é definido, em geral não se especifica um usuário individual para a execução da atividade, mas sim um papel, que pode ser assumido por várias pessoas dentro da organização. Logo, faz-se necessário que, quando a atividade for instanciada, seja determinado o participante que irá executá-la. Essa escolha pode ser feita, como será visto mais adiante, pelo próprio WFMS, ou então, manualmente, pelo gerente do *workflow* ou pelos próprios usuários aptos a executarem a atividade;

- **gerenciamento dos Procedimentos**: procedimentos em negócios são sujeitos a mudanças contínuas. Um sistema de *workflow* deve oferecer aos usuários finais meios para se redefinir (ou mesmo, definir) os passos do *workflow*, as seqüências dos passos, o caminho ao longo de certo passo e condições contendo regras as quais o caminho é determinado;

- **priorização de trabalho:** A capacidade de priorizar trabalho é uma função-chave dos WFMS. Já na primeira geração de sistemas de workflow, percebeu-se que uma política de execução de trabalho baseado no modelo FIFO (*first in first out*), isto é, o trabalho é entregue aos participantes na ordem exata de sua introdução no WFMS, pode não ser sempre o melhor mecanismo de prioridade, uma vez que determinadas instâncias de workflow devem possuir uma prioridade superior às demais;

- **geração de dados estratégicos:** A utilização de sistemas de workflow para o gerenciamento do trabalho na organização pode, a médio e longo prazo, reverter em benefícios estratégicos. Através do armazenamento de certos atributos de cada instância de *workflow* executada, pode-se criar uma base de dados que reflete a eficiência e a eficácia dos processos atualmente desempenhados pela organização. Com isso, pode-se responder a perguntas como “Quanto tempo em média uma determinada tarefa leva para ser executada?”, “Quanto tempo um item de trabalho espera em uma fila?” ou “Quantas transações podem ser processadas diariamente?”.

Com esses dados, pode-se, por exemplo, justificar economicamente a aquisição do sistema de *workflow*, mostrando seu impacto positivo sobre a produtividade da organização. Mas, mais importante que isso, com esses dados pode-se analisar profundamente o desempenho do processo atual, identificando gargalos e inconsistências, e, posteriormente, fazer melhorias e correções sobre ele.

2.5 Arquitetura de um Sistema de *Workflow*

[JOO95a] propõe uma arquitetura bastante genérica para sistemas de *workflow*. Ao mesmo tempo, ela é suficientemente flexível para possibilitar qualquer mecanismo de armazenamento e transferência de itens de trabalho. Qualquer componente da arquitetura pode ser armazenado em qualquer máquina, que inclui a possibilidade de dois ou mais componentes ocuparem a mesma máquina. Note-se que a estrutura apresentada é tipicamente a de uma arquitetura cliente/servidor (Figura 2.5).

Esta arquitetura, mostrada na Figura 2.5, é composta de cinco componentes de armazenamento (dados de definição, dados de instância, dados organizacionais, dados de gerência e dados da aplicação), três componentes ativos (processadores de interface, gerenciadores de eventos e gerenciadores de *workflow*) e três tipos de clientes que podem desempenhar três funções distintas na arquitetura: trabalhadores, desenvolvedores e gerentes.

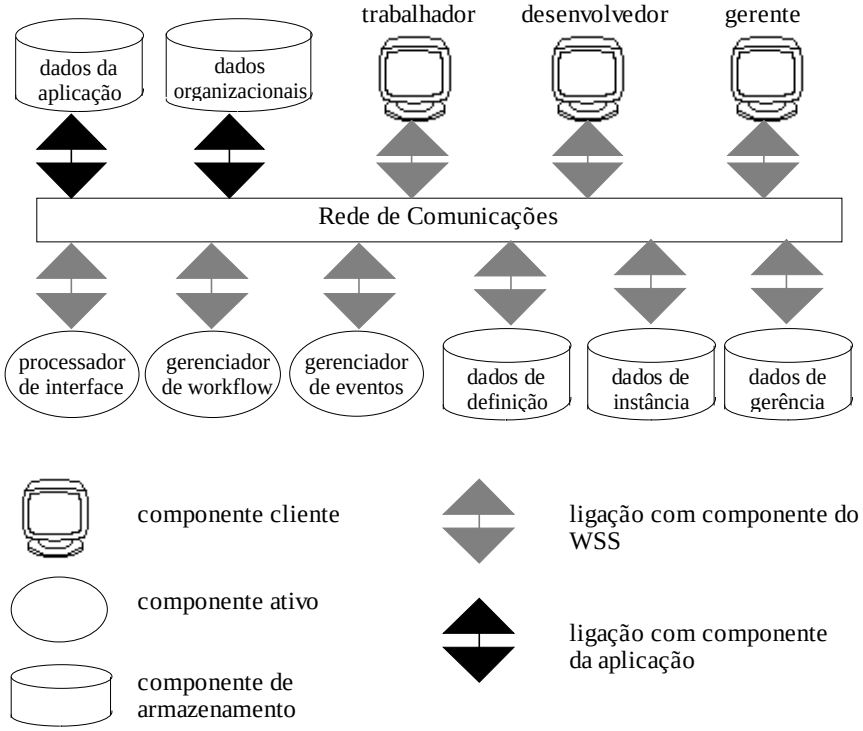


FIGURA 2.5 - Arquitetura genérica de um sistema de *workflow* [JOO95a]

Antes do detalhamento dos componentes, faz-se necessário diferenciar dois elementos da arquitetura: o sistema de suporte de *workflow* (WSS - *workflow support system*) e as aplicações de *workflow*. O sistema de suporte provê funções genéricas de gerência de *workflow*, como as descritas na seção anterior. As aplicações são

desenvolvidas para automatizar um processo específico, como o processamento de matrículas ou de pedidos de compra, e utilizam os serviços oferecidos pelo WSS. Na Figura 2.5, os elementos do sistema de suporte de *workflow* estão conectados com setas claras, enquanto os de aplicações estão com setas escuras.

Os componentes de armazenamento, como o próprio nome diz, são responsáveis pelo armazenamento de informações, tanto estáticas como dinâmicas, úteis tanto para o funcionamento do *workflow* como para o seu gerenciamento. São elas:

- **dados de definição**: componente responsável pelo armazenamento de informações estáticas referentes a estrutura dos *workflows*. Corresponde à etapa de especificação (modelagem) do *workflow*, portanto este componente é preenchido e mantido pelo especificador do *workflow;*

- **dados de instância**: componente responsável pelo armazenamento de informações dinâmicas referentes ao *status* corrente e ao histórico das instâncias do *workflow*. Esses dados são gerados durante a execução do *workflow*, e também são, em geral, consultados durante esta execução, para que decisões sobre roteamento e distribuição de trabalho sejam tomadas. Além disso, esses dados podem ser consultados pelos participantes do *workflow* para fins de acompanhamento, podendo assim informar outras pessoas (em geral, os clientes da organização) sobre o *status* atual da instância;

- **dados organizacionais**: componente responsável pelo armazenamento de informações referentes ao contexto organizacional onde o sistema de *workflow* está operando, como a estrutura da organização, funções e recursos (humanos e materiais). Esses dados não pertencem a um *workflow* específico, mas sim, comuns a todos *workflows* da organização. Um sistema de *workflow* necessita desses dados para respeitar autorizações e qualificações que são definidas pela organização, como a sua estrutura hierárquica;

- **dados de gerência**: componente responsável pelo armazenamento de dados sobre a execução de *workflows*, que podem ser utilizados para decisões estratégicas. Eles consistem basicamente de estatísticas extraídas das instâncias de *workflow*. São utilizados, em geral, pelos gerentes, para otimizar o desempenho dos processos. Com esses dados, é possível detectar-se diversos aspectos do processo: existência de gargalos, ocorrência de picos de trabalho, medidas de produtividade, tempos de conclusão do processo, tempos de conclusão de cada atividade e tempos de espera entre atividades, entre outros;

- **dados da aplicação**: componente responsável pelo armazenamento de dados pertinentes às aplicações, que são utilizados para a execução das atividades, como, por exemplo, documentos, formulários, planilhas eletrônicas e módulos de software. Basicamente, os dados da aplicação constituem-se no trabalho efetivamente realizado pelos participantes do *workflow*. Os outros quatro componentes existem unicamente para suportar este componente.

Os componentes ativos são responsáveis pelo interfaceamento e gerenciamento das instâncias dos *workflows*. Como já comentado, os componentes ativos são divididos em três tipos, a saber:

- **processadores de interface**: o processador de interface é responsável por ligar as aplicações ao sistema de *workflow*, fornecendo uma API para que estas possam fazer uso de seus serviços;

- **gerenciador de eventos**: responsável por manter a lista de trabalho a ser realizada, controlando prazos (*deadlines*) e demais condições para a execução do trabalho. Caso algum destes eventos ocorra, este componente encarrega-se de notificar aos participantes envolvidos;

- **gerenciador de *workflow***: responsável em coordenar a execução das instâncias de *workflow*, disparando-as e monitorando-as, e também se comunica com outros WFMS quando necessário.

Finalmente, a arquitetura proposta em [JOO95a], apresenta três tipos de clientes que podem desempenhar três funções distintas na arquitetura: trabalhadores, desenvolvedores e gerentes. Vale ressaltar que apesar das funções serem distintas, na prática, esses papéis podem ser acumulados por um único participante.

- **trabalhadores**: são os que realizam o trabalho estipulado, tendo as aplicações de *workflow* como interface e o WSS como infra-estrutura de software, constituindo-se assim nos efetivos participantes do *workflow*;

- **desenvolvedores**: constróem as aplicações e, possivelmente, adaptam o componente processador de interface para permitir a comunicação do WSS com outras aplicações. Para tal, necessitam de ambientes de construção de aplicações;

- **gerentes**: podem supervisionar o andamento do trabalho dos funcionários, controlando o cumprimento dos objetivos traçados, e utilizando as informações acumuladas sobre a execução dos *workflows* para propor alterações nos processos correntemente executados pela organização.

2.6 Tipos de *Workflow*

2.6.1 Classificação segundo McCREADY

De acordo com [MCC92], sistemas de *workflow* são classificados em três categorias, a saber: *workflow* de produção, *workflow* ad hoc e *workflow* administrativo. [MCC92] usa como critérios de classificação a estruturação prévia das atividades e a flexibilidade na tomada das decisões.

2.6.1.1 *Workflow* de Produção (*Production Workflow*)

Os sistemas de *workflow* classificados nesta categoria, destacam-se por serem bastante estruturados. Tipicamente, estão envolvidos com processos de informação complexos, onde existe a necessidade de acesso a múltiplos sistemas de informação (banco de dados). Essas aplicações, além de serem bastante estruturadas, são consideradas de missão crítica para a empresa, ou seja, são consideradas como sendo o principal processo de negócio da organização. Processo de requisição de seguros (Figura 2.6) e avaliações de empréstimo são exemplos deste tipo de *workflow*.

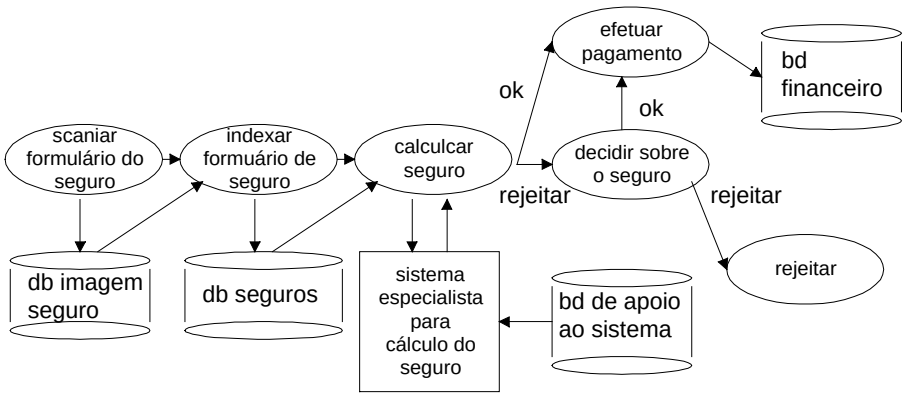


FIGURA 2.6 - Exemplo de um *Workflow* de Produção (requisição de seguro) [GEO95]

2.6.1.2 *Workflow* Ad Hoc (*Ad Hoc Workflow*)

Esses sistemas são caracterizados por possuírem processos menos estruturados que os *workflows* de produção. *Workflow* ad hoc envolve coordenação, colaboração e decisão humana. A ordenação e decisões são realizadas enquanto o *workflow* está sendo executado. Não há um padrão pré-determinado de movimentação de informações entre pessoas. Não são considerados de missão crítica, visto que falhas periódicas não interferem significadamente no processo de trabalho como um todo. Como exemplos pode-se citar: avaliação de trabalhos submetidos a um congresso onde não se conhece previamente o revisor (Figura 2.7) e o desenvolvimento de um software (processo de desenvolvimento de software).

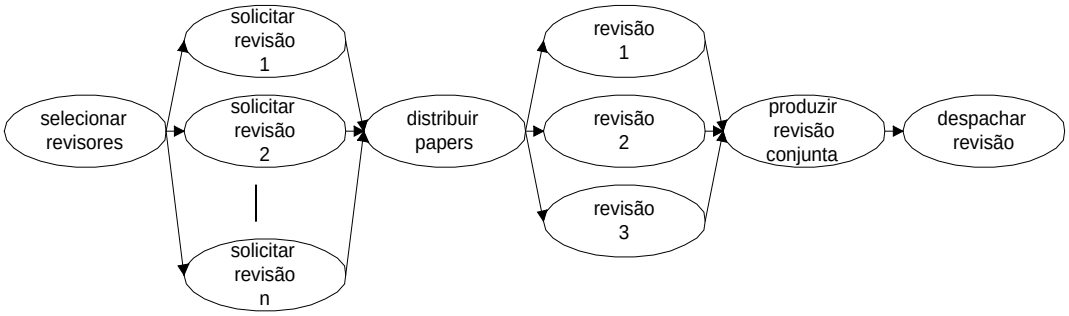


FIGURA 2.7 - Exemplo de um *Workflow* Ad hoc (revisão de trabalhos por avaliadores) [GEO95]

2.6.1.3 *Workflow* Administrativo (*Administrative Workflow*)

Possuem diversas características dos sistemas de produção (estruturados), porém são direcionados para atividades administrativas internas da organização. *Workflows* administrativos geralmente não são de missão crítica, visto que são menos exigentes em relação a confiabilidade, correção e integração com sistemas externos do que os *workflows* de produção. Como exemplo pode-se citar: avaliação de trabalhos submetidos a um congresso onde já se conhece previamente os revisores (Figura 2.8) e autorização de viagens.

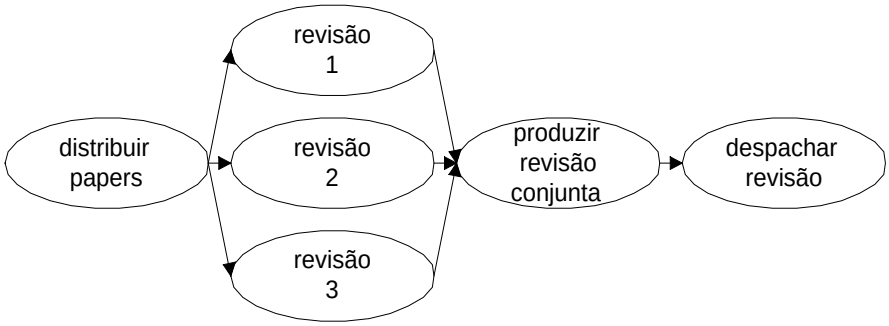


FIGURA 2.8 - Exemplo de um *Workflow* Administrativo (revisão de trabalhos por avaliadores) [GEO95]

2.6.2 *Workflow* orientado a pessoas e *workflow* orientado a sistemas

[GEO95] apresenta uma outra classificação para *workflows*, caracterizando-os dentro de dois aspectos: orientado a pessoas e orientado a sistemas. O primeiro tipo envolve humanos na execução e coordenação de tarefas, enquanto o segundo tipo, envolve sistemas de computadores que executam operações computacionais intensas e softwares especializados em tarefas.

A grande diferença entre estes dois tipos de *workflow*, está no fato do primeiro, orientado a humanos, controlar e coordenar tarefas humanas, e o segundo, orientado a sistemas, controlar e coordenar tarefas de softwares com pequena intervenção humana. Desta forma, faz-se necessário que os sistemas de *workflow* orientados a sistemas necessitem incluir softwares para controle de concorrência e técnicas de recuperação

para assegurar a consistência e a segurança das informações manipuladas por esse tipo de sistema.

Em um *workflow* orientado a pessoas, as principais questões a serem analisadas são:

- interação homem-máquina;
- combinar habilidades humanas para suportar as tarefas necessárias;
- modificar a cultura do escritório, isto é, como as pessoas preferem ou necessitam trabalhar.

Em um *workflow* orientado a sistemas, as principais questões a serem analisadas são:

- combinar as necessidades dos processos de negócios para a funcionalidade do sistema e providenciar dados a partir dos sistemas já existentes;
- interoperabilidade entre sistemas do tipo HAD (heterogêneo, assíncrono, distribuído);
- procurar softwares adequados para executar tarefas de *workflow*;
- determinar novas necessidades de software de forma a permitir automação dos processos de negócios;
- assegurar a execução correta e segura dos sistemas.

2.6.3 *Workflow* Transacional

Workflow transacional envolve a execução coordenada de múltiplas tarefas que:

- podem envolver humanos;
- requer acesso para sistemas do tipo HAD (heterogêneo, assíncrono, distribuído);
- suporta o uso seletivo de propriedades transacionais (atomicidade, consistência, isolamento e durabilidade).

O uso seletivo de propriedades transacionais é necessário para permitir a especialização das funcionalidades necessárias para cada *workflow* (por exemplo: permitir colaboração de tarefas e suportar estruturas complexas de *workflow*).

2.7 Classificação das Atividades

Em [BAR95a] é apresentada uma classificação sobre os tipos de atividades que podem existir na especificação de um *workflow*, são elas:

- **baseada na instância do *workflow* x atividades b*atch***: baseia-se no número de instâncias com as quais uma atividade lida;

- **estruturada x não-estruturada**: baseia-se no fato da atividade poder ser colocada ou não sob supervisão do WFMS;

- **internas x externas**: leva em consideração se a atividade será executada dentro ou fora do sistema de *workflow*;

- **quanto à finalidade**: baseia-se no tipo de função na qual a atividade está envolvida.

2.7.1 Baseada na Instância do *Workflow* X *Batch*

- **baseada na instância do *workflow***: são atividades que lidam exclusivamente com uma única instância de um *workflow*. A informação necessária para executar a instância da atividade está disponível para aquela particular instância do *workflow*, e o produto da instância da atividade é adicionado à instância do *workflow* correspondente. Como exemplo tem-se o preenchimento de uma ordem de compra. Algum funcionário precisa preencher os campos da ordem de compra, e talvez consultar o banco de dados da organização para buscar informações sobre preço e código dos itens. Nenhum acesso a outra instância do *workflow* é necessário, e o resultado da atividade só compete à instância do *workflow* em questão;

- ***batch*:** é uma atividade que somente pode ser completada se um conjunto de instâncias de um dado *workflow* estiverem sendo executadas em conjunto, e o produto da atividade for adicionado a um subconjunto das instâncias . Uma típica atividade *batch* é a classificação (ranking), por exemplo a classificação de candidatos após uma prova. A atividade de classificação será executada após todas as instâncias do *workflow* (prova) dos candidatos estiverem finalizadas. O produto desta atividade (classificação) será inserido em um campo de cada instância do *workflow* dos candidatos, ou seja, cada candidato receberá sua classificação.

2.7.2 Estruturada X Não-Estruturada

- **estruturadas**: são atividades bem entendidas pelo sistema e pelas pessoas que especificaram o procedimento. Essas atividades podem ser realizadas por uma única pessoa, trabalhando dentro do sistema de *workflow*. O preenchimento de uma ordem de compra, como visto anteriormente, também é um exemplo de atividade estruturada, sendo executada por uma única pessoa, e gerenciada pelo sistema de *workflow*;

- **não-estruturadas**: são atividades que não podem ser colocadas sob supervisão do sistema. É o caso de atividades que envolvem um trabalho de criatividade, por exemplo. Geralmente são executadas por um grupo de pessoas, trabalhando com ferramentas fora do *workflow*. Como exemplo, tem-se uma ordem de serviço para uma construtora, onde pode existir uma atividade de projeto que será executada por um grupo de engenheiros e arquitetos. Tal atividade demanda um trabalho de criatividade que não tem como ser supervisionada pelo sistema de *workflow*.

2.7.3 Internas X Externas

- **internas**: são atividades executadas dentro do sistema de *workflow*, e monitoradas pelo mesmo. Como exemplo, também pode-se citar uma ordem de compra, visto que a mesma é executada dentro do sistema de *workflow*, e sob seu monitoramento;

- **externas**: são atividades executadas fora do sistema de *workflow*. Isto significa que durante a execução deste tipo de atividade, o sistema de *workflow* não fornece ferramentas para sua execução, nem é capaz de monitorar seu progresso. Geralmente, atividades não-estruturadas são atividades externas, porém, as atividades estruturadas também podem ser externas. Por exemplo, o sistema de *workflow* é usado somente por alguns departamentos da organização, e as atividades de um procedimento são executadas por outros departamentos externos ao sistema de *workflow*. Documentos relevantes da instância do *workflow* devem ser impressos, manipulados pelos departamentos externos ao sistema e, de alguma forma, devem ser inseridos de volta ao sistema.

2.7.4 Quanto à Finalidade (Função)

- **busca de dados externos (*External Data Gathering*):** atividades envolvidas na obtenção e armazenamento de dados de fontes externas;

- **processamento de dados (*Data Processing*):** atividades que derivam novas informações de dados externos, classificados de acordo com critérios internos da organização;

- **decisão (*Decision*):** atividades que envolvem a escolha entre alternativas através de um julgamento subjetivo. A atividade de decisão irá determinar qual caminho o fluxo irá tomar;

- **autorização (*Authorization*):** a atividade de autorização é uma forma fraca de decisão, onde uma pessoa da organização assume a responsabilidade pelo estado da instância do *workflow*;

- **espera (*Waiting*):** é uma atividade a qual a instância do *workflow* fica esperando por algum *trigger* para dar seguimento à próxima atividade;

- ***schedule* Temporal (*Temporal Scheduling*):** são atividades que determinam um horário ou pelo menos a ordem na qual um conjunto de atividades será executado;

- **atividades proxy (*Proxy Activities*):** é um tipo de atividade de espera (*waiting*) que só será executada após a ocorrência de um número de execuções de uma atividade externa.

- **alocação (*Actor assignment*):** é uma atividade que determina quem executará a instância de uma outra atividade.

2.8 Limitações dos Sistemas de Gerenciamento do *Workflow*

Segundo [GEO95], os sistemas de gerenciamento de *workflow* possuem uma série de limitações, a saber:

1. **falta de Interoperabilidade entre *Workflow Management Systems* (WFMS)**: Diretamente ligado a falta de padrões para WFMS. Como comentado no capítulo anterior, este assunto já vem sendo tratado pela WfMC;

2. **performance inadequada**: Os WFMS suportam não mais do que 100 *workflows* por dia. Alguns processos necessitam manipular um grande número de *workflows*, um número bem maior do que 100 por dia;

3. **falta de suporte para correção e confiabilidade na presença de concorrência e falhas**: A execução de um *workflow*, assim como qualquer execução de aplicações que acessam recursos compartilhados, deve preocupar-se com: a consistência de tarefas individuais, consistência de *workflows* individualmente, e a execução concorrente de tarefas provenientes de *workflows* diferentes. O problema de confiabilidade do *workflow* envolve a consistência quando o *workflow* termina de forma anormal. Tarefas completadas de um *workflow* parcialmente completo devem ser desfeitas ou compensadas. Alternativamente, tarefas incompletas de um *workflow* parcialmente completo pode necessitar ser refeito ou uma tarefa contigencial necessitará ser executada. É importante saber quais tarefas foram completadas, quais estão ainda ativas, quais não começaram, e quais tarefas precisam ser desfeitas ou refeitas para garantir a consistência. Devido a esses problemas, WFMS contam com *workflow* designers para fornecer especificações que incluam tarefas e ações de compensação e, programadores de tarefas para elaborar códigos para controle da concorrência e log;

4. **ferramentas de suporte fracas para análise, teste, e *debug* de especificações e implementações**: Tais ferramentas são necessárias para avaliar a eficiência da especificação e implementação do *workflow*, simular uma execução, e determinar problemas. A sofisticação de tais ferramentas facilita a especificação e implementação do *workflow*, contribuindo diretamente para sua elaboração. Simulação é necessária para descobrir erros lógicos e conseguir um estimativa de quanto tempo um *workflow* leva para ser executado (completado).

2.9 Implicações

Pode-se dividir as implicações do *workflow* e do sistema gerenciador em dois grupos. O primeiro refere-se às implicações positivas que os mesmos podem trazer para o negócio como um todo da empresa. Já o segundo grupo, refere-se às fraquezas e obstáculos que tal tecnologia pode impor à empresa [GEO95][DUI94].

2.9.1 Implicações Positivas

1. **gerenciamento dos processos**: os processos são controlados pelo sistema, o que facilita seu gerenciamento por parte da empresa;

2. **eliminação de atrasos**: rotinas automatizadas e trabalho dividido eliminam atrasos desnecessários. Isto conduz ao aumento do *throughput*;

3. **diminuição de Falhas**: partes pequenas do processo são controladas pelo sistema. Validações são fáceis de serem implementadas;

4. **aumento da produtividade**: a produtividade pode aumentar devido ao aumento do *throughput*. Mais casos podem ser processados na mesma quantidade de tempo;

5. **melhorar o serviço aos clientes**: no caso de questionamentos, todas as informações essenciais de interesse de um certo cliente estão em mãos. Também, o sistema pode ser configurado para oferecer novos serviços para antecipar sinais do mercado;

6. **redução dos custos**: diminuição dos custos pode ser visto em custo de pessoal, uso de papel e impressora. Devido ao aumento do *throughput* o custo por caso pode reduzir;

7. **aumento do gosto pelo trabalho**: entre outros fatores, pode-se citar a diminuição da pressão do trabalho devido a forma estruturada do mesmo, bem como a flexibilidade de se alterar a rotina de trabalho mediante a sugestão de algum funcionário, o que o torna peça fundamental;

8. **redução da vulnerabilidade**: aumento do conhecimento do processo de trabalho, formas fáceis de se organizar o trabalho, são instrumentos que reduzem a vulnerabilidade do trabalho.

2.9.2 Fraquezas e Obstáculos

1. **processos rígidos**: o sistema pode ditar a forma de trabalho e a ordem que as atividades serão executadas. *Workflow* pode ser usado para colocar uma camisa de força nas pessoas;

2. **inspeção gerencial**: todos os trabalhadores podem ser monitorados com respeito a sua velocidade e forma de trabalho. O uso de tais informações pode prejudicar um bom ambiente de trabalho;

3. **redução do gosto pelo trabalho**: se as unidades de trabalho são muito pequenas, trabalhadores são forçados a executar muitas partes pequenas de uma atividade em altas quantidades. Isto reduz a motivação e o gosto pelo trabalho;

4. **sem volta**: a implementação de um sistema de *workflow* exige uma grande organização por parte da empresa, visto que esta sofrerá várias mudanças. Um projeto de *workflow* pode ser projetado de tal forma que o retorno seja impossível. Isto pode causar sérios problemas se o projeto de *workflow* falhar.

Um projeto de *workflow* deveria ser projetado para fazer surgir todas as vantagens, e prevenir-se contra os obstáculos. Várias precauções podem ser tomadas para aumentar as chances de sucesso:

- investigação preliminar;
- equipe multi-disciplinar;
- desenvolvimento evolutivo;
- participação dos usuários;
- utilizar um protótipo ou piloto;
- investigar os circuitos de comunicação formais e informais.

2.10 Modelos para Especificação de *Workflow*

2.10.1 Introdução

Segundo [BAR95], modelagem tem uma longa história. Várias metodologias de análise existem e são usadas para capturar aspectos relevantes. Modelagem do *workflow* adiciona a esses processos tradicionais (análise estruturada, análise orientada ao objeto) alguns novos ingredientes: papéis e sincronismo.

Além disso, o escopo dos sistemas de *workflow* é normalmente maior do que o dos sistemas tradicionais, uma vez que eles devem suportar não somente atividades automatizadas (estruturadas), mas também atividades não estruturadas e manuais, aquelas que não possuem uma solução algorítmica.

O sincronismo em um sistema de *workflow* estabelece dependência entre as atividades, e especifica quais tarefas devem ser executadas em paralelo e quais devem necessariamente ser prorrogadas até que uma dada atividade seja completada.

Já em sistemas tradicionais, usuários devem achar alguma forma de coordenar seus esforços, talvez pelo estabelecimento de um protocolo manual. Este protocolo será usado para avisar outros participantes que alguma atividade deve ser executada.

Sistemas de *workflow* já são construídos com a alocação de atividades para a pessoa certa, automaticamente ou com a ajuda de um agente humano. Para ser capaz de distribuir o trabalho, o sistema deve ter algum conhecimento da estrutura organizacional hierárquica, funcional, papéis e atores. Este modelo organizacional entre outras utilidades, será usado para associar atores a papéis.

Em sistemas construídos de acordo com os métodos tradicionais normalmente partem do princípio que os operadores/usuários serão treinados para saber qual, quando e como acessar cada função do sistema [BAR95].

Portanto, ao se especificar um *workflow*, deve haver uma preocupação com esses fatores, escolhendo uma metodologia que especifique de maneira clara e que não traga dúvidas sobre o funcionamento do *workflow*.

Ao se modelar um *workflow*, seja ele uma linha de produção ou um fluxo de documentos, está se modelando a maneira ou forma com que este trabalho é ou será executado. Como citado anteriormente, o *workflow* traz a promessa de integrar o trabalho dentro da organização, bem como melhorar o desempenho do mesmo. Para tanto, esta tarefa deve ser feita de maneira bem criteriosa e detalhista. Uma importante

parcela do esforço das organizações é despendido para o tratamento de exceções [BAR95], exceções essas que ou foram deixadas de lado num primeiro instante ou não foram percebidas/analisadas de forma criteriosa e acabaram não sendo incluídas no *workflow*. Se fosse possível expressar-se mais em um determinado nível, menos esforço seria gasto tratando essas situações como exceção [BAR95]. De fato, elas deixariam de ser exceções, tornando-se regras, e conseqüentemente, deixando o *workflow* mais preciso e robusto.

Visto que as organizações são normalmente compostas de pessoas com diferentes formas de se expressar, cada uma com sua própria linguagem e visão do mundo, um modelo único que especifique o fluxo de atividades (*workflow*) de uma organização terá, pelo menos, a importância de trazer uma homogeneidade no que tange ao entendimento do fluxo, possibilitando, desta forma, que todos trabalhem em função de seu aperfeiçoamento e/ou otimização.

