

5 Estudo de caso: utilizando o sistema para requisição de material

A fim de avaliar as características da arquitetura proposta e a corretude da implementação, realizamos experiências com cenários de complexidade crescente, que apresentamos a seguir.

5.1. Seleção simples

Iniciamos nosso estudo de casos apresentando um cenário onde se deseja apenas indicar o fornecedor a ser contatado. Dentro da aplicação de requisição de material, o administrador define uma lista de fornecedores contendo apenas um fornecedor, e associa esta lista como critério de uma composição, conforme mostramos na figura a seguir:



A imagem mostra uma interface web para "Requisição de Material". No topo, há uma barra de navegação com os links "Início", "Compor", "Requisitar" e "Acom". Abaixo, há uma seção intitulada "Nova Composição" com um sub-título "Requisitos". Nesta seção, há três opções de seleção: "Fornecedor A" (com uma caixa de seleção marcada), "Entrega 24hs" (com uma caixa de seleção desmarcada) e "ISO 14000" (com uma caixa de seleção desmarcada).

Figura 44 – Composição tipo seleção simples

Este caso simples nos permite explorar os aspectos mais fundamentais da mediação flexível. O módulo de materiais acessa a lista e define uma mensagem para o mediador, indicando o código do fornecedor desejado. Esta mensagem é avaliada contra todos os registros de fornecedores na fase de determinação de destinatários dentro do mediador. O *bpm* principal aguardará pela resposta do *bpm* do fornecedor, e há a possibilidade de haver resposta nula. Neste caso, o fluxo de agregação é desviado com a geração desta mensagem pela cláusula de exceção.

É evidente que a utilização do mediador para este tipo de processamento torna o processo muito custoso. Fizemos alguns experimentos com a ajuda de um cliente SOAP, o *SOAPUI*, avaliando o tempo para a obtenção do catálogo do fornecedor A. Escolhemos este método porque o tempo de processamento no *Bean* é muito menor do que o tempo de integração. A figura a seguir apresenta os resultados por tipo de chamada:

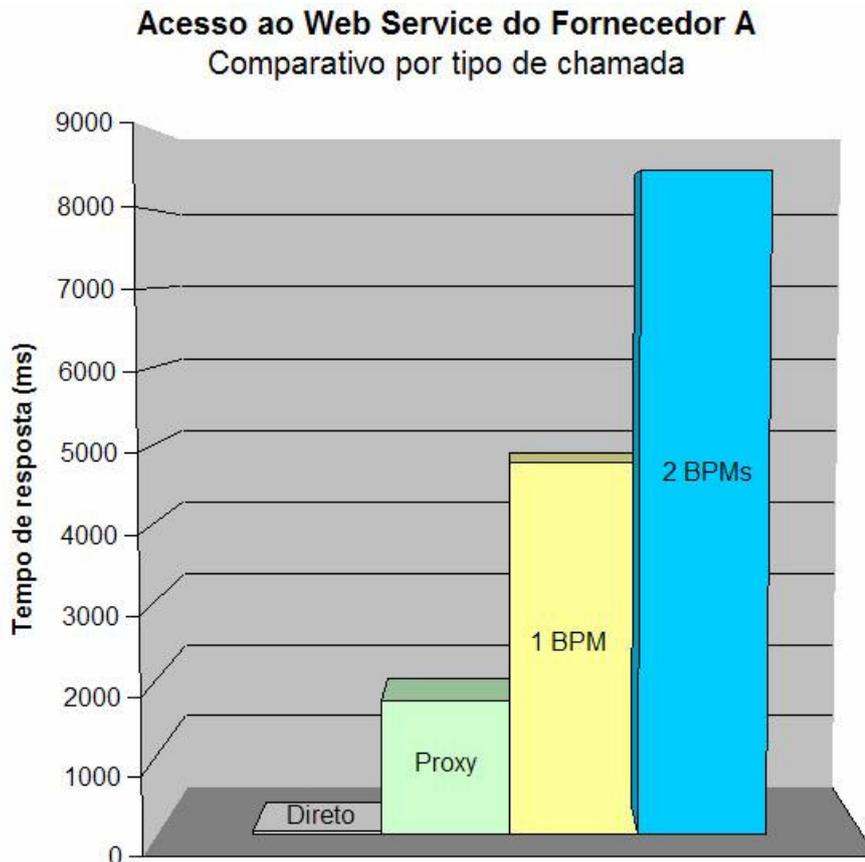


Figura 45 - Acesso ao Web service do Fornecedor A

Tempo (ms)	
Direto	32
Proxy	1739
1 BPM	4795
2 BPMs	8476

Tabela 3 - Tempos de resposta por tipo de chamada

- **Acesso direto:** realizamos uma chamada diretamente ao Web service do fornecedor, sem passar pelo mediador. O tempo de resposta foi excelente para uma aplicação interativa, 32 ms. De fato, o processamento mais lento ocorre na inicialização do *Bean*, quando o

catálogo é lido do arquivo. Depois disso, o sistema apenas retorna o conteúdo armazenado na classe *Helper*.

