

Workflow Management Systems

João Sequeira

Tecnologias de Middleware

28 Outubro 2004

Plano de Apresentação

Introdução

O que são WfMS

Background Histórico

Definição de Sistemas de Workflow

Execução de um Sistema de Workflow

O Standard Definido pela WfMC (Workflow management Coalition)

Modelo de Referência da WfMC

Programação com Sistemas de Workflow

Integração Com Outras plataformas Middleware

Benefícios e Limitações dos WfMS

Plano de Apresentação

Características do IBM WebSphere MQ Workflow V3.5

Estrutura Interna do MQ Workflow

- Componentes do MQ Workflow

- Componentes do Servidor

- Componentes do Cliente

Usando o MQSeries Workflow

- Definição do Exemplo

- Usando o MQSeries Workflow Buildtime

- Usando o MQSeries Workflow Runtime

Programando com as APIs fornecidas

- Criando um Processo

- Programa externo que pode ser chamado durante um workflow

Parte I

Modelo Teórico

O que são WfMS (Workflow Management Systems)?

Workflow

É a automatização de um processo de negócio, no seu todo, ou em parte, onde documentos, informação ou tarefas são passadas entre os diversos participantes, de acordo com um conjunto de regras procedimentais.

Workflow Management System (Sistema de gestão de Workflow)

É um sistema que define, cria e gere a execução de um ou vários Workflows, através do uso de software.

Background Histórico

- ▶ Origem com o *Office Automation*, baseados em e-mail, que mais tarde evoluíram para sistemas baseados na web.
Denominados **Workflows Administrativos**
- ▶ Aumento da necessidade de integrar aplicações leva a utilização de WfMS para definição da lógica de negócio tornando-se nos **Workflows de produção**
- ▶ Limitações nos sistemas de workflow levaram à sua integração com outras plataformas de EAI, como Message Brokers e afins

Definir um sistema de workflow

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 inicio e fim

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

Definir um sistema de workflow

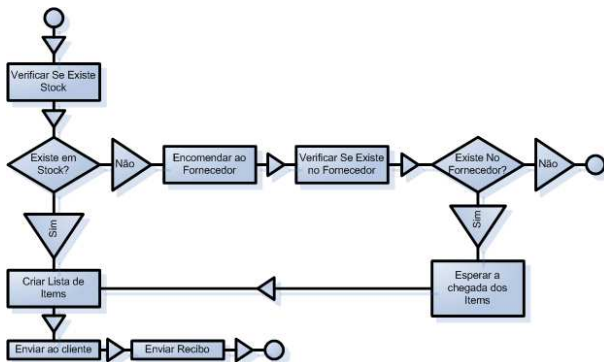


Figura: Processo da encomenda

Execução de um Sistema de Workflow

1. Obter a informação do processo da base de dados
2. Activa os nodos a serem executados. Caso seja um nó de encaminhamento, avalia-o e activa o proximo.
3. Selecciona o recurso e envia-lhe a tarefa.
4. Quando o Recurso está disponível processa a tarefa.

Modelo de Referência da Workflow Management Coalition

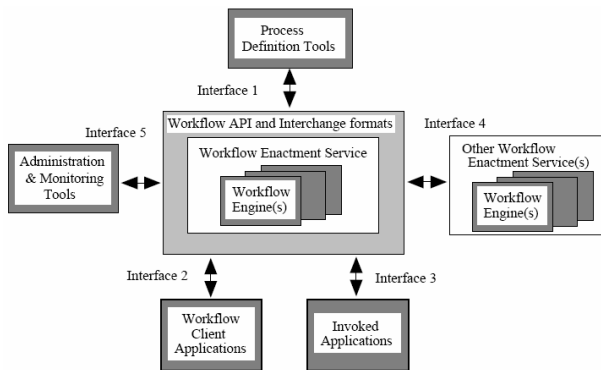


Figura: Modelo de Referência de Workflow

Workflow Enactment Services

Workflow Enactment Service

Um serviço de software que consiste num 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).

