

# Workflow Management Systems

João Sequeira

[jsequeira@fc.ul.pt](mailto:jsequeira@fc.ul.pt)

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
1.1	O Que são Workflow Management Systems (WfMS)	3
1.2	Background Histórico	3
<b>2</b>	<b>Definição de um Sistema de Workflow</b>	<b>4</b>
<b>3</b>	<b>Execução de um Sistema de Workflow</b>	<b>6</b>
<b>4</b>	<b>O Standard Definido pela WfMC (Workflow management Coalition)</b>	<b>6</b>
4.1	Modelo de referencia da WfMC	6
4.1.1	Workflow Enactment Services	7
4.1.2	Motor de Workflow	7
4.1.3	Process Definition Interface (Interface 1)	8
4.1.4	Workflow Client Application Interface (Interface 2)	10
4.1.5	Invoked Aplicativos Interface (Interface 3)	12
4.1.6	Workflow Interoperability(Interface 4)	13
4.1.7	Administration and monitoring interface(Interface 5)	14
<b>5</b>	<b>Programação com Sistemas de Workflow</b>	<b>15</b>
<b>6</b>	<b>Integração com Outras Plataformas de Middleware</b>	<b>16</b>
<b>7</b>	<b>Benefícios e limitações dos WfMS</b>	<b>16</b>
<b>8</b>	<b>Um Exemplo de um sistema de gestão de Workflow "Websphere MQ Workflow"</b>	<b>17</b>
8.1	Características do IBM WebSphere? MQ Workflow V3.5	17
8.2	Estrutura Interna do MQ Workflow	17
8.3	Componentes do Servidor	18
8.4	Componentes do Cliente	19
8.5	Ciclo de Vida de um Workitem	20
8.6	Usando o MQ Series Workflow (Um Exemplo)	21
8.6.1	Definição do exemplo	21
8.6.2	Usando o MQSeries Workflow Buildtime	22
8.6.3	Usando o MQSeries Workflow Runtime	25
8.7	Programando com as APIs fornecidas pelo MQ	28
<b>9</b>	<b>Links Úteis</b>	<b>29</b>

# 1 Introdução

Até agora temos vindo a falar de formas de esconder ao programador toda a heterogeneidade e distribuição de um sistema empresarial típico, fornecendo ao programador uma base uniforme a partir da qual pode desenvolver as aplicações. Os sistemas de gestão de workflow situam-se do outro lado da integração de aplicações, isto é servem para facilitar a definição e manutenção da lógica da integração, ou seja, a sua ordem relativa de execução das aplicações, quais a serem executadas, etc.

## 1.1 O Que são Workflow Management Systems (WfMS)

Começamos então por definir o que se entende por workflow. Workflow é o aspecto operacional de um processo de trabalho, ou seja, como estão estruturadas as tarefas, quem as executa, qual a sua ordem relativa, como se sincronizam entre si. O Workflow também se refere ao fluxo de informação de suporte a execução das tarefas, e como as monitorizar.

Segundo a Workflow Management Coalition Workflow, é a automatização de um processo de negócio, no seu todo, ou em partes, onde documentos, informação ou tarefas são passadas entre os diversos participantes, de acordo com um conjunto de regras procedimentais.

Existem várias formas de modelar o encaminhamento e processamento necessário a um processo de workflow típico. Como o processo Decision-chain em que são usadas milestones e pontos de decisão (condições que podem ou não ter sido satisfeitas) para mapear o processo. Um outro exemplo é o mapeamento por fluxo de eventos, que representa o processo por um numero de eventos, manuais ou automatizados, sendo as condições mapeadas como bifurcações no esquema. Como terceiro exemplo temos o grafo de fluxo no qual os diferentes tipos de nós representam os vários passos e bifurcações que o fluxo pode seguir.

Um sistema de gestão de Workflow (Workflow Management System- WfMS) é um sistema que, define, cria e gere a execução de um ou vários Workflows, através do uso de software. Software este que é capaz de interpretar a definição dos processos, ou seja como estes devem ser executados, e por que ordem. Um WfMS também deve ser capaz de comunicar com os vários participantes e, quando necessário, invocar as diversas ferramentas e aplicações. Muitas vezes os sistemas de gestão de workflow também permitem a monitorização dos vários processos para que estes possam ser continuamente melhorados. A maioria dos sistemas de workflow integram-se com outras aplicações como Bases de Dados, serviços de e-Mail e outras aplicações, para que seja possível estruturar um processo que envolva vários sistemas independentes.

Em resumo, um sistema de gestão de workflow como o próprio nome indica tem como objectivo a automatização do processo de Workflow, mantendo a lógica de execução entre os diversos participantes.

## 1.2 Background Histórico

Os sistemas de gestão de Workflow tem a sua origem no domínio do *office automation*, tendo as primeiras implementações como objectivo automatizar os processos administrativos que tinham como base documentos em papel. O objectivo seria a utilização de versões electrónicas destes, usar um WfMS para tratar do encaminhamento dos documentos entre os vários pontos. Muitas

destas implementações iniciais eram baseadas em sistemas de e-mail, que mais tarde foram substituídas por formulários em páginas Web. Sistemas de workflow que implementam este tipo de automatização administrativa, são denominados *Workflows Administrativos*.

À medida que o tempo ia passando e a tecnologia de workflow ia amadurecendo, e com o aumento da necessidade por parte das empresas de integrar as diversas aplicações, várias empresas começaram a aperceber-se que os WfMS serviam não apenas para definir o fluxo de informação entre pessoas, mas também para a definição da lógica de negócio necessária a integração de sistemas heterogéneos e distribuídos. Os sistemas de gestão de Workflow que tratam da gestão da lógica de negócio também são chamados de *Workflows de Produção*

Como os sistemas de workflow apesar de estarem desenhados para o encaminhamento de informação entre vários pontos numa empresa não tinham o suporte necessário a heterogeneidade que existe num ambiente empresarial normal, então estes foram integrados com sistemas de EAI (Enterprise Application Integration) como message brokers e afins. Onde a plataforma EAI esconde a heterogeneidade do sistema e a plataforma de Workflow suporta a definição da lógica de negócio. Isto torna-se possível uma vez que um WfMS é capaz de tornar a lógica de aplicação explícita através de linguagens de alto nível e muitas vezes gráficas. E hoje em dia a maioria dos vendedores de plataformas middleware tem no seu portfolio aplicações de WfMS bem como Message Brokers.

## 2 Definição de um Sistema de Workflow

Nesta secção explicam-se alguns dos principais conceitos subjacentes aos WfMS e, como é feita a modelação de um sistema de Workflow.

Chamamos processo de negócio ao conjunto de actividades realizadas por utilizadores humanos ou aplicações de software que têm de ser executados para se atingir um determinado objectivo do negócio. Como exemplo temos todo o processo necessário a sustentar uma determinada encomenda, desde verificar se há stock, levar a encomenda do armazém para o sítio onde vai ser embalada, verificar se o cliente já efectuou o pagamento e enviar a encomenda ao cliente.

Os termos *workflow processo de workflow* ou simplesmente *processo* referem-se a uma descrição formal e executável de um processo de negócio. E por fim um *Sistema de Gestão de Workflow* é uma plataforma software que permite o desenho, desenvolvimento, execução e análise de um processo de workflow.

Ainda se considera como uma *instância de workflow* como a execução de um workflow. Um workflow pode ser instanciada diversas vezes, e podem estar a ser executadas concorrentemente varias instâncias do mesmo ou de diferentes workflows.