- **Mediador como *proxy***: na primeira etapa do desenvolvimento com o mediador SAP XI, definimos um *business system* EBPIXI para receber as requisições e repassar para os fornecedores. Inicialmente, este *business system* apenas repassava as requisições para o fornecedor A e as respostas para o cliente SOAP. O tempo de resposta neste caso subiu para 1.739 ms., mais de 50 vezes o tempo sem o mediador. Isto se deve à arquitetura flexível deste, projetada para permitir uma grande variedade de cenários:

1. A requisição SOAP é recebida por um adaptador, que transforma a chamada em uma mensagem assíncrona para o mecanismo central de integração.
2. Esta mensagem é retirada da fila por este mecanismo e então processada, a fim de determinar o destinatário.
3. Com base nas configurações do *business system* do destinatário, o mediador passa a requisição para o adaptador SOAP por meio de uma mensagem assíncrona.
4. Esta mensagem é transformada em uma requisição SOAP e encaminhada ao fornecedor A.
5. A resposta deste fornecedor é recebida pelo adaptador SOAP, que a transforma e encaminha para o mecanismo de integração.
6. Este mecanismo retira a mensagem da fila e determina que se trata da resposta aguardada pelo *business system* do destinatário.
7. A mensagem é encaminhada ao adaptador SOAP como resposta à requisição síncrona indicada no item 1.
8. O adaptador SOAP transforma a mensagem no formato apropriado e envia a resposta, encerrando a conexão.

Além dos passos acima, todo processamento fica registrado para monitoração, e as mensagens trocadas são numeradas e arquivadas.

Monitor for Processed XML Messages

Status	Executed From	Start Time	End Time	Sender Service	Sender Interface	Receiver
✓	02.06.2008	11:58:54	11:58:54	FA	Catalogo_InboundSync	EBPXI
✓	02.06.2008	11:58:53	11:58:54	EBPXI	Catalogo_OutboundSync	FA

Figura 46 – Registro de mensagens no caso de mediador como *proxy*

- Mediador com BPM:** o processamento WS-BPEL é realizado fora do mecanismo central de integração. Para isso, definimos *integration processes* além dos *business services*, que resultam na duplicação das mensagens dentro do mediador. Além disso, os elementos da linguagem WS-BPEL são convertidos dinamicamente para o ambiente de *workflow* da SAP, e então executados. O tempo médio de resposta neste caso aumentou para 4.795 ms., 2,75 vezes maior do que sem o uso de *BPM*. Mostramos a seguir a troca de mensagens dentro do mediador:

Status	Executed From	Start Time	End Time	Sender Service	Sender Interface	Receiver Service
✓	02.06.2008	12:52:44	12:52:44	TST_BPM	Catalogo_AbstractSync	EBPXI
✓	02.06.2008	12:52:43	12:52:45	FA	Catalogo_InboundSync	TST_BPM
✓	02.06.2008	12:52:42	12:52:43	TST_BPM	Catalogo_AbstractSync	FA
✓	02.06.2008	12:52:41	12:52:45	EBPXI	Catalogo_OutboundSync	TST_BPM

Figura 47 – Registro de mensagens no caso de mediador com *BPM*

A seguir apresentamos a visão do *workflow* gerado e executado, em função da chamada ao *TST_BPM*:

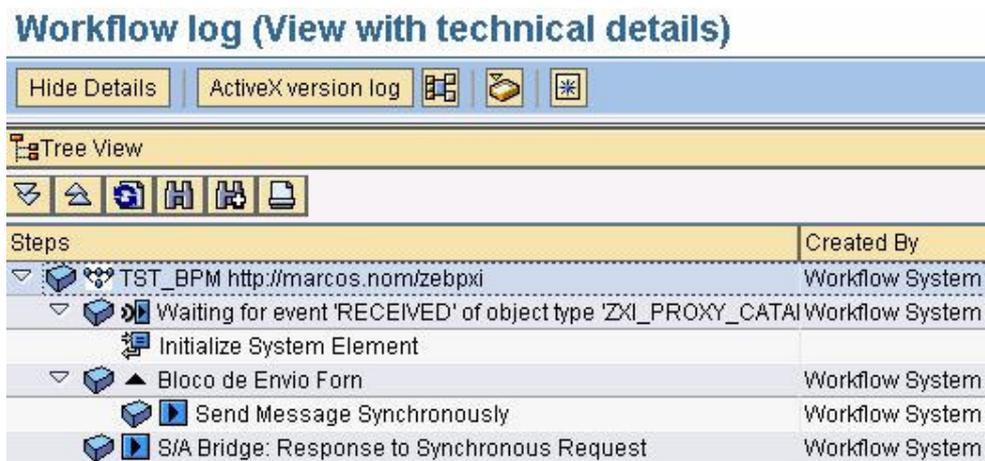


Figura 48 – Workflow gerado para execução do BPM

- Mediador com 2 BPMs:** a fim de superar a limitação do bloco *ParForEach* com relação ao processamento de mensagens síncronas, introduzimos um segundo *bpm*. Como estes são acionados pelo mecanismo central de integração, passamos a triplicar o número de mensagens necessárias para acionar um fornecedor. Além disso, com a generalização do *BPM* agora passamos a realizar a coleta e agregação de mensagens. Tudo isto para leva em média 8.476 ms. para processar o catálogo do fornecedor A, 76% a mais do que com apenas um *bpm*. Mostramos a seguir a troca de mensagens no mecanismo de integração e o log de processamento dos *workflows* gerados:

XML Messages						
Status	Executed Fro...	StartTI...	EndTime	Sender Servi...	Sender Interface	Receiver Service
✓	02.06.2008	10:55:28	10:55:29	FlexComp_PAI	CatalogoXI_AbstractSyr	CFWS
✓	02.06.2008	10:55:26	10:55:26	FA_BPM	Catalogo_RespAbstrac	FlexComp_PAR
✓	02.06.2008	10:55:25	10:55:26	FA	Catalogo_InboundSync	FA_BPM
✓	02.06.2008	10:55:24	10:55:25	FA_BPM	Catalogo_AbstractSync	FA
✓	02.06.2008	10:55:23	10:55:23	FlexComp_PAI	CatalogoXI_ReqAbstrac	FA_BPM
✓	02.06.2008	10:55:21	10:55:29	CFWS	CatalogoXI_OutboundS	FlexComp_PAR

Figura 49– Registro de mensagens no caso de mediador com 2 BPMs

Este *workflow* é gerado pela execução do *bpm* principal, FlexComp_PAR:



Figura 50 – Workflow gerado pela execução do bpm FlexComp_PAR

O workflow abaixo é gerado e executado a partir da chamada ‘Send Message Asynchronously’ do bpm principal:

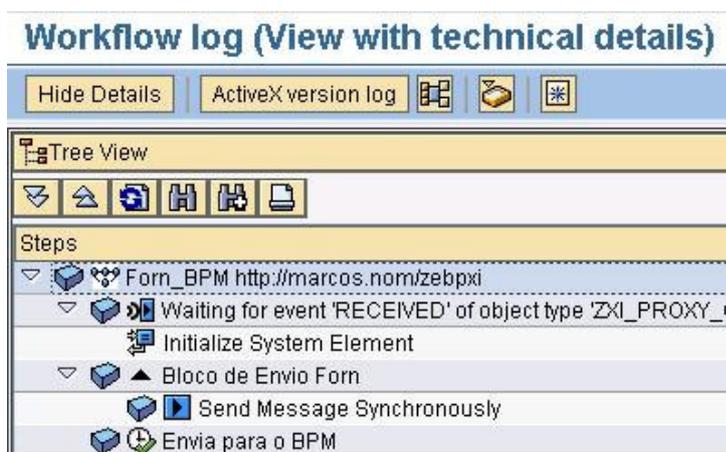


Figura 51 - Workflow gerado pela execução do BPM Forn_BPM

5.2. Composição interseção

A arquitetura que propomos comporta a interseção de vários requisitos. O administrador do sistema pode definir uma composição onde é requerida entrega em 24hs, com certificação ISO 9000 e ainda que o custo do frete esteja incluso no preço da mercadoria. Naturalmente, quanto mais requisitos forem exigidos, mais seletiva será a composição, e é possível que não sejam encontrados fornecedores que a satisfaçam. Caso isto aconteça, a aplicação retorna a mensagem “*Composição não retornou fornecedores ativos*”, e o mediador não é acionado.

A codificação deste tipo de composição foi facilitada pelo método *contains* da classe Java *ArrayList*. A complexidade da operação é $\Theta^{m.n}$, pois depende do número de fornecedores e de requisitos.

Para melhorar o desempenho, incluímos uma condição de parada antecipada no laço que varre a lista dos fornecedores, caso a lista fique vazia ($0 \cap A = 0$).

5.3. Composição união

Na definição de uma composição podem ser indicadas uma ou mais regiões de interesse. Internamente, o sistema calcula a união dos fornecedores que pertençam a estas regiões. A negação do método *contains* da classe *ArrayList* foi utilizada neste caso. Para melhorar o desempenho, também incluímos uma condição de parada antecipada do laço que varre a lista de condições, caso a lista de regiões resultante inclua todas as regiões possíveis.