[GEO95] identifica três etapas que envolvem o desenvolvimento de sistemas de *workflow* (Figura 2.9) , são elas:

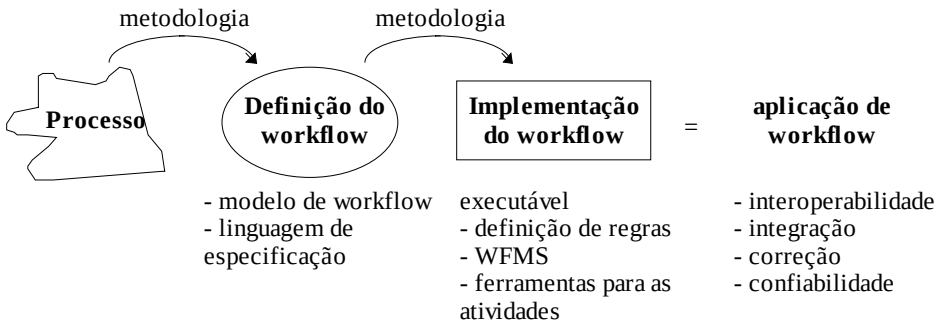


FIGURA 2.9 - Fases do desenvolvimento de sistemas de *workflow*

1. **modelagem de processos e definição de *workflow***: requer modelos de *workflow* e metodologias para capturar o processo como uma definição de *workflow*. Assim, em um passo inicial, é necessário compreender o processo que se deseja modelar, o que tipicamente é feito através de entrevistas com os participantes do processo. [GEO95] sugere a utilização de técnicas de entrevistas similares às utilizadas para o projeto de sistemas especialistas, como forma de obter-se um conhecimento mais aprofundado da natureza dos processos. Quando conhecimento suficiente for adquirido, pode-se capturar o processo em uma definição de *workflow*. Para realizá-la, é necessária a utilização de um modelo de *workflow*. Um modelo de *workflow* tipicamente inclui uma série de conceitos para descrever os processos, como os já vistos: atividades, dependências e papéis;

2. **reengenharia de processos**: requer metodologias para a otimização dos processos. Esta área é basicamente preocupação dos profissionais e pesquisadores de administração organizacional. [HAM94] define os princípios da reengenharia e diversas formas de executá-la, de acordo com os objetivos desejados, como aumento de produtividade, redução de custos, melhor atendimento aos clientes, entre outros;

3. **implementação e automação de *workflow***: requer metodologias e tecnologias para utilizar sistemas de informação e participantes humanos para implementar, escalonar,

executar e controlar as atividades definidas no *workflow*. [GEO95] apresenta uma extensa lista das arquiteturas e tecnologias necessárias para a efetiva implantação destes sistemas, como integração, interoperabilidade e confiabilidade.

A seguir, serão apresentados alguns modelos encontradas na literatura que possibilitam a modelagem/especificação do *workflow*.

2.10.2 Modelos Baseados em Atividades

Os modelos baseados em atividades enxergam o trabalho como uma seqüência de atividades, onde cada atividade recebe um certo conjunto de entradas e produz um certo conjunto de saídas.

Esta classe de técnicas caracteriza-se, em termos gerais, pela existência explícita de:

- um processo de transformação qualquer (atividade);
- um conjunto de entradas para este processo (eventualmente, pré-condições para sua ocorrência);
- um conjunto de saídas deste processo (eventualmente, pós-condições de sua ocorrência).

2.10.2.1 Modelagem por Gatilhos (*Trigger Modelling*)

A técnica de modelagem por gatilhos (*trigger modelling*) foi proposta por Stef Joosten [JOO94b]. O modelo é formal, onde o seu nível mais baixo de abstração pode ser convertido para redes de Petri, através de um mapeamento estabelecido pelos próprios idealizadores do modelo.

O propósito de se modelar um sistema de *workflow* baseado na dinâmica do comportamento do mesmo, ou seja, quem dispara quem, pode facilitar a modelagem dos aspectos dinâmicos.

Atividades são executadas (disparadas) em função da ocorrência de eventos (o evento é a pré-condição da atividade). Por outro lado, um evento ocorre como resultado da execução de uma atividade (o evento é a pós-condição da atividade).

Segundo [JOO94b], pelo fato de atividades, papéis e *triggers* serem importantes, eles devem ser tomados como ponto de partida para a análise e *design* do *workflow*. Este fato torna a modelagem do *workflow* diferente da modelagem de sistemas de informação, que convencionalmente começam pela modelagem das interfaces, das estruturas de dados ou pelos processos.

Cada atividade é representada por um retângulo, contendo o nome da atividade. Uma seta apontando em direção a uma atividade significa que a atividade é disparada (*triggered*) por um evento que ocorreu como resultado da atividade colocada no início da seta.

O modelo de *triggers* proposto em [JOO94b] é dividido em colunas, cada uma delas contendo uma atividade associada a um particular papel.

O exemplo a seguir representa um *workflow* de uma clínica radiológica na confecção de uma documentação ortodôndica para uma dado paciente. O *workflow* é composto de 8 atividades a saber: pedir documentação, tirar radiografia cefalomêtrica, tirar radiografia panorâmica, laudo técnico, traçado, fotos, modelos e a entrega da documentação. O *workflow* também possui 5 papéis, são eles: cliente, radiologista, operador do computador, auxiliar e office boy (Figura 2.10).

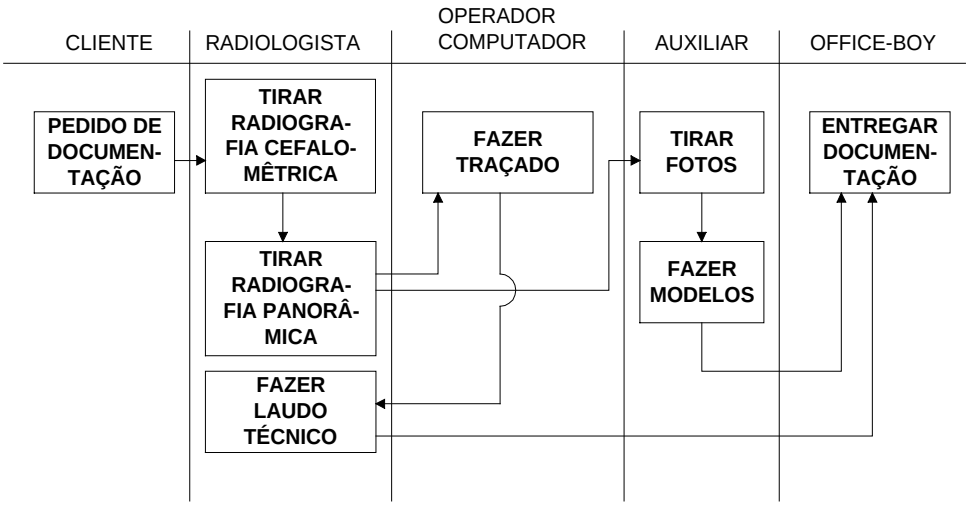


FIGURA 2.10 - *Workflow* Documentação Ortodôntica a nível de usuário leigo

Num primeiro instante, a modelagem do *workflow*, quando envolve analistas e clientes, deve ter um enfoque voltado para atividades, papéis e *triggers*, utilizando para tanto, retângulos e setas.

Em estágios mais avançados, o enfoque pode ser dado somente às atividades. Desta forma, deve-se substituir os retângulos por:

- **círculos**: quando a atividade é atômica, ou seja, ela não possui estrutura interna;
- **triângulos**: quando a atividade sincroniza *triggers*;
- **outro *workflow***: quando uma atividade é complexa, o retângulo é mantido para indicar tal complexidade.

Sendo assim, o exemplo da Figura 2.10 ficaria da seguinte forma (Figura 2.11):

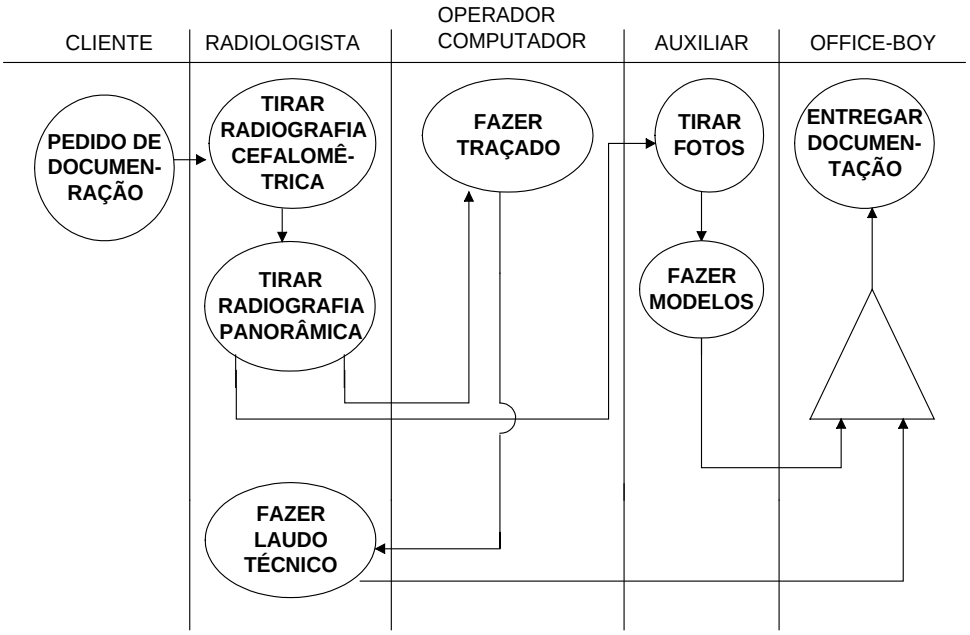


FIGURA 2.11 - *Workflow* Documentação Ortodôntica Avançado

Devido a presença de círculos e triângulos, esta representação é menos amigável para se analisar o *workflow* junto a clientes, quando comparada à representação que se utiliza somente de retângulos e setas. Em contra partida, ela traz mais semântica ao *workflow*, fazendo com que o modelo fique mais fácil de ser compreendido e implementado, principalmente pelo fato de mostrar o sincronismo entre as atividades.

2.10.2.2 Modelo de Barthelmes/Wainer

Segundo [BAR95], as linguagens de descrição existentes normalmente suportam somente a especificação da linha principal da organização, isto é, os procedimentos dos casos mais previsíveis. Aqui, a criação de uma descrição é poderosa e capaz de expressar mais do que a linha principal, que pode ser alcançada por uma melhor integração da manipulação de exceções.

O modelo proposto é baseado na definição do seus componentes conceituais, que, quando possível, seguem a definição proposta pela *Workflow Management Coalition* [WMC96d], são eles:

- **plano**: especifica estados e a transição entre eles, estabelecendo uma ordem;
- **objetos**: encapsulam os dados necessários durante a execução do *workflow*;
- **atividades**: especificam funções que transformam objetos;
- **ator**: participante do *workflow*, pode ser uma pessoa, um programa ou um equipamento;
- **estado**: indica um estágio no ciclo de vida de um *workflow*;

- **sub-Planos**: corresponde a uma decomposição hierárquica de um plano;

- **definição do Processo**: modela a solução do objetivo de um negócio.

Durante a execução de um *workflow*, situações excepcionais podem causar mudanças tanto no plano, como também no tipo dos objetos e nas atividades. Segundo [BAR95], tais mudanças irão modelar o *workflow* a situação de exceção. Tais exceções irão causar mudanças nas necessidades do *workflow*, trazendo às vezes a necessidade de uma re-análise e/ou re-planejamento no momento da execução do *workflow*.

As ocorrências dos sistemas de *workflow* são assíncronas, isto é, não é possível antecipar o exato momento de sua ocorrência. Dependendo do momento da ocorrência do evento, respostas diferentes podem ser geradas [BAR95].

Devido ao fato de que quase todas as linguagens de descrição não serem capazes de lidar com eventos assíncronos [BAR95], o tratamento dos mesmos será de exceção, ou seja, a adequada resposta irá depender do conhecimento do usuário em lidar com a situação. Usuários com conhecimento do assunto irão lidar com sucesso, porém usuários sem treinamento ou experiência terão problemas para tratarem a exceção.

Sendo assim, para resolver este problema, o modelo proposto em [BAR95], deve suportar eventos assíncronos tanto no momento da modelagem como no momento da execução do *workflow*.

A principal diferença entre o modelo proposto em [BAR95] e outros modelos baseados em *trigger* está em se lidar com eventos assíncronos tanto na especificação como na execução do *workflow*.

A idéia é ter transições entre eventos e não entre atividades. Isto permite representar estados de espera e eventos externos de uma maneira uniforme, mesmo que esses estados não tenham nenhuma atividade associada, isto é, eles são puramente uma saída para o sincronismo do *workflow*. Isto torna a visão do fluxo de conexão mais clara, fazendo com que a análise do *workflow* seja mais precisa.

Em [BAR95] não é apresentado uma forma gráfica ou textual de se especificar o *workflow*, porém é levantada uma comparação com os modelos de *triggers* que fazem transições entre atividades e não entre eventos. Como pode ser visto no exemplo a seguir (Figura 2.12), determinadas atividades não disparam (*trigger*) as próximas, dando a impressão que essas são disparadas por atividades *batch*. De acordo com [BAR95], usando-se estados, pode-se criar atividades que representam algum tipo de espera, ou mesmo, uma atividade *batch*.

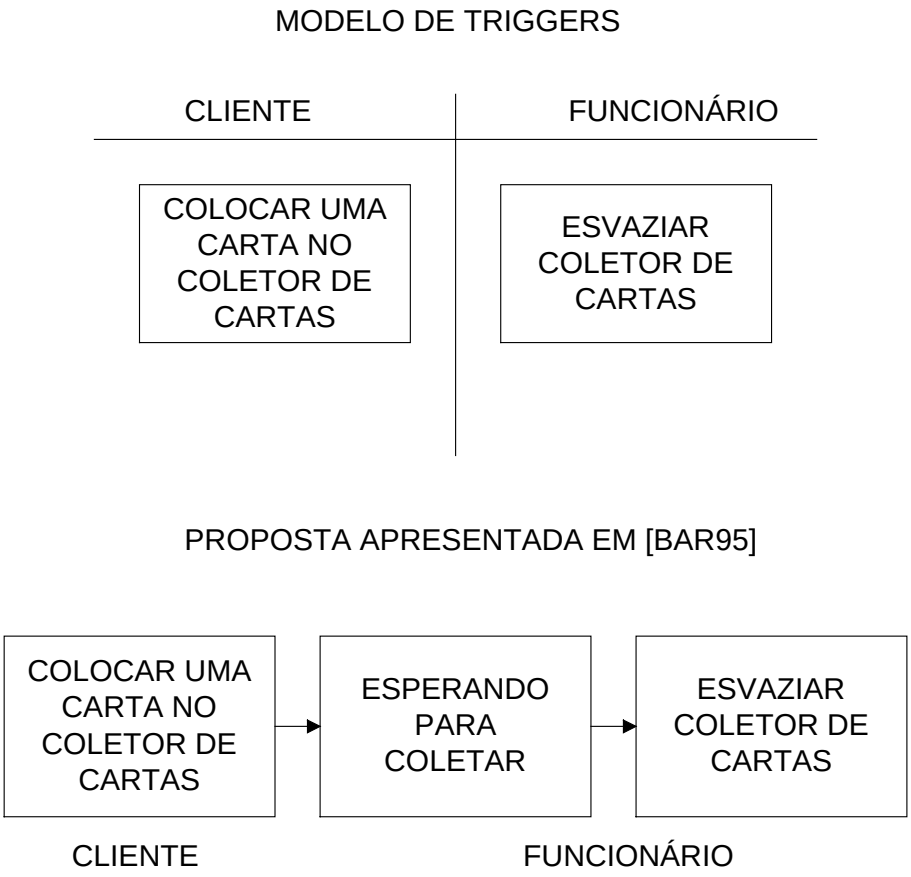


FIGURA 2.12 - Modelo de *Triggers* X Proposta apresentada em [BAR95]

Como pode ser visto no exemplo apresentado, usando-se atividades, não há como representar a existência de um estado de espera entre as duas atividades. Usando-se eventos, pode-se inserir um evento entre as duas atividades, demonstrando uma seqüência de disparados (*triggers*) de atividades.

2.10.2.3 Modelo de Casati

O modelo proposto por Casati, Ceri, Pernici e Pozzi [CAS95a] é um dos modelos mais completos para a especificação de *workflows*. As preocupações do modelo estão em três tópicos:

- descrição formal do "comportamento interno" do *workflow*, como a definição, interação e cooperação entre atividades;
- relacionamento entre o *workflow* e o seu ambiente, como a alocação de atividades a atores;
- acesso a bases de dados externas (através de comandos SQL). Assim, os autores almejam a integração entre sistemas de *workflow* e os sistemas de informação atuais, fortemente assentados em bases de dados.

Segundo [CAS95a], *workflows* são atividades que envolvem a execução coordenada de múltiplas tarefas desempenhadas por diferentes entidades processadoras

(atores). A especificação de um *workflow* envolve descrever aspectos de cada tarefa e das entidades processadoras aptas pela execução, aspectos estes que são relevantes para controlar e coordenar sua execução, bem como a relação entre elas.

A especificação sugerida é composta por uma linguagem de descrição do *workflow*, que combina tanto a especificação quanto o acesso a banco de dados externos, sendo este último fator visto como ponto diferencial do modelo. A linguagem de descrição descreve as tarefas a serem executadas durante a execução da especificação do *workflow*, e mecanismos que são usados para ativá-las e terminá-las, tanto em soluções normais ou excepcionais.

Segundo [CAS95a], para tornar efetiva a convergência entre o gerenciamento do *workflow* e os bancos de dados é necessário uma melhora e um reforço na especificação do nível conceitual do *workflow*, pela formalização dentro de um único modelo do comportamento interno do *workflow* (interação e cooperação entre tarefas), seu relacionamento com o ambiente (alocação de pessoas às tarefas) e o acesso a banco de dados externos.

Para especificar o *workflow* e seu relacionamento com bancos de dados, [CAS95a] propõe uma linguagem de descrição composta por símbolos (Figura 2.13) e textos. A saber:

Tasks: tarefas são unidades de trabalho elementares que coletivamente alcançam o objetivo da especificação do *workflow*. O sistema de gerenciamento do *workflow* toma cuidado em determinar quando uma certa tarefa deve começar a ser executada e alocar um agente executador, de acordo com alguma política de alocação. Cada tarefa possui as seguintes características:

- **nome**: nome da tarefa;
- **descrição**: descrição sucinta da tarefa;
- **pré-condições**: condições de início da tarefa;
- **ações**: uma ação é uma seqüência de declarações que definem como dados temporários e persistentes são manipulados pela tarefa;
- **exceções**: um conjunto de pares <exceção, reação> usado para se lidar com eventos anormais; toda vez que uma exceção ocorrer, a reação correspondente será executada.

Conexões: conexões descrevem interação entre tarefas; conexões possuem tanto uma linguagem de descrição como uma descrição gráfica. Duas tarefas A e B podem ser diretamente conectadas, neste caso elas são unidas por um fio; o significado intuitivo é que, tão logo A termine, B está pronta para execução. Em outros casos, conexões entre tarefas são executadas por uma tarefa de roteamento/direcionamento. Neste caso, pode-se ter uma fork task, para iniciar a execução concorrente de tarefas, ou uma join task, para sincronizar tarefas depois de uma execução concorrente.

- fork task
    - total: depois que o predecessor termina, todos os sucessores estão prontos para execução;

- não determinístico: o fork é associado a um valor k; depois que o predecessor termina, k sucessores selecionados de maneira não determinística estão pronto para execução;
- condicional: cada sucessor está associado a uma condição; depois que o predecessor termina, condições são instantaneamente avaliadas e somente sucessores com condições (*true*) estarão prontos para execução;
- condicional com exclusão mútua: agrega à situação anterior a restrição de que somente uma condição pode ser verdadeira; desta forma, somente um sucessor estará apto para execução.

- join task
  - total: o sucessor torna-se pronto somente depois do término de todos predecessores;
  - parcial: o join é associado a um valor k; o sucessor torna-se apto depois do término de k predecessores; términos subsequentes de predecessores não têm efeito;
  - interativo: o join é associado a um valor k; o sucessor fica apto toda vez que k predecessores terminam.

Start e Stop símbolos: possibilitam a criação e a finalização de uma execução do *workflow*. Cada *workflow* possui um símbolo de começo e vários símbolos de término; o símbolo de começo tem somente uma tarefa sucessora (possivelmente uma tarefa de conexão) e cada símbolo de parada possui vários símbolos predecessores. Quando qualquer símbolo de término tornar-se pronto, o *workflow* está completo. Tarefas que ainda estiverem ativas são canceladas.

Supertarefas: usado para agrupar várias tarefas relacionadas, bem como para introduzir a noção de modularização e para definir pré-condições e exceções comuns para um grupo de tarefas. Elas possuem as características tanto do *workflow* como das tarefas. Como o *workflow*, elas são internamente decompostas em tarefas.

Multitarefas: em vários *workflows* é necessário definir um conjunto de tarefas que vão executar o mesmo trabalho em paralelo, mas serão alocadas para diferentes agentes. Cada multitarefa está associada com um valor j indicando o número de tarefas que tornam-se aptas a serem executadas quando o antecessor termina. Também é possível especificar quando uma multitarefa deve ser considerada completa, pela associação de um valor limiar chamado quorum. Quando o número de componentes finalizados alcançam o quorum, a multitarefa também é terminada, e o sucessor torna-se apto a ser executado. Finalizações subsequentes de componentes não terão efeito.

Agentes: um *workflow* é povoado por tarefas, tipos, variáveis, e também por pessoas. Esses são chamados de agentes do *workflow*.

Para exemplificar a linguagem de especificação proposta, será utilizado o seguinte exemplo (Figura 2.14): um aluno, ao se matricular na disciplina de estágio, deve, no momento da matrícula, apresentar seu plano de estágio. O funcionário que efetua a matrícula distribui o plano para um número de professores que ele julgar ideal (não existe nenhum critério). Cada professor deve dar uma nota de 1 a 10 ao plano de

estágio, caso o aluno, na soma das três notas obtiver soma igual ou superior a 21, o plano de estágio é aprovado, do contrário, o aluno é reprovado. Em ambas situações, o aluno é comunicado do resultado.

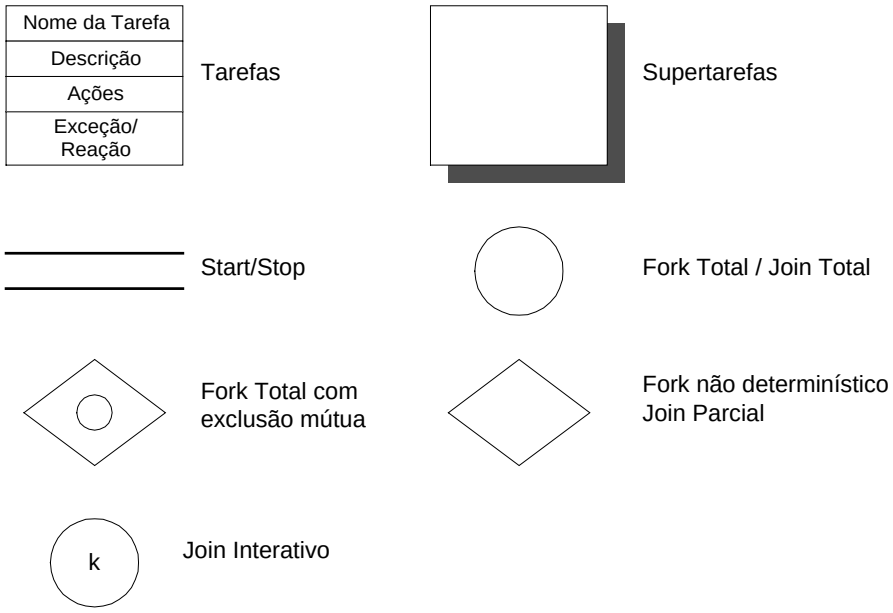


FIGURA 2.13 - Símbolos usados na Linguagem de Especificação

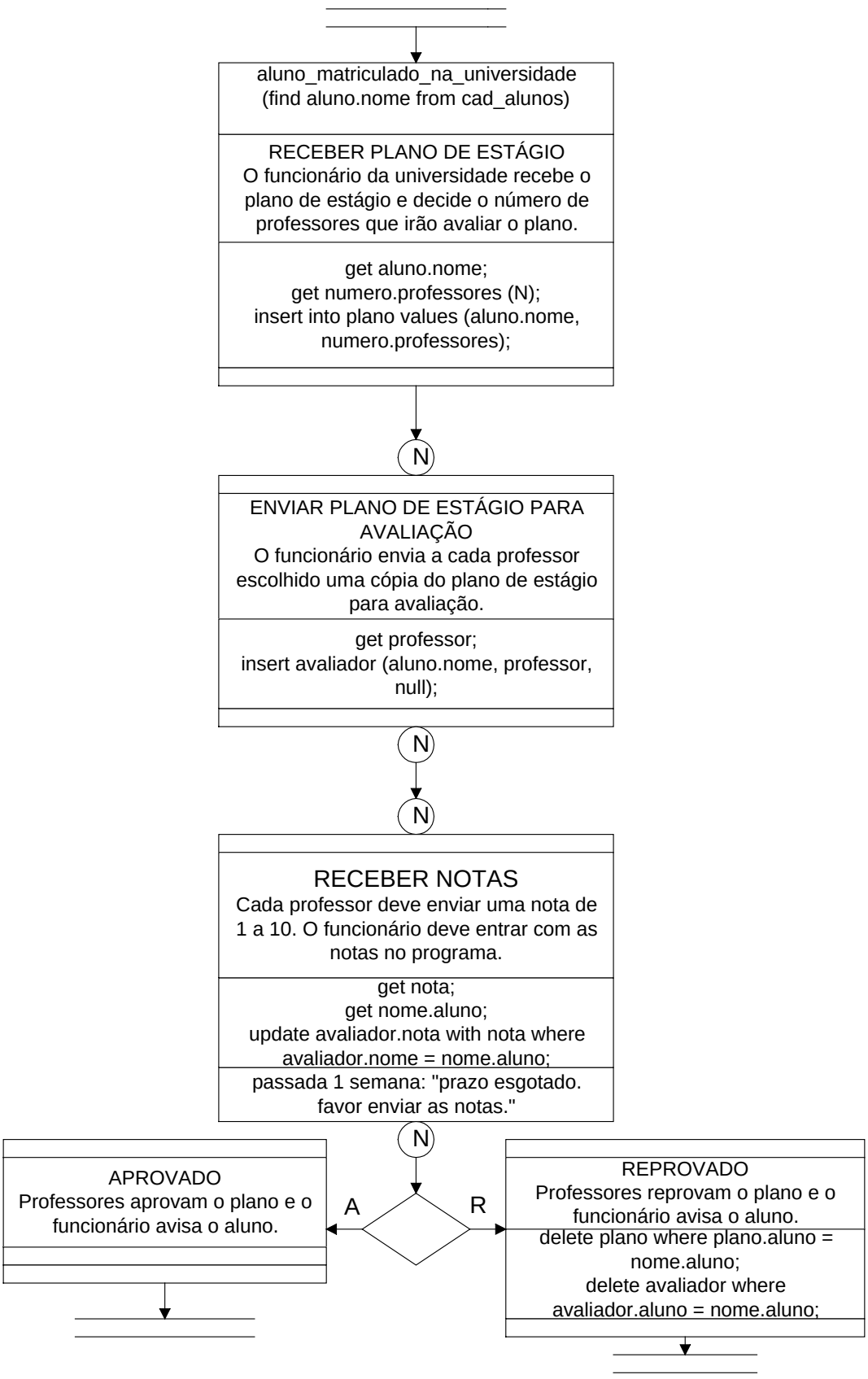


FIGURA 2.14 - *Workflow* da matrícula de um aluno no estágio

2.10.2.4 Redes de Petri

Redes de Petri têm sido usadas para modelar sistemas num variado domínio de aplicações [AAL95b].

Sistemas podem ser modelados por Redes de Petri de acordo com duas abordagens diferentes, mas, profundamente relacionadas: a abordagem F, cujo conceito fundamental é o fenômeno (estado ou transição), e a abordagem C/E, cujos elementos fundamentais são condição e evento. A relação entre as abordagens está no fato de que fenômenos tipo estado são classificados em classes que recebem a denominação condição, e os fenômenos tipo transição são classificados em classes que recebem a denominação evento. A abordagem F serve apenas como base conceitual para melhor compreensão da abordagem C/E que é aquela destinada à aplicação na prática [HEU90].

Uma Rede de Petri, normalmente representada através de um gráfico, é composta por elementos de dois tipos: lugares, indicados no gráfico através de círculos, e conexões (transições) entre lugares, indicadas no gráfico através de retângulos ligados por setas aos círculos representativos aos lugares conectados. Cada lugar de uma rede é considerado como um depósito de conteúdo variável, que pode ou não estar marcado, o que é representado graficamente pela presença ou ausência de uma ficha (*token*) no lugar. A idéia é que cada marca sirva para modelar um estado do sistema, servindo uma marcação da rede para modelar um estado global.

A possibilidade de ocorrência de uma alteração em uma marcação é dada pela regra de habilitação. Uma alteração está habilitada frente a uma marcação, quando:

- suas marcas de entrada, caso a conexão as tenha, estão presentes;
- suas marcas de saída, caso as tenha, estão ausentes.

Um pressuposto da abordagem C/E é o de que usando-se apenas três tipos básicos de relações causais entre eventos, pode-se descrever qualquer sistema [HEU90].

- **seqüência**: diz-se que dois eventos estão em seqüência, quando, para que um evento ocorra, é necessário que um outro evento tenha ocorrido;

- **concorrência**: diz-se que dois eventos são independentes, quando eles não possuem condições de entrada, nem de saída comum;

- **opção**: diz-se que dois eventos são opcionais, quando eles possuem condições de entrada comuns ou condições de saída comuns, e quando existem estados no modelo, nos quais ambos eventos se encontram habilitados.

Segundo [AAL95a] , o conceito chave do *workflow* (gerenciamento do *workflow*) é a tarefa. Tarefa é uma parcela de trabalho para ser feito por um ou mais recursos num intervalo de tempo pré-determinado. Uma tarefa pode ser: um exame clínico por um médico, a digitação de uma carta pela secretária, ou a computação de uma equação numérica por um computador.

Já um procedimento, segundo [AAL95a], é um conjunto ordenado de atividades de controle, tarefas, recursos e sub-procedimentos. Uma atividade de controle especifica o caminho a ser seguido dentro de um procedimento e também sincroniza tarefas.

Do ponto de vista do sistema de gerência do *workflow* (WFMS), a tarefa é executada externamente, isto é, o WFMS está relacionado somente no que diz respeito ao gerenciamento das tarefas, tais como: que recursos irão executar a tarefa e quais documentos serão necessários pelo recurso para executar a tarefa [AAL95a].

Mapeando estes conceitos para Redes de Petri temos em [AAL95a] que: uma tarefa corresponde a um sistema de transições p1, p2, p3, p4 e p5, e quatro lugares s1, s2, s3 e s4. Uma tarefa modelada tal como um sistema interage com um procedimento (via conector de início e conector de término), com o gerenciador de recursos (via solicita, associa e libera recurso), e com o recurso (via executa tarefa e termina tarefa) A interação é dada da seguinte forma: no início, é solicitado um recurso. Este recurso, por sua vez, é associado a tarefa. Neste ponto, a tarefa já pode ser executada. Com o fim da tarefa o recurso é liberado (Figura 2.15).

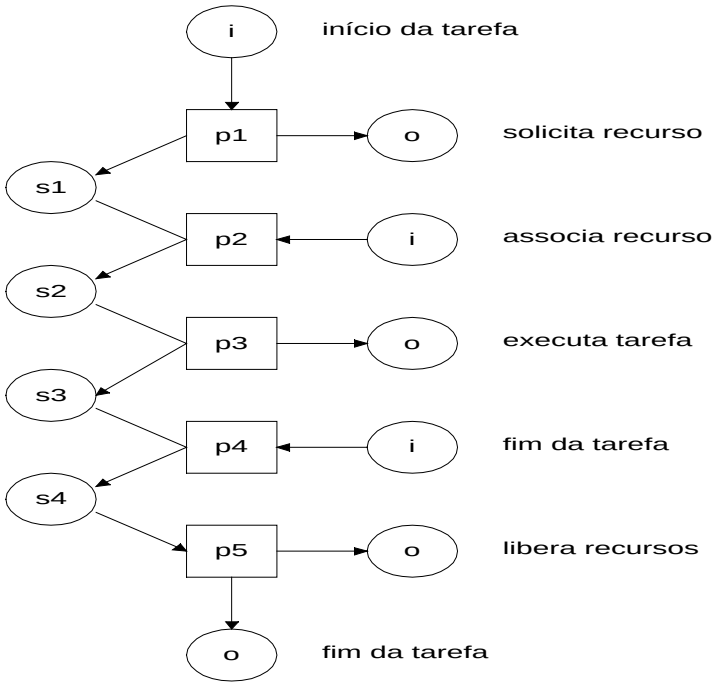


FIGURA 2.15 - Estrutura de uma tarefa

Um procedimento é também representado por um sistema. Atividades de controle são representadas por transições marcadas com um C. Tarefas e procedimentos são marcados por um T e P respectivamente. Atividades de controle não interagem com o gerenciador de recursos ou com o próprio recurso. Se uma atividade de controle requer tal interação, então ela é precedida por uma tarefa.