Motor de Workflow

Workflow Engine

Um Serviço de software ou "Motor" que disponibiliza o ambiente de execução para uma instância de workflow

Permite:

- ▶ Interpretação da definição do processo
- ▶ Controlo de instâncias de processos
- ▶ Navegação entre as várias actividades do processo
- ▶ Entrada e saída dos participantes
- ▶ Identificação de itens que requerem atenção do utilizador
- ▶ Manutenção dos dados relativos ao workflow, e encaminhamento destes entre os vários participantes
- ▶ Uma interface com capacidade de invocar aplicações externas
- ▶ Acções de supervisão, administração e auditorias.

Process Definition Interface (Interface 1)

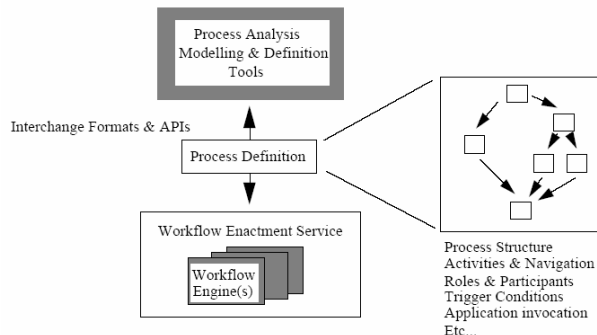


Figura: Modelo de Referência de Workflow(Pormenor)

Meta-modelo de Definição de Processos

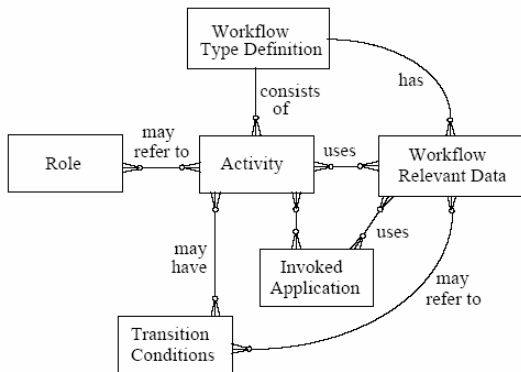


Figura: Meta-Modelo Básico de Definição de processos

Workflow Client Application Interface (Interface 2)

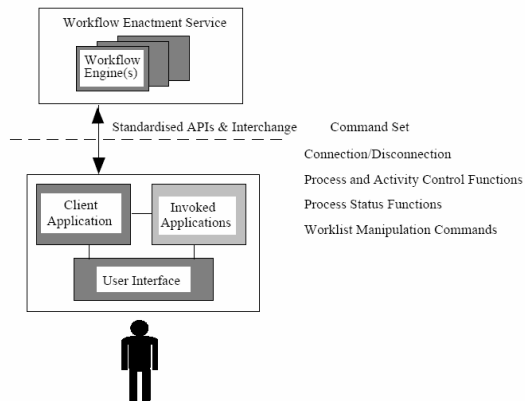


Figura: Modelo de Referência de Workflow(pormenor)

Invoked Applications Interface

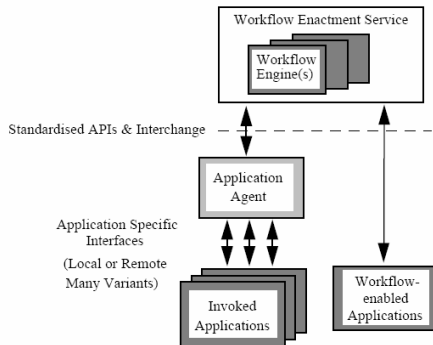


Figura: Modelo de Referência de Workflow(Pormenor)

Workflow Interoperability (Interface 4)

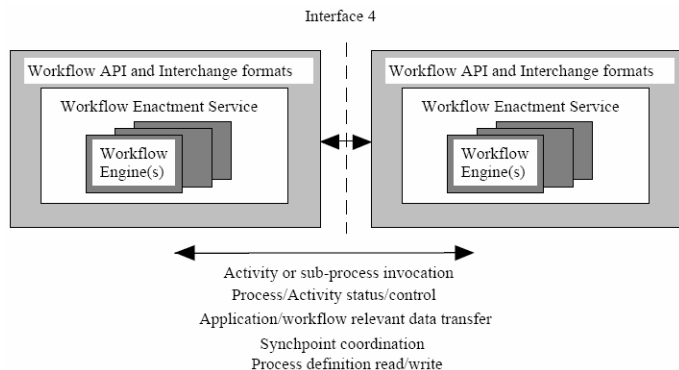


Figura: Modelo de Referência de Workflow(Pormenor)