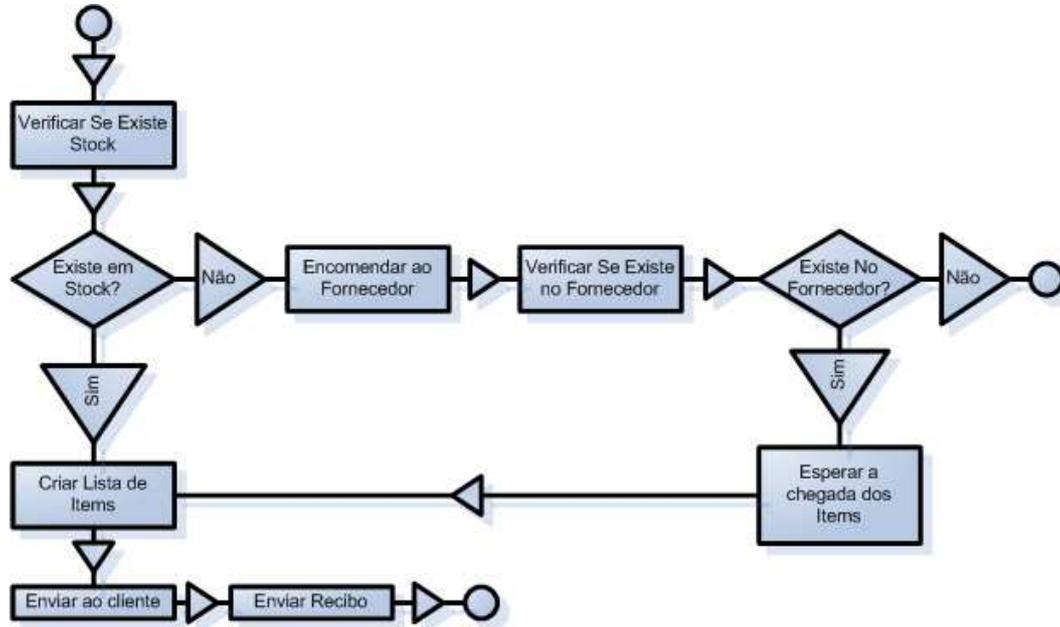
Um processo de workflow pode ser representado, como foi indicado anteriormente, por um grafo de fluxo que define a ordem de execução entre os vários nós no processo. Os nos podem ser de 3 tipos:

**Nó de Trabalho** Os Nós de Trabalho representam items de trabalho que devem ser executados quer por um recurso humano quer automático.

**Nó de Encaminhamento** Nós de encaminhamento definem a ordem pela qual os items de trabalho devem ser executados, e permitir a definição da activação de nós de trabalho em paralelo ou mediante determinadas condições.

**Nós de início e fim** Os Nós de início e de fim denotam o início ou o fim de um determinado fluxo de trabalho.

Através do arranjo dos diversos nós é possível especificar o trabalho a ser feito, a ordem relativa das diversas actividades, e a quem, ou ao quê o trabalho deve ser atribuído.



**Figura 1.** Exemplo de uma especificação de workflow

A Figura 1 representa um hipotético grafo de workflow que representa os passos que uma encomenda deve passar desde que o cliente executa o pedido até que o produto lhe é finalmente entregue. Na figura os quadrados representam nós de trabalho (*work nodes*) os losangos representam nós de encaminhamento (*routing nodes*), e os círculos representam os nós de início e fim.

A sequência de eventos representada no grafo é a seguinte:

1. Quando chega um pedido ao sistema este verifica se o produto existe em stock
2. Caso exista em stock passa para o passo 6 caso contrário para o passo 3.
3. Como não existe em stock então encomenda ao fornecedor
4. Verifica se o produto ainda existe no fornecedor Se existe vai para 5, caso contrário aborta
5. Espera a chegada dos itens do Fornecedor, quando chegarem vai para 6
6. Cria uma lista de Items a serem enviados ao cliente
7. Envia os itens ao cliente
8. Envia um recibo ao cliente

9. Termina o processo

Aqui foi apresentado este tipo de grafo, e não nenhum dos restantes apresentados na introdução uma vez que a maioria dos sistemas de gestão de workflow comerciais, pelo menos ao nível de GUI (*Graphical User Interface*), usam este tipo de grafos.

### 3 Execução de um Sistema de Workflow

As instâncias de workflow são executadas pelo *Motor de Workflow*. Este motor é basicamente um escalonador, isto é, organiza o trabalho a ser feito e atribui-o aos diversos recursos disponíveis ao sistema.

Pode descrever-se a operação de um motor de workflow da seguinte forma:

Quando é instanciado um novo workflow, o motor vai a um repositório buscar a definição desse workflow. Em seguida activa os nodos a serem executados, no caso os directamente ligados aos nós de início. Se forem nós de encaminhamento vai avaliar as condições presentes e escolher qual dos caminhos a seguir, e o nó a ser executado a seguir. Se for um nó de trabalho então selecciona um recurso ao qual atribuir o trabalho, colocando-o la numa fila atribuída aquele recurso. E quando o recurso estiver pronto a processar uma nova tarefa, vai busca-la a sua fila, executa-a e devolve o resultado ao motor de workflow (colocando-o na fila do motor de workflow).

O trabalho referido anteriormente pode assumir variadas formas, como enviar uma SMS, enviar um formulário a ser preenchido por um utilizador, executar um comando numa base de dados, mas para o WfMS são todos iguais, não passando de mensagens que podem ser executadas por um determinado recurso. A maneira como o recurso as trata, é completamente ignorada pelo WfMS.

Quando se refere que cada recurso tem uma fila de espera de trabalho não implica que o trabalho seja executado de forma first in first out, podendo este escolher o que quer fazer. Esta comunicação entre partes do sistema e efectuada através de apis que serão discutidas mais a frente.

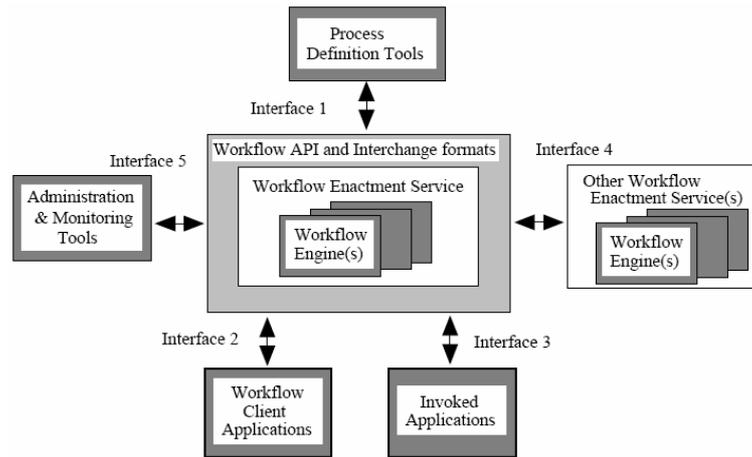
## 4 O Standard Definido pela WfMC (Workflow management Coalition)

### 4.1 Modelo de referencia da WfMC

O Modelo de referencia de workflow foi criado a partir da estrutura de uma aplicação genérica, identificando as interfaces que permitem aos produtos interagir aos mais diversos níveis. Como diversos produtos apresentam níveis de capacidade diferentes em cada modulo é necessário definir interfaces entre estes, de forma a que possa existir interoperabilidade entre produtos.

Na figura estão identificados os principais componentes e interfaces, estes serão analisados em seguida.

A interface que envolva o *Workflow Enactment Service* designa-se por WAPI - Workflow API, e possui formatos de intercâmbio. Esta interface pode ser considerada como um conjunto de pontos por onde os serviços do sistema de workflow podem ser acedidos. E que regulam as interacções entre o software de controlo de workflow e os restantes componentes do sistema. Uma vez que como muitas das funções das diversas interfaces são comuns torna-se mais apropriado considerar a WAPI como uma interface de serviço unificada que suporta funções de gestão de workflow sobre 5 areas funcionais.



**Figura 2.** Modelo de Referência de Workflow

#### 4.1.1 Workflow Enactment Services

O Workflow Enactment Service disponibiliza um ambiente onde ocorre a inicialização e activação de processos (onde os vários processos de workflow podem correr). Usando um ou mais motores de gestão de workflow (Workflow Management Engines), responsáveis por interpretar e activar parte ou todo o processo, e a comunicação com recursos externos necessários a realização das várias actividades.

*Definição - Workflow Enactment Service* Um serviço de software que consiste em um ou mais motores de workflow com o objectivo de criar, gerir e executar instâncias de workflow. As diversas aplicações podem interagir com este serviço pela Workflow Application Programming Interface (WAPI).

#### 4.1.2 Motor de Workflow

O Motor de Workflow (*Workflow Engine*) é responsável por parte ou pelo total controlo do ambiente dentro de um Enactment Service. Sendo capaz de controlar a execução de um determinado conjunto de instâncias de processos ou sub-processos com um fim. Determinado por uma quantidade de objectos e seus atributos, que consegue interpretar de entre as definições do processo.