Este tipo de composição pode ser utilizado para calcular outros atributos opcionais, como “Permite retirada de mercadoria OU entrega em 24 h”. Uma versão ainda mais generalizada da aplicação que implementamos poderia permitir o uso das listas de atributos obrigatórios neste tipo de operação.

Na definição da classe *Solicitante*, armazenamos a propriedade *localização* a fim de possibilitar um critério de composição baseado em uma distância máxima, expressa em quilômetros. Desta forma o sistema poderia determinar quais regiões estão contidas num raio informado pelo solicitante, e utilizá-las como critério para seleção de fornecedores.

5.4. Composição complexa

Tal como o concebemos, o módulo de materiais avalia as composições da seguinte forma:

$$\begin{array}{c} (\text{Requisito 1} \cap \text{Requisito 2} \cap \dots \cap \text{Requisito n}) \\ \cap \\ (\text{Região 1} \cup \text{Região 2} \cup \dots \cup \text{Região n}) \\ \cap \\ (\text{conceito} \geq \text{conceito}) \end{array}$$

Esta expressão permite ao administrador definir uma composição como por exemplo “*Fornecedores com ISO 14000, das regiões Sul ou Sudeste, com conceito mínimo maior ou igual a 8*”. A fim de tornar o uso do programa mais intuitivo, criamos uma função que gera o nome da composição em função de seus requisitos. Em uma aplicação real, pode ser definido que cada solicitante só tenha acesso a um tipo de composição, ou que estas sejam apresentadas a estes por meio de códigos.

O critério conceito também pode ser interpretado como um requisito. Neste caso não haveria uma avaliação numérica dentro do sistema, apenas uma lista de fornecedores considerados prioritários.

5.5. Desempenho, escalabilidade e segurança

Conforme apresentamos anteriormente, o desempenho da aplicação foi impactado negativamente pela adoção de dois *bpms* simultâneos na consulta aos catálogos. Para a função de acompanhamento de requisições, optamos por efetuar a composição na própria aplicação, disparando múltiplas chamadas por meio do mediador e agregando as respostas em uma tabela interna. Desta forma, transferimos a tarefa de agregação do mediador para a aplicação, que ficou apenas com a função de roteamento de mensagens, que não depende dos *bpms*. Obtivemos assim tempos de resposta consistentemente inferiores a 5 segundos, mesmo para requisições envolvendo chamadas a 3 fornecedores. Várias razões contribuem para este bom desempenho:

- Evitamos a multiplicação de mensagens no mediador;
- Evitamos o alto custo da geração dinâmica de *workflows* relativos aos *bpms*.

- Continuamos nos beneficiando do paralelismo no processamento do mediador;
- Uma parte maior do processamento passou a ser feita diretamente em Java.

Mesmo não sendo explicitamente informado pela SAP, ficou claro para nós que o mecanismo de processamento WS-BPEL do XI foi pensado para aplicações assíncronas, onde o volume, a garantia de entrega e o controle são mais importantes que um baixo tempo de resposta.

Com relação à escalabilidade da arquitetura proposta, consideramos os seguintes cenários:

- **Elevado número de fornecedores:** se as composições forem definidas indicando um número máximo de catálogos a considerar, não haverá problema de desempenho. Mesmo que uma composição considere todos os fornecedores, o mediador só irá considerar as *n* primeiras respostas recebidas. A fase de determinação de destinatários no mediador, que tem custo linear em função do número de fornecedores, é realizada pelo mecanismo de integração, não pelo *bpm*. Com o aumento do número de fornecedores, aumenta também a possibilidade de ocorrerem erros em função da indisponibilidade destes ou mesmo falhas de comunicação. Para isto limitamos o tempo para conexão em 5 segundos e definimos um critério de suspensão dos fornecedores que não respondem às requisições.
- **Elevado número de critérios:** as composições são processadas localmente no módulo de materiais, com base em listas armazenadas no banco de dados e em memória. O mediador só recebe uma mensagem com o resultado da composição, isto é, os fornecedores a serem contatados. Sendo assim, mesmo um elevado número de critérios não impactará sensivelmente o desempenho da aplicação.

Na implementação do módulo de fornecedores, definimos seu Web service como público, para todas as funções. Em aplicações *B2B*, os Web services normalmente utilizam autenticação por meio de certificado digital e transporte seguro. Internamente, o Web service do módulo de materiais é acessado com autenticação de usuário e senha, para não comprometer o desempenho da aplicação com o uso de criptografia em chamadas na rede interna de servidores, que normalmente é confinada.