Administration And Monitoring Interface (Interface 5)

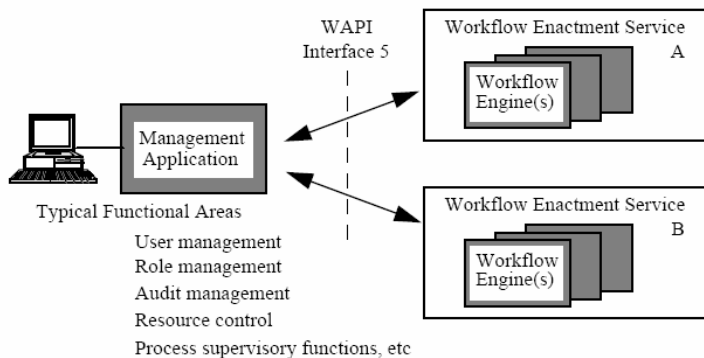


Figura: Modelo de Referência de Workflow(Pormenor)

Programação com Sistemas de Workflow

Semelhante a linguagens de 3a geração

- ▶ Invocar Funções de acordo com uma lógica de Fluxo
- ▶ Condições e ciclos
- ▶ Declaração de Variáveis

No entanto...

- ▶ Longa duração (horas ou mesmo dias)
- ▶ Módulos de grandes dimensões e complexos
- ▶ Necessitam de se integrar com o que já existe
- ▶ Necessitam de um forte suporte à recuperação de faltas

Suporte a Faltas dos Sistemas de Workflow

- ▶ Forward Recovery
- ▶ Backward Recovery
- ▶ Linguagens com tratamento de exceções
- ▶ Definição de prazos

Integração Com Outras plataformas Middleware

- ▶ Possuem características de monitores transaccionais
- ▶ Qualquer WfMS possui pelo menos um ORB ou MOM
- ▶ Plataformas Middleware (EAI) tornam o sistema homogéneo, e os WfMS mantêm a lógica organizacional e da integração dos sistemas
- ▶ Pode então ter-se um WfMS em cima de um Broker de Mensagens, sendo o broker o único recurso para o WfMS e estando responsável pela distribuição das tarefas (Mensagens)

Benefícios e Limitações dos WfMS

Benefícios:

- ▶ Fácil codificação do processo de negócio, quando composto por aplicações de baixa granulosidade e interações com os utilizadores
- ▶ Proporciona um processo rápido de desenho e manutenção, bem como tratamento de falhas e excepções
- ▶ Desenho do sistema tipicamente por ambiente gráfico que facilita a transposição dos processos de negócio

Limitações:

- ▶ Difíceis de implementar
- ▶ Não cumpriam expectativas
- ▶ Licenças muito caras
- ▶ Onde se provavam úteis já existiam aplicações middleware para suportar esses casos.