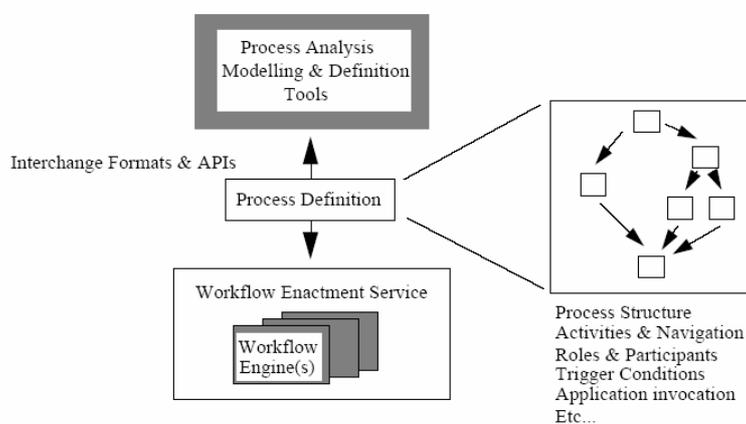
*Definição - Workflow Engine* Um Serviço de software ou "Motor" que disponibiliza o ambiente de execução para uma instância de workflow

Tipicamente este tipo de software disponibiliza primitivas para tratar de:

- Interpretação da definição do processo
- Controlo de instâncias de processos - criação, activação, suspensão, terminação, etc.
- Navegação entre as várias actividades do processo, que podem envolver operações sequenciais ou paralelas, imposição de prazos, etc.
- Entrada e saída dos participantes

- Identificação de itens que requerem atenção do utilizador, e uma interface para a intervenção deste
- Manutenção dos dados relativos ao workflow, e encaminhamento destes entre os vários participantes
- Uma interface com capacidade de invocar aplicações externas, e passagem de dados importantes
- Acções de supervisão para controlo, administração e auditorias.

#### 4.1.3 Process Definition Interface (Interface 1)



**Figura 3.** Modelo de Referência de Workflow(Pormenor)

Esta interface permite uma ligação entre as ferramentas de definição de processos e os ambientes onde são executados (runtime workflow management).

Chamadas API para aceder as definições de processo:

*Estabelecimento de Sessões*

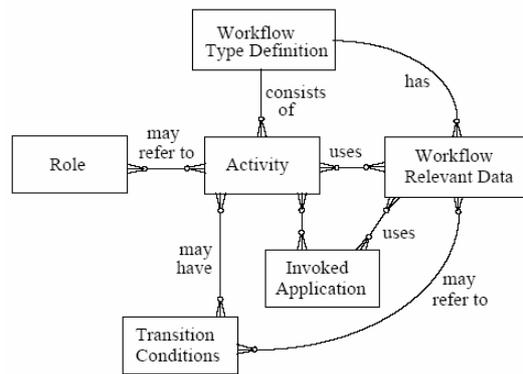
- Estabelecer/cortar as ligações entre os vários componentes

*Operações de Definição de Workflow*

- Obtenção das listas de definição de processos de workflow
- Selecção/De-Seleccção da definição de um determinado processo para obter uma handle de sessão para outras operações a nível de objectos (outros workflows, etc)
- Leitura e escrita de objectos de definição de workflow de topo.

*Operações sobre objectos de definição de Workflow*

- Criar, remover e consultar objectos dentro de uma definição de workflow



**Figura 4.** Meta Modelo Básico de Definição de processos

- Consultar, alterar, e remover atributos de atributos.

Meta-Modelo básico.

Para o Meta-Modelo básico de definição de workflow, foram definidos vários componentes, com os mais variados atributos, que são enumerados em seguida: *Definição de tipo de Workflow*

- Nome do processo de workflow
- Número de Versão
- Condições de início e fim
- Informação de Segurança, Auditoria e outro controle

*Actividade*

- Nome da Actividade
- Tipo de actividade
- Pré e pos condições da actividade
- Outro tipo de restrições de escalonamento

*Condições de transição*

- Condições de fluxo ou de execução

*Dados relevantes de Workflow*

- Nome e caminho para os dados
- Tipos de Dados

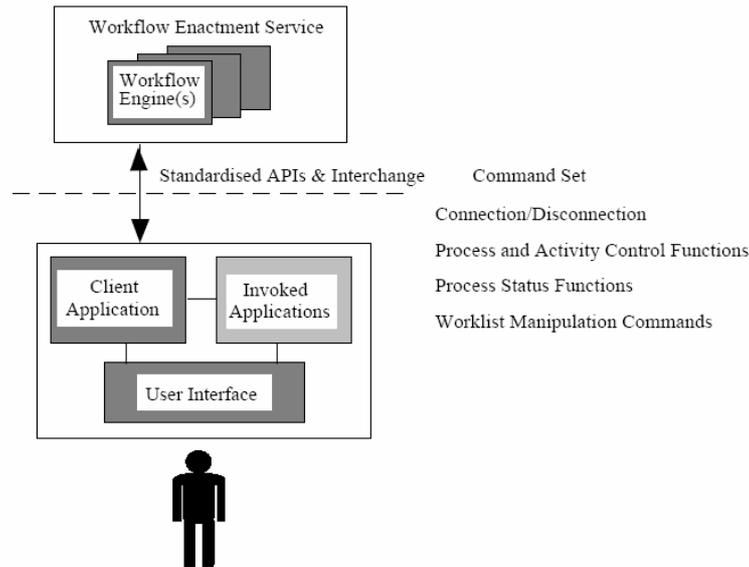
*Papel*

- Nome e entidade Organizacional

### *Aplicação Invocada*

- Tipo ou nome Genérico
- Parâmetros de execução
- Localização ou caminho de acesso

#### 4.1.4 Workflow Client Application Interface (Interface 2)



**Figura 5.** Modelo de Referência de Workflow(pormenor)

Estas aplicações interagem directamente com o utilizador quando é necessário algum tipo de intervenção por sua parte. Podendo ser distribuídas como parte do pacote de software de WfMS, ou escritas pelos utilizadores.

A interface entre a aplicação cliente e o motor de workflow deve ser suficientemente flexível para permitir vários tipos de implementações uma vez que é dada a opção aos utilizadores de a modificarem.

Em termos funcionais as funções da API subdividem-se em:

#### *Estabelecimento de sessões*

- Estabelecimento/corte de sessões (ligações) entre os sistemas

#### *Operações de definição de Workflow*

- Obtenção/Consulta (que pode ou não ter critérios de selecção) de nomes ou atributos

#### *Controlo de Processo*

- criar/iniciar/terminar uma determinada instância de um processo

- suspender/resumir uma instância de um processo
- forçar uma mudança de estado de uma instância de um processo
- alterar/adicionar/consultar um atributo(ex: prioridade) de um processo

#### *Estado de processo*

- abrir/fechar um processo ou actividade, Consultar usando filtros
- Obtenção de detalhes de instância(s) de processo(s), ou uma actividade(s), com possibilidade do uso de filtros

#### *Tratamento de Worklists/Workitems*

- Abrir/fechar uma consulta a uma Worklist (lista de tarefas), podendo ser filtradas
- Obtenção de itens da Worklist, com opções de filtragem
- Notificação da selecção/redistribuição/finalização de um workitem específico
- Alteração ou consulta de um atributo de um Workitem

*Supervisão de processos* (Estas funções têm como objectivo permitir a supervisão e alteração de todos os processos, e tem como intenção de ser acessíveis apenas a utilizadores com um nível de privilégios elevado)

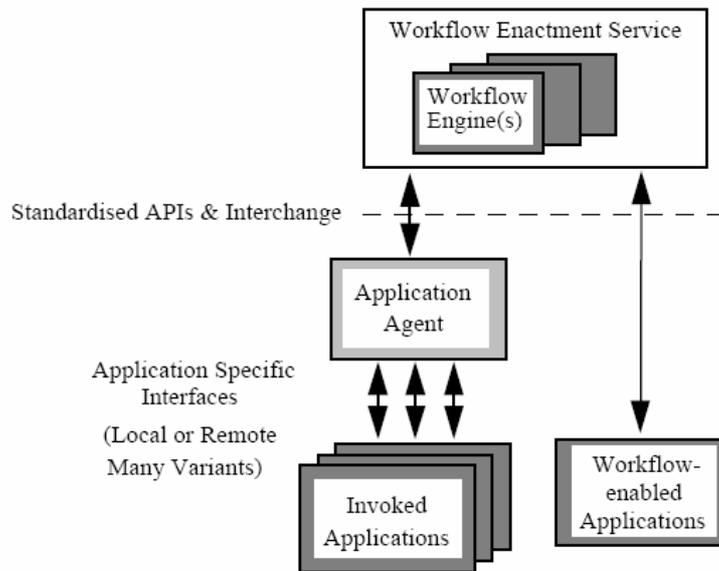
- Alterar o estado operacional de uma definição de um processo de workflow e/ou as instâncias desse processo
- Alterar o estado de todas as instâncias de processos ou actividades de um determinado tipo
- Adicionar/Alterar atributos a todos os processos de um determinado tipo
- Terminar todas as instâncias de um processo

#### *Tratamento de Dados*

- Obtenção / retorno de dados de aplicação ou informação necessária a continuação do workflow.