Dado este mapeamento, pode-se dizer que para uma tarefa (conexão) ser executada, os eventos (lugares) a ela conectados devem possuir marcas de entrada, e os eventos (lugares) aos quais ela se conecta não devem possuir marcas.

Segundo [AAL95a], o uso de Redes de Petri para se modelar *workflows* favorece a formalização de importantes conceitos. A formalização de conceitos como tarefas, procedimento, recursos, atividades de controle e jobs usando-se Redes de Petri traz semântica ao *workflow* e ao WFMS, deixando-os mais claros e compreensíveis.

Para exemplificar a modelagem do *workflow* usando redes de Petri, será utilizado o seguinte exemplo: o processo consiste no recebimento, distribuição, avaliação e julgamento de artigos submetidos a um congresso. O artigo, uma vez submetido, é distribuído para três avaliadores. Esses devem dar um parecer sobre o artigo. Quando os três pareceres forem recebidos, um julgamento deve ser feito. Independente do resultado, aceito ou rejeitado, o autor do artigo deve ser comunicado do resultado (Figura 2.16).

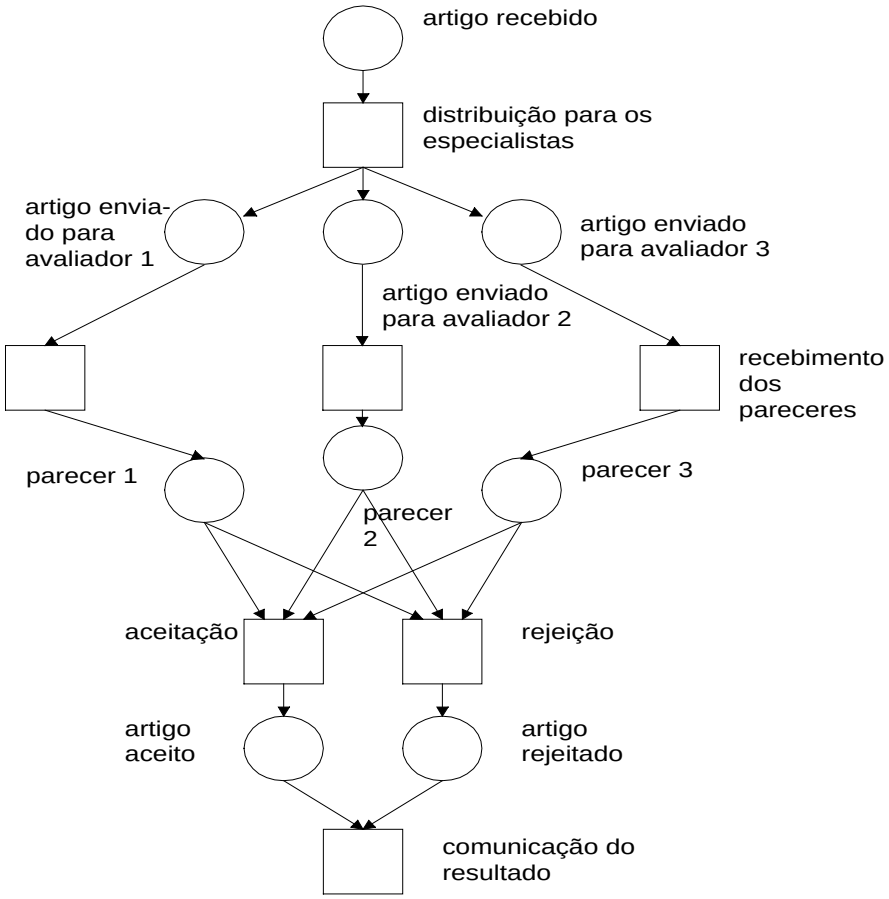


FIGURA 2.16 - Rede de Petri para seleção de artigos

Dentre as vantagens gerais da modelagem de sistemas de *workflow* com redes de Petri destacam-se a semântica formal e a modelagem explícita dos estados. Redes de Petri têm sua semântica formalmente definida, dispensando anotações ou pseudocódigos externos ao modelo. Isto traduz-se em uma grande vantagem para a modelagem de sistemas de *workflow*, pois pode-se passar o modelo obtido diretamente para um WFMS (que, evidentemente, compreenda redes de Petri), sem ambigüidades. Modelagem explícita dos estados: ao contrário de outras técnicas de modelagem de processos, o estado de uma determinada instância de *workflow* pode ser modelado

explicitamente em uma rede de Petri. Técnicas como diagramas de fluxo de dados, por exemplo, são baseadas em eventos, isto é, as transições são modeladas explicitamente enquanto os estados entre elas são modelados implicitamente. Esta diferenciação é muito importante, pois distingue claramente a habilitação da execução de uma atividade. Assim, pode-se modelar as diversas situações onde a habilitação e a execução representam momentos diferentes [AAL95a].

2.10.3 Modelagem baseada em Comunicação

Os modelos baseados em comunicação enxergam o trabalho como um conjunto de interações humanas bem definidas (laços de trabalho), representando compromissos realizados entre as pessoas envolvidas.

2.10.3.1 O Modelo *Action Workflow*

O modelo *Action Workflow* assume que o objetivo de um processo é aumentar a satisfação do cliente deste processo. Para tal, todas as ações realizadas no *workflow* são reduzidas a um conjunto limitado de atos da fala, classificados e ordenados de determinada forma, e que representa as interações possíveis entre o cliente e o provedor do processo.

Um aspecto único da técnica *Action Workflow* é a compreensão particular do conceito de trabalho. É sustentado em [FLO79], que o trabalho humano pode ser visto como uma cadeia de criações e cumprimentos de compromissos. Além disso, a coordenação eficaz das atividades é o mesmo que comunicação eficaz, e que o fluxo de trabalho pode ser acompanhado pela observação dos atos da fala empregados por estas pessoas para comunicarem-se.

Ao basear-se na comunicação entre as pessoas para representar a coordenação entre elas, este modelo tenta capturar não somente as atividades desempenhadas, mas também levantar diversos aspectos culturais da organização. [DEN95] considera que esta técnica é capaz de mostrar as inconsistências ('*breakdowns*') do trabalho, revelar as relações de confiança, e guiar a restruturação dos processos de trabalho rumo a uma maior satisfação e produtividade.

A partir da teoria dos atos da fala, e da compreensão dos autores sobre como se desenrola o processo de comunicação dentro das organizações, foi proposto um *loop* básico de trabalho, também denominado *laço de workflow*. Esse laço, segundo [MED92], representa fielmente a estrutura da comunicação humana, sendo genérico para qualquer situação de trabalho que se deseje modelar e universal, no sentido em que é independente de qualquer cultura, linguagem ou meio de comunicação utilizado para conduzi-lo. Esse laço identifica quatro fases na comunicação entre o cliente (aquele que solicita que algo seja realizado) e o provedor (aquele que executará algo para o cliente) do processo, na ordem abaixo (Figura 2.17):

1. **requisição**: o cliente requisita ao provedor que uma ação seja executada ou o provedor se oferece para executar alguma ação;
2. **negociação**: o cliente e o provedor concordam sobre a ação a ser executada e definem as condições para a satisfação do cliente (como condições pode-se citar, por exemplo, prazo de entrega, nível de qualidade e preço);
3. **execução**: a ação é realizada (pelo provedor) de acordo com os termos estabelecidos. Ao final desta fase, o provedor declara ao cliente que a tarefa está pronta;
4. **aceitação**: o cliente relata sua satisfação (ou insatisfação) com a ação realizada. Caso haja insatisfação, até o final do loop, a situação deve ser resolvida.

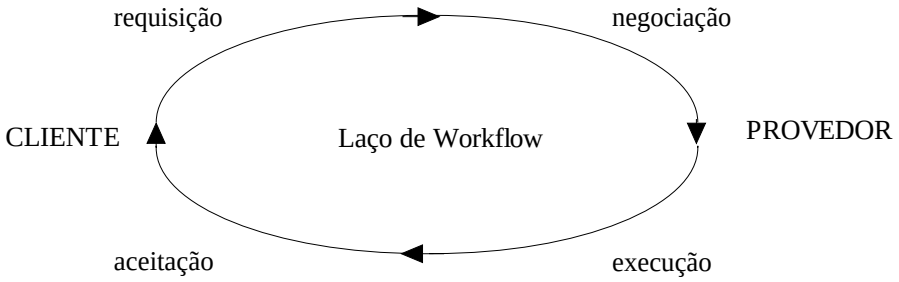


FIGURA 2.17 - Estrutura do laço de *workflow* do modelo *Action Workflow*

Cada conjunto completo destas quatro fases é denominada laço de *workflow*. Cada laço de *workflow* entre um cliente e um provedor pode ser ligado a outros laços de *workflow* para modelar um processo de negócios completo. Um provedor em um determinado laço de *workflow* pode ser um cliente em outro laço. O modelo resultante revela a rede social dentro da qual um grupo de pessoas, assumindo diversos papéis, executam um processo da organização.

2.11 Workflow Management Coalition Specification (WfMC)

Segundo [ALO95], os produtos de *workflow* comerciais estão tendo considerável sucesso mesmo não havendo uma sólida fundamentação teórica.

De acordo com [AAL95a], não há uma definição clara dos sistemas de *workflow*. Além do mais, não há um modelo conceitual para sistemas gerenciadores de *workflow*, como o modelo relacional para a maioria dos sistemas gerenciadores de banco de dados.

A falta de um padrão faz com que sistemas de fornecedores distintos não troquem informações, não trabalhem de forma cooperativa, impossibilitando, desta forma, uma maior produtividade na execução do trabalho.

Com o intuito de desenvolver especificações para softwares que irão permitir produtos gerenciadores de *workflow* trabalharem de forma interoperativa, foi criada a Workflow Management Coalition Specification (WfMC).

A WfMC propõe um modelo de interoperabilidade em [WMC96a], através do qual seria possível modelar o *workflow* em uma ferramenta qualquer que suporte este padrão, e implementá-lo em qualquer WFMS que também o entenda. Assim, um dos maiores problemas do atual mercado de *workflow* - a incompatibilidade entre os produtos de modelagem e de run-time - poderia ser solucionado. A Figura 2.18 apresenta esquematicamente como seria a relação entre as ferramentas de modelagem e os WFMS.

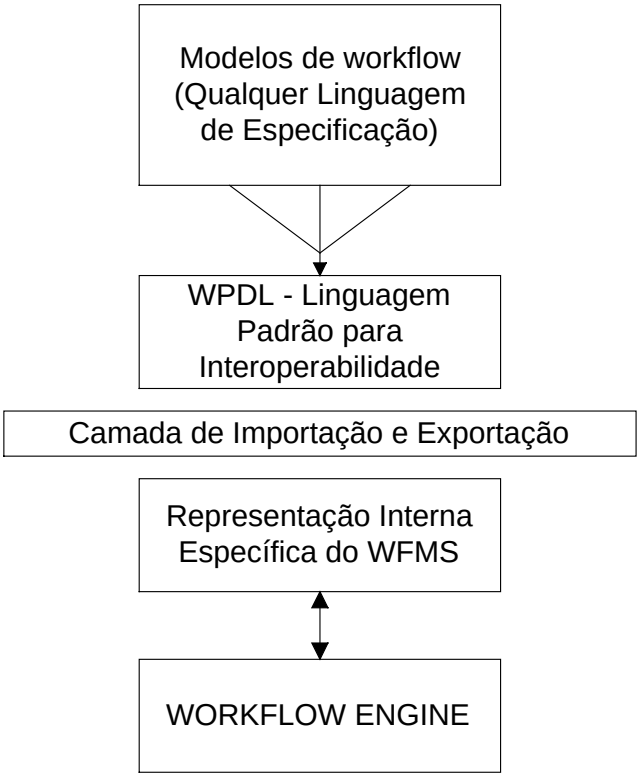


FIGURA 2.18 - Esquema do modelo de interoperabilidade da WfMC

A interoperabilidade entre estes softwares é obtida através de dois elementos:

- um meta-modelo, contendo as informações que devem ser armazenadas sobre o *workflow* (Figura 2.19);
- uma linguagem padrão para interoperabilidade, a WPDL (*Workflow Process Definition Language*).

É importante observar que tanto o meta-modelo da WfMC quanto sua linguagem associada são claramente fundamentados nos modelos baseados em atividades, já discutido neste capítulo. Este fato, sem dúvida, resulta do maior número de representantes das empresas dessa classe que das representantes dos modelos baseados em comunicação, também já discutidos neste capítulo.

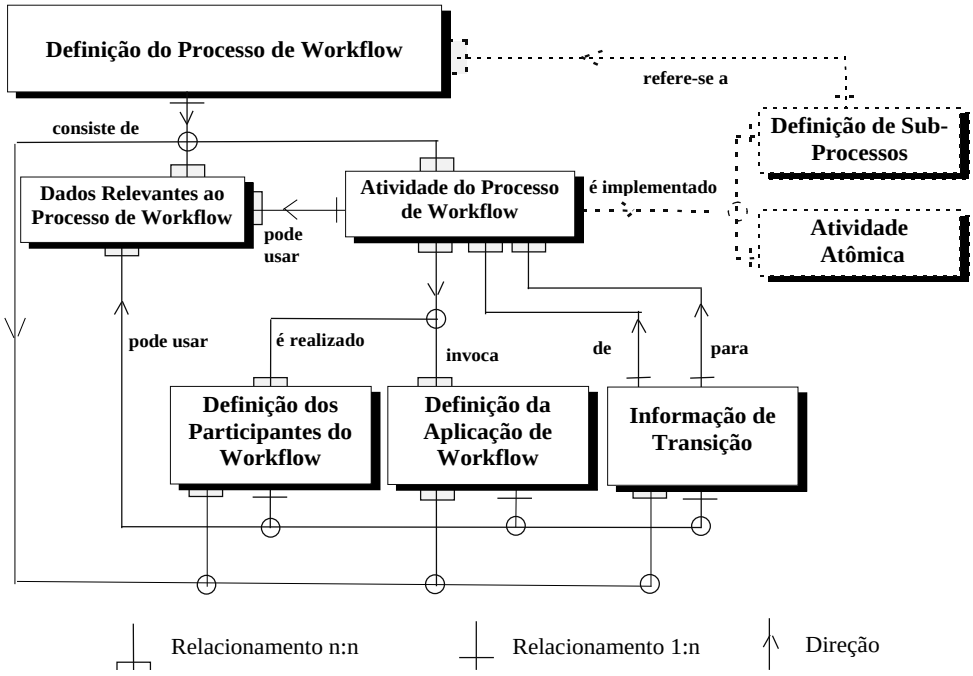


FIGURA 2.19 - Meta-modelo proposto pela WfMC [WMC96a]

Algumas das entidades do meta-modelo, estão descritas a seguir:

- **processo de *workflow***: tem como função identificar o processo de *workflow*. Nome do processo e descrição do mesmo são seus atributos principais;

- **atividade de processo de *workflow***: identifica as diversas atividades que compõem o *workflow*. O responsável pela execução da atividade, a ferramenta invocada, bem como as pré-condições e as pós-condições de sua execução, são alguns dos atributos desta entidade;

- **definição de participante de *workflow***: define os participantes e papéis que executam o processo. Estratégia de alocação de atividade, capacidade de execução e custo são alguns dos atributos desta entidade;

- **informação de transição**: armazena os relacionamentos de dependência entre as atividades: Os principais atributos são: atividades predecessoras, atividades sucessoras e condições para a transição;

- **definição de aplicação de *workflow***: descreve as ferramentas disponíveis para serem invocadas pelo WFMS. Aplicação e os parâmetros utilizados são os principais atributos desta entidade;

- **dados relevantes para o processo de *workflow***: dados gerados por uma determinada ação e que serão necessários para outras atividades a seguir, ou para uma transição, ou para uma ferramenta. Atributos principais: tipo do dado e valor.

Esta organização também definiu a seguinte terminologia para a modelagem de um *workflow*:

- *and-join*: Quando duas ou mais atividades executadas em paralelo convergem para um ponto de controle, existindo assim, um sincronismo para a execução desta atividade;

- *and-split*: Quando um ponto de controle divide-se para executar duas ou mais atividades em paralelo, existindo assim um sincronismo para o início das mesmas;

- *interation*: Uma atividade ou atividades de um *workflow* ficam em *looping* até que uma dada condição de saída seja satisfeita;

- *or-join*: Quando duas ou mais atividades do processo de *workflow* conectam-se ou convergem para uma atividade simples/única. Neste caso não existe um sincronismo de controle para cada uma das atividades convergentes;

- *or-split*: Quando uma linha simples de controle toma uma decisão sobre qual ramificação tomar quando se choca com múltiplos ramos de atividades de um processo de *workflow*;

- *sequencial routing*: Um segmento do processo de *workflow* onde as atividades são executadas seqüencialmente.

Após esta introdução à tecnologia de *workflow*, no próximo capítulo será detalhada e discutida a alocação de atividades nos sistemas de gerência de *workflow*, sendo este o assunto deste trabalho. Os problemas que envolvem esta funcionalidade, bem como os fatores que interferem na mesma, serão apresentados e exemplificados para uma maior compreensão.

3 Alocação de Atividades

Este capítulo apresenta e discute os problemas envolvidos no processo de alocação de atividades em um sistema de *workflow*, bem como uma classificação dos fatores que influenciam neste processo, sendo esta uma das contribuições deste trabalho.

3.1 Introdução

Dentre as funcionalidades que um sistema de *workflow* deve oferecer está a alocação de atividades às pessoas [DUI94]. Alocação é a tarefa de se definir quem será o responsável pela realização de uma atividade pertencente a uma dada instância de um *workflow*. Segundo [BAR95], os sistemas de *workflow* são construídos para lidar especificamente com a distribuição de atividades para a pessoa certa.

De acordo com [BAR95], este processo fica sob responsabilidade do WFMS. Porém, de uma maneira mais ampla, este processo também pode ser executado manualmente pelo gerente do *workflow*, como também pode ser executado pelos próprios atores, quando estes tiverem que tomar a iniciativa em escolher qual atividade irão executar.

Como já comentado anteriormente no segundo capítulo, uma empresa pode possuir mais de um ator apto pela execução de uma dada atividade. Para tanto, usa-se o conceito de papel. O papel nada mais é do que um cargo responsável pela execução de determinadas atividades, uma posição na hierarquia da empresa, enfim, o papel engloba atores que se encaixam no perfil preestabelecido para este papel.

Segundo [CAS95a], o uso de papel traz as seguintes vantagens para a especificação e execução do *workflow*:

- independência entre indivíduos e definição do processo;
- oferece um caminho para o balanceamento de carga de trabalho.

Quando o ator já é especificado no modelo de *workflow*, considera-se que o processo de alocação deve alocar esta atividade para este ator. Quando a atividade não possui um ator específico, e sim, um papel (conjunto de atores), entende-se que qualquer ator pode executar a atividade, resguardando-se, é claro, os demais fatores que interferem na alocação, como a carga de trabalho e aspectos temporais, por exemplo. Neste caso, considera-se que o processo de alocação pré-aloca a atividade para um dado ator. Futuramente, caso for necessário, esta atividade pode ser retirada deste ator e alocada para outro (realocação de atividades).

Entretanto, para esta alocação ocorrer, vários fatores devem ser analisados, visto que uma escolha aleatória ou equivocada, pode trazer prejuízos para a execução do processo como um todo. Conclui-se, portanto, que a alocação de atividades interfere diretamente na execução da mesma. Uma política errada de alocação pode levar a uma execução sem qualidade. Uma execução sem qualidade significa uma execução que não atingiu seus objetivos, ou seja, não resultou num produto (informação) correto e/ou

ultrapassou o prazo estipulado para o seu término, o que pode trazer prejuízos ou problemas para a empresa (Figura 3.1).

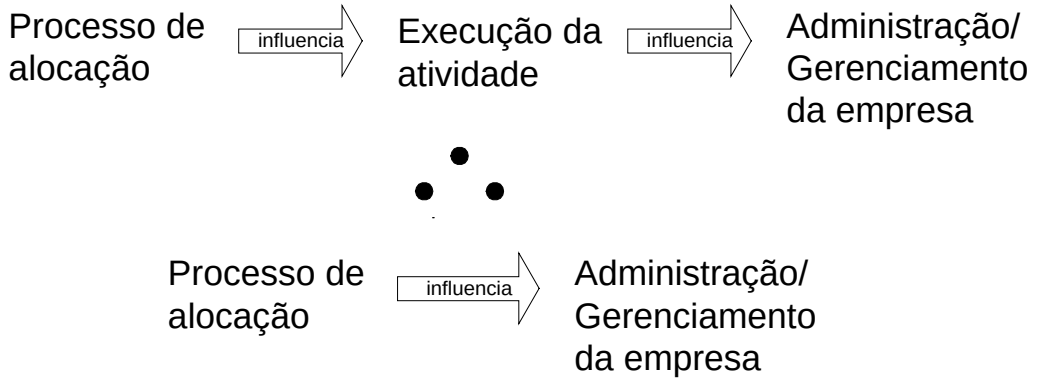


FIGURA 3.1 - Influência do Processo de Alocação

Porém, poderia ser questionado: Por que se preocupar com a alocação? Somente por causa de atrasos ou novas execuções? No primeiro capítulo deste trabalho, foi discutido sobre o novo cenário no qual as empresas estão atuando. Sabe-se que o dinamismo, a flexibilidade, a qualidade no atendimento aos clientes, rapidez nas tomadas de decisão, entre outros fatores, ditam as empresas que sobrevivem neste cenário. Sendo assim, execuções sem qualidade (que geram atrasos, por exemplo) são inadmissíveis neste cenário. Portanto, a alocação de atividades, que neste trabalho também será vista como uma atividade que auxilia o empresário na organização de suas atividades, como também uma ferramenta de auxílio nas tomadas de decisão, é de suma importância para o bom andamento da empresa, ou seja, para que a mesma se mantenha competitiva neste mundo globalizado.

Um ponto a ser destacado, diz respeito à política de alocação de atividades que a empresa emprega. Como poderá ser visto no decorrer deste capítulo, esta política de alocação deve ser definida pela empresa, ou seja, a mesma deve determinar quais fatores ou atributos deverão ser levados em consideração. O que um sistema de gerência de *workflow* deve oferecer, é a liberdade para que a empresa configure esta política da maneira que melhor lhe convier.

Antes de dar continuidade ao trabalho, uma ressalva deve ser feita. Este trabalho levará em consideração que a alocação só deve se preocupar com os atores. É claro, que existem atividades, que para serem executadas, não dependem somente dos atores, mas também de ferramentas, materiais, máquinas, etc., enfim, outros requisitos que devem estar à disposição dos atores para a execução das atividades. E ainda, existem atividades que serão executadas por máquinas, sem intervenção humana. Porém, para este trabalho, a alocação destes atributos não será considerada, assim como as atividades executadas por máquinas, devido às seguintes considerações:

- entende-se que o ator é o principal requisito para a execução de uma atividade, daí a escolha em somente se preocupar com este requisito;

- de uma maneira geral, pode-se afirmar que as situações que ocorrem com os atores, também ocorrem com os outros requisitos, porém sobre um contexto diferente, por exemplo: a impossibilidade de um funcionário não poder trabalhar devido ao fato do mesmo estar enfermo, eqüivale a dizer que uma dada máquina ou ferramenta não pode ser usada por causa de uma manutenção preventiva ou corretiva. Portanto, as mesmas considerações feitas ao ator, também poderão ser feitas aos demais requisitos.

3.2 Papel X Modelos de Especificação de *Workflow*

A noção de papel está relacionada com o processo de alocação de atividades. Antes de discutir e detalhar este processo, será feita uma comparação superficial entre os modelos de especificação de *workflow* apresentados no capítulo anterior, levando-se em consideração o conceito de papel (Tabela 3.1).

TABELA 3.1 - Comparação entre os modelos de especificação em relação ao conceito de papel

	Gatilhos	Casati	Redes de Petri	*Action Workflow*	WfMC
Definição de Papéis	sim	sim	não explicitamente	sim	sim, com alguns atributos

Dentre os modelos apresentados, somente o modelo usando-se Redes de Petri não traz explicitamente o conceito de papel, ou seja, devido ao fato deste modelo não ser voltado exclusivamente para a modelagem de workflow, o mesmo não faz nenhuma referência à necessidade ou obrigação de se definir o papel.

Já os modelos Gatilhos, Casati e *Action Workflow*, possibilitam a modelagem de papéis, porém de maneira pobre. Não existe nenhuma referência em relação a características, habilidades ou capacidades necessárias aos participantes para executarem as atividades relacionadas ao papel.

Já o modelo da WfMC mostrou ser o mais completo dos modelos cobertos por este trabalho. Este modelo inclui diversos atributos que podem ser utilizados na alocação de atividades: número máximo de itens de trabalho que o ator pode processar na unidade do tempo (capacidade), quanto custa o ator para a empresa na unidade do tempo (custo) e estratégias de alocação dinâmica de tarefas (por exemplo, FIFO).

3.3 Os estados de uma atividade segundo [CAS95a][CAS96a] e o processo de alocação

Segundo o modelo apresentado em [CAS95a][CAS96a], um atividade é iniciada quando a análise do fluxo determina que ela está pronta para execução, neste momento, a atividade passa para o estado *waiting*. Então, a atividade é associada (alocada) a um dado ator pelo WFMS, indo para o estado de *running.*

A atividade permanece em estado *running* até o usuário decidir por completá-la ou cancelá-la. Ele também pode rejeitá-la, neste caso a atividade vai para o estado *rejected*. Posteriormente, o WFMS retoma sua execução associando-a a um novo ator adequado (realocando-a). No estado *running*, a atividade também pode ir para o estado *delayed*, neste caso, basta o usuário decidir por adiar sua execução.

A Figura 3.2 mostra o diagrama de estado de uma atividade segundo [CAS95a][CAS96a]. Neste caso, como será visto mais adiante, o processo de alocação escolhe o ator no momento da execução da atividade (alocação dinâmica). Como visto anteriormente, este ator também já pode ter sido escolhido no momento da especificação (pré-alocada) do workflow, sendo que esta estratégia tira a flexibilidade do WFMS, na medida em que impede o mesmo em optar por esse ou aquele ator, com o intuito de se balancear a carga de trabalho entre eles.

As possíveis operações sobre as atividades são:

- ***dela****y*: atrasa a execução da atividade;
- ***selec****t*: seleciona uma atividade a ser executada;
- ***reject***: o usuário rejeita a execução de uma dada atividade;
- ***cancel***: a atividade é cancelada pelo usuário;
- ***complete***: a atividade é finalizada pelo usuário;
- ***modify***: o usuário decidiu por executar alguma ação, ou seja, a atividade está sendo executada.

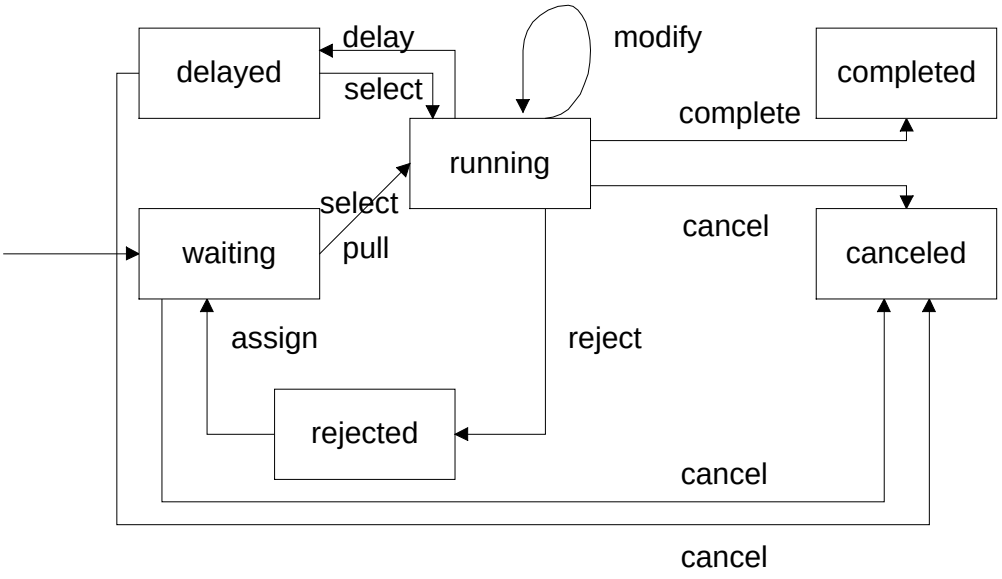


FIGURA 3.2 - Diagrama de estados de uma atividade [CAS95a][CAS96a]

3.4 Os tipos de alocação encontrados na bibliografia

Segundo [CAS95a], “a alocação de uma atividade a um agente pode ser predefinida na descrição do *workflow*, ou pode ser selecionada pelo WFMS. Neste último caso, que é mais interessante e também mais freqüente na prática, o WFMS executa a seleção aplicando regras específicas, tais como: equalização da carga de trabalho, uso da experiência dos agentes para tarefas críticas. Para tanto, os agentes são descritos pelo seu comportamento num dicionário de informações gerenciado pelo WFMS”.

Já em [BAR95a], tem-se: ”especificações padrões incluem: selecionar randomicamente um dos atores que preenchem o papel; usar o algoritmo de *round-robin*; associar a atividade ao ator com menos trabalhos alocados capaz de preencher o papel”.

[CAS96a] propõe o modelo *Push/Pull* para alocar as atividades. A diferença entre essas duas modalidades, está na maneira ativa ou passiva do usuário (ator) em relação a sua capacidade de selecionar um atividade para ser executada.

No modelo *Push*, o usuário se comporta passivamente, ou seja, quando o WFMS ou o gerente do *workflow* decidem qual usuário será responsável pela execução de uma dada atividade, a atividade é colocada na lista de trabalho deste usuário para que o mesmo a execute. Portanto, a atividade está lhe sendo imposta.

No modelo *Pull*, o usuário se comporta ativamente, ou seja, o sistema possui uma lista das atividades que devem ser executadas, e que são compartilhadas por vários usuários. Quando um usuário decide pela execução de uma dada atividade, ele deve requerê-la junto ao WFMS. Neste instante, a atividade é retirada da lista de atividades compartilhadas, e colocada na lista de trabalho do usuário. O sistema deve ter uma atenção especial para as atividades que ficam na fila e podem ter seu prazo de execução estourado. Caso isto ocorrer, as atividades devem ser impostas a algum usuário.

Segundo [BAR95], os papéis devem ser preenchidos por atores em tempo de execução. Este escalonamento pode ser feito automaticamente através de algum algoritmo ou feito de maneira manual pelo próprio ator. Vale salientar que a aceitação de uma atividade por um ator pode envolver uma negociação entre atores, sendo que tal negociação deve contar sempre com a intervenção do gerente.

A seguir, alguns comentários sobre as formas de alocação citadas na bibliografia consultada:

- **alocação imposta no momento da especificação do *workflow*** (Ex.: João será o responsável): se a idéia de inserir o conceito de papel, é obter uma independência entre indivíduos e definição do processo, esta estratégia faz com que a figura do papel desapareça, vinculando-se a atividades sempre a um mesmo ator. Esta estratégia é justificável, se a atividade realmente só puder ser executada por aquele ator. Mesmo assim, a demanda de atividades deve ser compatível com a capacidade de trabalho do ator, além do mais, o responsável pelo *workflow* deverá saber gerenciar as situações nas quais o indivíduo ficará impossibilitado de trabalhar, como no caso de um imprevisto, e o mesmo vier a faltar;

- **alocação randômica**: esta estratégia pode levar a um desbalanceamento da carga de trabalho. Como a alocação é aleatória, ou seja, sem qualquer tipo de política, determinados atores poderão acumular mais atividades do que outros, trazendo, desta forma, um desequilíbrio na carga de trabalho;

- **utilização de algoritmos para equalização da carga de trabalho**: mostra-se uma boa estratégia, visto que a carga será distribuída entre os atores de maneira a equilibrar a carga de trabalho;

- **uso de experiência dos atores para atividades críticas**: determinadas instâncias de uma atividade podem ser críticas, ou seja, casos mais complexos que os normais, onde a experiência dos atores é extremamente importante para a execução da atividade;

- **negociação**: como já visto anteriormente, a negociação pode ser uma boa saída quando existe um impasse em relação ao responsável pela execução de uma atividade. A negociação pode ser entre gerente e ator, quando o segundo, por exemplo, não possui todas as habilidades para executar uma atividade, como também pode ser entre gerentes, quando um deles requisita um ator vinculado ao outro gerente, para atuar com ele na execução de alguma atividade;

- **alocação dinâmica**: a alocação quando feita em tempo de execução, ou seja, uma vez instanciada a atividade, será alocado um ator para executá-la, mostra-se uma boa solução. Porém, duas situações não são possíveis neste tipo de alocação:
    - não se consegue estimar o término da instância do workflow, visto que não se sabe logo no início de sua instanciação quando as atividades terão seu início;
    - não se consegue antever problemas (ex.: os atores estão com carga de trabalho máxima), os mesmos só serão conhecidos quando a atividade for instanciada. Isto pode gerar atrasos e realocações, o que pode vir a prejudicar o *workflow* como um todo.

