

► Aprenda a disciplina,
busque a arte e contribua
com idéias em
www.ArchitectureJournal.net
**Recursos nos quais você
pode confiar.**

THE ARCHITECTURE JOURNAL™

Idéias para melhores resultados

Jornal 7

Geração Workflow

Desenvolva aplicativos
em uma plataforma de
workflow

A metáfora Amazing
Race

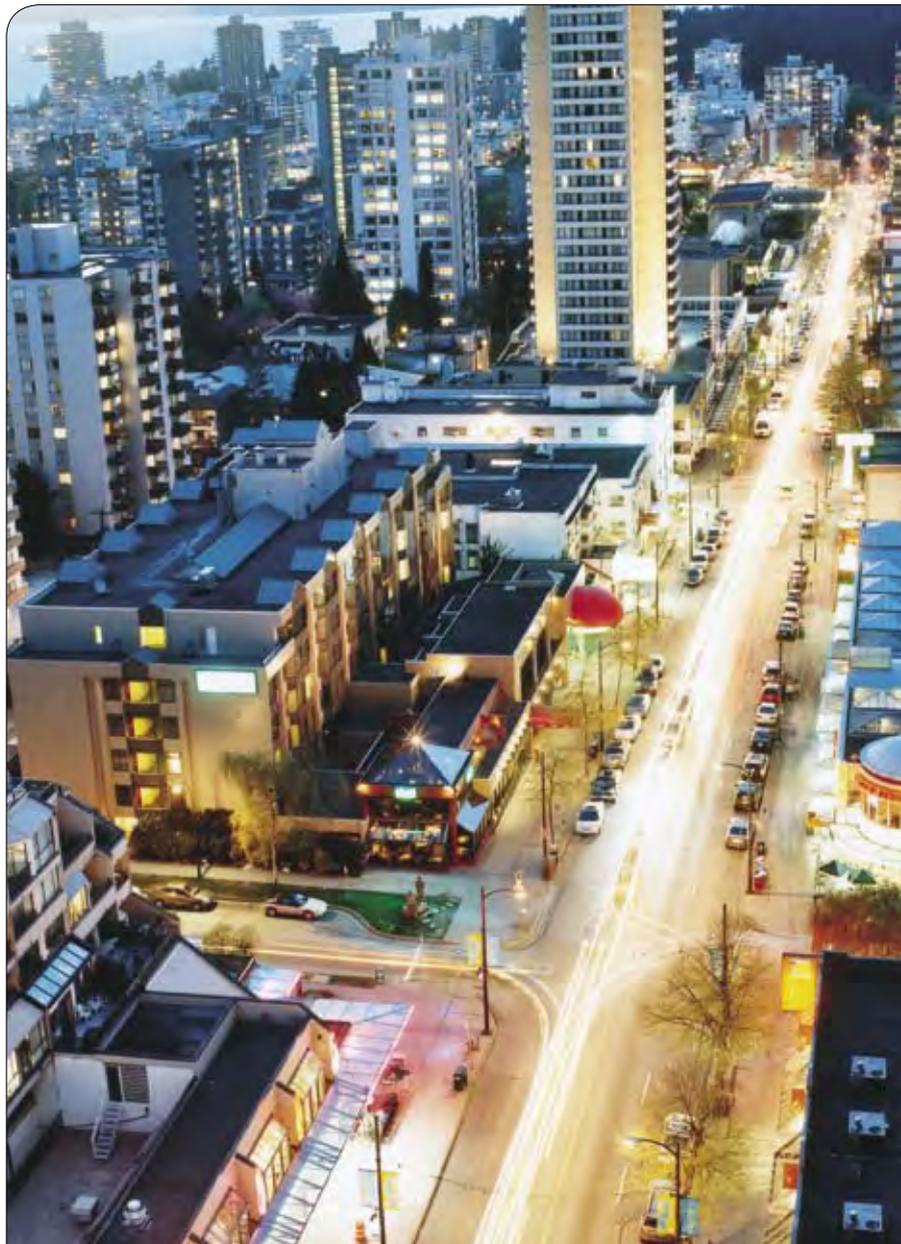
Explore as arquiteturas
do workflow humano