*Funções de Administração* Podem ser incluídas funções de administração adicionais caso se verifique ser apropriado para um determinado tipo de cliente.

*Invocação de Aplicações* As funções referidas acima permitem um nível básico para invocação de aplicações, por permitirem acesso a atributos de processos/actividades.



**Figura 6.** Modelo de Referência de Workflow(Pormenor)

#### 4.1.5 Invoked Aplicativos Interface (Interface 3)

Pode-se assumir com algum grau de certeza que qualquer implementação de um WfMS não tem capacidade para invocar todas as aplicações potencialmente necessárias. Então para um determinado tipo de aplicações que têm tipos de dados bem definidos pode usar-se um mecanismo de comunicação standard (o X.400 é dado como exemplo :)). Para um outro tipo de aplicações cria-se uma "aplicação middleware" que usa uma API standard para comunicar com o Workflow Enactment Service, ou então é criada uma nova aplicação de raiz que use estas APIs.

Em traços gerais as APIs devem ter funções para:

*Estabelecimento de sessões*

- Estabelecimento/corte de sessões (ligações) entre os sistemas

*Gestão de Actividades Direcção Workflow engine - Aplicação*

- Iniciar actividade
- Suspende/Retomar/Abortar actividade

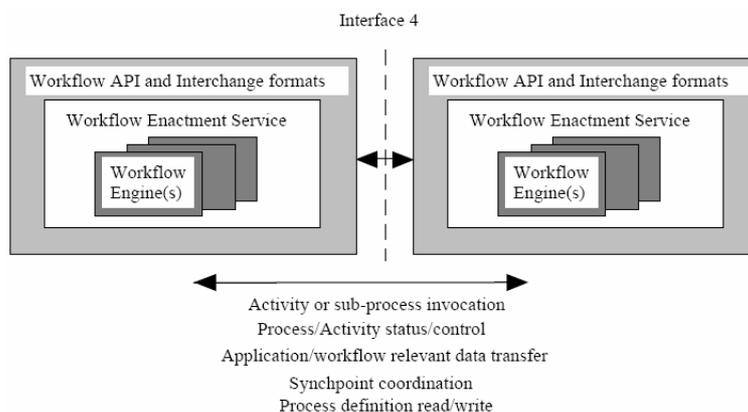
*Direcção Aplicação - Workflow engine*

- Notificação de conclusão de actividade
- Sinalização (para sincronização por exemplo)
- Obtenção de atributos da actividade

*Tratamento de Dados*

- Trocar informação relevante (parâmetros necessários a aplicação, e informação necessária a continuação do processo de workflow)
- Envio de dados à aplicação

#### 4.1.6 Workflow Interoperability(Interface 4)



**Figura 7.** Modelo de Referência de Workflow(Pormenor)

Um dos objectivos chave desta definição de standards é definir como podem 2 WfMS de fabricantes diferentes interagir entre si.

Existem 2 aspectos necessários para que possa haver interoperabilidade:

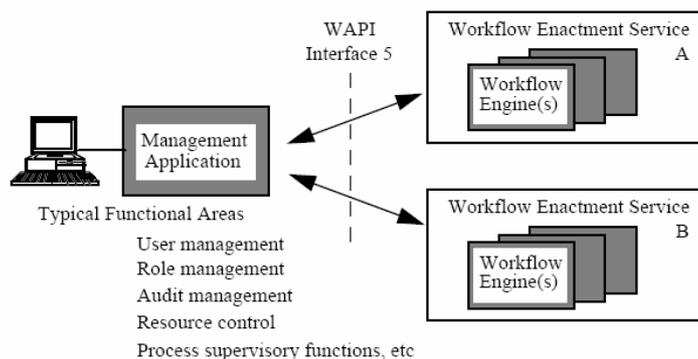
- É necessária uma interpretação comum da definição dos processos, que pode ser obtida através do uso de um mesmo process building tool, exportando atributos dos processos, ou através do uso de uma gateway que faz um mapeamento dos objectos e atributos.
- Suporte para transferência de dados de controlo, informação relativa ao estado do processo de workflow. Durante a execução são usadas funções da WAPI para acesso a serviços, falando os workflow engines directamente uns com os outros, caso isto não seja possível constrói-se então uma gateway.

Algumas das funções referidas atrás podem ser usadas para a interoperabilidade entre servidores como:

- Estabelecimento de Sessões
- Operações sobre definição de workflow e objectos
- Controlo e auditoria de processos
- Gestão de Actividades
- Tratamento de dados

Também se torna necessário um certo nível de gestão centralizada entre os vários WfES

#### 4.1.7 Administration and monitoring interface(Interface 5)



**Figura 8.** Modelo de Referência de Workflow(Pormenor)

Também foi definida uma interface para administração e auditoria/monitorização dos sistemas *Operações de gestão de Utilizadores*

- Estabelecer/apagar/suspender/corrigir privilégios de utilizadores ou Workgroups

*Gestão de Papeis*

- Estabelecer/apagar/corrigir papéis; relações entre participantes

*Auditoria a operações de gestão*

- Operações de pesquisa, apagar, impressão e começar de novo sobre logs

*Operações de controlo de recursos*

- Adicionar, remover ou modificar as características de concorrência de uma actividade ou processo

*Operações de Supervisão de Processos*

- Mudar o estado de uma definição de um processo de workflow, e/ou das suas instâncias
- Escolher de entre as várias versões da definição de um processo
- Mudar o estado de todas as instâncias de processos ou Actividades de um certo tipo
- Atribuir novos atributos a todas as instâncias de processos ou actividades de um certo tipo
- Terminar todas as instâncias de processos.

*Operações de Auditoria de Processos*

- Abrir ou fechar Consultas a instâncias de actividades ou processos, com filtros opcionais
- Consulta de detalhes sobre instâncias de processos ou actividades, com filtros

## 5 Programação com Sistemas de Workflow

Um sistema de workflow é semelhante a programar numa linguagem de 3ª geração. É de facto caracterizado por invocar determinadas funções de acordo com uma dada lógica de fluxo. As condições e ciclos são modeladas num sistema de workflow por nós de encaminhamento. Em adição, assim como nas linguagens de programação, um sistema de workflow permite a declaração de variáveis, que podem ser passadas como input ou recebidas como output de invocações referidas pelo nó de trabalho. Estas variáveis são então usadas para avaliar as condições de encaminhamento, e transferência de dados entre nós.

Apesar das semelhanças um sistema de workflow apresenta um grande número de diferenças para com as linguagens de programação ditas normais. Uma primeira é em termos de escala, tipicamente um processo invocado num sistema tem uma curta duração, enquanto que um sistema de workflow as invocações são esporádicas mas de longa duração, estendendo-se por horas ou dias. Uma outra e também importante diferença prende-se com a granulosidade da composição de um sistema de gestão de workflow. Sistemas de workflow são tipicamente compostos por módulos de software de grandes dimensões, tipicamente aplicações completas e de complexidade elevada.

Os sistemas de workflow, tipicamente não são começados de início para um determinado propósito, dando uso as aplicações que já se encontravam em produção anteriormente no sistema, tornando-se assim eles também sistemas de EAI (Enterprise Application Integration).

Como linguagem para EAI os sistemas de workflow vêm-se obrigados a implementar funcionalidades que vão para além das tipicamente encontradas numa linguagem de programação dita normal. Uma das consequências da longa duração de um processo de workflow é a necessidade de implementar primitivas de recuperação de falhas. Num programa normal, se há uma falha, reinicia-se, numa transacção aborta-se e faz-se rollback. No entanto nenhuma destas soluções é viável num sistema de workflow devido aos seus custos de execução (quer em termos de tempo ou financeiros), fazer rollback e reiniciar, mesmo que apenas uma pequena parte não é desejável, pois o roll back pode não ser fácil, se por exemplo se envia um cheque a um cliente, e depois o sistema falha, torna-se difícil fazer rollback do envio do cheque, ou a retransmissão do cheque, apesar de bom para o cliente, não o é para a empresa.

Para fornecer algum suporte a falhas um sistema de workflow permite tipicamente:

**Forward Recovery:** os WfMS mantêm o estado da instância de wf numa unidade de armazenamento persistente, sendo assim quando há uma falha, é passível retomar a execução da instância activa no momento da falha.