3.5 Alocação de Atividades X Tipos de *Workflow*

Como visto no capítulo 2, em [MCC92] encontra-se uma categorização dos sistemas de *workflow*. Esses sistemas são classificados em três categorias: *workflow* de produção, *workflow* ad hoc e *workflow* administrativo.

Por essa classificação, tem-se que os sistemas de *workflow* do tipo produção são bastante estruturados, e são considerados de missão crítica para a empresa, ou seja, são considerados o principal processo de negócio da organização.

Sendo de missão crítica, implica em dizer que falhas em tais sistemas interferem significativamente na empresa como um todo. Como o processo de alocação interfere diretamente nos objetivos da empresa, ou seja, um processo de alocação mau executado, sem levar em consideração os fatores citados e comentados anteriormente, com certeza acarretará problemas de atraso, uma nova execução de atividade, enfim, falhas, as quais são inadmissíveis em tais sistemas, uma vez que seus objetivos poderão não ser alcançados como o planejado.

Portanto, conclui-se que *workflows* de produção devem ter um processo de alocação rígido, ou seja, o processo de alocação deve se preocupar em alocar a pessoa certa, dentro de um intervalo de tempo hábil, para que a mesma possa executar a atividade a contento.

Se a empresa conseguir planejar a execução das instâncias de seus *workflows*, ou seja, conseguir determinar quando os mesmos devem iniciar ou terminar, o processo de alocação poderia se antecipar, alocar as atividades, e alertar o gerente do *workflow* se os prazos são possíveis de serem alcançados. Caso não sejam possíveis, pelo menos o problema será conhecido antes do mesmo acontecer, e uma solução para o problema poderá ser encontrada antes do mesmo acontecer.

Já os sistemas do tipo *Ad Hoc*, que não são de missão crítica, já aceitam falhas periódicas, sem essas atrapalharem o andamento da empresa. Pela sua definição, a alocação de atividade neste sistema pode ser feita informalmente, ou seja, a decisão pode ser tomada em tempo de execução. Caso o ator escolhido não puder imediatamente ou no momento desejado executar a atividade, o que certamente atrasará o andamento do *workflow*, este fato não trará nenhum problema maior para a empresa.

Apesar de geralmente não serem de missão crítica, visto que são menos exigentes em relação a confiabilidade, correção e integração com sistemas externos do que os *workflows* de produção, os *workflows* administrativos requerem que os executores das atividades sejam conhecidos previamente, o que os leva a ter um processo de alocação de atividades equivalente ao *workflow* de produção, ou seja, fazer uma planejamento prévio das instâncias dos *workflows* a fim de se antever problemas com a alocação.

3.6 Alocação de atividades e alguns produtos comerciais

Alguns sistemas comercias foram analisados em relação ao processo de alocação de atividades, com o intuito de se verificar como estes sistemas tratam este processo.

No ambiente MS Windows 95, existe o MS Mail, que permite o envio de mensagens com *attachment.* Esta ferramenta permite ao ator (usuário) enviar formulários ou informações como mensagens de correio eletrônico utilizando-se de sistemas de e-mail. Esta solução é adequada para sistemas do tipo *ad hoc*, onde não se existe um padrão pré-determinado de movimentação entre as pessoas. Também, não há preocupação em se verificar, pelos menos formalmente, a situação atual do destinatário, se o mesmo está ou não com uma carga de trabalho elevada, ou mesmo, se o destinatário possui habilidades para executar a atividade.

Como pode-se ver, esta ferramenta não traz o conceito de papel ou ator. Este fato é justificável visto que a mesma não é uma ferramenta voltada para a tecnologia de *workflow*.

Outro produto comercial analisado, foi o *AutoManager WorkFlow 5*, desenvolvido pela *CYCO International*. Este sistema de *workflow*, voltado para ambientes de engenharia, possui o conceito de papel e ator. A definição do responsável

quando da especificação do *workflow*, pode ser feita de duas maneiras: ou se define previamente o ator, ou se define o papel.

Como o sistema é voltado para ambientes de engenharia, onde geralmente existe um grupo trabalhando num projeto, a alocação já é praticamente definida no momento em que se define o grupo responsável pelo projeto. O sistema é bastante flexível no que tange a definição desde ou daquele ator. Fica a cargo do gerente, e não do sistema, alocar a atividade. O gerente aloca, de acordo com seus critérios, o ator que melhor lhe convier. Um detalhe, o sistema permite que o gerente visualize a carga de trabalho dos atores, suas agendas. Desta forma, ele tem uma idéia para a tomada de decisão.

O Sistema FORO [CAS96a] desenvolvido pelo projeto WIDE (*Workflow on Intelligent and Distributed Environment*), utiliza-se dos mecanismos de *PULL* e *PUSH*, já descritos anteriormente neste capítulo. O modulo responsável pela alocação de atividades usa critérios diferentes para escolher entre o conjunto de atores (agentes) que podem executar a atividade. Carga de trabalho, habilidades dos atores e prioridades são alguns dos mecanismos disponíveis. A configuração fica a cargo do gerente do *workflow*, que também pode instruir o sistema a alocar a atividade diretamente a um dado ator.

A alocação é feita dinamicamente, ou seja, quando a atividade estiver apta para execução, o processo de alocação escolhe o ator que a executará. Neste caso, o processo de alocação está usando o mecanismo *PULL*, ou seja, ele está determinando que a atividade seja executada pelo ator. No caso do processo utilizar o mecanismo *PUSH*, o ator é que escolherá a atividade, porém, também em tempo de execução.

A Tabela a seguir, faz uma comparação entre os três sistemas analisados levando-se em consideração quatro características, são elas (Tabela 3.2) :

- **conceito de papel**: existe o noção de papel no sistema, ou seja, existe uma entidade que engloba todos os atores ou entidades processadoras que possuem características ou habilidades similares;

- **alocação pré-definida**: verifica se o sistema possibilita a alocação de atividades a uma dado ator no momento da especificação do *workflow*, ou mesmo, ao invés de se alocar a atividade a um papel, alocar a atividade a uma dado ator;

- **alocação baseada em alguma política (carga de trabalho, prioridade, etc.)**: verifica se o sistema possui ou permite a criação de alguma política de alocação;

- **alocação por iniciativa do usuário**: verifica se o sistema oferece a possibilidade do ator requer uma atividade, ao invés da atividade ser atribuída para ele;

TABELA 3.2 - Comparação entre produtos comerciais

	MS Mail	*AutoManager*	FORO
Conceito de Papel	não	sim	sim

Alocação pré-definida	sim, de maneira informal	sim	sim
Alocação baseada em alguma política (carga trabalho, prioridade, etc.)	sim, de maneira informal	sim, de maneira informal	sim
Alocação por iniciativa do usuário	não	não	sim, modelo *PUSH*

3.7 Os Problemas em torno da alocação das atividades

Os problemas que giram em torno do processo de alocação de atividades podem ser divididos em dois grupos: problemas a serem resolvidos pelo processo quando da instanciação da atividade, e problemas a serem resolvidos pelo processo advindos da execução da atividade.

O primeiro grupo, o de problemas a serem resolvidos pelo processo de alocação quando da instanciação da atividade, ocorre quando este processo deve alocar uma atividade pertencente a uma instância de um dado *workflow* pela primeira vez, ou seja, logo após a sua instanciação. Nem sempre este processo é fácil, visto que nem sempre existirá um ator disponível e apto para executar a atividade no momento desejado. Neste caso, alguma decisão deverá ser tomada, ou uma atividade, executada por uma ator apto, é interrompida momentaneamente para se executar esta nova atividade, ou esta nova atividade sofrerá um atraso para o início de sua execução. Outra hipótese, seria alocar a atividade para um ator que não possuísse todas as habilidades desejadas para a execução da atividade, o que pode ser um risco, visto que ele não possui as características necessárias para uma execução com sucesso garantido.

O segundo grupo, o de problemas advindos da execução da atividade, é uma conseqüência de uma execução sem qualidade e que, conseqüentemente, não só prejudicará a instância em questão, pois irá gerar um atraso no término da mesma, como também poderá prejudicar outras instâncias. Logo, o processo de alocação deverá resolver estes problemas de atrasos ou dar condições para que o gerente tome decisões e resolva estes problemas em prol de se alcançar os objetivos da empresa.

Resolver estes problemas de atraso nada mais é do que realocar as atividades que sofrerão este atraso para novas datas e horários de início e término. Resolvendo estes problemas, tem-se condições de se alocar novas atividades sem que estas sofram um atraso embutido (possivelmente seu horário de início não seria aquele, pois já existiam atrasos e esses não foram computados) e, como um forma de antever problemas em relação ao término da instância, visto que o primeiro prazo (objetivo) não será mais alcançado, a realocação possibilita visualizar uma nova estimativa de término para as instâncias que sofrerão atrasos, como também um estimativa correta para as novas instâncias.

3.7.1 Situações que causam atrasos e interferem no processo de alocação das atividades

Como comentado anteriormente, o processo de alocação de atividades em um sistema de *workflow* interfere diretamente na execução das instâncias de um *workflow*, como também sofre influências desta execução. Caso algum problema ocorra tanto no processo de alocação, como na execução da atividade, pode-se ter um atraso na execução da instância do *workflow*, ocasionando problemas para a empresa.

Para melhor compreender estas situações de atraso, veja o exemplo ilustrado na Figura 3.3:

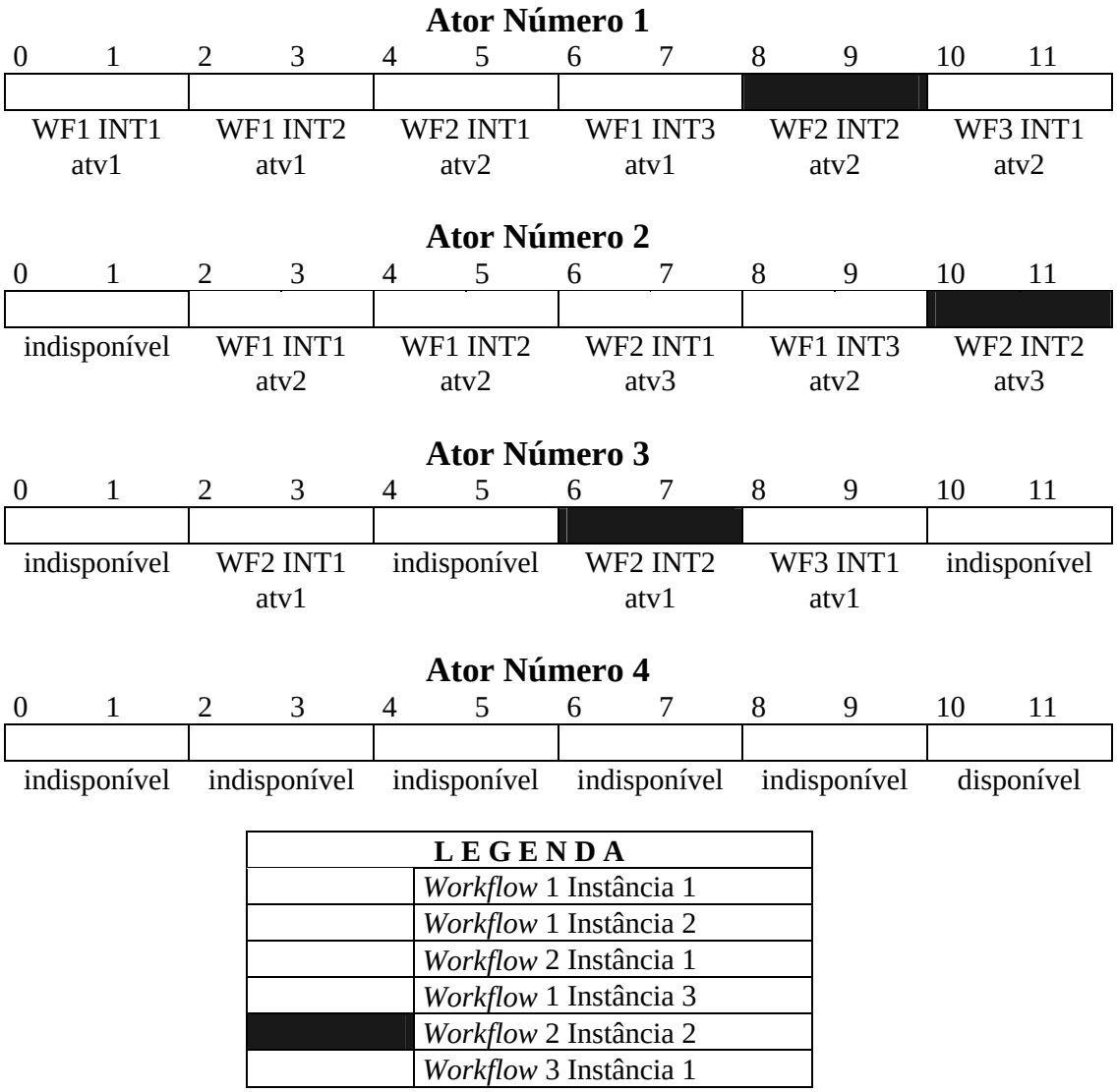


FIGURA 3.3 - Exemplo da agenda de quatro atores e suas respectivas atividades

Neste exemplo são apresentados quatro atores e suas respectivas agendas. Existem três workflows, e seis instâncias, sendo três do *workflow* 1, duas do *workflow* 2 e uma do *workflow* 3.

Como pode-se verificar, se WF1 INT1 atv1 (*workflow* 1, instância 1, atividade 1) alocada para o Ator Número 1 não estiver pronta no prazo estabelecido, conseqüentemente, a WF1 INT2 atv1, WF2 INT1 atv2, WF1 INT3 atv1, WF2 INT2 atv2 e WF3 INT1 atv2, pertencentes ao mesmo ator, irão sofrer um atraso. Da mesma forma, as atividades WF1 INT1 atv2, WF1 INT2 atv2, WF2 INT1 atv3, WF1 INT3 atv2 e WF2 INT2 atv3, pertencentes ao Ator Número 2, também irão sofrer um atraso.

Portanto, um atraso cometido pelo Ator Número 1, na atv1, pertencente a instância 1, do *workflow* 1, como repercussão causará um atraso nas demais atividades de sua agenda, e nas atividades da agenda do Ator Número 2, visto que este possui atividades sucessoras das atividades da agenda do Ator Número 1. Um atraso em uma atividade provocando atrasos em outras atividades, tem-se um atraso em cadeia. Veja, pode-se ter situações que envolverão um número n de atores, o que demonstra a complexidade do problema.

Outras situações que podem vir a ocorrer, são elas:

1. o usuário pode ter sua agenda de atividades completa para a semana inteira. Caso ele venha a faltar por alguma razão qualquer (ficou doente, pediu demissão, foi demitido), e nenhum outro ator puder assumir suas atividades (isto pode ser sempre verdade, visto que os outros atores também podem estar com suas agendas completas), atrasos irão ocorrer, prejudicando os objetivos da empresa;

2. no momento da execução de uma atividade, seu recurso (computador, impressora, teclado, terminal, software) não está 100%, não lhe permitindo, conseqüentemente, a perfeita execução da atividade. Caso não haja uma peça de reposição à disposição, um atraso na execução da atividade pode ocorrer, atrasando, conseqüentemente, o *workflow* como um todo;

3. uma sobrecarga de trabalho não prevista também pode trazer problemas para a empresa. Caso a capacidade de trabalho dos integrantes de um dado *workflow* for atingida, e uma nova instância do *workflow* surgir, atrasos poderão vir a ocorrer, acarretando possíveis problemas para a empresa;

4. um dado ator, na execução da sua atividade, não cumpriu o prazo estabelecido de entrega do trabalho por razões de incompetência ou por um mau planejamento do tempo de execução da atividade por parte do especificador do *workflow*. Conseqüentemente, as atividade que dependiam desta para iniciar também sofrerão um atraso, e assim, sucessivamente, até o término da instância do *workflow*;

5. por razões políticas da empresa, uma dada atividade foi alocada para uma pessoa que não preenchia todas habilidades desejadas. Pode-se ter um resultado não desejado, ou seja, na execução do *workflow*, por não possuir as habilidades necessárias, o produto resultante da atividade não estará de acordo com o estabelecido e deverá ser refeito, o que pode trazer atrasos para o *workflow*.

3.7.2 Situações que ocasionam um reorganização na alocação das atividades

Quando uma atividade é instanciada, uma pessoa deve ser alocada para executá-la. Este processo nem sempre é executado de maneira fácil, ou seja, nem sempre haverá uma pessoa disponível ou uma pessoa com todas as habilidades desejadas e completamente disponível para sua execução. Pelo contrário, o processo de alocação de atividades deve se basear numa série de fatores para tomar uma decisão, ou seja, alocar a atividade para a pessoa que melhor atenda todas as necessidade desejadas, e neste caso, poderá se utilizar da realocação de atividades para buscar uma otimização de tempo, recursos e das pessoas.

Além do mais, para um melhor aproveitamento do tempo, dos recursos, e das pessoas, determinados eventos que acontecem durante o desenrolar das atividades, podem fazer com que o gerente do *workflow* deseje reorganizar a distribuição de atividades, ou seja, realocar as atividades dentro de um novo cenário.

A seguir, alguns exemplos de situações que ocasionam um reorganização na alocação das atividades (realocação):

1. cancelamento de uma instância de um dado *workflow*: por uma razão qualquer, decidiu-se cancelar uma instância do *workflow* "A", composto das atividades "A1", "A2" , "A3" e "A4". Se a política da empresa for de alocar previamente as atividades, sabe-se, portanto, os responsáveis pela execução das atividades "A1", "A2" , "A3" e "A4". Sendo assim, tais atividades sairão da lista de trabalho (agenda) dos seus respectivos executores, possibilitando a alocação de atividades que ainda não foram alocadas, ou mesmo, o adiantamento de atividades já alocadas e que já podem ser executadas;

2. instanciação de um *workflow* com prioridade máxima: Caso surja uma instância com prioridade máxima, e cujo início deve ser imediato, a alocação deverá procurar uma pessoa disponível, o que nem sempre é possível. Sendo assim, alguma pessoa deverá ser interrompida para dar início às atividades da instância em questão. Para isto ser possível, o WFMS ou o gerente do *workflow*, deverá reorganizar a alocação das atividades, viabilizando uma reorganização otimizada das atividades;

3. atrasos: Por uma razão qualquer, sabe-se que uma dada atividade sofrerá um atraso. Sendo assim, deve-se reorganizar as atividades alocadas para a pessoa responsável pela atividade, bem como todas as pessoas que sofrem interferência direta ou indireta das atividades alocadas para essa pessoa, mesmo àquelas atividades não pertencentes ao mesmo *workflow*.

Este processo de reorganização das atividades alocadas (realocação de atividades), visa uma otimização da execução das tarefas (não ocorrer atrasos), bem como um aumento de produtividade, visto que as pessoas não ficarão paradas esperando o início de suas atividades.

Voltando ao exemplo da Figura 3.3, tem-se: se WF2 INT2 atv1 alocada para o Ator Número 3 tiver que ser realocada para o Ator Número 4, conseqüentemente WF2 INT 2 atv2 pertencente ao Ator Número 1 e WF2 INT2 atv3 pertencente ao Ator Número 2, deverão ser realocadas para outros atores ou alocadas para os mesmos, porém em horários posteriores, visto que as mesmas são sucessoras de WF2 INT2 atv 1, e o Ator Número 4 só possui horário disponível em horários posteriores aos horários

nos quais WF2 INT 2 atv2 e WF2 INT2 atv3 estão alocadas no momento. Uma realocação causando outras realocações, tem-se uma realocação em cadeia.

3.8 Fatores que interferem na alocação de atividades

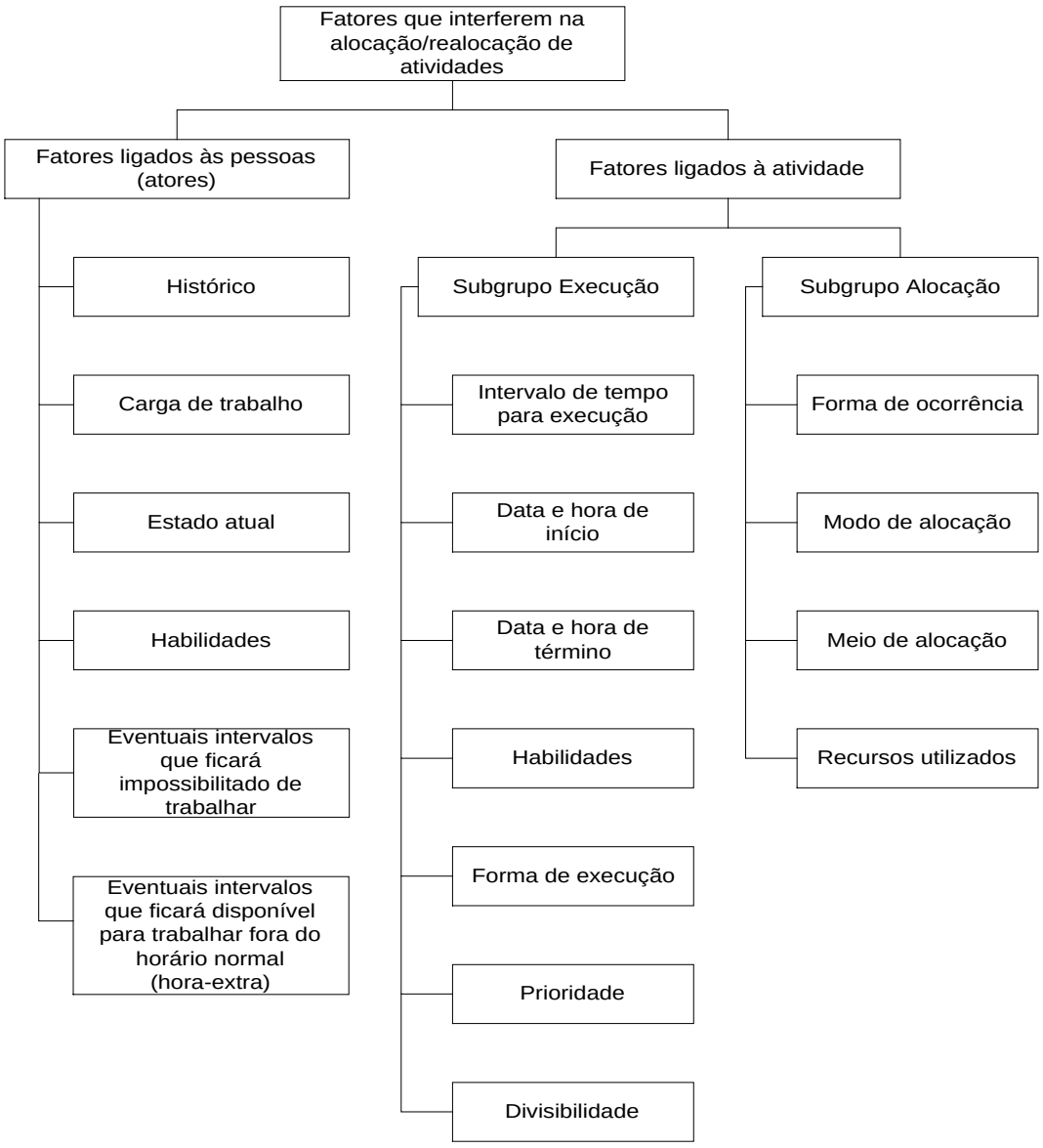


FIGURA 3.4 - Fatores que interferem na alocação de atividades

Fatores são atributos pertencentes à atividade e ao ator, sendo opções à disposição da empresa ou do gerente do *workflow* para configurar a alocação de atividades, ou seja, são opções que, uma fez configuradas, irão ditar a política de alocação de atividades da empresa.

Estes fatores podem ser divididos em dois grupos: fatores ligados aos atores aptos a executarem a atividade e fatores ligados diretamente à atividade (Figura 3.4). Os fatores ligados aos atores dizem respeito diretamente às suas condições técnicas e sobre

sua carga de trabalho. Já os fatores ligados à atividade dizem respeito aos atributos que ditam sua execução e alocação.

3.8.1 Fatores ligados às pessoas

Este grupo de fatores possibilita ao responsável pela alocação (gerente/WFMS/usuário) decidir qual a pessoa certa para executar a atividade, baseado nos dados dos possíveis atores.

Os fatores descritos a seguir, podem ser usados de acordo com a política ou necessidade da empresa.

- **histórico da performance da pessoa na execução da atividade ao longo do tempo:** este fator indica como determinada pessoa vem se comportando na execução da atividade. Dados sobre o tempo gasto, estouros de tempo (prazo), nível de dificuldade encontrado, podem ser armazenados e consultados para um maior subsídio na decisão quando da alocação;

TABELA 3.3 - Informações referentes ao histórico do ator na execução das atividades

ATOR (PESSOA)			
Atividades	**Data**	**Horário**	**Resultado**
atividade 1	01/05/97	10:00	o.k.
atividade 1	02/06/97	14:00	atrasou a atividade em 5 minutos
atividade 2	03/06/97	08:00	o.k.
atividade 3	03/06/97	10:00	atividade cancelada pelo gerente por incapacidade técnica para executá-la.

A Tabela 3.3 traz um exemplo do desempenho de uma dada pessoa na execução de atividades. Por exemplo, na execução da atividade 1, o ator cumpriu o prazo estabelecido. Já na atividade 3, a mesma foi cancelada, pois o ator não possuía capacidade técnica suficiente para executá-la.

O resultado e sua forma de representação devem ser padronizadas pela empresa, mostrando se o ator está tendo ou não uma boa performance.

- **carga de trabalho atual e futura:** a distribuição ou balanceamento de cargas é definido como sendo a atribuição de processos aos elementos de processamento, de tal maneira que não ocorra situações onde alguns elementos fiquem com carga de trabalho elevada, enquanto outros permanecem com um mínimo [LIN92]. Trazendo tal definição para o contexto do *workflow*, pode-se dizer que os processos são as atividades a serem alocadas a uma pessoa, e os elementos de processamento são os atores (pessoas) aptos a executarem a atividade. Portanto, busca-se um equilíbrio na alocação de atividades entre o conjunto de pessoas pertencentes ao papel;

- **estado atual:** uma pessoa pode possuir vários estados durante seu dia-a-dia, são eles:

    - **disponível (livre)**: no momento não está responsável por nenhuma atividade;
    - **ocupado**: está respondendo por alguma atividade;
    - **ocupado em estado de espera**: está respondendo por alguma atividade, porém, no momento, tal atividade está em estado de espera pela falta de algum recurso ou esperando algum gatilho ser disparado;
    - **ocupado com carga de trabalho**: está respondendo por alguma atividade, e já existem outras lhe esperando;
    - **indisponível**: não está respondendo por nenhuma atividade, não está ocupado e não pode tornar-se responsável por nenhuma atividade.

- **habilidades:** determinadas atividades necessitam de níveis de habilidade (níveis de características) para serem executadas. Portanto, as características dos atores e seus respectivos níveis devem ser checados pelo processo de alocação para se alocar o ator que melhor se encaixa na execução da atividade.

TABELA 3.4 - Características e habilidades do ator

ATOR (PESSOA)	
Características a serem avaliadas	**Nível**
Domínio da língua inglesa	bom
Domínio de legislação tributária	ótimo
Conhecimento de Excel	razoável
Domínio do Windows95	total

A Tabela 3.4 traz exemplos das características e o respectivos níveis apresentados pelo ator. Estes níveis devem ser estabelecidos pela empresa, de acordo com o desempenho da pessoa ao longo do tempo.

- **eventuais intervalos de tempo que a mesma poderá ficar ausente do trabalho:** datas e horários que a pessoa não estará disponível para responder por alguma atividade. Sabendo-se previamente que uma pessoa estará impossibilitada de trabalhar (ex.: férias, gravidez, cursos), conseqüentemente, ela não estará apta a executar atividades. Portanto, deve-se alocar as atividades para outras pessoas;

- **eventuais intervalos de tempo que a mesma ficará trabalhando (disponível) fora do seu horário de trabalho (hora-extra):** datas e horários que a pessoa estará disponível para responder por alguma atividade fora do seu horário normal de trabalho (ex.: hora-extra).

3.8.2 Fatores ligados à atividade

Este grupo de fatores possibilita ao responsável pela alocação (gerente/WFMS/usuário) decidir qual a melhor pessoa apta a executar a atividade, baseado nos dados da atividade.

Para tanto, este grupo é subdividido em dois subgrupos. O primeiro, está relacionado com a execução da atividade, e o segundo está relacionado com sua alocação propriamente dita, numa perspectiva administrativa.

3.8.2.1 Fatores ligados à atividade/ subgrupo execução

Fatores relacionados à sua execução dizem respeito aos atributos que a atividade carrega que ditam exclusivamente a sua execução, são eles:

- **intervalo de tempo para execução:** O tempo de execução (duração) de uma atividade, representa quanto tempo a mesma levará para ser executada sob condições normais. O mesmo pode ser classificado em:
    - **determinado**: Sabe-se previamente quanto tempo a atividade leva para ser executada.
        1. **Exato**: a atividade possui um tempo exato de execução. Ex.: 10 dias, 1 hora, 10 minutos;
        2. **Não Exato**: não se conhece o tempo exato de duração (execução), mas se conhece os tempos mínimo e máximo de execução. Ex.: de 10 a 15 dias, de 1 a 10 horas, de 5 a 20 minutos.
    - **indeterminado**: não se consegue num primeiro instante definir o tempo de execução (não existe condições de se determinar o tempo de execução), nem de forma exata, nem os tempos mínimo e máximo. Em *workflows* do tipo *ad hoc* isto pode ser possível, visto que um objetivo pode ser alcançado várias vezes e de maneiras distintas, ou seja, sem padronização.
- **data e hora de início**: indica que a atividade deve ter seu início na data e hora preestabelecidas, ou, terá sua contagem de execução iniciada naquele momento. Quando este atributo existir, a execução do *workflow* como um todo perde em flexibilidade, visto que sua execução ficará presa a determinadas condições, ou seja, a data e hora de início da atividade em questão. A data e hora de início da atividade pode ser classificada em:
    - **dependente de uma data precisa**: possui data e hora preestabelecidos. Ex.: a atividade deve começar no dia 23/11/97 às 13:00 horas;
    - **dependente de uma data em um intervalo de tempo**: possui uma data e horário de início dentro de um intervalo de tempo. Ex.: a atividade deve começar no dia 23/11/97 entre às 08:00 e 12:00 horas;
    - **independente do tempo**: não possui data e hora preestabelecidos.
- **data e hora de término**: indica que a atividade deve ter seu término até a data e hora preestabelecidos. Quando este atributo existir, a execução do *workflow* como um todo também perde em flexibilidade, visto que sua execução ficará presa a

determinadas condições, ou seja, a data e hora de término marcada para a atividade. A data e hora de início da atividade pode ser classificada em:

- **dependente de uma data precisa**: possui data e hora preestabelecidos. Ex.: a atividade deve estar pronta até o dia 23/11/97 às 13:00 horas;

- **dependente de uma data em um intervalo de tempo**: possui uma data e horário de término dentro de um intervalo de tempo. Ex.: a atividade deve estar pronta entre os dias 23/11/97 e 30/11/97;

- **independente do tempo:** não possui data e hora preestabelecidos.