Workflow na integração
de aplicativos

Simplifique modelos de
workflow complexos

Prepare a empresa
orientada a serviços

Modelagem orientada
a serviços para sistemas
conectados



Microsoft



Sumário

Apresentação

1

por Simon Guest

Desenvolva aplicativos em uma plataforma de workflow

2

por David Green

Um fluxo de trabalho é importante para resolver problemas de negócios. Examine uma série de aplicações que demonstram as decisões e o raciocínio que os arquitetos defrontam ao criar uma plataforma de fluxo de trabalho.



A metáfora Amazing Race

9

por Vignesh Swaminathan

A empresas atualmente percebem o potencial de automatizar os seus processos de negócios. Descubra como gerenciar processos de negócios de alto nível por meio de uma analogia com um programa de televisão de jogos baseados na realidade.



Explore as arquiteturas do workflow humano

16

por Jesus Rodriguez e Javier Mariscal

Existem dois componentes nos sistemas de fluxo de trabalho humano e padrões representativos das interações dos processos das pessoas com negócios. Descubra como aplicar esses componentes para implementar esses processos.



Workflow na integração de aplicativos

19

por Kevin Francis

A integração de aplicações é um dos maiores desafios que os arquitetos enfrentam hoje. Observe uma estrutura para a integração de aplicações por meio da utilização de ferramentas como tecnologias de fluxo de trabalho.



Simplifique modelos de workflow complexos

24

por Andrew Needleman

O projeto sólido de fluxos de trabalho exige muitas habilidades que tornam o fluxo de trabalho um desafio até mesmo para arquitetos experientes. Aprenda uma abordagem para simplificar o processo de projeto de sistemas complexos utilizando um novo tipo de diagrama



Prepare a empresa orientada a serviços

27

por William Oellermann

Embora criar muitos serviços da Web possa ser difícil, gerenciá-los pode ser realmente difícil. Explore utilizando um modelo que pode ajudá-lo a planejar os recursos de uma empresa ativada por serviços



Modelagem orientada a serviços para sistemas conectados – Parte 1

33

por Arvindra Sehmi e Beat Schwegler

Os arquitetos desejam identificar os artefatos corretamente e no nível de abstração certo. Verifique uma abordagem para modelar sistemas conectados orientados a serviços que promove um alinhamento íntimo entre soluções de TI e as necessidades de negócios na Parte 1 de uma série de duas partes



Fundador

Arvindra Sehmi
Microsoft Corporation

Editor-chefe

Simon Guest
Microsoft Corporation

Conselho Editorial Microsoft

Gianpaolo Carraro
John deVadoss
Neil Hutson
Eugenio Pace
Javed Sikander
Philip Teale
Jon Tobey

Editor

Marty Collins
Microsoft Corporation

Design, impressão e distribuição

Fawcette Technical Publications

Jeff Hadfield, vice-presidente de publicação
Terrence O'Donnell, editor-geral
Michael Hollister, vice-presidente de arte e produção
Karen Koenen, diretor de circulação
Brian Rogers, diretor de arte
Kathleen Sweeney Cygnarowicz, gerente de produção