**Backward Recovery:** este tipo de recuperação serve para os casos onde não é possível continuar a execução e torna-se necessário desfazer uma execução parcial de uma instância de workflow. Isto normalmente é obtido através da atribuição a cada actividade, uma outra que tem o efeito oposto e serve para desfazer o que foi feito por essa actividade.

**Linguagens com tratamento de Excepções:** Alguns WfMS especialmente a nível académico oferecem primitivas de tratamento de excepções, incluindo as que podem ocorrer de modo assíncrono, como por exemplo o cancelamento de uma encomenda. Estes mecanismos detectam excepções, e tratam-nas ao estilo do try-catch do java.

**Prazos:** Como os WfMS são ferramentas para integrar diversas aplicações, têm de lidar com o problema de uma vez invocada a aplicação nunca chegar a responder, impondo então prazos no tempo de resposta destas. Sendo então posteriormente tratada a falha.

Uma outra diferença entre sistemas de programação típicos e sistemas de gestão de workflow são os recursos usados. Enquanto que num sistema típico os recursos são sempre o computador, num sistema de workflow os recursos podem variar conforme a tarefa a ser executada, de acordo com parâmetros da acção (diferentes encomendas de produtos podem ser tratadas por pessoas diferentes). Sendo para isto necessário que haja suporte a definição de regras de recursos, para identificar o recurso certo para uma dada actividade.

## 6 Integração com Outras Plataformas de Middleware

Os sistemas de workflow têm uma relação muito estreita com os conceitos apresentados nas aulas anteriores, possuindo características de Monitores transaccionais, por exemplo executam transacções em cima de aplicações autónomas, atribuição de recursos, gestão de performance e gestão de carga, permitindo a execução de milhares de operações complexas em simultâneo. Os WfMS dão ênfase à programação em larga escala, com atribuição dinâmica de recursos, e ao tratamento de excepções, em vez da utilização de RPCS transaccionais. Os WfMS também elevam a um novo patamar os brokers do tipo ORB, MOMS ou arquitecturas produtor/consumidor. Pelo menos uma destas tecnologias faz parte de qualquer WfMS para permitir interoperabilidade entre os vários participantes.

Como foi dito antes os WfMS e plataformas EAI são complementares, as plataformas EAI tratam da heterogeneidade das aplicações, dando uma plataforma homogénea sobre a qual trabalhar. E os WfMs mantêm a lógica geral das aplicações e da integração dos sistemas.

Pode então constituir-se uma arquitectura com uma plataforma EAI, por exemplo um broker de mensagens que esconde a heterogeneidade do sistema, e colocando um adaptador WfMS, que funciona como único recurso para o sistema de gestão de workflow, estando este responsável pela distribuição das tarefas pelos diversos recursos disponíveis.

## 7 Benefícios e limitações dos WfMS

As primitivas apresentadas anteriormente ofereceram muitos benefícios no que respeita à codificação do processo de negócio no que respeita à composição de aplicações de baixa granulosidade e interacções com utilizadores. Proporcionando um processo rápido de desenho e manutenção, tratamento de falhas e excepções, tendo em atenção a performance e elevada disponibilidade. O desenho de sistemas de workflow é tipicamente feito por um ambiente de desenvolvimento gráfico, que facilita em muito a transposição de processos de negócio complexos para o sistema.

Pelas razões anteriormente referidas os sistemas de gestão de workflow na altura geraram um grande interesse, mas tinham algumas falhas, tornando-se difíceis de implementar, não cumprindo as expectativas que se tinham neles, e licenças extremamente caras, levaram a um declínio da sua utilização na altura. No entanto provaram o seu valor em sistemas repetitivos e bem definidos, facilmente formalizáveis usando linguagens de workflow, mas estes já eram suportados por outras tecnologias de middleware.

Hoje em dia o modelo de workflow está a aumentar a sua popularidade de novo, especialmente para computação em grelha, e clusters, e na composição de web services.

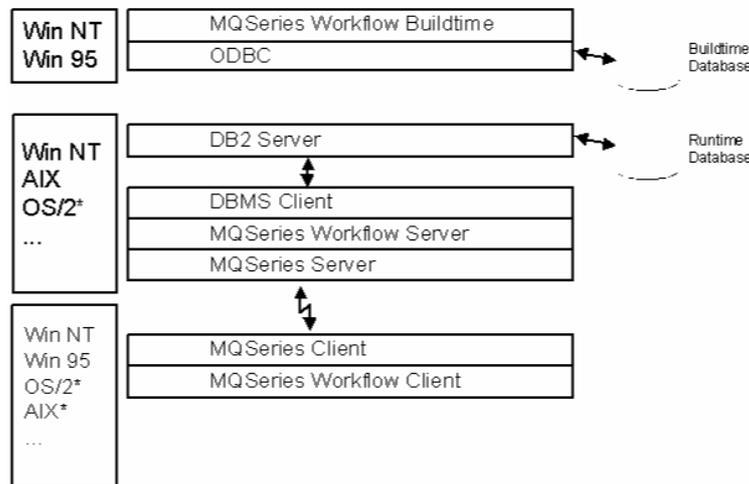
## 8 Um Exemplo de um sistema de gestão de Workflow "Websphere MQ Workflow"

Vamos agora ver com algum detalhe uma plataforma que implementa algumas das especificações apresentadas anteriormente. Trata-se do IBM "Websphere MQ Workflow". Esta plataforma, assenta sobre qualquer coisa.

### 8.1 Características do IBM WebSphere? MQ Workflow V3.5

- Suporte a processos de negócio de longa duração
- Fornece integração de processos, com amplo suporte a interacções por parte de utilizadores
- Permite um ambiente único com capacidade de integrar sistemas e pessoas
- Oferece uma ampla conectividade, com interfaces para XML, J2EE, Web Services
- Permite integração com o WebSphere Business Integration Modeler and Monitor, para análise, simulação e monitorização de melhoramentos no processo.

### 8.2 Estrutura Interna do MQ Workflow



**Figura 9.** Estrutura interna do MQ Workflow

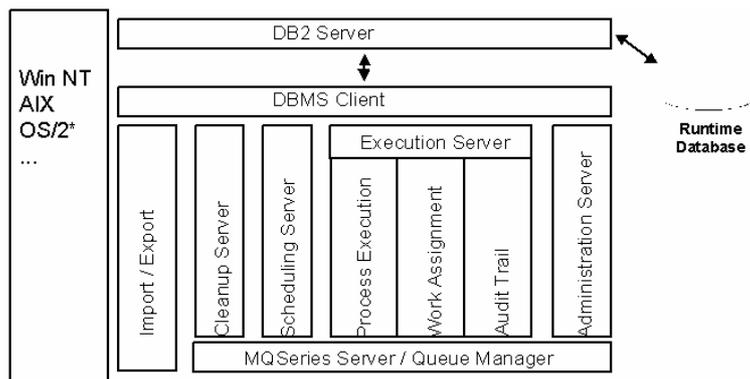
Como se pode reparar pelo esquema, uma das principais características do MQ Workflow é que usa duas bases de dados diferentes, em Tempo de construção, e em tempo de execução.

Em tempo de execução existe uma instância da base de dados a correr no servidor de workflow, assim as características de acesso a base de dados podem ser melhor optimizadas.

O facto de haver duas bases de dados separadas permite:

- Cada uma das bases de dados está otimizada para o seu propósito. a base de dados de tempo de configuração está preparada para suportar transacções de longa duração. enquanto que a base de dados usada em tempo de execução deve estar preparada para padrões transaccionais, ou seja grande quantidade de transacções mais curtas.
- Pode trabalhar-se na modelação sem qualquer impacto nos processos que estão em execução

### 8.3 Componentes do Servidor



**Figura 10.** Componentes do Servidor MQ Workflow

As comunicações entre clientes e servidor estão a cargo do MQ Series, sendo para o efeito usada as primitivas de filas deste, quando um componente deseja comunicar com outro, a mensagem é enviada usando o nome da fila do componente apropriado.

Porquê usar o MQ Series para o tratamento das mensagens:

- Suporte a múltiplos protocolos.
- Independência de rede/ protocolos.
- Robustez da comunicação.
- Permite mensagens voláteis ou não voláteis.
- Permite o uso do triggering implementado nesta camada para arranque dos componentes do sistema

Todos os servidores de um sistema têm de partilhar um "MQ Series Queue Manager", que tem obrigatoriamente de estar na mesma maquina.