- **habilidades requeridas para a execução**: determinadas atividades, para obterem um resultado satisfatório, necessitam níveis de habilidade (níveis de características) por parte dos atores, para serem executadas a contento pelos mesmos.

TABELA 3.5 - Características e habilidades requeridas pela atividade

ATIVIDADE	
Características a serem avaliadas	**Nível**
Domínio da língua inglesa	bom
Domínio de legislação tributária	ótimo
Conhecimento de Excel	razoável
Domínio do Windows95	total

A Tabela 3.5 traz exemplos de características e os respectivos níveis desejados pela atividade. Estes níveis devem ser estabelecidos pela empresa, de acordo com a necessidade apresentada.

- **forma da execução (rígida/flexível):** uma atividade é rígida no que tange a sua execução, quando a mesma deve ser executada num determinado momento. Tais atividades demandam, como o próprio nome diz, uma rigidez por parte da pessoa responsável por sua execução, visto que o seu prazo deve ser cumprido sem atrasos. Uma atividade é dita flexível em relação à sua execução, quando o intervalo de tempo para sua execução é superior ao tempo de execução. Sendo assim, fica a cargo da pessoa responsável pela sua execução, definir quando ela irá executá-la. Porém, de maneira similar à outra forma, ela também não deve sofrer atrasos;

- **prioridade**: indica a urgência com que a atividade deve ser executada. O nível de urgência pode depender da instância do *workflow*, bem como, pode ser uma constante;

- **divisibilidade da atividade**: a atividade pode ser atômica ou não atômica. Quando atômica, após ser iniciada, ela não pode ser interrompida até seu final. Quando não atômica, após ser iniciada, sua execução pode ser interrompida e reinicializada quantas vezes for necessário.

Em relação às datas e horários de início e término, a seguinte consideração deve ser feita: se uma atividade for dependente do tempo, tanto em relação ao seu início,

quanto em relação ao seu término, esta atividade irá ditar o início e o término das demais atividades pertencentes ao *workflow*. Esta afirmação é facilmente explicada: sendo o *workflow* um fluxo de atividade, onde, para uma atividade iniciar, ela deve esperar o término da sua atividade predecessora, e, uma vez finalizada esta atividade, sua atividade sucessora está apta a iniciar, conclui-se que a data e horário de término de uma atividade vai implicar na data e horário de início de sua sucessora, e assim por diante.

Portanto, se uma instância tiver seu início ou término pré-determinado, a data e horário da primeira atividade da instância ou a data de horário da última atividade da mesma, respectivamente, serão pré-determinadas e, conseqüentemente, as atividades serão dependente do tempo.

Outra situação, seria o caso de uma atividade ter seu horário determinado na especificação do *workflow*. Neste caso, a atividade seria executada sempre no mesmo horário, somente variando o dia (ex.: rotinas de *backup*, alguma rotina envolvendo a abertura de um cofre, etc.).

3.8.2.2 Fatores ligados à atividade/ subgrupo alocação

Já os fatores ligados à sua alocação, dizem respeito aos atributos que a atividade carrega que ditam a forma com que a mesma deverá ser alocada, são elas:

- **forma da ocorrência (aleatória/periódica):** um *workflow* pode ter sua execução de maneira aleatória, ou seja, não existe um momento certo do dia ou da semana para sua execução, nem o número de vezes que o mesmo poderá ser executado. Quando periódico, o *workflow* é executado todo dia ou toda semana, no mesmo horário ou em horário diferenciado, porém, sabendo-se previamente da sua execução;

- **modo de alocação:** segundo [CAS96a], o modo de alocação de atividades em um sistema de workflow pode ser feito de duas maneiras, através dos modelos *PUSH* e *PULL*:

    - ***push:*** os usuários comportam-se de maneira passiva em relação à alocação de atividades. Um gerente ou o próprio WFMS decide, baseado em alguma política da empresa, quem será o responsável pela execução da atividade. Como o usuário possui em lista das tarefa que o mesmo deve executar, quando o sistema (ou o gerente) decide quem será o responsável pela execução da atividade, ele coloca tal atividade na lista de tarefas do usuário;

    - ***pull:*** os usuários comportam-se de maneira ativa em relação à alocação de atividades, ou seja, ele mesmo decide qual atividade irá executar. O sistema possui a lista de atividades a serem executadas pelos agentes. Quando este se conecta ao sistema, as atividades compartilhadas lhe são apresentadas, e lhe cabe pegar uma tarefa para executar.

- **meio de alocação (gerente/WFMS/usuário):** existem três meios de se alocar uma atividade. No primeiro, o gerente, através do sistema gerenciador de *workflow*, decide quem será o responsável pela execução das atividades. No segundo, o próprio

WFMS, é quem decide pelo responsável. Finalmente, no terceiro meio, o próprio usuário decide pela alocação desta ou daquela atividade;

- **políticas de alocação utilizadas pelo meio de alocação (imposta dinamicamente, imposta, requerida, negociação):** os recursos utilizados pelos meios de alocação (gerente/WFMS/usuário) são as maneiras pelas quais eles efetivamente alocam as atividades às pessoas. Esses recursos só serão válidos quando o responsável pela especificação do *workflow* não definir previamente (pré-determinar) qual a pessoa que executará a atividade. Caso seja definido na especificação, a atividade seria pré-alocada para um determinado ator para qualquer instância do *workflow*, ou seja, toda vez que a atividade for executada, o ator responsável pela execução será o mesmo (o responsável pela atividade "A" será o João). Sendo assim, quando a mesma for instanciada, não haverá a preocupação de se alocar algum ator para executá-la. Porém, o *workflow* torna-se rígido (perde-se em flexibilidade de escolha), visto que a atividade está sendo pré-alocada àquele ator, não possibilitando ao gerente definir um outro ator ou mesmo balancear a carga de trabalho entre eles. As quatro políticas de alocação são:

    i. **imposta:** a atividade é imposta pelo gerente ou pelo WFMS a uma dada pessoa previamente, ou seja, não em tempo de execução;

    ii. **imposta dinamicamente**: a atividade é imposta pelo gerente ou pelo WFMS a uma dada pessoa de maneira dinâmica, ou seja, em tempo de execução;

    iii. **requerida:** a atividade fica a disposição dos atores que, ao seu modo, optam pela execução desta ou daquela atividade. Neste caso, poderia ser utilizada uma lista de trabalho geral para mostrar as atividades aptas a serem executadas. Quando um ator escolhe uma das atividades, esta sai da lista de trabalho geral, e entra na lista de trabalho do ator;

    iv. **negociação**: caso o WFMS não consiga alocar a atividade, a incumbência é repassada ao gerente. Este, por sua vez, caso também não consiga alocar a atividade a uma pessoa julgada ótima para aquela atividade, pode negociar entre os atores aptos a executarem a atividade e fazer alguma realocação de atividades para resolver o problema, ou mesmo, buscar um ator pertencente a outro workflow para executar esta atividade.

Dentro desse subgrupo, pode-se fazer combinações dos atributos. Veja as combinações a seguir:

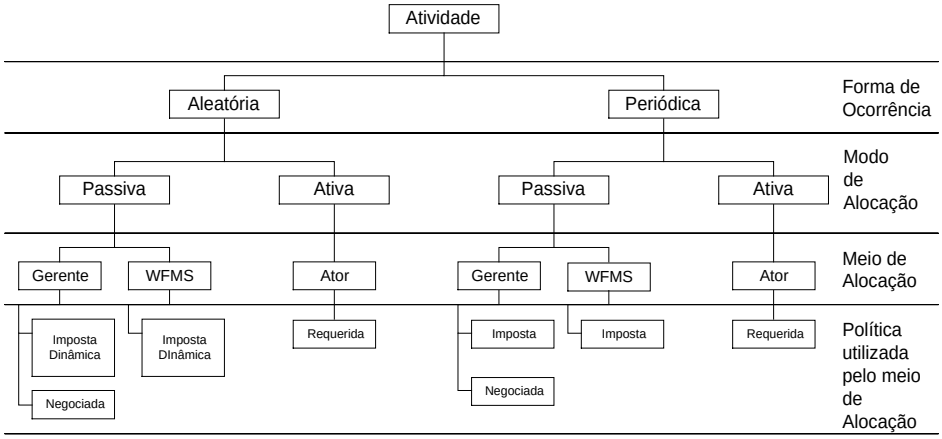


FIGURA 3.5 - Possíveis Combinações do Subgrupo Alocação

A Figura 3.5 mostra as possíveis combinações que uma atividade pode possuir de acordo com cada um dos fatores do subgrupo alocação descritos anteriormente. Algumas considerações devem ser feitas, a saber:

1. quanto ao modo de alocação, tanto a atividade instanciada de forma aleatória, como a instanciada de forma periódica, podem ser passiva ou ativa. Como o modo de alocação nos diz respeito à passividade dos usuários (gerente/WFMS/ator) em relação à alocação de atividades, nada impede de se ter as duas opções para os dois modos de alocação;

2. quanto ao meio de alocação, como as atividades ativas em relação ao modo de alocação permitem que o usuário tome a iniciativa, ou seja, decida qual a atividade que o mesmo deseja executar, o meio de alocação só pode ser o próprio ator;

3. no caso do modo de alocação ser passivo, ou seja, o ator fica a espera de que lhe seja imposta uma atividade, os meios de alocação podem ser tanto o gerente do workflow, como o próprio WFMS. Baseados na política da empresa, ambos podem decidir quem será o responsável pela execução da atividade. Neste caso, o ator não participa, pois ele é passivo ao processo, ficando a espera da atividade lhe ser imposta;

4. quando a atividade for aleatória quanto a sua forma de ocorrência, e passiva quanto ao seu modo de alocação, tanto o gerente como o WFMS, deverão, quando usando como política a imposição da atividade a um dado ator, fazê-la de maneira dinâmica, ou seja, em tempo de execução do *workflow*, visto que sua ocorrência não pode ser previamente planejada. Portanto, o recurso utilizado pelo meio de alocação será a imposição dinâmica;

5. no caso da forma de ocorrência ser periódica, ou seja, planejada, a imposição também será usada, porém de uma forma planejada, não imposta em tempo de execução. Neste caso, a política utilizada será a imposição, e não a imposição dinâmica;

6. Como política de alocação, o gerente ainda possui a negociação. Quando uma solução depender de mudanças que fogem do escopo de seu cargo (mudança da prioridade de um *workflow* o qual ele não é o gerente, trazer um ator de outro papel

para executar uma atividade que não lhe diz respeito), o gerente deve negociar com os atores ou outros gerentes para efetuar a alocação.

3.9 Como os fatores ligados às pessoas interferem na alocação

- **histórico:** pode ser usado para se averiguar a performance de um ator para uma possível alocação, bem como para alteração do nível de suas habilidades. Como comentado anteriormente, o resultado da atividade, ou seja, o tempo gasto na execução da atividade, o nível de dificuldade, podem ser consultados para se obter um maior nível de precisão na escolha desta ou daquela pessoa. Caso perceba-se que ao longo do tempo existam muitos problemas com estouro de tempo por exemplo, cabe ao especificador do workflow redefinir o tempo de execução da atividade;

- **habilidades**: com esses dados em mãos, o processo de alocação buscará o melhor ator para executar a atividade, ou seja, o ator que possui as melhores condições (habilidades) para executar a atividade. Como as habilidades ditam, juntamente como outros fatores (ex.: tempo), o grau de dificuldade da atividade, caso o processo aloque um ator que não atenda às habilidades desejadas, o processo ficará comprometido e, conseqüentemente, a execução e o resultado do mesmo;

- **eventuais intervalos de tempo que o ator poderá ficar ausente do trabalho**: este atributo contribui para a organização da empresa. Sabendo-se previamente que um ator estará impossibilitado para trabalhar, e, caso não houver outro ator para executar uma dada atividade, recursos deverão ser buscados para que seja possível a realização da mesma. Do contrário, problemas poderão surgir, como atrasos, sobrecarga de trabalho, etc.;

- **eventuais intervalos de tempo que o ator ficará trabalhando (disponível) fora de seu horário normal de trabalho:** este atributo também contribui para a organização da empresa. Sabendo-se previamente que um ator estará apto (disponível) para trabalhar fora de seu horário normal, atividades que devem ser finalizadas com urgência poderão ser alocadas para o mesmo, desta forma a empresa conseguirá manter seus compromissos e atingir seus objetivos;

- **Carga de trabalho atual e futura**: sabendo-se a carga de trabalho dos atores, tanto atual como futura, o processo de alocação buscará um equilíbrio na distribuição das atividades entre os atores.

TABELA 3.6 - Agenda dos Atores

HORÁRIO	ATOR1	ATOR2	ATOR3
08:00 10:00	atv1		atv3

10:00	12:00	atv4	atv2	atv6
14:00	16:00	atv7	atv5	
16:00	18:00	atv10	atv8	atv9
18:00	20:00		atv11	
20:00	22:00			

A Tabela 3.6 representa a agenda de três atores responsáveis pela execução de várias atividades. Caso uma nova atividade for instanciada, para quem ela será alocada? Levando-se em consideração a carga de trabalho (atividades já alocadas) dos atores, verifica-se que os atores 1 e 2 possuem um maior número de atividades (4 atividades) e, neste caso, uma maior quantidade de tempo a ser gasto na execução das atividades que o ator 3, sendo que este possui somente três atividades. Portanto, levando-se em consideração somente a carga de trabalho, a atividade será alocada para o ator 3.

- **Estado atual do ator**: o estado atual do ator indica sua situação, ou seja, se o mesmo está ocupado ou ocioso. Levando-se em consideração somente o estado atual do ator, o processo de alocação verificará qual ator está disponível ou estará disponível para a execução da atividade instanciada dinamicamente ou planejada, respectivamente.

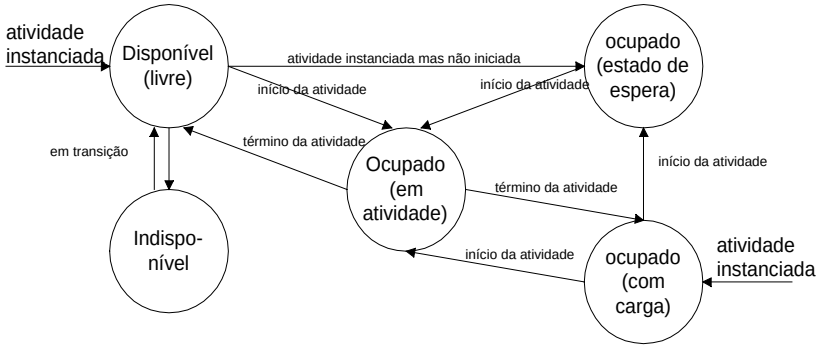


FIGURA 3.6 - Possíveis Estados de um Ator

A Figura 3.6 mostra os possíveis estados de um ator e a transição do mesmo entre esses estados. Como comentado anteriormente, o processo de alocação buscará uma pessoa disponível no momento, caso a alocação seja imediata, ou uma pessoa disponível a longo prazo, quando a alocação for planejada. Sendo assim, este processo deverá, baseado na forma de ocorrência da atividade, verificar o estado do ator. Se ele estiver disponível, poderá ser responsável pela execução, do contrário, como ele estará ocupado ou indisponível, ele não poderá executá-la.

3.10 Como os fatores ligados à atividade interferem na alocação

- **intervalo de tempo para execução**: como citado anteriormente, este intervalo representa quando tempo o ator levará para executar a atividade. Como o processo de alocação das atividades leva em consideração este intervalo, a programação das atividades do atores seguirá este intervalo para iniciar/terminar as atividades. Qualquer erro no dimensionamento deste intervalo, ou atraso na execução da atividade por parte do ator atrasará, conseqüentemente, as demais atividades. Caso

alguma das atividades for de missão crítica, este atraso trará problemas para a empresa como um todo;

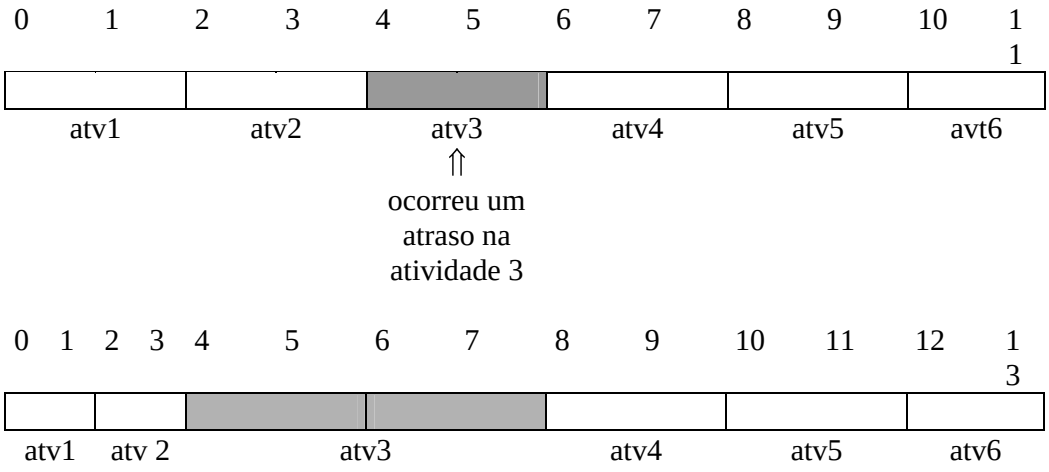


FIGURA 3.7 - Atraso no início da execução das atividades 4, 5 e 6 provocado por um atraso na execução da atividade 3

A Figura 3.7, na sua primeira parte, traz a agenda de um ator, suas respectivas atividades, e a indicação da ocorrência de um atraso na atividade 3. Já na segunda parte, a figura mostra como ficou a agenda com o atraso provocado pela atividade 3. Como pode ser verificado, as atividades 4,5 e 6, tiveram seu início prorrogados por 2 (duas) horas.

- **data e hora de início/data e hora de término**: refletem diretamente na seqüência das demais atividades do ator, bem como na seqüência do *workflow* do qual a atividade faz parte. De maneira similar ao item anterior, se esses valores forem dimensionados erroneamente, ou acontecer um atraso no início ou no término da atividade, atrasos irão acontecer nas demais atividades da instância (atrasos em cadeia);

- **forma de execução**: se a atividade for rígida em relação a sua forma de execução, nenhuma outra atividade poderá ser alocada para o ator no mesmo intervalo de tempo. No caso dela ser flexível, o ator terá mais flexibilidade para executá-la, visto que o intervalo de execução é maior que o tempo de execução. Porém, tal intervalo pode ser reduzido se sua carga de trabalho for aumentada;

TABELA 3.7 - Início/Término e tempo de execução das atividades de um ator

ATOR			
ATIVIDADE	INÍCIO	TÉRMINO	TEMPO DE EXECUÇÃO

1	12:00	20:00	1H
2	12:00	20:00	1H
3	12:00	20:00	1H
4	12:00	20:00	1H
5	12:00	20:00	1H
6	12:00	20:00	1H
7	12:00	20:00	1H

Na Tabela 3.7 pode-se ver um exemplo de atividades flexíveis. O ator possui 7 atividades com tempo de execução total de 7 horas. Sendo seu dia de 8 horas, o processo de alocação pode lhe dar mais uma atividade flexível de 1 hora ou menos, visto que ainda lhe sobra 1 hora de trabalho.

- **prioridade**: A prioridade interfere diretamente no processo de alocação, visto que uma prioridade elevada indica urgência, e nem sempre a empresa possui recursos disponíveis para suprir tal urgência. Caso não tiver, realocações poderão ser feitas. Se isso não puder ser feito, atividades poderão ser atrasadas, comprometendo o *workflow* e, conseqüentemente, a empresa;

- **tipo de atividade (divisibilidade)**: se a atividade puder ser interrompida e reinicializada (atividade não atômica), o processo de alocação pode alocar uma atividade com urgência para um ator responsável por uma atividade deste tipo, desde que ela não esteja comprometida em relação ao tempo. No caso dela ser atômica, uma interrupção não poderá ser feita, visto que a mesma não pode ser paralisada e reinicializada. A possibilidade de se poder interromper a execução da atividade torna o processo flexível, possibilitando aos responsáveis pela alocação uma alternativa a mais para possíveis impasses;

- **forma de ocorrência:** se a ocorrência for periódica (planejada), o WFMS ou o gerente do workflow poderá fazer a alocação previamente, buscando a melhor pessoa para responder pela atividade. Uma das vantagens de se ter ocorrências planejadas, é o fato de se antever algum tipo de problema, dentre os quais destacam-se:

    - falta de recursos (máquina, impressora, terminal, etc.);
    - sobrecarga de trabalho das pessoas aptas a executarem a tarefa, onde a compra de novos equipamentos ou a contratação de novos funcionários são fundamentais para a efetivação das atividades em tempo hábil.

  Sendo assim, pode-se prevenir problemas e evitar que os mesmos venham a acontecer, mantendo-se, assim, os compromissos estabelecidos. Em contrapartida, tem-se o seguinte problema: suponha que a pessoa responsável pela atividade alocada previamente não compareceu ao trabalho pelos mais variados motivos (ficou doente, simplesmente faltou, pediu demissão ou foi demitido). O que fazer? As atividades deverão ser alocadas para outras pessoas ou suspensas até a contratação de outra pessoa? E se todas outras pessoas aptas a executarem as atividades do faltoso estiverem com carga de trabalho máxima? Portanto, um imprevisto surgiu, e nem sempre haverá uma solução fácil para solucionar o impasse.

Para fazer com que a alocação de atividades não seja somente uma funcionalidade responsável pela distribuição de atividades aos participantes do *workflow*, e sim, uma funcionalidade que também ajude na organização e planejamento dos recursos da empresa (no caso, os recursos humanos), sugere-se o uso de atividades planejadas.

Embora não se possa assegurar que atividades planejadas (periódicas) terão seu êxito garantido, indiscutivelmente, o planejamento aumenta a possibilidade de se trabalhar dentro de uma estratégia preventiva.

Se a ocorrência for aleatória, não haverá condições de antever a sua ocorrência, conseqüentemente, não haverá condições de um planejamento prévio da alocação. Desta forma, após a criação da instância da atividade, deve-se determinar qual o participante responsável por sua execução. Esta estratégia, dependendo da situação que se encontra o conjunto de pessoas vinculados ao papel, pode trazer prejuízos para o processo, principalmente em relação a tempo de execução, visto que todas as pessoas num dado momento poderão estar ocupadas e sobrecarregadas, por exemplo. Um *workflow* caracterizado como sendo de missão crítica para a empresa, não pode ficar esperando um participante ficar livre ou desocupado para que o mesmo execute alguma atividade de seu processo. Já um *workflow* que não tem o fator tempo como primordial, pode usar tal estratégica. Caso acontecer algum atraso, o mesmo não deve trazer muitos problemas à empresa.

- **modo de alocação X meio de alocação**: se o modo de alocação for passiva, ou seja, imposta a uma dada pessoa, tem-se dois meios de alocação: o gerente do *workflow* e o WFMS. O primeiro, possui duas maneiras de efetivar a alocação, a imposição e a negociação. Já o segundo, o WFMS, possui somente a imposição para efetivar a alocação.

Tanto a política de imposição feita pelo WFMS, como a feita pelo gerente do *workflow*, procuram, nas políticas da empresa, alocar a melhor pessoa para a execução da atividade. Porém, nem sempre isto é possível. Carga de trabalho, eventuais faltas, atividades atrasadas, falta de recursos, são alguns problemas que tornam tal processo um tanto quanto complicado. Por isso, quando, ambos gerente e WFMS, não forem capazes de impor a uma dada pessoa uma atividade sem terem que atrapalhar o andamento de outra, o gerente, e somente o gerente, pode abrir mão de outro maneira para contornar tal problema, a negociação. Este processo consiste em alterar prazos, alterar prioridades, alterar alocações já previamente estabelecidas, trazer recursos de outros *workflows*, trazer pessoas de outros setores da empresa para dar continuidade à atividade em questão, etc. Porém, esta negociação é feita de maneira consciente, ou seja, existe uma negociação entre os gerentes para resolver determinado problema, sendo que tal solução não deve trazer prejuízos.

3.11 Onde ficam armazenadas as informações necessárias ao processo de alocação de atividades?

O processo de alocação de atividades toma uma decisão, ou seja, escolhe por esse ou por aquele ator, baseado numa série de informações. Essas informações são, por exemplo, as habilidades dos atores, a performance dos mesmos nas execuções de uma

dada atividade, o tempo em que o mesmo leva para executar uma dada atividade, sua agenda, suas ausências programadas, enfim, os atributos vistos nos itens anteriores. Para tanto, faz-se necessário que estas informações estejam armazenadas para possibilitar sua recuperação e análise por parte do processo de alocação.

Como visto no capítulo 2, [JOO95a] propõe uma arquitetura bastante genérica para sistemas de *workflow*. Conforme comentado, esta arquitetura, mostrada na Figura 2.5, é composta de cinco componentes de armazenamento (dados de definição, dados de instância, dados organizacionais, dados de gerência e dados da aplicação), três componentes ativos (processadores de interface, gerenciadores de eventos e gerenciadores de *workflow*) e três tipos de clientes que podem desempenhar três funções distintas na arquitetura: trabalhadores, desenvolvedores e gerentes.

Os componentes de armazenamento, como o próprio nome diz, são responsáveis pelo armazenamento de informações, tanto estáticas como dinâmicas, úteis tanto para o funcionamento do *workflow* como para o seu gerenciamento. São eles: dados de definição, dados de instância, dados organizacionais, dados de gerência, dados da aplicação. Dentre esses, destacam-se os dados organizacionais e os dados de gerência (ver item 2.5).

Um ponto importante a ser destacado, refere-se mais uma vez ao trabalho sendo feito pelo WfMC. Em [WMC96e], encontra-se as informações que um sistema de workflow deve capturar e armazenar a respeito das atividades que estão sendo executadas. Este documento não define como as informações devem ser armazenadas, e sim quais as informações que devem ser armazenadas.

Outra colocação feita em [WMC96e], diz respeito à forma de implementação e armazenamento das informações. Existe um conjunto de informações que devem ser implementadas e armazenadas de acordo como o padrão estabelecido pela WfMC. Em [WFC96e], tem-se: “produtos de workflow individuais serão responsáveis pela definição de quais informações e dados serão armazenados. Entretanto, todos os elementos especificados neste documento como requeridos ou mandatórios, devem ser implementados como definido”.

Sendo assim, informações extras à respeito das atividades e dos atores que não são especificadas pela WfMC podem ser armazenadas e usadas. Além disso, não existe uma imposição sobre onde (local físico) armazenar os dados.

Portanto, as informações que interferem no processo de alocação analisadas anteriormente, podem ser usadas e armazenadas pelo WFMS, desde que as informações requeridas pelo WfMC também sejam armazenadas.

Uma vez conhecidos os problemas e os fatores envolvidos no processo de alocação de atividades em um WFMS, o próximo capítulo irá apresentar uma proposta para tornar este processo mais flexível, objetivando a diminuição dos problemas já apresentados.

4 Proposta para a alocação de atividades

Este capítulo apresenta uma proposta para a alocação de atividades em um sistema de gerência de *workflow*, sendo esta proposta a principal contribuição deste trabalho. Além desta proposta, é apresentada uma série de funcionalidades que o processo de alocação deve oferecer, para que este processo não seja somente responsável pela alocação de atividades, mas também, auxilie o WFMS e a gerência da empresa na tomada de suas decisões.

4.1 Alocação de atividades "*topdown*" e "completa"

Com o intuito de fazer com que a alocação de atividades não seja somente uma funcionalidade do WFMS responsável por associar uma atividade a um ator, mas também uma ferramenta que auxilie na administração da empresa, ajudando-a no seu planejamento, organização, controle e direção, este trabalho propõe uma alocação "*topdown*" e "completa" , onde a idéia básica da mesma, está no fato de se instanciar um *workflow* se e somente se todas as suas atividades forem alocadas.

Como o ator, na pior das hipóteses, sempre terá um horário livre no final de sua agenda, esta alocação sempre será possível. Dependendo de sua carga de trabalho, esta alocação ou será mais cedo ou mais tarde, porém, sempre haverá a possibilidade de alocação.

Esta alocação só trará algum problema para o processo de alocação, quando existir uma data e horário de término determinado, e estes não puderem ser atingidos na atual conjuntura da empresa, ou seja, as agendas dos atores não permitirem que os mesmos atendam esta instância. Nestas situações, visto que a empresa deve ser flexível e dinâmica, o sistema de *workflow* através do processo de alocação deve oferecer subsídios (funcionalidades) para o gerente do *workflow* tomar decisões a fim de alcançar seus objetivos.

Estas funcionalidades devem dar condições para que o gerente altere prazos, efetue realocações, interrompa ou cancele instâncias, sempre com uma única finalidade, alcançar os objetivos traçados pela empresa.

Veja bem, o sistema de *workflow* não tomará nenhuma decisão por si só, as decisões ficarão a cargo do gerente ou da administração da empresa. O que o WFMS fará é, simplesmente, dizer quais instâncias de quais *workflows* sairão perdendo em função da execução de uma nova instância estratégica para a empresa (prioritária), por exemplo.

Sendo assim, existe a necessidade de se estabelecer prioridades para as instâncias do *workflow*, desta forma, torna-se possível classificar as necessidades da empresa em ordem de prioridade, ou seja, atender esta ou aquela instância, e, se for o caso, este ou aquele cliente. A capacidade de priorizar o trabalho é uma função-chave dos WFMS. Os primeiros sistemas baseavam-se no modelo FIFO (*first-in-first-out*), ou seja, o trabalho é entregue ao participante na ordem exata de sua introdução no WFMS.

Esta estratégia não se mostrou ideal, visto que determinadas instâncias de um *workflow* possuem uma prioridade superior às demais. Por exemplo, o processo de compra de materiais para a produção na semana seguinte provavelmente deve ter prioridade sobre outra cuja produção é referente ao próximo mês.

Como a prioridade é uma fator subjetivo, ou seja, cada empresa possui suas prioridades, e além do mais, a prioridade de hoje pode não ser a mesma de amanhã, fica claro que a prioridade deve ser definida pela empresa de acordo com suas necessidades, objetivos e responsabilidades.

Como vantagens deste modelo de alocação, tem-se:

- **antever problemas**: como a idéia é alocar todas as atividades ou nenhuma delas, alguns problemas (falta de recursos, sobrecarga de trabalho, o não cumprimento de um prazo estabelecido para término, etc.) já serão conhecidos. Conhecendo-se antecipadamente os problemas, a busca por soluções também pode ser feita antecipadamente, visando a não ocorrência dos mesmos;

- **estimar prazo de término**: alocando-se previamente as atividades, torna-se possível estimar a data e horário de término da instância. Isto é muito útil, tanto para a empresa, como para o cliente. Por parte da empresa, esta estimativa mostra que a mesma possui uma qualidade no planejamento de suas atividades. Por parte do cliente, este consegue se organizar melhor, visto que já possui uma estimativa de quando poderá contar como o produto/serviço a ser produzido/realizado pela instância;

- **agendar antecipadamente as atividades para os atores**: com as atividades agendadas antecipadamente, os atores possuem uma idéia de sua carga de trabalho e de sua responsabilidade sobre as instâncias que o seu trabalho interfere. Portanto, pretende-se que o mesmo venha a ter mais responsabilidade em cumprir os prazos de início e término das atividades, para que não haja problemas para a empresa como um todo.

Esta forma de alocação ("*topdown*" e "completa") é mais indicada para os tipos de *workflow* administrativo e de produção. O primeiro pela característica de se conhecer previamente o responsável pela execução das atividades, e o segundo pelo planejamento antecipado das atividades. Como já foi comentado, um planejamento prévio pode evitar problemas que possam, de uma forma ou de outra, retardar o andamento do processo. Sendo este tipo de *workflow* de missão crítica, tal fato é inconcebível. Uma análise prévia pode resolver esses tipos de problemas, possibilitando uma realocação de atividades, não somente em relação ao responsável pela execução, como também em relação a datas e demais recursos solicitados.