Parte II

Um Exemplo Real - IBM Websphere MQSeries Workflow

Características

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

Estrutura Interna do MQ Workflow

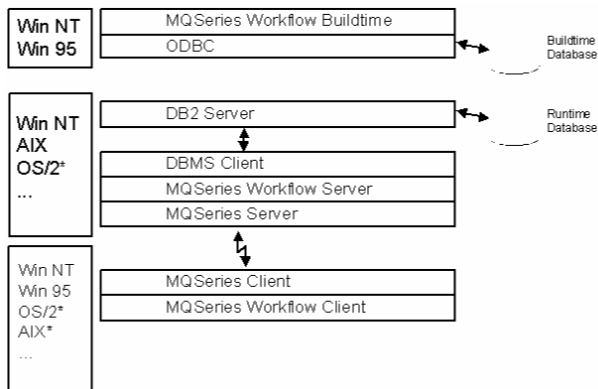


Figura: Estrutura interna do MQ Workflow

Modelo do Cliente

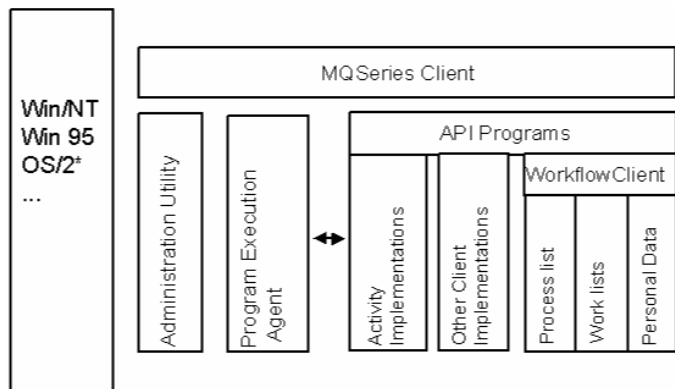


Figura: Componentes do cliente do MQ Workflow

Definição do processo de Workflow

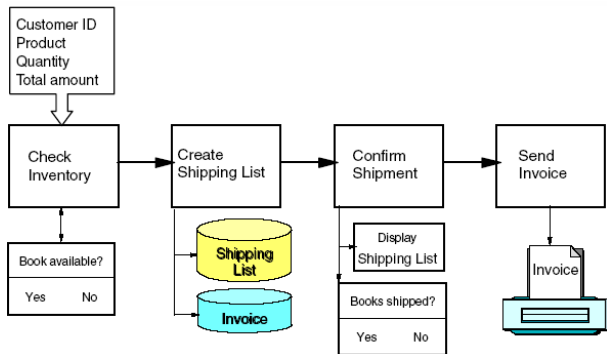


Figura: Processo da encomenda

MQSeries Buildtime

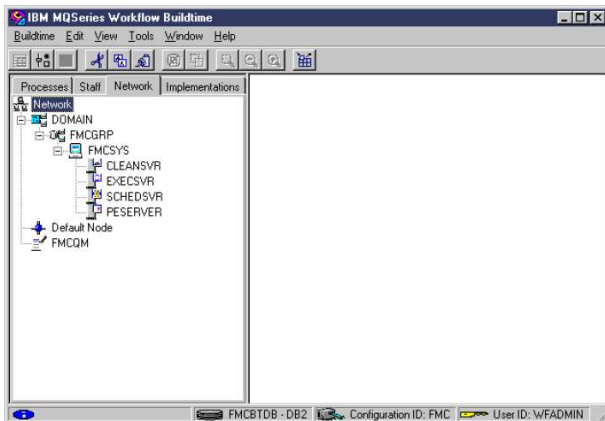


Figura: MQSeries Workflow Buildtime - GUI (Vista da Rede)

Definindo o Staff

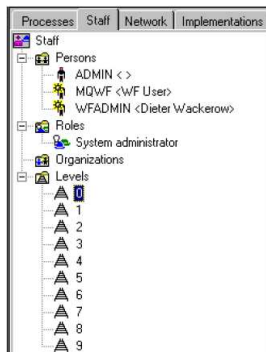


Figura: MQSeries Workflow Buildtime - GUI (Staff)

Definindo estruturas de dados

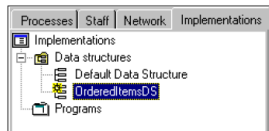
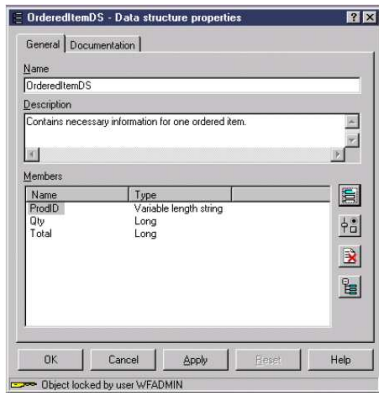