Tarefas dos componentes do servidor:

#### 1. Administration Server:

- Autenticação
- Gestão de Sessões

2. Cleanup Server é responsável por periodicamente apagar instâncias de processos de itens de trabalho
3. O Scheduling Server é responsável por:
  - Envio de Notificações
  - Continuação ao fim de um intervalo de tempo ou segundo um plano.
4. Execution Server sendo o componente central do Websphere MQ Workflow tem como responsabilidades:
  - Execução de processos
  - Navegação entre processos
  - Distribuição e Atribuição de trabalho
  - Tratamento de Itens de Trabalho
  - Queries
  - Logging

Quando um Servidor é inicializado ocorrem as seguintes acções:

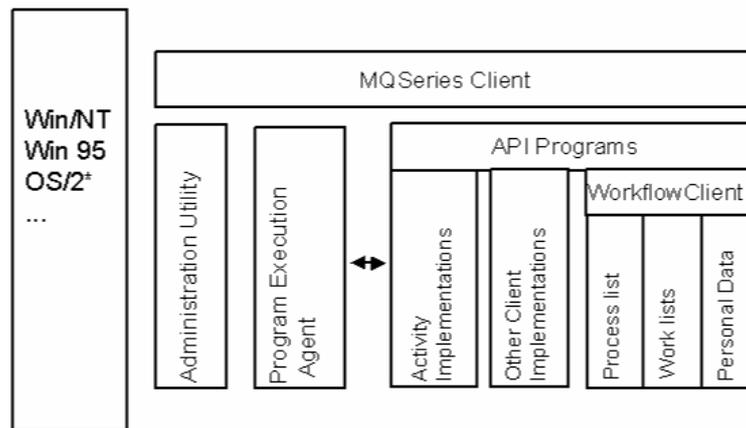
- Inicialização
  - Ligar-se ao MQ Series
  - Ligar-se a Base de Dados
  - Obter as definições do servidor
- Loop Principal
  - Começar a transacção global
  - Obter uma mensagem da fila de entrada
  - Processar a mensagem, e opcionalmente, actualizar a base de dados
  - Por a resposta e outras mensagens na fila de saída do servidor
  - Fazer um two fase commit a transacção.

#### **8.4 Componentes do Cliente**

Como plataforma para comunicar os clientes usam o MQ Series como cliente, estando o queue manager do lado dos servidores, sendo necessário fornecer os dados do canal MQ durante o arranque do cliente.

Características do cliente MQ Workflow:

- Acedem à base de dados através do servidor
- Usam mensagens não persistentes na comunicação com o servidor



**Figura 11.** Componentes do cliente do MQ Workflow

- Usam a API de Workflow internamente, e providenciam tratamento de pedidos assíncronos
- São implementados usando ActiveX

O *Program Execution Agent* é executado apenas através de uma chamada a API, sendo também necessário uma API para o terminar, de forma a manter o servidor sincronizado. Este módulo usa mensagens persistentes para a comunicação, sendo também efectuada uma autenticação perante o servidor no início.

A *Administration Utility* é um executável que fornece uma CLI para funções administrativas, comunicando com o Servidor Administrativo, e permite as seguintes operações:

- Iniciar/Parar Servidores
- Consultar o estado de Servidores
- Recolher e visualizar mensagens de estado

### 8.5 Ciclo de Vida de um Workitem

1. Um workitem é iniciado pelo cliente de Workflow, através das APIs providenciadas
2. O Workitem é colocado na fila, com o Execution Server como destino
3. O Execution server apanha-o
4. O Execution Server actualiza a base de dados, e envia-o para a fila do Program Execution Agent
5. A partir da fila o PEA apanha o workitem e processa-o
6. Através das APIs o cliente é informado com facto que acabou o processamento daquele item

## 8.6 Usando o MQ Series Workflow (Um Exemplo)

Como exemplo aqui vamos desenvolver um processo de negócio para uma empresa que vende livros. Tendo como objectivo implementar uma solução de workflow que permita o processamento de uma ordem de compra. Para este exemplo não se vão construir programas, usando apenas programas já existentes como o Windows Notepad e uma GUI simples já providenciada pelo MQ Workflow.

### 8.6.1 Definição do exemplo

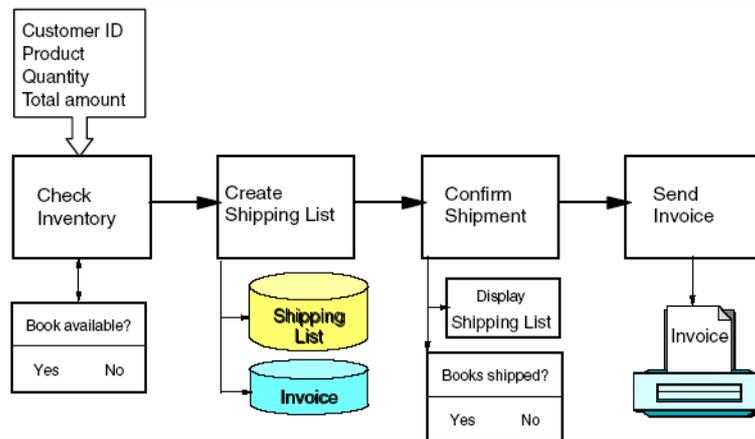


Figura 12. Processo da encomenda

1. Em primeiro lugar temos a entrada da encomenda, que pode ser feita usando uma das características do cliente do MQSeries Workflow. Que permite a introdução de dados quando uma instância é iniciada.
2. A primeira actividade é verificar se existe em stock, isto será feito por uma pessoa. Um programa pergunta se a quantidade encomendada pelo cliente existe em stock, com a opção de resposta sim/não
3. Após verificar o inventário, cria-se uma lista de items a enviar, isto será feito sem intervenção humana, por um programa Windows NT que também cria um recibo, para ser usado mais tarde
4. Com a lista de items a enviar um empregado então embala-os e envia-os ao cliente, Dando uma resposta de sim/não ao sistema por uso de um programa para o efeito. Para a consulta da lista de items, usará o notepad.
5. O ultimo passo será o envio do recibo ao cliente, um empregado abre-o usando o notepad, imprime e envia por correio ao cliente.

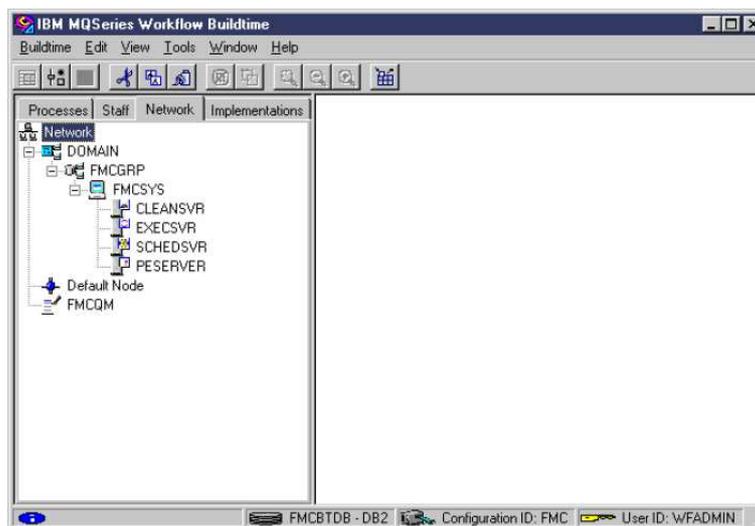
### 8.6.2 Usando o MQSeries Workflow Buildtime

Para utilizar o MQSeries Workflow Buildtime o MQSeries Workflow Server não necessita de estar em execução, devido à separação de bases de dados existentes entre estes. No entanto a base de dados de Buildtime tem de se encontrar em execução.