Entretanto, esta estratégia, também pode trazer problemas, uma vez que quando foi prevista a alocação, a pessoa estava em condições de executá-la no momento desejado, mas, por um motivo qualquer (a pessoa não veio trabalhar, pediu demissão, etc.), o ator ficou impossibilitado de executar a atividade, e o processo pode parar e ficar esperando uma solução, o que também pode trazer prejuízos para o processo e, por conseguinte, para a empresa. Porém, vivendo neste mundo dinâmico e cheio de

imprevisto, a empresa tem que ser flexível e rápida na tomada de decisões, e é isto que o WFMS deve oferecer à empresa através do processo de alocação de atividades, funcionalidades de se realocar atividades, interromper ou atrasar instâncias menos importantes em prol de suas prioridades, como citado anteriormente.

Sendo assim, a idéia seria tentar alocar todas as atividades como primeira tentativa de alocação. Uma vez instanciada a atividade e suas pré-condições satisfeitas, ela mudaria de status, e passaria a estar pronta para execução (ativa). Caso contrário, no momento de tornar-se apta para execução, o ator previamente alocado não estiver disponível, o WFMS (através do processo de alocação) entraria em ação, e tentaria alocar um outro ator para executá-la.

Caso não seja possível alocar algum ator, o WFMS passaria o controle da alocação desta atividade ao gerente do *workflow*, que, baseado nas listas de trabalho de todos os participantes, utilizaria de funcionalidades fornecidas pelo WFMS através do processo de alocação para reorganizar a alocação das atividades entre os participantes, ou negociaria com outros atores pertencentes a outros papéis a possibilidade desses executarem temporariamente as atividades.

Como já discutido no capítulo 2, um ator pode ser responsável pela execução de uma ou mais atividades pertencentes a mais de um *workflow*. Por exemplo, ele pode ser comprador e digitador no processo de compra de materiais, como também conferencista, no processo de conferência das mercadorias quando da chegada das mesmas. Desta forma, um atraso na execução de uma de suas atividades, seja este atraso por motivos de falta ou problemas na execução da atividade, não somente trará um atraso na execução da instância do *workflow* em questão, como também para as outras instâncias do mesmo *workflow*, e para as instâncias de outros *workflow*s que este ator porventura estiver envolvido.

Neste caso, um atraso provocado na execução de uma atividade, provocará atrasos na execução de todas as atividades deste ator, como também, atrasos em todas as instâncias que este ator participa, ou seja, em todas as atividades predecessoras das atividades que este ator é o responsável.

Atrasando as atividades predecessoras, que pertencem a outros atores, estes também atrasarão suas atividades, e assim sucessivamente. Este atraso sucessivo das atividades predecessoras, já discutido no capítulo 3, dá-se o nome de atrasos em cadeia.

Outro fato complexo da alocação de atividades ocorre quando existe a necessidade de uma realocação de atividades. O dinamismo de uma empresa, e por conseqüência, a flexibilidade que a mesma deve ter, faz com que o responsável pela gerência dos *workflows* necessite realocar atividades, ou seja, alterar o responsável pela execução desta ou daquela atividade. Por exemplo, uma atividade A1, atualmente alocada para o ator L1, deverá ser alocada para o ator L2. Porém, este ator, no intervalo de tempo no qual a atividade A1 deverá ser executada, já possui uma atividade A2 alocada. Para alocar a atividade A1 para o ator L1 neste horário, o gerente deverá alocar a atividade A2 para outro ator no mesmo horário, para que a instância a qual a atividade A2 pertença não atrase. Porém, tem-se duas situações:

- existe um ator que está livre neste intervalo, portanto mais nenhuma realocação será necessária;
- não existe um ator livre neste horário, portanto, para que esta atividade não atrase, alguma outra atividade deverá ser atrasada em prol desta. Sendo assim, pode-se ter atrasos e/ou realocações sucessivas.

O WFMS deve oferecer funcionalidades para que estas realocações e/ou atrasos em cadeia (sucessivo) sejam facilmente gerenciados, ou seja, o WFMS deve informar quais as atividades e instâncias que sofrerão atrasos e de quanto é este atraso. Desta forma, providências poderão ser tomadas em prol de diminuir estes atrasos, ou comunicar antecipadamente aos interessados (cliente, por exemplo) que haverá um atraso na finalização da instância.

4.2 Os estados de uma atividade segundo a alocação "*topdown*" e "completa

Os estados de uma atividade desde a sua instanciação até o seu término (ou finalizada ou cancelada) indicam sua situação momentânea em relação ao sistema de gerência de *workflow*.

A Figura 4.1 mostra os estados de uma atividade, na visão do ator, levando-se em consideração as colocações feitas neste trabalho, em especial, sobre a forma de alocação "*topdown*" e "completa".

De acordo com a Figura 4.1, uma vez instanciada e, para tanto, associada a um dado ator, a atividade pode permanecer no estado **alocada com pré-condições falsas**, ou no estado **alocada com pré-condições verdadeiras**. Em ambos estados, a atividade pode ir para o estado **finalizada/cancelada**, para tanto, basta o gerente do *workflow* ou alguma imposição do *workflow* (atividades concorrentes) mudá-la de estado.

Uma vez suas pré-condições verdadeiras, a atividade sai do estado **alocada com pré-condições falsas** e muda para o estado **alocada com pré-condições verdadeiras**. Neste estado, a atividade pode voltar ao estado anterior, **alocada com pré-condições falsas**, desde que alguma condição torne-se falsa, como também pode ir para o estado **em execução.**

Já no estado **em execução**, a atividade pode ser realocada para outro ator, desta forma indo para o estado **alocada com pré-condições verdadeiras**, como também pode ir para o estado **alocada com pré-condições falsas**. Em ambas situações, pode-se ter a intervenção gerencial que determine tais mudanças, como uma realocação de atividades devido a uma instância com prioridade máxima.

Também neste estado, a atividade pode ser finalizada ou cancelada, indo para o estado **finalizada/cancelada**, como também pode ser suspensa temporariamente, indo para o estado **suspensa**.

Portanto, tem-se:

- **alocada com pré-condições falsas**: a atividade já foi instanciada e alocada a um ator, porém suas pré-condições não foram satisfeitas. Portanto, ainda não pode ser executada pelo ator;
- **alocada com pré-condições verdadeiras**: a atividade já pode ser executada, pois suas pré-condições já foram satisfeitas;
- **em execução**: a atividade está sendo executada pelo ator;
- **finalizada/cancelada**: a atividade foi finalizada ou cancelada, ou pelo ator, ou pelo gerente do *workflow*;
- **suspensa**: a atividade estava em execução, e foi suspensa momentaneamente pelo ator ou por imposição do sistema.

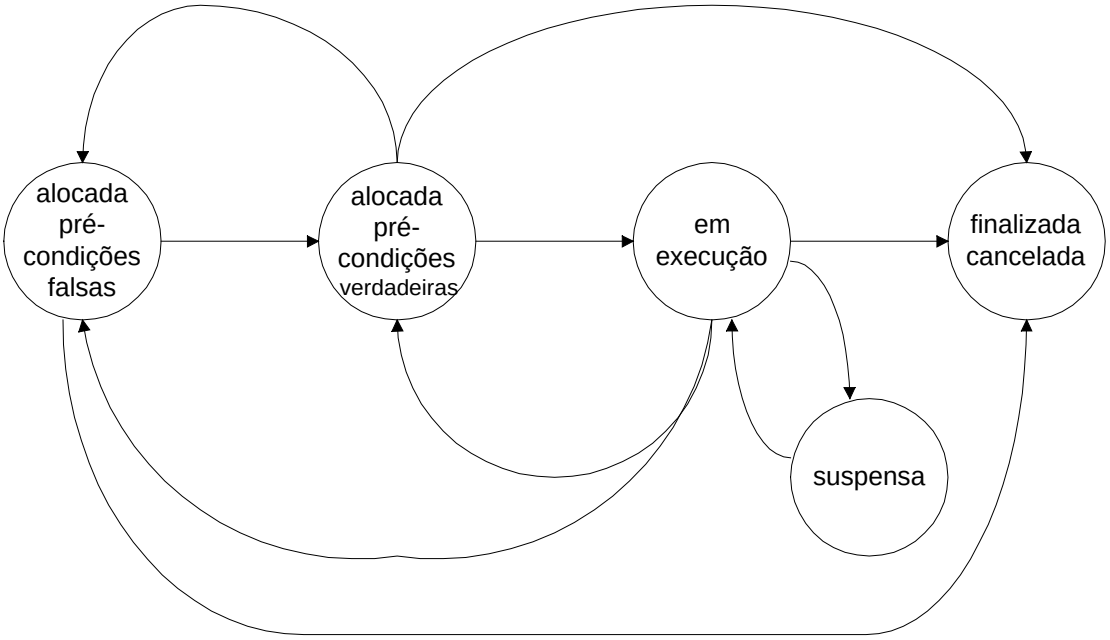


FIGURA 4.1 - Estados de uma atividade na visão do ator

As pré-condições nada mais são do que as condições necessárias para que uma atividade possa ser executada. Como pré-condições, pode-se ter n fatores, que podem variar de atividade para atividade. Como exemplo de pré-condições, tem-se:

- o término de atividades predecessoras;
- recursos disponíveis (ex.: máquinas, computadores, ferramentas);
- locais disponíveis (ex.: laboratórios, oficinas);
- ator para sua execução.

Aqui vale fazer uma ressalva. Como citado anteriormente, este modelo, apesar de alocar as atividades antecipadamente (planejadas), faz com que o WFMS certifique-se que o ator alocado esteja realmente pronto para executar a atividade. Portanto, apesar de se partir do princípio que a atividade já está alocada, logo, a alocação do ator não seria uma pré-condição para seu início, este modelo deve executar uma confirmação para verificar se o ator realmente está disponível. Caso ele não estiver disponível, o WFMS tomará as devidas providências, já discutidas anteriormente (realocações, por exemplo).

A razão de se ter criado um novo modelo de estados, e não ter usado o modelo apresentado no item 3.3 [CAS95a][CAS96a], está no fato deste último alocar uma atividade se e somente se a análise do fluxo determinar que a mesma esteja pronta para ser executada. No modelo proposto neste trabalho, isto não ocorre, pelo contrário, o *workflow* só será instanciado se todas as suas atividades forem alocadas. Para tanto, foi criado dois estados para tornar possível esta forma de alocação: o estado **alocada com pré-condições falsas** e **alocada com pré-condições verdadeiras**. Com estes dois estados, apesar da atividade já ter sido alocada para um dado ator, ela só poderá ser executada, quando suas pré-condições forem verdadeiras. Neste caso, a atividade mudaria de estado **alocada com pré-condições falsas** para **alocada com pré-condições verdadeiras**.

4.3 Alterando o fluxo de atividades

De uma maneira geral, o processo de alocação deve, baseado no fluxo de atividades, alocar as atividades entre os possíveis atores, ou permitir que os atores tomem iniciativa, e requisitem uma atividade para os mesmos executarem. A Figura 4.2 ilustra este cenário:

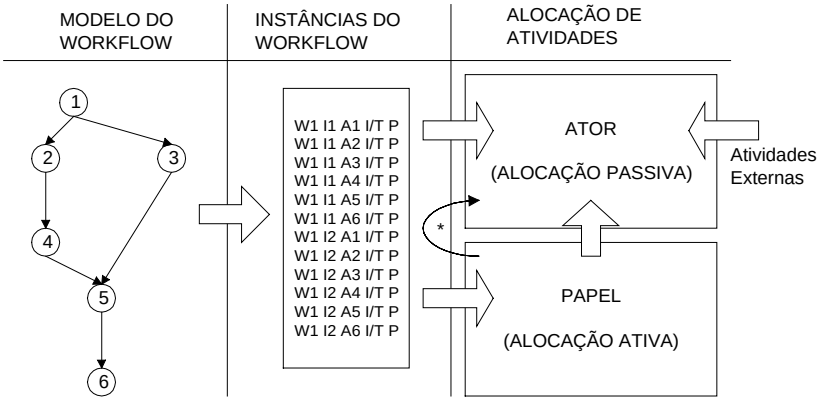


FIGURA 4.2 - Cenário do processo de alocação

A partir do modelo do *workflow* são criadas as instâncias dos *workflows* e, conseqüentemente, das atividades. A atividade pode ser alocada para um dado ator (alocação passiva), como também pode ser requerida pelo ator (alocação ativa). O processo de alocação sofre influências de atividades externas, que nada mais são do que compromissos ou atividades que não fazem parte de nenhum *workflow* da empresa, que devem ser registradas nas agendas dos atores, para que não seja alocada nenhuma atividade para o mesmo neste intervalo. Caso uma atividade que deve ser requerida por um ator, já tiver seu prazo de início espirando, esta atividade deve ser imposta a um ator. Este situação está representada na Figura 4.2 pelo asterisco (*).

Conforme comentado, o processo de alocação baseia-se em um modelo de *workflow* (fluxo de atividades) para fazer a alocação. Este processo busca a melhor forma de se alocar as atividades, para que elas alcancem os objetivos da empresa. Porém, como já visto, nem sempre este objetivo é facilmente alcançado. Há situações, por exemplo, onde uma instância de um *workflow* não pode ser iniciada imediatamente,

pois não existem atores disponíveis para executar a(s) primeira(s) atividade(s) do mesmo.

Um dos conceitos fundamentais em sistemas de gerência de *workflow* é a possibilidade de predefinir a seqüência em que as atividades serão executadas. Tipicamente, o participante do *workflow* recebe um item de trabalho e, quando termina o processamento deste, a atividade seguinte pode ser iniciada.

Especificando-se o *workflow*, tem-se a possibilidade de se saber quanto tempo este levará para ser finalizado. Para tanto, basta atribuir unidades de tempo a cada atividade a ser executada. Esta informação é de suma importância para a empresa, pois desta forma, ela obtém uma estimativa de término, podendo passá-la ao cliente, estipular prioridades, fazer sua programação de compras, enfim, a empresa consegue se organizar melhor para atingir seus objetivos.

Como se sabe, alguns sistemas de *workflow* comerciais possibilitam que as empresas simulem situações de demanda de trabalho, com o intuito de estipular o número ideal de atores para suprirem as necessidades que mais ocorrem, redefinir tempos de execução para cada atividade, projetar situações de mercado, enfim, estes sistemas possibilitam à empresa conhecer melhor seu fluxo de atividades. Porém, estas situações podem nunca ocorrer, como também, situações não simuladas podem ocorrer a qualquer momento. Desta forma, os sistemas de *workflow* devem possuir funcionalidades que possibilitem as empresas interromperem instâncias e alterarem prazos de término das mesmas, dispararem instâncias em situações que as mesmas deveriam esperar, desalocar atores, redefinirem prioridades, enfim, funcionalidades que possibilitem a empresa ser flexível e dinâmica, alterando suas prioridades e objetivos de acordo com suas necessidades.

Pelas definições apresentadas no segundo capítulo deste trabalho, pode-se verificar que um *workflow* é um conjunto de atividades, sendo que estas atividades são executadas em uma determinada ordem com o intuito de se obter um objetivo comum, sendo que estas atividades podem ser seqüenciais, paralelas ou optativas.

Porém, existem situações em que para se alcançar um mesmo objetivo, pode-se ter várias opções, ou seja, um mesmo objetivo pode ser alcançado por mais de um fluxo de atividades. Para que esta situação seja possível, a ordem de execução das atividades pode ser alterada, para tanto, uma atividade não necessariamente precisa ser executada posteriormente à antecessora.

Já que existe mais de uma seqüência válida, então por que não definir vários *workflow*s? Por que não adotar a política de um *workflow* do tipo ad hoc, onde as atividades não possuem necessariamente um fluxo pré-determinado? O que também está sendo colocado nesta situação, é o fato da empresa estimar um prazo de término para uma instância de *workflow*, estimativa que um *workflow* do tipo ad hoc não necessita e não contempla, visto que falhas periódicas são plausíveis neste tipo de *workflow*, e a alocação é feita dinamicamente, o que inviabiliza um planejamento da alocação. Portanto, incluindo-se esta possibilidade de alterar/inverter o fluxo de atividades, está sendo dado mais uma ferramenta ao gerente, ou melhor, à empresa, para esta atingir seus objetivos.

Como outra vantagem, se não a principal, situações de gargalo poderão ser minimizadas ou até evitadas, visto que ao tentar instanciar a primeira atividade, o WFMS deverá, baseado nas opções de fluxo e agenda dos atores, buscar o melhor caminho, ou seja, o caminho mais rápido para atingir o objetivo do *workflow*.

Como pode-se ver, a alteração do fluxo de atividades é uma técnica visando a otimização da alocação dos recursos (no caso, os atores). A principal dificuldade desta técnica é identificar que atividades podem ser invertidas. Esta identificação deve ficar sob responsabilidade do especificador do workflow, que deve identificar quais são as atividades que não possuem pré-condições e que podem ter sua ordem de execução alteradas.

4.3.1 Um exemplo da alteração do fluxo

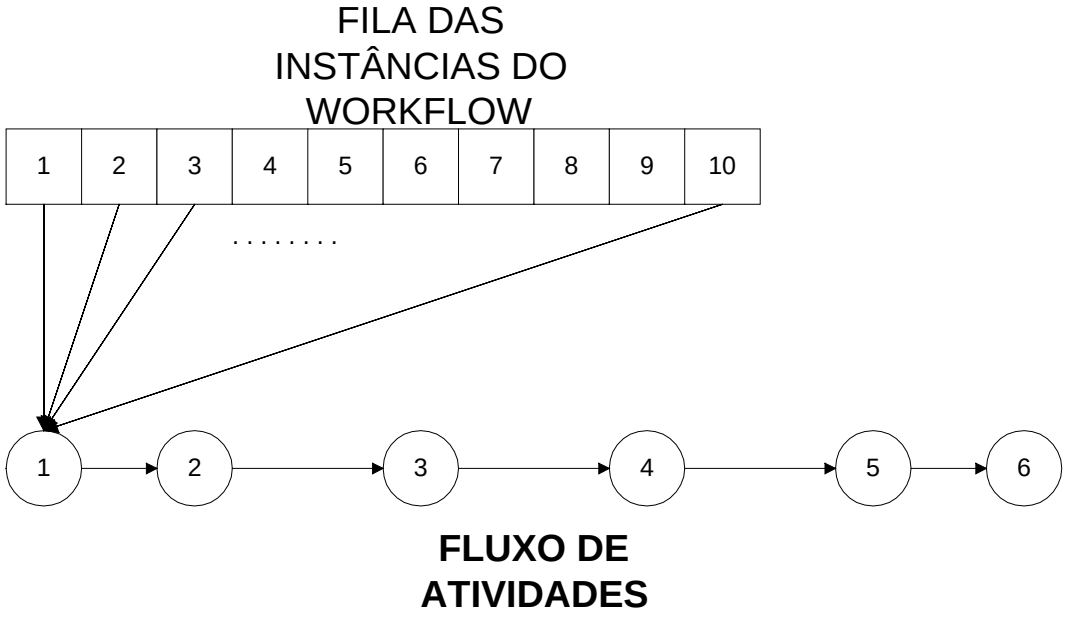


FIGURA 4.3 - Fluxo normal de atividades

A Figura 4.3 representa um *workflow* com somente um fluxo de atividades. As instâncias, numeradas de 1 a 10, deverão percorrer as atividades seqüencialmente, sempre começando pela de número 1. Levando-se em consideração que cada atividade leva 10 minutos para ser executada, e que as 10 instâncias foram disparadas ao mesmo tempo, tem-se (Tabela 4.1):

TABELA 4.1 - Comparação Instância X Tempo de Execução X Atraso para um fluxo de atividades

Instância	Tempo de execução	Atraso em relação aos 60 minutos previstos

1	60	0
2	70	10
3	80	20
4	90	30
5	100	40
6	110	50
7	120	60
8	130	70
9	140	80
10	150	90

Pela Tabela 4.1, pode-se ver que a instância 7, por exemplo, demora 120 minutos para ser executada, sendo que 60 minutos, ela fica na fila de espera para começar a ser executada. Na média, cada instância demorou 105 minutos para ser executada.

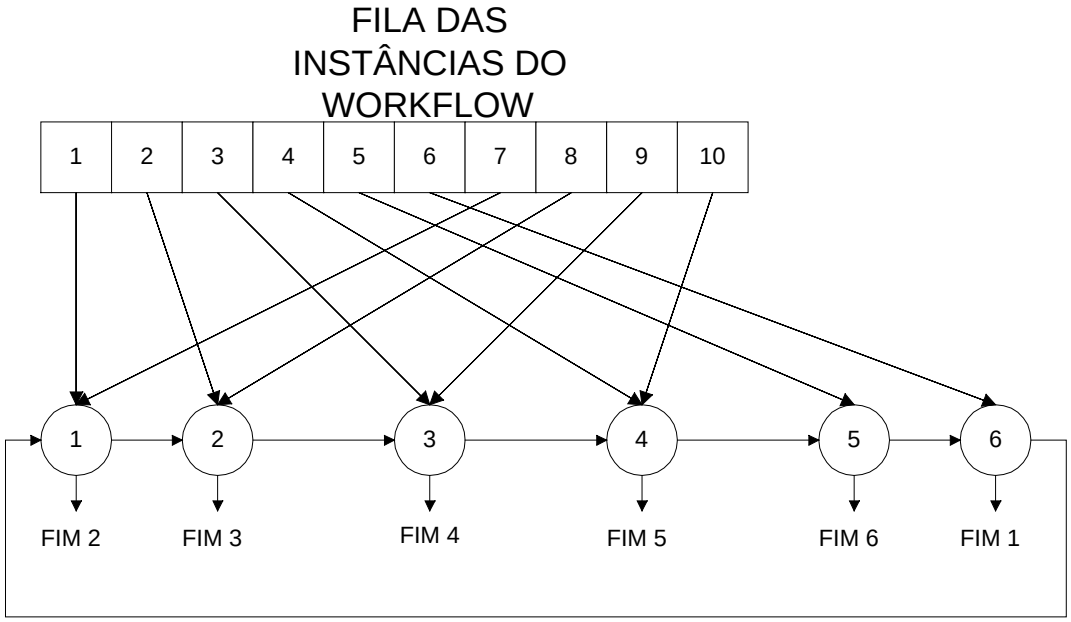


FIGURA 4.4 - Vários fluxos de atividades

A Figura 4.4 representa um *workflow* com seis fluxos de atividades. As instâncias, numeradas de 1 a 10, deverão percorrer as atividades seqüencialmente, independente da primeira a ser executada. Também levando-se em consideração que cada atividade leva 10 minutos para ser executada, e que as 10 instâncias foram disparadas ao mesmo tempo, tem-se (Tabela 4.2):

TABELA 4.2 - Comparação Instância X Tempo de Execução X Atraso para seis fluxos de atividades

Instância	Tempo de execução	Atraso em relação aos

		60 minutos previstos
1	90	30
2	90	30
3	90	30
4	90	30
5	100	40
6	100	40
7	100	40
8	100	40
9	100	40
10	100	40

Pela Tabela 4.2, pode-se ver que a instância 7, por exemplo, demora 100 minutos para ser executada, sendo que 40 minutos, ela fica na fila de espera para começar a ser executada. Na média, cada instância demorou 96 minutos para ser executada.

Comparando-se as duas estratégias, verifica-se que na média as instâncias executadas com somente um fluxo, demoraram em média 105 minutos para serem executadas. Já as instâncias que foram executadas com 6 tipos de fluxo, obtiveram uma média de 96 minutos de execução.

Portanto, caso exista mais de um fluxo para se alcançar o mesmo objetivo, a estratégia de se utilizar tais fluxos para diminuir a média de tempo de execução mostra-se uma boa opção.

Exemplo:

- as atividades a serem executas pelos mecânicos na revisão padrão de um carro são padronizadas pela oficina. Hipoteticamente, a revisão consta das seguintes atividades, que devem ser executas na seguinte ordem: regular motor, checar bateria, checar freios, checar faróis, checar amortecedores, checar nível do óleo, checar alinhamento e balanceamento. Com essa seqüência definida, com o tempo gasto na execução de cada uma destas atividades e com a agenda dos atores, o WFMS consegue alocar as atividades levando-se em consideração a política de alocação da empresa, e mais, consegue estimar quando a revisão estará pronta. Porém, neste caso, nada impede que o fluxo de atividades seja alterado, ou seja, outros fluxos seja criados para a revisão de um automóvel.

4.4 Utilizando Redes PERT/CPM

Um fluxo de trabalho, que é chamado simplesmente de *workflow* aqui neste texto, é composto de atividades. A realização de uma atividade consome recursos materiais, humanos, financeiros, tempo, espaço, etc. A programação de projetos abrange traçar as necessidades de recursos e antecipar o desenvolvimento das tarefas a fim de se atingir o horizonte do projeto. As técnicas para programar projetos incluem os

tradicionais gráficos de Gantt e técnicas de rede. Os gráficos de Gantt são facilmente entendidos e atualizados, mas não refletem a inter-relação entre recursos ou conexões de precedência entre as atividades do projeto. As técnicas de rede, tais como CPM e PERT, embora mais complexas, mostram conexões de precedência e produzem valiosas informações comerciais para melhor uso dos recursos.

Um diagrama de rede é um modelo matemático que usa círculos pequenos (nódulos) ligados por elos ou ramos (arcos) para representar conexões de precedência.

O método de caminho crítico (CPM e PERT) são técnicas de rede para avaliar um sistema em termos de atividades e eventos que devem ser completados numa seqüência especificada, a fim de realizar uma meta. Algumas atividades podem ser realizadas simultaneamente, ao passo que outras exigem de precedência.

A Figura 4.5 ilustra, num diagrama de rede, parte das atividades de trabalho necessárias para montagem de um carro. Por exemplo, colocar a tampa do tanque de combustível (atividade 3-4) só será iniciada depois de se colocar as rodas do lado direito (atividade 2-3), indicando uma precedência entre as atividades 3-4 e 2-3.

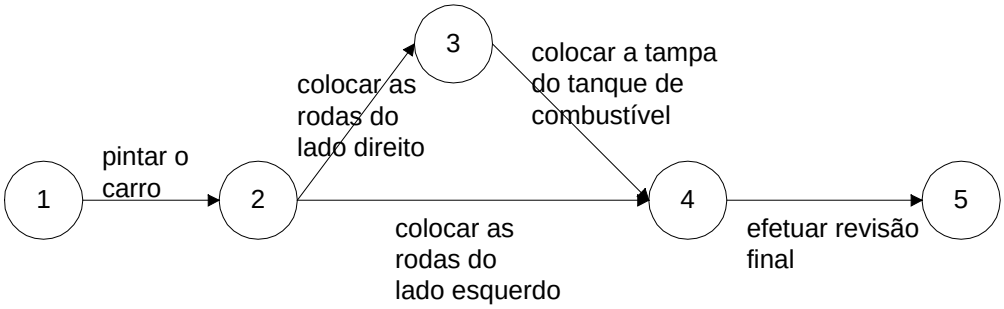
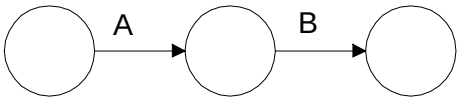
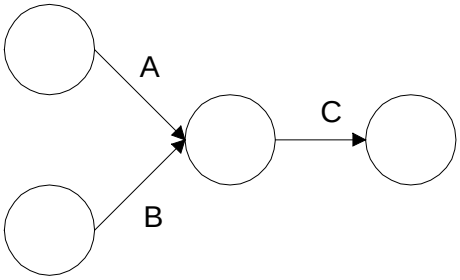


FIGURA 4.5 - Exemplo de uma rede com cinco atividades

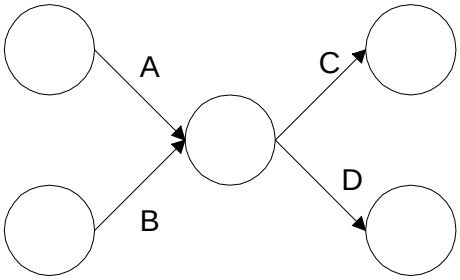
Um resumo das seqüências de diagrama de rede é ilustrado na Figura 4.6.



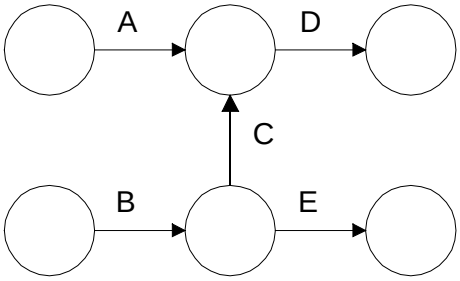
Atividade A deve ser completada antes da atividade B começar.



As atividades A e B podem ocorrer ao mesmo tempo, mas ambas precisam ser completadas antes de C começar.



As atividades A e B precisam ambas ser completadas antes de C e D começar, mas C pode começar independente de D ou vice-versa.



As atividades A e B podem ocorrer ao mesmo tempo, mas ambas precisam ser completadas antes que D possa começar. A atividade C, com linha tracejada é uma atividade imitativa que tem precedência de duração zero.

FIGURA 4.6 - Situações contempladas pelas redes PERT/CPM

4.4.1 Método de Caminho Crítico (CPM)

Segundo [MON87], as etapas envolvidas na execução CPM são:

1. defina o projeto em termos de atividade e eventos;
2. construa um diagrama de rede mostrando as relações de precedência;
3. desenvolva a estimativa de duração de cada atividade;
4. compute a exigência de tempo para cada caminho da rede;
5. mude os recursos para otimizar a consecução de objetivos.

No exemplo da Figura 4.5, tem-se as seguintes estimativas de duração das atividades (Tabela 4.3):

TABELA 4.3 - Duração das Atividades

Atividade	Duração
1-2	10 horas
2-3	2 horas
2-4	2 horas
3-4	1 hora
4-5	4 horas

Para calcular a exigência de tempo para cada caminho da rede, basta somar os tempos de cada atividade pertencente ao caminho. Sendo assim, tem-se (Tabela 4.4):

TABELA 4.4 - Duração dos Caminhos

Via	Tempo
A: 1-2-3-4-5	17 horas
B: 1-2-4-5	16 horas

Sendo o caminho crítico, o caminho cuja soma dos tempos de execução das atividades é o maior, tem-se que o caminho A (1-2-3-4-5) é o caminho crítico (17 horas). Os demais caminhos terão excesso de tempo (ou folga), e a folga associada com qualquer caminho é simplesmente a diferença entre o tempo do caminho crítico e o tempo para o caminho dado.

Logo, o caminho B, composto das atividades 1-2-4-5, possui uma folga de 1 hora. Isto quer dizer, que a atividade 2-4, que não faz parte do caminho crítico, pode sofrer um atraso de até 1 hora, que não comprometerá a execução do trabalho como um todo.

4.4.2 Primeiras e Últimas Datas das Atividades

A administração das atividades de um projeto é, às vezes, útil para saber quanto antes ou quanto depois uma atividade individual pode ser iniciada ou terminada sem afetar a data de conclusão programada do projeto total. Quatro símbolos são comumente usados para designar as primeiras e últimas datas das atividades.

1. PDI - Primeira Data de Início de uma atividade. A hipótese é a de que todas as atividades precedentes iniciam na PDI;
2. PDT - Primeira Data de Término de uma atividade. A hipótese é a de que a atividade começa na sua PDI e leva o tempo esperado t. Assim, PDT = PDI + t;
3. UDT - Última Data de Término de uma atividade. A hipótese é a de que as atividades sucessivas levam seu tempo esperado;
4. UDI - Última Data de Início de uma atividade. UDI = UDT - t.

PDI e PDT são calculadas numa seqüência da esquerda para a direita (às vezes chamada “um passo à frente”). A PDI de uma atividade é a soma dos tempos de todas as

atividades precedentes naquele caminho. Onde dois caminhos convergem a um nódulo, o caminho de tempo mais longo prevalece.

Os tempos posteriores são computados ao contrário. Começam com o tempo crítico, ou final, e subtraem cada atividade precedente até a atividade especificada.