Figura: Vista da nova estrutura de dados

Figura: Propriedades de uma estrutura de dados

Definindo Programas a usar

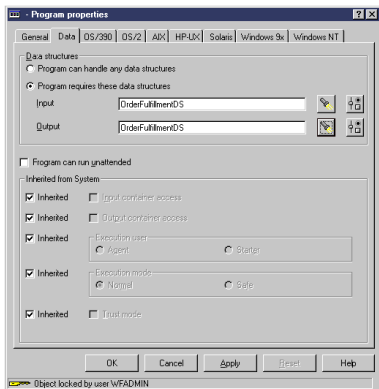


Figura: Tipos de Dados a tratar

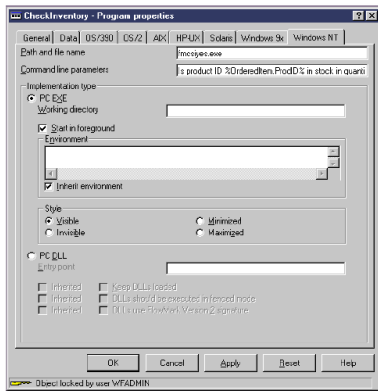


Figura: Propriedades de definição de um programa - WindowsNT

Modelando o Processo

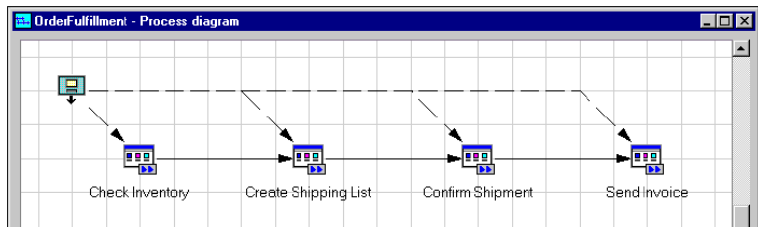


Figura: Processo de Encomenda já completo

Modelando o Processo

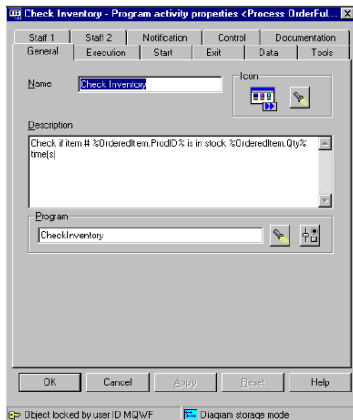


Figura: Propriedades de uma actividade

Vista Geral

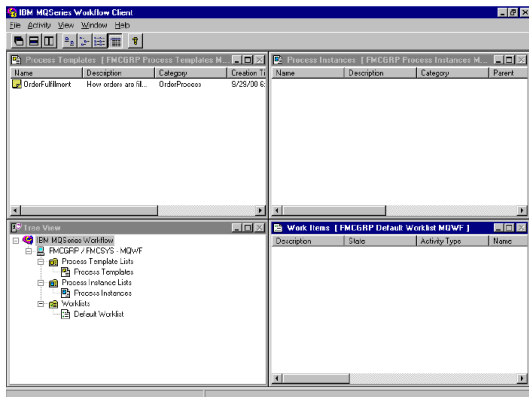
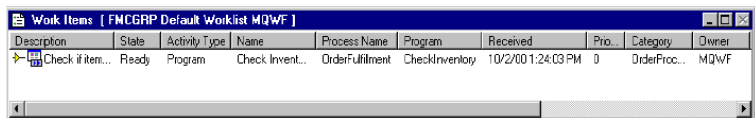


Figura: MQSeries Workflow Client

Work Item



The screenshot shows a window titled "Work Items [FMCGRP Default Worklist MQWF]". It contains a table with the following data:

Description	State	Activity Type	Name	Process Name	Program	Received	Pro...	Category	Owner
Check if item...	Ready	Program	Check Invent...	OrderFulfilment	CheckInventory	10/2/00 1:24:03 PM	0	OrderProc...	MQWF

Figura: Worklist Mostrando um Item

Código Java usando a API do MQSeries 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( , );
service.logon2(userid, passwd, SessionMode.PRESENT_HERE, AbsenceIndicator.LEAVE );

ProcessTemplate[] proctempl = service.queryProcessTemplates("NAME = 'OrderFulfillment'", "NAME", myint);

// get the input container
ReadWriteContainer inputContainer = proctempl[0].inContainer();

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

// create and start a process instance and pass input container
ProcessInstance procinst = proctempl[0].createAndStartInstance2( , , inputContainer, false);

// log off from MQW
service.logoff();
```

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
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();
```

Parte III

Perguntas?