**Em seguida são necessários os seguintes passos para definir um processo:**

1. Fazer login no Servidor de Buildtime
2. Definir as pessoas (staff) que participam no processo
3. Definir estruturas de dados (Containers)
4. Especificar os programas que participam no processo
5. Modelar o processo de negócio

**Após feito o login aparece o seguinte écran:**



**Figura 13.** MQSeries Workflow Buildtime - GUI (Vista da Rede)

Os diversos componentes do sistema que aparecem na janela são descritos abaixo:

**DOMAIN** Top Level domain (por exemplo empresa)

**FMCGRP** Grupo a que a maquina pertence (por exemplo departamento)

**FMCSYS** Maquina onde está instalado

**CLEANSVR** Cleanup Server

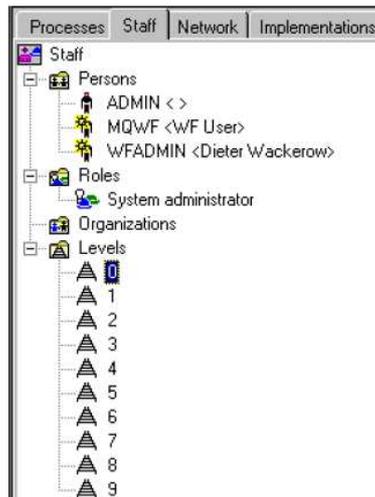
**EXECSVR** Execution Server

**SCHEDSVR** Scheduling Server

**PESERVER** Process Execution Server

**FMCQM** Queue Manager

## Definindo o staff



**Figura 14.** MQSeries Workflow Buildtime - GUI (Staff)

Na figura 14 é possível ver os vários membros do Staff e definições de papéis, Organizações a que pertencem e níveis. Estes componentes estão detalhados mais abaixo. Neste ecrã também é possível alterar e definir novos membros de staff, roles, etc...

**Persons** São as pessoas que participam num processo, pertencem a uma organização e tem determinadas tarefas

**Roles** São as descrições abstractas das tarefas

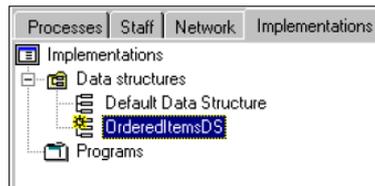
**Organizations** São unidades organizacionais, as quais as pessoas pertencem

**Levels** Um numero que pode ser usado para o que se entender, como rank, ou capacidades

## Definindo estruturas de dados

Na figura 15 pode ter-se a vista das estruturas de dados definidas. Para criar uma nova estrutura de dados, basta usar o menu que se acede com o botão direito do rato, e em seguida escolher "Properties", para definir os atributos da estrutura.

Na Figura 16 pode ver-se o que uma estrutura de dados pode conter, tendo vários atributos, de vários tipos, para o caso uma String e dois longs. Trata-se da informação de um produto. Nesta janela é possível adicionar ou remover campos a uma determinada estrutura de dados.



**Figura 15.** Vista da nova estrutura de dados



**Figura 16.** Propriedades de uma estrutura de dados

Será agora necessário criar outra estrutura de dados para a encomenda, que ha-de conter mais campos como ClientID quantidade, e uma estrutura do tipo produto.

### Definindo Programas a usar

Na janela da Figura 15 também é possível criar novos programas a serem usados no workflow, da mesma forma que se criam novas estruturas de dados.

Na figura 17 é possível definir que tipos de dados o programa trata.

Nesta figura 18 é possível definir qual o programa a ser executado e parâmetros de entrada.

**fmcibes** trata-se de um programa capaz de receber uma string, fazer uma pergunta ao utilizador e retornar a resposta do utilizador.

os % servem para delimitar uma variável, que se encontram na estrutura de input.

### Modelando o Processo

Em primeiro lugar é necessário definir o processo a ser usado, para o caso, o MQSeries Buildtime fornece uma GUI do tipo point and click (Fig:20).

A cada uma das actividades é possível alterar as suas propriedades, tipos de dados, etc usando a seguinte dialog box(Fig: 21):

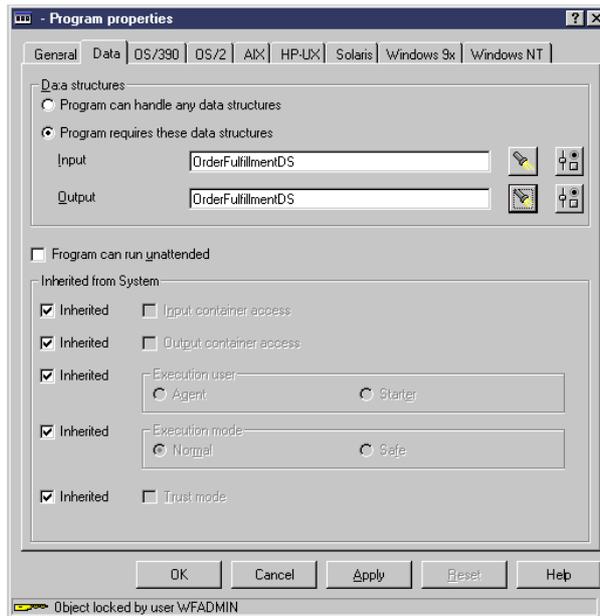


Figura 17. Tipos de Dados a tratar

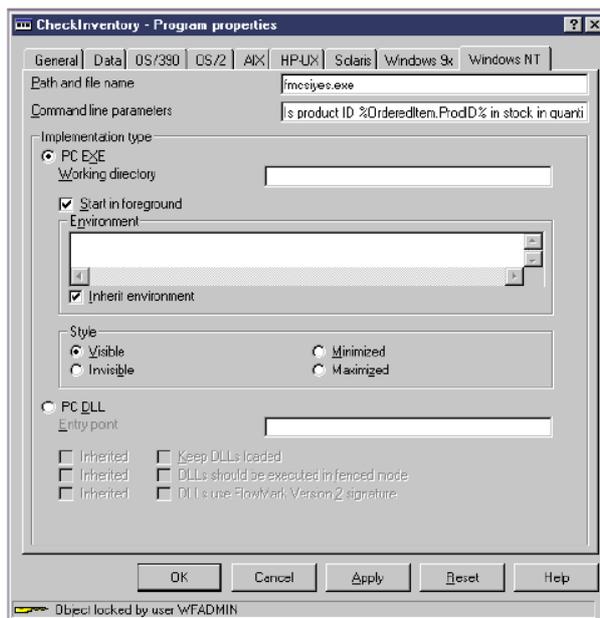
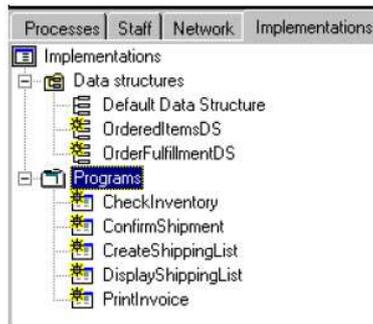


Figura 18. Propriedades de definição de um programa - WindowsNT

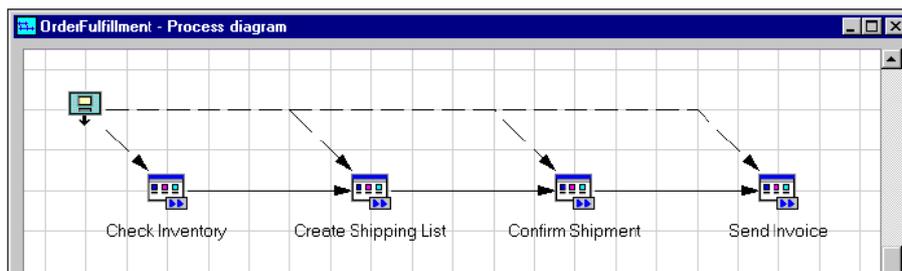
### 8.6.3 Usando o MQSeries Workflow Runtime

Para passar a definição de processo de buildtime para runtime, é feito através do uso do menu buildtime - export na janela de Buildtime, e em windows o import para runtime é feito apenas com um duplo click no ficheiro exportado pela ferramenta de buildtime(Fig: 22) .