O cálculo das datas da Figura 4.5 é dado na tabela a seguir (Tabela 4.5):

TABELA 4.5 - Cálculo das datas da Figura 4.5

Atividade	Duração	PDI	UDI	PDT	UDT	Folga
1-2	10	0	0	10	10	0
2-3	2	10	10	12	12	0
2-4	2	10	11	12	13	1
3-4	1	12	12	13	13	0
4-5	4	13	13	17	17	0

4.4.3 *Workflow* X Redes CPM

Nem sempre é possível alocar uma atividade a um dado ator no intervalo de tempo desejado: ou o ator já possui uma atividade neste intervalo, ou parte do intervalo já está reservado para outra atividade, ou já se sabe previamente que o ator não estará disponível neste intervalo.

O fato de ter que se alocar uma atividade no intervalo de tempo especificado em função de não se atrasar a execução do *workflow* como um todo, torna o processo de alocação em um sistema de gerência de *workflow* muito rígido.

Com o intuito de tornar este processo um pouco mais flexível, este trabalho sugere o uso de redes CPM para o cálculo das datas e horários de início e término das atividades. A idéia é tornar o processo de alocação mais flexível em relação ao início das atividades, ou seja, tornar possível que uma atividade seja executada em outro intervalo de tempo, sem prejuízo para a instância do *workflow* como um todo. Em outras palavras, atividades que não fazem parte do caminho crítico e tiverem uma folga no que tange ao seu início e término, podem ter sua alocação facilitada quando, numa primeira tentativa, não puderem ser alocadas no intervalo de tempo desejado. Neste caso, a atividade poderia ser alocada em um intervalo posterior, sem prejudicar a instância, desde que este intervalo posterior não ultrapasse a folga que lhe é atribuída.

Atividades optativas (decisórias), ou seja, atividades que serão executadas dependendo do resultado da(s) atividade(s) antecessora(s), e atividades que somente se conhecerá o número de instâncias a serem executas em tempo de execução, deverão ser tratadas como atividades normais, ou seja, tratadas como atividades que serão executadas de uma forma ou de outra. No caso das atividades que somente se conhecerá o número de instâncias a serem executadas em tempo de execução, a especificação do *workflow* deve fornecer o número máximo de instâncias que podem ocorrer.

Para o cálculo da PDI, PDT, UDT e UDI, as atividades decisórias serão mapeadas como se fossem atividades obrigatórias. Isto não traz nenhum prejuízo, pois, levando-se em consideração que esta decisão só será tomada em tempo de execução, e, deseja-se saber uma estimativa de término para a instância do *workflow* antes do início da instância, então deve-se passar uma estimativa de término mais pessimista, ou seja, para o cálculo do caminho crítico, todas as opções serão analisadas.

As atividades que serão executadas um número x de vezes (onde x é o número de instâncias da atividade, e só se conhece este número em tempo de execução), também não representam nenhum problema, pois as mesmas serão executadas em paralelo, e, como possuem o mesmo tempo de execução, para o cálculo do caminho crítico isto não fará diferença.

Como exemplo, será analisada a Figura 4.7, que apresenta as seguintes situações:

1. A atividade 2-7 será executada um número indeterminado de vezes, porém em paralelo;
2. Ou a atividade 4-5 ou a atividade 4-6 será executada, esta decisão será tomada também em tempo de execução;
3. Conseqüentemente, ou será executada a atividade 5-8, ou a atividade 6-8.

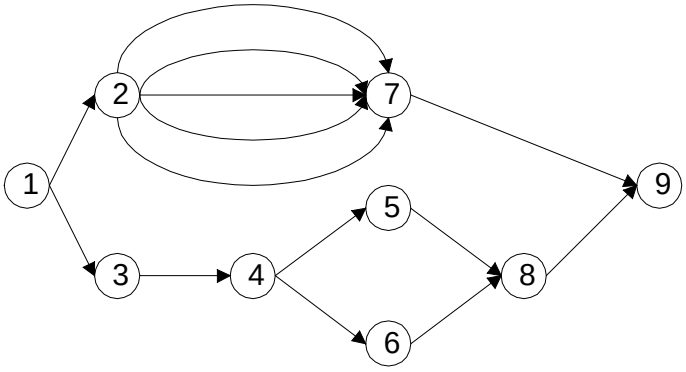


FIGURA 4.7 - Exemplo para calcular as primeiras e últimas datas

Para o exemplo da Figura 4.7, tem-se as seguintes estimativas de duração das atividades (Tabela 4.6):

TABELA 4.6 - Atividades e suas respectivas durações para o exemplo da Figura 4.7

Atividade	Duração
1-2	10 min
1-3	10 min
2-7	20 min
3-4	10 min
4-5	10 min
4-6	20 min
5-8	10 min
6-8	20 min
7-9	10 min
8-9	10 min

Para calcular a exigência de tempo para cada caminho da rede, basta somar os tempos de cada atividade pertencente ao caminho. Sendo assim, tem-se (Tabela 4.7):

TABELA 4.7 - Vias e os respectivos tempos

Via	Tempo
A: 1-2-7-9	40 min
B: 1-3-4-5-8-9	50 min
C: 1-3-4-6-8-9	70 min

Sendo o caminho crítico, o caminho cuja soma dos tempos de execução das atividades é o maior, tem-se que o caminho C (1-3-4-6-8-9) é o caminho crítico (70 min).

O cálculo das datas da Figura 4.7 é dado na tabela a seguir (Tabela 4.8):

TABELA 4.8 - Resultado do cálculo das PDI, UDI, PDT, UDT e Folga para o exemplo da Figura 4.7

Atividade	Duração	PDI	UDI	PDT	UDT	Folga
1-2	10	0	30	10	40	30
1-3	10	0	0	10	10	0
2-7	20	10	40	30	60	30
3-4	10	10	10	20	20	0
4-5	10	20	40	30	50	20
4-6	20	20	20	40	40	0
5-8	10	30	50	40	60	20
6-8	20	40	40	60	60	0
7-9	10	30	60	40	70	30
8-9	10	60	60	70	70	0

Pela tabela acima, pode-se verificar que a atividade 2-7, apesar de não se saber o número de vezes que a mesma deverá ser executada (em paralelo), sabe-se que a mesma

pode ter sua data de início prolongada em até 30 minutos. Isto é extremamente importante, visto que, no horário previsto inicialmente para a mesma começar, nenhum ator poderia estar disponível. Com esta folga de 30 minutos, o processo de alocação de atividades possui mais flexibilidade para trabalhar.

Também pode-se ver, que o caminho crítico é composto por uma das atividade alternativas, a 4-6. Sendo assim, a estimativa de término é dada sobre o caso mais demorado. Se no momento da execução, a atividade escolhida for a 4-5, o tempo de execução poderá ser inferior ao estimado.

Portanto, independente do número de atividades a serem executadas e das alternativas entre atividades, o cálculo do caminho crítico mostra-se uma boa técnica para ajudar no processo de alocação de atividades, permitindo uma maior flexibilidade para o mesmo.

4.4.4 Como distribuir a folga entre as atividades que não fazem parte do caminho crítico?

As atividades que não pertencerem ao caminho crítico terão uma folga em termos de tempo de início e término. Porém, esta folga não é só para uma única atividade, e sim, para as demais atividades que pertencem ao caminho e que não fazem parte do caminho crítico. Veja o exemplo anterior (Figura 4.7), a atividade 1-2 não detém sozinha a folga de 30 minutos, e sim o caminho 1-2-7-9 é quem detém esta folga. Já o caminho 1-3-4-5-8-9, que não é o crítico, mas que contém atividades que fazem parte do crítico possui 20 minutos de folga. Porém como citado anteriormente, esta folga pertence as atividades que não fazem parte do caminho crítico, portanto, neste caso, esta folga pertence as atividades 4-5 e 5-8. Isto pode ser verificado, analisando a tabela de tempos do exemplo (Tabela 4.8).

Sendo assim, como distribuir esta folga entre as atividades? Esta discussão está fora do escopo deste trabalho, porém um estudo preliminar foi feito.

A princípio, pode-se utilizar uma distribuição eqüitativa entre as atividades. Outra forma seria ponderar a distribuição, ou seja, as atividades que possuem um tempo de execução maior, usufruirão de mais folga, as que possuem menos tempo de execução, terão menos folga. Uma última sugestão, seria primeiramente tentar alocar as atividades sem a folga, aí sim, as que não fossem alocadas, repartiriam, ou eqüitativamente ou ponderosamente, a folga entre si.

4.4.5 O que fazer com as atividades alocadas que não serão executadas? Como tratar as atividades que somente será conhecido o número de execução da mesma em tempo de execução?

Se a idéia é somente instanciar um *workflow* se e somente se todas as suas atividades forem alocadas, como proceder com as atividades que não serão executadas visto que uma decisão foi tomada e tal decisão optou por outra atividade?

A atividade em questão não deve ser executada, e não será, pois suas pré-condições não deverão estar satisfeitas. Logo, o WFMS não permitirá sua execução. Porém, como a própria atividade saberá que ela não será executada, ou como informar ao ator desprezar tal atividade? A princípio, o usuário poderia pensar que a atividade predecessora está atrasada, e portanto, a sua atividade não está apta.

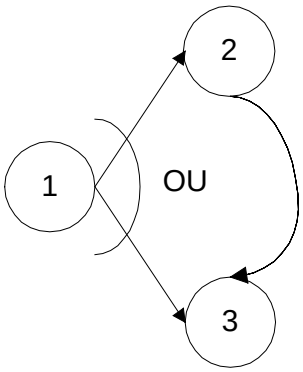
Pensando assim, o ator ficaria em estado de espera aguardando a liberação desta atividade até o seu limite de tempo se esgotar, e a próxima atividade de sua agenda, caso existir, ficar apta para execução. Caso uma atividade necessitar ser executada neste momento, ela não poderia, porque o usuário estaria falsamente ocupado, aguardando (em estado de espera) a liberação da sua atividade.

Ou mesmo, ele não poderia adiantar seu trabalho, pois poderiam existir atividades alocadas em sua agenda já aptas a serem executadas que, no momento da alocação, foram alocadas adiante, pois não havia um horário anterior disponível para sua execução.

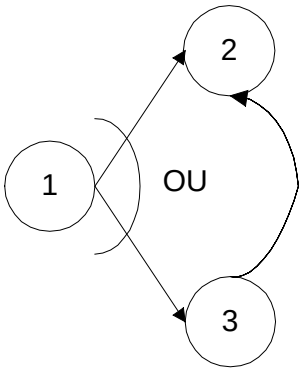
Sendo assim, algum processo deve ser disparado para cancelar esta atividade, liberando o ator para que o mesmo possa adiantar seu trabalho, ou possibilitar ao WFMS alocar alguma atividade neste horário no qual o ator ficará ocioso.

Esta discussão também não pertence ao escopo deste trabalho, porém um estudo superficial foi feito, e se obteve as seguintes propostas para o problema apresentado:

1. como já visto anteriormente, algumas técnicas de modelagem de *workflow* contemplam o uso de alarmes (uma exceção ocasionando uma ação). O alarme entra em ação para comunicar ao responsável pela atividade, que o tempo de execução da mesma está se esgotando ou já esgotou. Baseado nesta idéia de alarme, sugere-se criar um alarme para verificar se a atividade que também poderia ser executadas já iniciou, ou, a atividade que for escolhida, avisar a outra, para a mesma se auto cancelar (Figura 4.8).



A atividade 2, escolhida para ser executada, dispara um alarme e cancela a atividade 3.

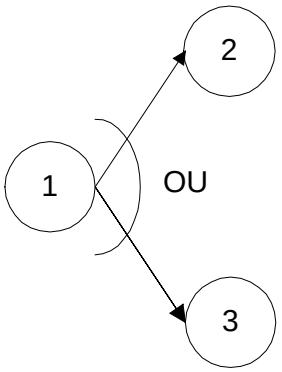


A atividade 3, de tempos em tempos, dispara um alarme para verificar se a atividade 2 já está em estado de execução. Se estiver, ela se auto cancela. Do contrário, ela se programa para disparar o alarme novamente.

FIGURA 4.8 - Cancelando atividades que não serão executadas

Para tanto, as atividades devem trazer junto aos seus atributos, as atividades que são concorrentes, ou seja, quais as atividades que serão executadas se elas não forem. Com este atributo em mãos, basta implementar um alarme que faz a verificação ou quando a atividade entra em estado de execução, ou quando esta atinge o limite de início, porém, sem estar com suas pré-condições satisfeitas.

2. uma outra solução (Figura 4.9) seria a própria atividade antecessora, após decidir por essa ou aquela atividade, cancelar as atividades que não serão executadas. Para tanto, basta a atividade antecessora conhecer as atividades predecessoras, condição que já é contemplada nas metodologias de especificação (modelagem) vistas neste trabalho.



Apartir da atividade 1, o workflow partiu para a execução da atividade 2. Automaticamente, a atividade 1 deverá cancelar a atividade 3, visto que esta não foi escolhida.

FIGURA 4.9 - Cancelando atividades a partir da atividade predecessora

Sobre as duas soluções apresentadas, vale fazer uma ressalva. Ao se cancelar a atividade ou atividades que não serão executadas, suas atividades sucessoras também

deverão ser canceladas. Só não serão canceladas, as atividades sucessoras que também tiverem outras atividades antecessoras.

4.5 Tratando atividades que podem ser executadas em locais diferentes

Como já foi comentado, independente do ramo de atuação, sejam elas indústrias ou prestadoras de serviço, as empresas buscam cada vez mais aumentar sua produtividade, aumentar a qualidade de seus produtos, alcançar uma participação maior no mercado, enfim, tornarem-se mais competitivas, flexíveis e dinâmicas neste mundo cada vez mais globalizado.

Algumas delas, em busca de alcançar seus objetivos, avançam suas fronteiras, e montam filiais em outros regiões.

Em relação ao seu processo administrativo e/ou produtivo, as empresas passam a executar parte deste processo ou todo este processo nas filiais. Quando feito totalmente, o processo pode ser igual ou sofrer alterações para se adaptar à nova realidade apresentada na região instalada, ou na região que se pretende atingir com esta filial. Quando feito parcialmente, algumas atividades deverão ser feitas na matriz ou em alguma outra filial que contemple estas atividades.

Sendo assim, surge uma nova realidade para os sistemas de gerência de *workflow*. Uma atividade, que antes era somente executada localmente, agora deve ou pode ser executada em outra localidade.

Neste novo cenário, o processo de alocação de atividades pode dispor de mais opções para efetuar a alocação de atividades. Dependendo das prioridades da empresa, uma atividade não podendo ser alocada localmente, pois os atores responsáveis por ela estão sobrecarregados de trabalho, o processo de alocação pode alocá-la remotamente, ou seja, um ator, estabelecido em outra filial, poderá tornar-se o responsável por sua execução.

Algumas considerações devem ser feitas para que esta situação se torne possível, são elas:

1. quando da especificação da atividade, deve-se especificar se a mesma pode ou não ser executada em outra localidade;
2. se for possível, estabelecer em quais filiais (levando-se em consideração que está se especificando a partir da matriz) ela pode ser executada;
3. levar em consideração que o tempo de execução pode variar de filial para filial;
4. levar em consideração o fator transporte, ou seja, o tempo gasto para se enviar e receber de volta um documento ou uma peça que será trabalhada em outra localidade.

Em relação ao transporte, este pode ser embutido na estimativa de tempo de término da instância do *workflow*, ou pode ser criada uma atividade intermediária chamada atividade de transporte, que deve ser disparada, e também deve dar

continuidade à instância do *workflow*. É claro que esta atividade terá um responsável, sempre local, e também terá um tempo de execução, que nada mais é que o próprio tempo de transporte.

Para uma melhor compreensão desta situação, analisar a Figura 4.10:

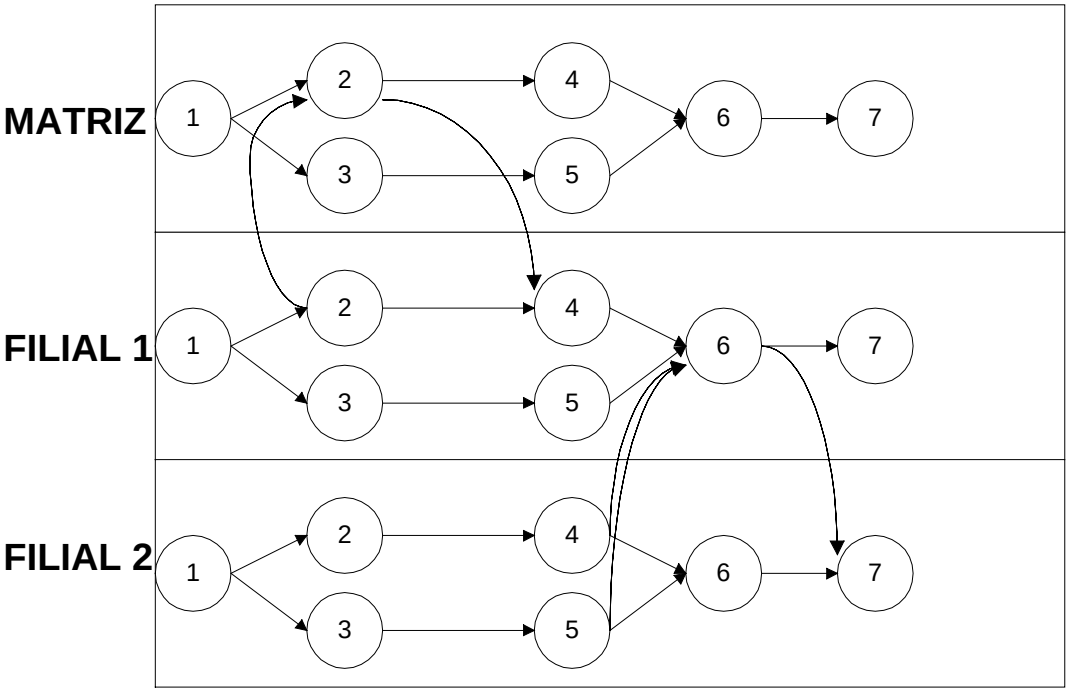


FIGURA 4.10 - Atividades sendo executadas em locais remotos

Como pode-se verificar, a atividade 2 pertencente a instância do *workflow* sendo executado na filial 1, será executada na matriz. Após sua execução, ele dá início a atividade 4 pertencente a filial.

Já as atividades 4 e 5 pertencentes a instância do *workflow* sendo executado na filial 2, dispararão a atividade 6 na filial 1. Após o término da atividade, a execução volta para a filial 2, com o início da execução da atividade 7.

A tabela a seguir (Tabela 4.9) mostra algumas informações necessária para o efetivo uso desta idéia. As colunas representam o local aonde será executada a atividade, e as linhas representam o local de origem da atividade, ou seja, o local no qual elas foram instanciadas. Como pode-se ver, a atividade2, instanciada na MATRIZ e com execução na FILIAL 2, levará 20 unidades de tempo para ser executada. A mesma atividade, com origem na FILIAL 2, e com execução FILIAL 1, levará 10 unidades de tempo para ser executada.

TABELA 4.9 - Tempo de execução da atividade 2 de acordo com a origem da atividade

ATIVIDADE 2				
		EXECUÇÃO DA ATIVIDADE		
		MATRIZ	FILIAL 1	FILIAL 2
ORIGEM DA ATIVIDADE	MATRIZ	10	15	20
	FILIAL 1	15	10	25
	FILIAL 2	20	25	10

Como comentado anteriormente, a possibilidade da execução de uma atividade instanciada num local, ser executada em outro local, quando o local de sua origem seria o mais provável, torna o processo de alocação mais flexível. Outro vantagem a ser destacada, é o fato de poder contar com os locais remotos para se buscar a melhor tempo de execução, quando o próprio local não possibilita. Se isto for possível remotamente, realocações não serão necessárias localmente e, conseqüentemente, não atrasarão nenhum *workflow* já instanciado.

4.6 O modelo de estados e a agenda do ator

Levando-se em consideração os estados que uma atividade possui, a forma com que a alocação está sendo tratada neste trabalho ("*topdown*" e "completa"), e as demais considerações feitas anteriormente, a agenda do ator (suas atividades e respectivos horários de início e término) pode variar durante o seu dia-a-dia.

Esta mudança é justificada pelos atrasos, realocações, atividades alocadas e que não serão executadas devido a uma opção no fluxo de atividades, enfim, devido ao dinamismo da empresa. Dinamismo este que o ator sofre influências. Tais influências refletem na sua agenda, ou seja, no que ele deve fazer e quando.

Veja o exemplo a seguir (Figura 4.11):

08:00 09:00	09:00 10:00	10:00 11:00	11:00 12:00	12:00 13:00
W1 I1 atv35	W1 I2 atv35	W2 I1 atv1	W2 I2 atv1	W1 I5 atv35
atividade cancelada	pré-condições verdadeiras	pré-condições falsas	pré-condições falsas	pré-condições falsas

FIGURA 4.11 - Exemplo da agenda de um ator

Neste exemplo, o ator possui 5 atividades agendadas, sendo que a atividade a ser executada no horário das 08:00 horas foi cancelada. Desta forma, ou o ator fica ocioso, esperando pela atividade das 09:00 horas, ou alguma atividade lhe é alocada neste intervalo, ou ele antecipa alguma das atividades de sua agenda. Esta última opção só é válida, se esta atividade estiver com pré-condições *true*, ou seja, estiver apta para execução.

Uma possível configuração da agenda do ator para esta situação seria (Figura 4.12):

08:00 09:00	09:00 10:00	10:00 11:00	11:00 12:00	12:00 13:00
W1 I2 atv35		W2 I1 atv1	W2 I2 atv1	W1 I5 atv35
em execução		pré-condições falsas	pré-condições falsas	pré-condições falsas

FIGURA 4.12 - Exemplo de uma configuração para o exemplo da Figura 4.11

Como pode ser visto, a atividade das 09:00 horas foi antecipada para ser executada às 08:00 horas. Isto só foi possível, porque suas pré-condições estavam verdadeiras. Portanto, este exemplo demonstra como a agenda do ator sofre alterações devido à flexibilidade e dinamismo da empresa.

4.7 Situações impostas pelo gerente as quais devem ser tratadas pelo WFMS

Quando um *workflow* for ser instanciado, pode-se ter ou não algumas imposições sobre a sua data e horário de término. Essas imposições vão ditar a possibilidade da empresa executar ou não a instância, ou mesmo, executando-a, alterar o prazo de término de outras instâncias.

Aqui, cabe descrever possíveis situações impostas pelos usuários, que o WFMS, através do processo de alocação, terá que tratar. Veja, nem sempre estas situações trarão problemas (alteração de término/início de atividades, realocações em cadeia), porém, o WFMS deve estar preparado (através das funcionalidades do processo de alocação) para tratá-los:

1. **sem data e horário previstos**: este tipo de situação, a princípio, não causa nenhum problema para o processo de alocação, pois, como já foi comentado anteriormente, a idéia é alocar todas as atividades, e não havendo data nem horário de término, sempre haverá uma data e horário para as atividades serem executadas nas agendas dos atores;

2. **o mais cedo possível**: este tipo de alocação também, a princípio, não causa nenhum problema para o processo de alocação. Neste situação, o processo fará uma alocação mais otimizada, buscando uma combinação entre dois fatores, a saber: quando um dado ator pode executar a atividade, e em quanto tempo ele leva para executá-la. Levando-se em consideração esses dois fatores, o processo escolherá os atores que, em conjunto, acabarem a instância do *workflow* o mais cedo possível. Vale ressaltar, que nenhuma realocação será necessária. Um algoritmo que cheque todas as possíveis situações, mostra-se uma boa estratégia para resolver esta situação;

3. **data e horário impostos**: esta situação traz três possibilidades. A primeira, e mais tranqüila, ao efetuar uma previsão de alocação, constatou-se a possibilidade de alocação das atividades e entrega no prazo estimado sem a necessidade de se fazer qualquer realocação. A segunda, mostrou-se impossível, porém, realocando-se atividades, existe a possibilidade de alocação das atividades e entrega no prazo estimado. Em contra partida, outras instâncias terão sua estimativa de término prolongadas. A terceira e última possibilidade, mostrou-se impossível de ser

realizada. Após fazer uma alocação prévia, constatou-se que, mesmo fazendo realocações, não existe a possibilidade de cumprir o prazo. A viabilidade de se aumentar a carga de trabalho dos atores, modificar a especificação do *workflow*, alterar as especificações das atividades ou mesmo do objetivo a ser atingido, ficam a carga da administração da empresa, mas são soluções que podem ser tomadas em prol de se cumprir o prazo imposto. Estas decisões não competem ao WFMS, ele só poderá ajudar, prevendo a possibilidade de se atingir o objetivo dentro do prazo imposto.

4.8 O Modelo Proposto

4.8.1 Considerações Iniciais

O modelo proposto neste trabalho para o processo de alocação envolve a alocação "*topdown*" e "completa", a alteração do fluxo de atividades e o uso de Redes CPM.

Em relação aos fatores que interferem na alocação, já vistos e discutidos no capítulo 3, deverá ficar a cargo da empresa definir quais serão relevantes para a mesma. Neste trabalho serão levados em consideração os seguintes fatores, já citados no capítulo 1: a agenda dos atores (carga de trabalho), o tempo de execução da atividade, a data e horário de início e término da atividades e a prioridade da instância.

O tempo de execução da atividade é um atributo da mesma e deve ser atribuído quando da especificação da atividade. Já a data e horário de início e término das atividades, bem como a prioridade da instância, somente serão definidos no momento da instanciação do workflow.

O diagrama entidade-relacionamento representado na Figura 4.13, ilustra o relacionamento das entidades (*workflow*, fluxo de atividades, atividades, ator, agenda) que serão usadas pelo modelo de alocação proposto.. Neste diagrama, tem-se:

- um *workflow* possui um ou mais fluxos de atividades;
- um fluxo de atividades pertence a um *workflow*;
- um fluxo possui uma ou mais atividades;
- uma atividade pode pertencer a vários fluxos;
- uma atividade é executada por um papel;
- um papel pode executar várias atividades;
- um ator pode pertencer a vários papéis;
- um papel pode possuir vários atores;
- um ator possui uma agenda;
- uma agenda pertence a um ator.

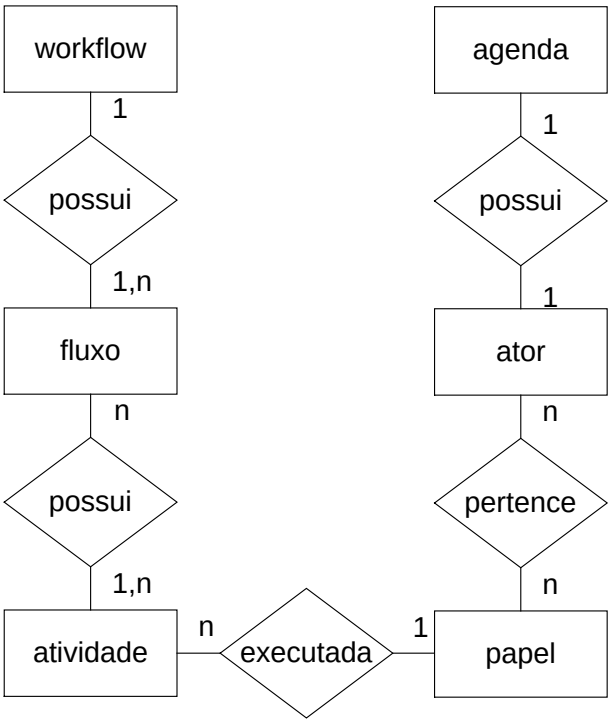


FIGURA 4.13 - Diagrama Entidade-Relacionamento do modelo de alocação proposto

Outro ponto a ser destacado, refere-se ao modo de alocação, ou melhor, como o ator se comportará, de maneira passiva (*PUSH*) ou de maneira ativa (*PULL*) quando do momento da alocação das atividades. Nesta proposta serão utilizados os dois modos, de acordo com a seguinte política:

- se a atividade fizer parte do caminho crítico, a atividade será imposta ao ator, ou seja, este será passivo em relação ao processo de alocação;
- se a atividade não fizer parte do caminho crítico, tem-se duas opções:
  - ou a atividade é imposta ao ator;
  - ou a atividade é imposta a um grupo de atores, e estes, de maneira passiva, poderão requerer a atividade para executá-la. Neste caso, a atividade não terá, pelo menos a princípio, um horário determinado para iniciar. Ela deverá ser iniciada dentro de um intervalo, sendo que este será determinado pela sua folga.

Esta última opção só será possível, se a soma dos tempos de execução das atividades impostas ao grupo não for superior a soma dos tempos que os atores do grupo possuírem para executar as atividades.

Uma vez alocada a atividade para o grupo, o WFMS, através do processo de alocação, deverá verificar se as atividades alocadas para o grupo não estão se aproximando do seu horário de início e, porventura, ainda não foram requeridas por nenhum ator do grupo. Caso isto venha a acontecer, as atividades alocadas para o grupo deixarão de serem requeridas, e serão impostas a um dado ator pelo processo de alocação.

4.8.2 O Modelo

Para efetuar a alocação de atividades, o modelo proposto deverá ter como entrada um pedido de instanciação de um *workflow* , as imposições do usuário e a prioridade da instância. Baseado nestas informações, o processo de alocação deverá verificar quantos fluxos de atividades existem para este *workflow* e, para cada um deles, o processo deverá calcular o caminho crítico, PDI, UDI, PDT, UDT e a Folga.

Com o resultado dos cálculos em mãos, e as imposições do usuário, o processo de alocação deverá aplicar a alocação "topdown" e "completa", e escolher uma configuração (ator X atividade) que satisfaça tais imposições, ou melhor, uma data e horário de término que satisfaça tais imposições. A Figura 4.14 ilustra o modelo proposto.

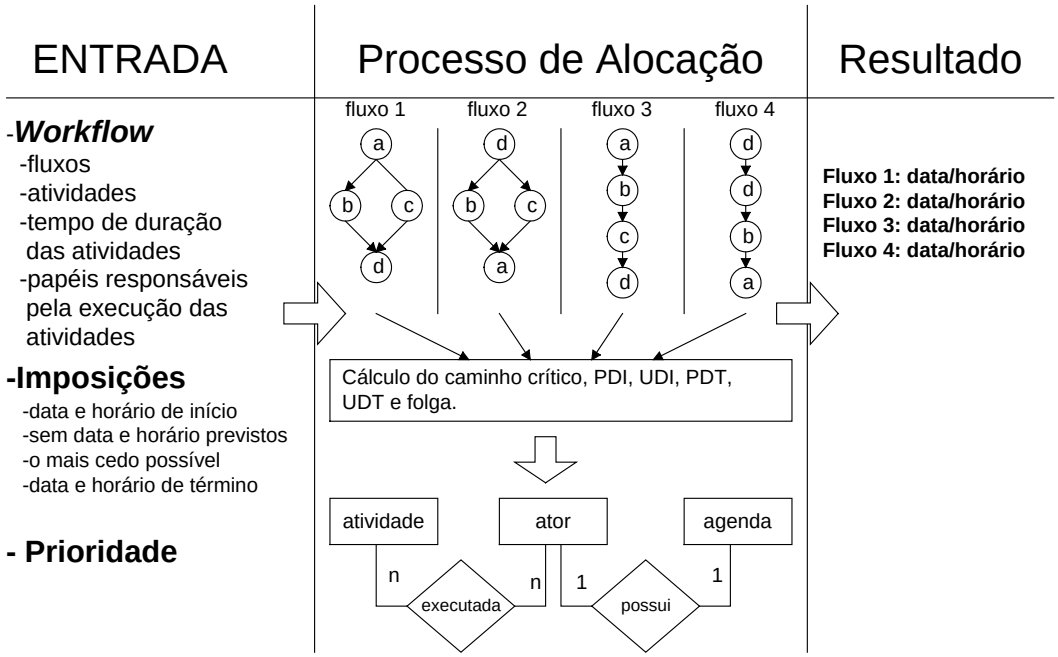


FIGURA 4.14 - Modelo de alocação proposto pelo trabalho

Como vantagens deste modelo tem-se:

1. **possibilidade de se ter uma previsão de término da instância do workflow**: como a alocação será "*topdown*" e "completa", haverá condições de se conhecer, antes do início da instância, a data e horário de término da mesma. Se a locação fosse dinâmica, tal fato não seria possível, visto que a alocação da atividade é feita em tempo de execução, ou seja, quando a mesma estiver com suas pré-condições satisfeitas;

2. **possibilidade de se antever algum problema mediante uma imposição do usuário**: caso o usuário desejar que uma instância termine em determinado horário, este modelo possibilita verificar antecipadamente se isso é possível ou não. Caso não seja possível devido a algum problema de carga de trabalho, por exemplo, o gerente

do *workflow* poderia providenciar soluções para tal problema antes do mesmo acontecer;

3. **várias opções de fluxo**: como o modelo possibilita que um *workflow* tenha mais de um fluxo de atividades, isto faz com que se tenha mais de uma opção para atingir o mesmo objetivo. Dependendo da situação e da carga de trabalho dos atores, uma imposição do usuário pode não ser atingida por um fluxo, e facilmente atingida por outro. Desta forma, tem-se uma flexibilidade para se alcançar um objetivo;

4. **flexibilidade na alocação das atividades**: com o cálculo do caminho crítico, PDI, PDT, UDI, UDT e da folga, tem-se uma flexibilidade na alocação das atividades para os atores. Se uma atividade que não faz parte do caminho crítico não puder ser alocada a nenhum ator logo na sua primeira tentativa, o processo de alocação pode utilizar a folga, para tentar alocar a atividade. Com isso, tem-se uma maior flexibilidade, visto que o processo de alocação não se torna rígido.

4.9 Funcionalidades do processo de alocação de atividades

Para atender as situações apresentadas na seção anterior, o WFMS, através do processo de alocação, deve oferecer uma série de funcionalidades, que permitirão à empresa tomar suas decisões e fazer da alocação de atividades uma ferramenta de auxílio para a administração de seus objetivos e prioridades.

1. **possibilitar ao responsável pela gerência dos *workflows* prever a data e horário de término de uma instância.**

A estratégia de alocação proposta neste trabalho, prevê a alocação de todas as atividades do *workflow*, para que o mesmo seja iniciado. Desta forma, como discutido anteriormente, sempre haverá uma previsão de término para a instância do *workflow*.

Esta previsão de término talvez não seja a ideal, portanto, esta funcionalidade, que a partir de agora será chamada de **previsão de término**, poderá ter as seguintes opções, que já foram discutidas na sessão anterior:

- sem data e horário previstos;
- o mais cedo possível;
- data e horário impostos.

2. **possibilitar ao responsável pela gerência dos *workflows* atrasar e adiantar atividades em prol dos objetivos e prioridades da empresa.**

Esta funcionalidade deve auxiliar o gerente atrasando e/ou adiantando atividades em prol dos objetivos da empresa. Como as prioridades de uma empresa podem mudar de acordo com as variáveis que a cercam (variáveis internas e variáveis externas), conseqüentemente seus objetivos também. Portanto, uma instância que ontem tinha prioridade máxima para a empresa, hoje, por mudanças estratégicas, pode passar a ser uma instância sem maiores importâncias. Em contra partida, uma instância sem

relevância para a empresa, também pode, devido a mudanças de estratégia administrativa, ter sua prioridade alterada para máxima.

Devido a esta flexibilidade que uma empresa deve possuir, o WFMS deve trabalhar em prol de permitir e ajudar a empresa a ser flexível. Portanto, uma funcionalidade que permita atrasar ou adiantar atividades em prol dos objetivos e prioridades da empresa é de suma importância para que a mesma seja flexível, dinâmica e competitiva.

3. **possibilitar ao responsável pela gerência dos *workflows* realocar todas as atividades de todas as instâncias dos *workflows*, para se buscar uma melhor alocação, mediante uma instância estratégica para a empresa.**

Esta funcionalidade é de suma importância, pois poderá alterar as estimativas de término das instâncias que estão sendo executadas. Como já visto e discutido, a empresa deve ser flexível o bastante para poder se adaptar as novas realidades. Com o intuito de verificar previamente as alterações de estimativas que poderão ocorrer mediante uma instância estratégica para a empresa, esta funcionalidade efetua uma realocação de todas as atividades em prol de se contemplar a instância estratégica, ou seja, a instância estratégica será atendida, as demais poderão ter sua estimativa de término alterada.

Porém, como visto anteriormente, uma realocação pode trazer o problema da realocação em cadeia, ou seja, uma realocação pode causar uma mudança na agenda do ator, atrasando alguma atividade que por sua vez atrasará suas atividades sucessoras, sendo que estas pertencem a outros atores, que também terão suas atividades atrasadas, enfim, possivelmente, as estimativas de término das instâncias mudarão. Sendo assim, esta funcionalidade deve alertar o gerente dos possíveis atrasos que poderão ocorrer antes do mesmo efetuar a realocação.

Outro detalhe a ser considerado, determinadas instâncias não podem ter suas estimativas de término alteradas, visto que são prioridades para a empresa ou a empresa já se comprometeu em entregá-las dentro da estimativa inicial. Desta forma, a instância deve possuir um atributo que indique a possibilidade de se alterar suas alocações. Porém, este atributo não pode ser rígido ao ponto de não permitir uma realocação de atividades perante a ausência de um ator.

Entretanto, após contemplar a instância estratégica, como alocar as demais instâncias, qual ordem será seguida? Esta pergunta deve ser respondida pelo gerente dos *workflow*s ou pela administração da empresa. O WFMS simplesmente deve possibilitar que o desejo do gerente ou da empresa seja satisfeito.

Alguns fatores poderiam ser analisados, são eles:

- **prioridade**: as instâncias com maior prioridade seriam alocadas primeiro;
- **tempo de término**: as instâncias que estão mais próximas de serem finalizadas teriam o direito de serem alocadas primeiro;
- **tempo fixo para término**: as instâncias que tiverem tempo fixo (imposto) de término, teriam o direito de serem alocadas primeiro. Dentro deste item,

primeiramente as que não possuem tempo de folga, e em seguida, as que possuem tempo de folga.

A partir do sistema de alocação proposto, sendo este a maior contribuição deste trabalho, foi implementado um protótipo objetivando demonstrar a viabilidade das idéias apresentadas.

5 O protótipo

5.1 Introdução

A implementação do protótipo tem a finalidade de testar na prática o conjunto de funcionalidades previstas ao processo de alocação de atividades em um sistema de gerência de *workflow*, funções estas que permitirão não só o entendimento das idéias propostas neste trabalho, bem como a avaliação das mesmas.

Estas funções estão divididas em três grupos. O primeiro deles, representa o grupo de funções que, ao nosso ver, todos os sistemas de workflow devem oferecer. A implementação destas funções se faz necessária visto que este protótipo não está baseado em nenhum sistema de workflow já implementado (existente). O segundo, representa o grupo de funções que darão suporte à implementação das funcionalidades propostas no capítulo anterior, sendo que estas formam o terceiro grupo de funções. De uma maneira geral, pode-se dizer que estes dois últimos grupos de funções darão as reais condições para a avaliação do protótipo.

5.2 O cenário do protótipo

No capítulo 4, foi mostrado o cenário no qual o processo de alocação efetua suas atividades (Figura 4.2). Baseado em um único fluxo de atividades, o WFMS (processo de alocação) aloca atividades à atores. Tal alocação pode ser do tipo ativa ou passiva. Existe também intervenções externas, que nada mais são do que atividades (compromissos) que não pertencem a nenhum *workflow*, e que devem ser registradas nas agendas dos atores.

Para o efetivo uso das idéias propostas neste trabalho, este cenário sofrerá uma modificação substancial: dependendo do *workflow*, e dos objetivos que o mesmo vier a buscar, poderão existir dois ou mais fluxos para se alcançar o mesmo objetivo.

Para um melhor esclarecimento das idéias aqui colocadas, veja a figura 5.1 a seguir:

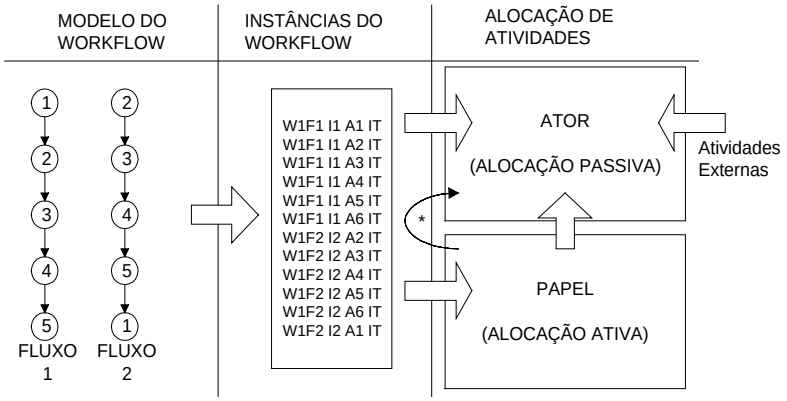


FIGURA 5.1 - Cenário de alocação do protótipo

Como pode ser verificado neste novo cenário (Figura 5.1), o modelo de um dado *workflow* pode possuir mais de um fluxo de atividades, sendo que estes buscam os mesmos objetivos. Baseado na agenda dos atores e nas opções de fluxo, o processo de alocação buscará o melhor caminho para se chegar ao objetivo desejado. Para esta busca, será utilizado a técnica do caminho crítico da rede CPM.

Duas situações podem ocorrer quando se busca o melhor caminho, são elas:

1. o caminho no qual a instância termina o mais cedo possível: deseja-se que a instância termine o mais cedo possível. Neste caso, após analisar as opções de fluxo, o WFMS deverá escolher qual das opções encerra-se primeiro;
2. o caminho no qual a instância termina na data e horário desejado: deseja-se que a instância termine em determinada data e horário. Talvez esta data e horário sejam facilmente alcançados, como também, pode ser que, para serem alcançadas, realocações sejam necessárias.

Estas situações representam duas imposições que são colocadas ao processo de alocação, para que o mesmo possibilite à empresa alcançar seus objetivos. Portanto, o processo de alocação deverá oferecer funcionalidades que permitam tais análises e, caso necessário, as realocações.

5.3 Os grupos de funções

Conforme comentado no início deste capítulo, este protótipo está dividido em três grupos de funções. Para um melhor entendimento desta divisão, foi atribuído uma denominação a cada um deles, a saber:

1. **grupo básico**: fazem parte deste grupo funções elementares para o funcionamento do processo de alocação dentro de um sistema de gerência de *workflow*;

2. **grupo de suporte**: funções que efetivamente serão responsáveis pela alocação, ou, que de uma maneira ou de outra, interferem neste processo;

3. **grupo gerencial**: funções que servirão de ferramenta para o responsável pelo *workflow* gerenciar e configurar as instâncias (data e hora de início e término), com o intuito de atingir seus objetivos.

A seguir, serão apresentadas as funções de cada um dos grupos:

Grupo Básico		
Função	Parâmetros	Descrição sucinta
Criar_*workflow*()	• Id_workflow • Descrição	Função responsável pelo cadastramento de um *workflow*.
Criar_fluxo()	• Id_workflow • Id_fluxo	Função responsável pelo cadastramento de um fluxo de um dado *workflow*.
Criar_atividade()	• Id_atividade • Tempo_execução • Id_papel	Função responsável pelo cadastramento de uma atividade.
Criar_papel()	• Id_papel • Descrição	Função responsável pelo cadastramento de papéis.
Criar_ator()	• Id_ator • Nome • Habilidades	Função responsável pelo cadastramento dos atores. Estes atores deverão ser associados à papéis e, também, poderão ser associados às atividades.
Associa_Papel_Ator( )	• Id_papel • Id_ator	Função responsável por associar um ator a um papel.

Grupo de Suporte		
Função	Parâmetros	Descrição sucinta
Criar_instância()	• Id_workflow • Data_início • Data_término • Horário_início • Horário_término • Prioridade	Função responsável pela criação de uma instância de um dado *workflow*. Não pode ser esquecido que um *workflow* será instanciado se e somente se todas as suas atividades forem alocadas.
Atv_externa()	• Id_ator • Data • Horário • Duração • Descrição	Função responsável por verificar e, se possível, confirmar, situações nas quais o ator ficará ocupado.
Agendar_ausência()	• Id_ator • Data • Horário • Duração • Descrição	Função responsável por indicar na agenda do ator, que o mesmo ficará impossibilitado de trabalhar.
Agendar_extra()	• Id_ator • Data • Horário • Duração • Descrição	Função responsável por indicar na agenda do ator, dias e horários que o mesmo ficará disponível para trabalhar, sendo que estes são horários e dias extras, ou seja, fora de sua jornada normal de trabalho.
Cancel_Instância()	• Id_workflow • Id_instância	Função responsável por cancelar uma instância de um dado *workflow*.
Ver_agenda()	• Id_ator • Período	Função responsável por disponibilizar a agenda de uma dado ator, mostrando suas atividades, datas e horários.

Grupo Gerencial		
Função	Parâmetros	Descrição sucinta
Prever_término()	• Id_workflow • Data_início • Data_término	Função que possibilita o gerente do *workflow* conhecer a previsão de término de uma dada instância.
Prever_estratégica()	• Id_workflow • Data_término • Horário_término	Função que possibilita o gerente do *workflow* impôr uma situação de término e saber se isto é possível, e quais as conseqüências, ou seja, saber quais as instâncias que sofrerão atrasos para que a mesma seja executada a contento.
Realocar()		Função que permite ao gerente do *workflow* realocar atividades entre os participantes do *workflow*. Porém, antes de efetuar tal realocação, a função deve informar se alguma instância sofrerá alguma conseqüência.
Alterar_prazo()	• Id_workflow • Id_Instância • Id_atividade • Nova_data • Novo_horário	Função que permite ao gerente do *workflow* alterar prazos de início e término de determinadas atividades, com o intuito de beneficiar outras atividades.
Alterar_Prioridade()	• Id_workflow • Id_instância • Nova_prioridade	Função que permite alterar a prioridade de uma instância de um dado *workflow*.
Alocar_manual()	• Id_workflow • Id_instância • Id_atividade • Id_ator • Data • Horário	Função que permite ao gerente alocar uma atividade a um dado ator manualmente.

5.4 A arquitetura do protótipo

Para concretizar a implementação deste protótipo foi criada uma arquitetura simples, porém funcional, que desse condições para colocar em prática as idéias apresentadas neste trabalho. A figura a seguir mostra esta arquitetura (Figura 5.2):

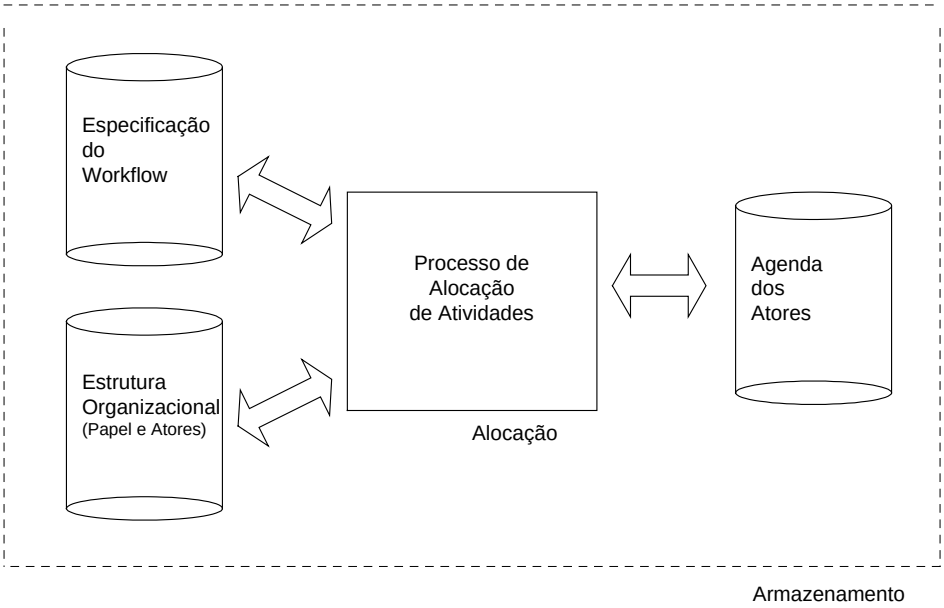


FIGURA 5.2 - Arquitetura do protótipo

A arquitetura do protótipo (Figura 5.2) é composta de dois componentes, o componente de armazenamento e o componente de alocação (funcional).

O componente de armazenamento é responsável pelo registro de informações relacionadas com o *workflow*, com a agenda dos atores e com a estrutura organizacional da empresa.

As informações sobre o w*orkflow* traz dados tais como atividades, possíveis fluxos do *workflow*, tempo de execução das atividades, habilidades necessárias para a execução de cada atividade, papel dos atores que podem executar cada atividade, enfim, informações relacionadas à especificação do *workflow*.

Já as informações armazenadas sobre a agenda dos atores, referem-se aos seus horários disponíveis e os já comprometidos com alguma atividade. Para cada horário comprometido, são armazenadas informações sobre qual a atividade a ser executada, a instância e o *workflow* a qual ela pertence.

Por fim, a estrutura organizacional, é responsável pelo armazenamento das informações dos atores, tais como suas habilidades e papéis que o mesmo pode executar.

O segundo componente, o componente de alocação, é o responsável por disponibilizar ao gerente e ao WFMS, as funções que lhe permitirão efetuar e gerenciar o processo de alocação de atividades. Este componente seguirá as regras traçadas no capítulo anterior, mas especificamente, no item 4.8.

6 Conclusões e Trabalhos Futuros

Um bom sistema de gerência de *workflow* é fundamental para aumentar a competitividade das empresas e assim contribuir para melhorar a eficiência do fluxo de atividades. Apesar do mercado apresentar uma série de sistemas de gerenciamento de *workflow*, não existe um padrão para esta tecnologia que seja aceito tanto por usuários, fornecedores e acadêmicos.

Como foi tratado nesta dissertação, através do consórcio WfMC, se observa um esforço na busca de uniformização dos termos e conceitos que até pouco tempo não existia. Esta uniformização permite, por exemplo, que sistemas de fornecedores distintos troquem informações e trabalhem de forma cooperativa, possibilitando o uso efetivo e completo da tecnologia. O esforço despendido pela WfMC pode ser visto com mais detalhes em [WMC96a] [WMC96b] [WMC96c] [WMC96d] [WMC96e].

Outro exemplo da falta de uniformização pode-se constatar na diversidade dos modelos de *workflow* descritos e analisados neste trabalho. Estas constatações decorrem da existência de diferentes linguagens, terminologias e simbologias empregadas em cada modelo para se especificar um *workflow*.

Esta dissertação trata de umas das principais funcionalidades de um WFMS, a alocação de atividades. A alocação de atividades é responsável pela distribuição das atividades para os atores. Como uma execução sem qualidade traz problemas para a empresa, conclui-se que o processo de alocação interfere diretamente no bom andamento da empresa, daí a importância em se preocupar com tal processo.

Dentro deste contexto, a primeira contribuição deste trabalho foi uma classificação dos fatores que interferem no processo de alocação. Fatores tais como tempo de execução, habilidades, prioridade, foram estudados visando detalhar a forma e mostrar como os mesmos interferem neste processo. Esta classificação e detalhamento permitiram uma visualização da complexidade do trabalho.

A segunda contribuição, e a mais importante, é o modelo proposto de alocação de atividades envolvendo três aspectos: (i) forma e estratégia de alocação "*topdown*" e "completa", (ii) possibilidade de escolha do fluxo de atividades, e (iii) o uso do método do caminho crítico CPM.

Este modelo de alocação mostrou-se bastante flexível na identificação de opções de alocação dos possíveis responsáveis pela execução de uma dada atividade. A partir dos dados do *workflow* (fluxos, atividades, tempo de execução das atividades, papéis, agenda dos atores, etc.), da prioridade da instância, e das exigências de início e término da mesma, o algoritmo de alocação definido para o modelo busca uma opção que melhor se encaixa nas exigências apresentadas.

As possibilidades de alteração do fluxo de atividades definidas na especificação do *workflow* é usada pelo algoritmo como opções na busca de uma solução que atenda as exigências impostas. A comparação dos resultados da alocação para cada fluxo faz com que de se diminua ou até mesmo se evite gargalos.

O uso da estratégia de alocação "*topdown*" e "completa" é definida nesta dissertação como uma forma de trabalhar com planejamento e previsão do fluxo de atividades.

O método do caminho crítico possibilita que durante o processo de alocação se identifiquem as atividades que fazem parte deste caminho onde o fator folga é mais crítico. Por outro lado, as atividades que não fazem parte deste caminho podem ter uma data/horário alternativo, sem que isto traga qualquer prejuízo para a execução do *workflow* como um todo.

Durante o desenvolvimento desta dissertação, foram identificados temas de estudos que não foram tratados por falta de tempo ou por não serem o foco principal do trabalho. São eles:

1. atividades executadas em locais diferentes da origem do *workflow: e*ntende-se que o processo de alocação de atividades, com base nesta possibilidade, pode ser ainda mais flexível e, sem dúvida, subsidiar a empresa em busca da competitividade neste cenário globalizado;

2. alteração do fluxo de atividades: pretende-se criar algoritmos que chequem os fluxos alterados, verificando se existem caminhos sem fim, *loops*, etc., ou seja, algoritmos que comprovem o alcance dos objetivos do *workflow*;

3. segurança temporal: sincronismo de mecanismos de segurança com as autorizações necessárias para a execução de uma atividade alocada;

4. modelagem do *workflow*: o desenvolvimento de um modelo de especificação de *workflow* que dê mais semântica no que tange aos atributos relacionados com a alocação de atividades. Como pode ser constatado, os modelos apresentados são pobres quanto a estes atributos. Este modelo deverá estar de acordo com os padrões estabelecidos pela WfMC;

5. negociação: a negociação entre atores e gerentes é uma artifício que pode ser usado para resolver problemas de alocação que não puderam ser solucionados pelo WFMS ou pelo próprio gerente do *workflow*. Um estudo sobre esta negociação deve ser feito para responder perguntas do tipo: Que atributos são necessários para se automatizar esta negociação?, Como e quando disparar este processo?, Quais são os efeitos da negociação?, etc.

BIBLIOGRAFIA

[AAL95a] AALST, W.M.P. **Petri-net-based Workflow Management Software.** Eindhoven: Eindhoven University of Technology, 1995. Relatório de Pesquisa.

[AAL95b] AALST, W.M.P. **Modelling and analysing workflow using a Petri-net based approach.** Eindhoven: Eindhoven University of Technology, 1995. Relatório de Pesquisa.

[ALO95] ALONSO, G. et al. **Advanced Transaction Models in Workflow Contexts.** San Jose: IBM Almaden Research Center, 1995.

[AMB96] AMBERG, M. **The Benefits of Business Process Modeling for Specifying Workflow-Oriented Application Systems**. Universidade de Bamberg, 1996. Disponível por WWW em: http://elwood.seda.sowi.uni-bamberg.de/workflow/document/wfmc02/wfmc-cfp.htm

[BAI93] BAIR, James H. Constrasting Workflow Models: Getting to the Roots of Three Vendors**.** In: GROUPWARE, 1993, San Jose. **Proceedings** … San Jose: [s.n.], 1993.

[BAR95] BARTHELMESS, P.; WAINER, J. Workflow Modeling. In: CYTED-RITOS INTERNATIONAL WORKSHOP ON GROUPWARE, CRIWG, 1., 1995, Lisboa. **Proceedings** … Lisboa: Junta Nacional de Investigação Científica e Tecnologica, 1995.

[BAR95a] BARTHELMESS, P.; WAINER, J. Workflow Systems: A few definitions and a few suggestions. In: COOCS'95 ACM CONFERENCE ON ORGANIZATIONAL COMPUTING SYSTEMS, 1995, San Jose. **Proceedings** … San Jose: [s.n.], 1995.

[BOR95] BORGERS, M.R.S. et al. **Suporte por Computador ao Trabalho Cooperativo.** In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 15., JAI95, JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA, 14., Canela, RS, Brasil, Julho 1995. 45p.

[BUL92] BULL CORPORATION. **FlowPath Functional Specification.** Paris: BULL, 1992.

[CAS95a] CASATI, F. et al. Conceptual Modeling of Workflows. In: OO-ER CONFERENCE, 1995, Gold Coast, Australia. **Proceedings** … Gold Coast, Australia: [s.n.], 1995.

[CAS96] CASATI F. et al. Workflow Evolution. In: ER'96 INTERNATIONAL CONFERENCE, 15., 1996, Germany. **Proceedings** … Germany: [s.n.], 1996.

[CAS96a] CASATI F. et al. **WIDE** Workflow model and architecture. April-1996. Disponível em: http://www.sema.es/projects/WIDE/Documents/ase30_4.ps.gz

[CHE90] CHEN P. **Modelagem de Dados -** A Abordagem Entidade-Relacionamento para Projeto Lógico**.** São Paulo: Makron Books, 1990. 80p.

[DEN95] DENNING, P.; MEDINA-MORA, R. Case Study: George Mason University. In: **New Tools for New Times: The Workflow Paradigm**. Lighthouse Point: Future Strategies, 1995. 348 p. p.59-74.

[DIE96] DIETRICH, E. **Projeto de um Sistema de Suporte à Autoria Cooperativa de Hiperdocumentos.** Porto Alegre: CPGCC da UFRGS, 1996. Dissertação de Mestrado.

[DUI94] DUITSHOF M. **Workflow Automation in Three Administrative Organizations.** Twente: University of Twente, 1994. Master's Thesis.

[ELL79] ELLIS, Clarence A. Information control nets: A mathematical model of office information flow**.** In: ACM CONFERENCE SIMULATION, MODELING AND MEASUREMENT OF COMPUTER SYSTEMS, 1979. **Proceedings** … [S.l.:s.n.], 1979.

[ELL80] ELLIS, Clarence A.; NUTT, Gary J. Office Information Systems and Computer Science. **ACM Computing Surveys**, v. 12, n. 1, p. 27-60, Mar. 1980.

[ELL93] ELLIS, Clarence A.; NUTT, Gary J. Modelling and Enactment of Workflow Systems. In: APPLICATION AND THEORY OF PETRI NETS, 1993, Berlin. **Proceedings** … Berlin: Springer-Verlag, 1993.

[FLO79] FLORES, Fernando. **Management and communication in the office of the future** Berkeley: University of California at Berkeley, 1979. Master's Thesis.

[GEO95] GEORGAKOPOULOS, D.; HORNICK, M.; SHETH, A. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. **ACM Distributed and Parallel Databases**, [S.l.], n. 3, p. 119-153, Mar. 1995.

[HAM94] HAMMER, M.; CHAMPY, J. **Reengenharia:** Revolucionando a Empresa. Rio de Janeiro: Campus,1994. 198 p.

[HEU90] HEUSER, C.A. **Modelagem Conceitual de Sistemas: Redes de Petri**. IV EBAI, 1990.

[IOC96] IOCHPE, C.; MANGAN, M.; ZIELINSKI, J. **First Report on Modelling Software Development through Project and Workflow Management with Integrated Human Interactions**. Porto Alegre: CPGCC/UFRGS, 1996. 32 p. RP-264).

[JOO94a] JOOSTEN, S. et al. **WA-12: An Empirical Study about the Practice of Workflow Management.** Research Monograph. The Netherlands: University of Twente,1994.

[JOO94b] JOOSTEN, S. Trigger Modelling for Workflow Analysis. In: CON'94: WORKFLOW MANAGEMENT, CHALLENGES, PARADIGMS AND PRODUCTS, 1994, Viena. **Proceedings** … Viena: [s.n.],1994.

[JOO95a] JOOSTEN, S. A Method for Analysing Workflows. In: ECSCW CONFERENCE, 1995, Stockholm. **Proceedings** … Stockholm: [s.n.], 1995.

[JOO95b] JOOSTEN, S.; BRINKKEMPER, S. **Fundamental Concepts for Workflow Automation in Practice**. Twente: University of Twente, 1995.
Disponivel em:
http://www_is.cs.utwente.ni:8080/~joosten/documents/artikel.ICIS.ps

[KHO95] KHOSAFIAN, S.; BUCKIEWICZ, M. Workflow: Computer-Supported Collaborative Work-Processing. In: **Introduction to Groupware, Workflow and Workgroup Computing.** New York: John Wiley & Sons, 1995. Chap.5, p.207-258.

[LIN92] LIN, Hwa C.; RAGHAVENDRA, C. S. A Dynamic Loading Policy with a Central Job Dispatcher (LBC). **IEEE Transactions on Software Engennering**, New York, v. 18, n.2, p. 148-158, Feb. 1992.

[MAR94] MARSHAK, R. Software to Support BPR - The Value of Capturing Process Definitions. **Workgroup Computing Report**, New York, v.17, n.7, p. 3-15, July 1994.

[MAR95] MARSHAK, R. Rethinking Workflow. **Workgroup Computing Report**, New York, v.18, n.3, p. 8-12, Mar. 1995.

[MCC92] McCREADY, S. There is more than one kind of workflow software**.** **Computerworld**, [S.l.], n. 2, p.151-155, Nov. 1992.

[MED92] MEDINA-MORA, R. et al. The Action workflow approach to workflow managemente technology. In: CSCW, 1992, [S.l.]. **Proceedings** … [S.l.:s.n.], 1992.

[MON87] MONKS, J.G. **Administração da Produção.** São Paulo:McGraw-Hill, 1987. 502p.

[NIC96] NICOLAO, M.; OLIVEIRA, J.P.M. **Caracterizando Sistemas deWorkflow**. Read Revista Eletrônica de Administração: Porto Alegre: CPGCC da UFRGS, v. 2, n. 4 (Dez 1996), p. 1-8. il.

[RAM94] RAMAGE, M. **Engineering a smooth flow? A study of workflow software and its connections with business process reengineering.** Brighton: University of Sussex, School of Cognitive & Computing Sciences, 1994. Master's Thesis.

[SIL95] SILVER, Bruce. Automating the Business Environment. In: **New Tools for New Times: The Workflow Paradigm.** Lighthouse Point: Future Strategies, 1995. 348 p. p.173-196.

[STR84] STRACK, J. **Modelagem e Simulação de Sistemas.** Rio de Janeiro: Livros Técnicos e Científicos, 1984. 160p.

[TAK95] TAKAHASHI, K.; HIGUCHI, M. **Hypermedia Support for Workflow Management in Collaborative Document Production**. [S.l.]: NTT Software Laboratories.
Disponível em http://www.nttlabs.com/people/kt/WF_pp.html (1995).

[THE95] THE, L. Workflow Tackles the Productivity Paradox. **Datamation**, [S.l.], v. 40, n. 3, Mar. 1995.

[WIN87a] WINOGRAD, Terry. A language/action perspective on the design of cooperative work. **Human-Computer Interaction**, [S.l.], v.3, n. 1, p. 3-30, 1987.

[WIN87b] WINOGRAD, Terry; FLORES, Fernando. **Understanding Computers and Cognition:** A New Foundation for Design**.** Reading: Addison-Wesley, 1987. 324 p.

[WIN88] WINOGRAD, Terry et al. Computer Systems and the Design of Organizational Interaction. **ACM Transactions on Office Information Systems**, New York, v.6, n.2, p.126-152, Apr. 1988.

[WMC96a] WORKFLOW MANAGEMENT COALITION. **Interface 1: Process Definition Interchange.** Bruxelas, Jun. 1996. 42p.
Disponível em: http://www.aiai.ed.ac.uk/project/wfmc.index.html

[WMC96b] WORKFLOW MANAGEMENT COALITION. **Draft Workflow Standard - Interoperability: Abstract Specification.** Bruxelas, Jun. 1996. 66p.
Disponível em: http://www.aiai.ed.ac.uk/project/wfmc.index.html

[WMC96c] WORKFLOW MANAGEMENT COALITION. **Interface 2: Application Programming Interface (WAPI) Specification.** Bruxelas, Out. 1996. 79p.
Disponível em: http://www.aiai.ed.ac.uk/project/wfmc.index.html

[WMC96d] WORKFLOW MANAGEMENT COALITION. **Terminology & Glossary.** Bruxelas, Jun. 1996. 52p.
Disponível em: http://www.aiai.ed.ac.uk/project/wfmc.index.html

[WMC96e] WORKFLOW MANAGEMENT COALITION. **Audit Data Specification.** Bruxelas, Nov. 1996. 31p.
Disponível em: http://www.aiai.ed.ac.uk/project/wfmc.index.html