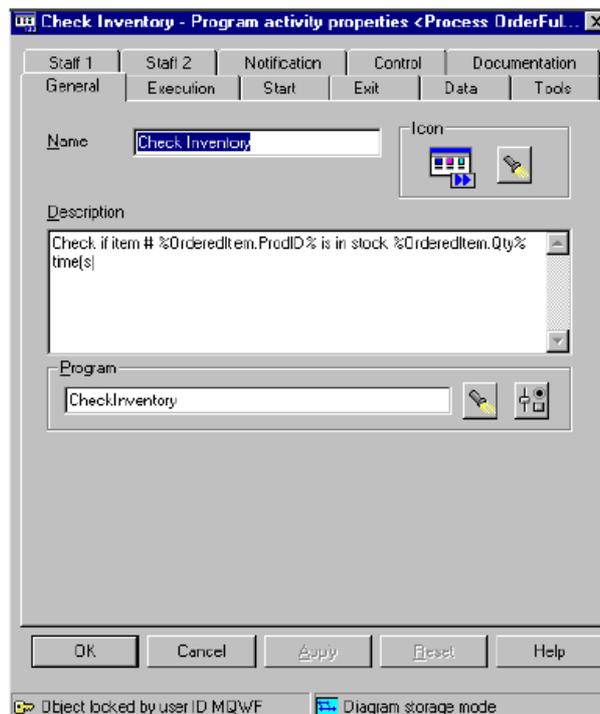
Para iniciar um novo processo basta fazer um duplo click no template e imediatamente é criada



**Figura 19.** Estruturas de dados e programas usados neste exemplo

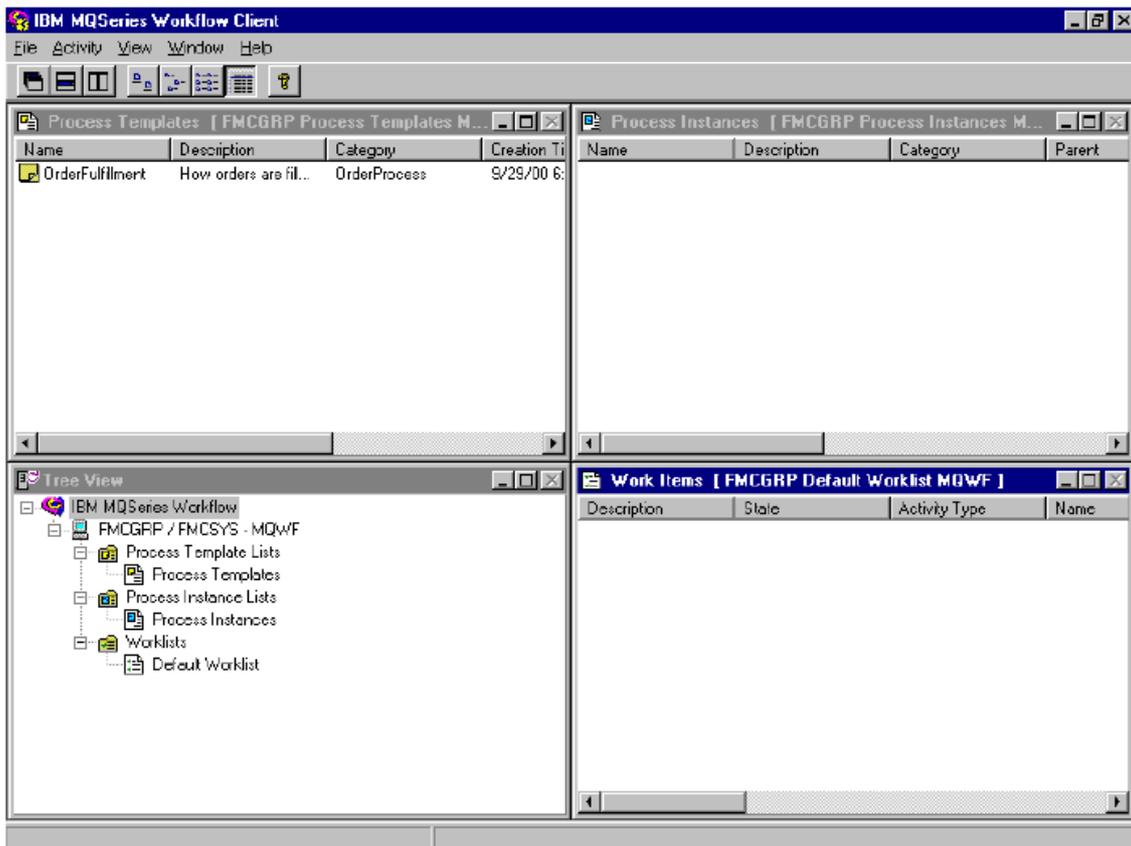


**Figura 20.** Processo de Encomenda já completo



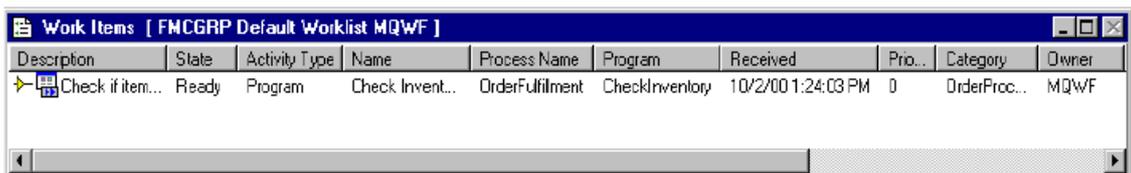
**Figura 21.** Propriedades de uma actividade

uma nova instância do processo. Sendo então pedido ao utilizador os dados da encomenda.



**Figura 22.** MQSeries Workflow Client

Nesta altura nos clientes apropriados aparece um novo workitem (Fig: 23) (depois de ter sido feito refresh a janela, uma vez que esta não é actualizada automaticamente), sendo esse workitem a verificação do item em stock



**Figura 23.** Worklist Mostrando um Item

Para iniciar o tratamento da worklist basta fazer duplo click neste.

Para um workitem que seja executado automaticamente o servidor encarrega-se disso.

Caso algum workitem necessite de alguma ferramenta adicional, então basta fazer click com o botão direito sobre o workitem e escolher *Start Support Tool* e em seguida escolher a ferramenta para o efeito, no caso de haver várias.

## 8.7 Programando com as APIs fornecidas pelo MQ

Durante a definição do exemplo anterior não se utilizou nenhum programa extra, mas nem todos os processos são assim tao simples, e apesar do MQSeries Workflow permitir a definição de parâmetros a serem passados aos programas que ele chama, também existe uma API que permite aos programas interagirem directamente. Passo a exemplificar um exemplo usando essa API abaixo:

```
/**
 * Este programa obtém a definição de um processo de workflow
 * e cria uma nova instância desse workflow
 */

import com.ibm.workflow.api.*; // import the workflow API's
import com.ibm.workflow.api.ServicePackage.*;

(...)

Agent agent = new Agent();

int locator = Agent.RMI_LOCATOR
String name = "MQWFAGENT";
agent.setLocator(locator);
agent.setName(name);

ExecutionService service = agent.locate("", "");

//executa o login no servidor de workflow
service.logon2(userid, passwd, SessionMode.PRESENT_HERE,
  AbsenceIndicator.LEAVE );

//obtem a definição do workflow relativo ao workflow desejado
ProcessTemplate[] proctempl = service.queryProcessTemplates
("NAME = 'OrderFulfillment'", "NAME", myint);

// obter o input container para introduzir os dados de entrada do workflow
ReadWriteContainer inputContainer = proctempl[0].inContainer();

// fill the input container: field CustID
inputContainer.setString("CustID", Order.custID);

// criar e iniciar um processo de workflow e introduzir la o input container
ProcessInstance procinst =
proctempl[0].createAndStartInstance2("", "", "", inputContainer, false);
```

```

// log off from MQW

service.logoff();

#####
/*
 * Programa externo que pode ser chamado durante um workflow
 * Código C usando a API do MQSeries Workflow
 */

//header files da API
#include <fmcjcrun.h>

//Inicializa a API, é obrigatório e deve ser a 1a chamada num programa
rc = FmcjGlobalConnect();

//Obter o input container
//Esta chamada é obrigatória se houver dados de input
rc = FmcjContainerInContainer( &hdlinctnr );

//Obter o output container (opcional)
rc= FmcjContainerOutContainer( &hdloutctnr );

//Obter uma string do input Container
//O valor do "CustID" é colocado na variável CustomerID
rc = FmcjContainerStringValue( hdlinctnr, "CustID", CustomerID, 5 );

//Obter informações sobre os dados do input container
len = FmcjContainerStringLength( hdlinctnr,"CustID" );

Escrever para um output container
rc=FmcjReadWriteContainerSetStringValue(hdloutctnr,"CustID",Customer,42);

//Enviar o output container para o PEA
rc = FmcjContainerSetOutContainer( hdloutctnr );

//Finalizar as ligações, etc
rc = FmcjGlobalDisconnect();

```

## 9 Links Úteis

Workflow Management Coalition:  
<http://www.wfmc.org/>

Workflow Management Coalition Reference Model:  
<http://www.wfmc.org/standards/docs/tc003v11.pdf>

IBM RedBooks:  
(<http://www.redbooks.ibm.com/>)

MQSeries Workflow for Windows NT for Beginners (ref: SG245848)  
(<http://www.redbooks.ibm.com/abstracts/sg245848.html?Open>)

Image and Workflow Library: MQSeries Workflow Concepts, Installation and Administration  
(ref: SG245375)  
(<http://www.redbooks.ibm.com/abstracts/sg245375.html?Open>)