Microsoft

As informações contidas neste Melhor do *Jornal de Arquitetura* da Microsoft ("*Jornal*") são fornecidas apenas para fins informativos. O material no *Jornal* não constitui a opinião nem a orientação da Microsoft, e você não deve confiar em nenhum material contido neste *Jornal* sem buscar orientação independente. A Microsoft não fornece nenhuma garantia ou representação com relação à precisão ou adequação para a finalidade de qualquer material contido neste *Jornal*, sob nenhuma circunstância, a Microsoft se responsabiliza por qualquer descrição, incluindo responsabilidades por negligências (exceto no caso de ferimentos ou morte), por quaisquer danos ou perdas (incluindo, sem limitação, perda de negócios, renda, lucros ou perdas consequentes) ou por qualquer outro item que possa resultar deste *Jornal*. O *Jornal* pode conter imprecisões técnicas ou erros tipográficos. Este *Jornal* pode ser atualizado periodicamente, mas, às vezes, pode estar desatualizado. A Microsoft não se responsabiliza por manter atualizadas as informações contidas neste *Jornal* ou por qualquer falha ao fazê-lo. Este *Jornal* contém materiais enviados e criados por terceiros. Até o máximo permitido pela lei aplicável, a Microsoft isenta-se de todas as responsabilidades por qualquer ilegalidade que possa surgir ou por erros, omissões ou imprecisões neste *Jornal*, e a Microsoft não se responsabiliza pelos materiais de terceiros.

Todos os direitos autorais, marcas registradas ou qualquer outro direito de propriedade intelectual contido no *Jornal* pertencem, ou estão licenciados, à Microsoft Corporation. Fica proibida a realização de cópias, reproduções, transmissões, armazenamentos, adaptações ou alterações no layout ou no conteúdo deste *Jornal* sem a autorização prévia e por escrito da Microsoft Corporation e de seus autores individuais.

© 2005 Microsoft Corporation. Todos os direitos reservados.

Apresentação

Caro Arquiteto,

Lembro-me bem daquela manhã. O dia estava claro e frio, e eu estava no escritório da Microsoft em Londres, sentado com Arvindra Sehmi, que me explicava seu ponto de vista e me mostrava um dos primeiros protótipos estimulantes do *Jornal*. Como você deve saber, daquela manhã em diante o *Jornal* passou a caminhar com cada vez mais força. Você está lendo a 7ª edição do *Jornal*, que atinge a mais de 30 mil assinantes no mundo todo – tanto no formato impresso quanto no formato on-line, por meio do nosso site, ArchitectureJournal.net, que foi lançado no mês passado. Enquanto conversava com Vin, mal sabia eu que, três anos depois, eu teria a honra de assumir o cargo de editor da revista.

Embora não haja grandes mudanças no formato, como editor, pretendo apresentar um novo direcionamento para O *Jornal de Arquitetura*. Em primeiro lugar, com a base crescente de assinaturas, vamos traduzi-lo para oito idiomas diferentes. Se você estiver lendo esta edição em outro idioma que não o inglês, gostaria primeiro de lhe apresentar com prazer o novo formato e esperar que ele estabeleça uma tendência contínua.

Em segundo lugar, a partir desta edição estamos nos encaminhando para uma abordagem temática, segundo a qual a maioria dos artigos em cada edição se enquadra em um tema ou padrão comum. Como você pode ter visto na capa, o tema desta edição é "Geração Workflow". Para dar o pontapé inicial no tema, vamos começar com o artigo de Dave Green, arquiteto do nosso produto WF (Windows Workflow Foundation). Dave compartilhará as decisões e as conclusões de sua equipe durante o desenvolvimento de uma plataforma de workflow, que, no final, acabou dando forma ao produto que estamos desenvolvendo.

Depois temos Vignesh Swaminathan, gerente de produto da Cordys R&D. Em seu artigo, Vignesh o conduzirá em uma "Amazing Race" e analisará as semelhanças entre as matrizes de decisão do workflow nas empresas e um programa popular de televisão. Para adicionar um pouco mais de toque humano ao *Jornal*, Jesus Rodriguez e Javier Mariscal apresentarão os principais componentes de um sistema de workflow humano e descreverão os padrões que podem ser usados para modelar as tarefas relacionadas às pessoas.

Para juntar as peças desse quebra-cabeça, Kevin Francis discutirá a integração dos aplicativos com o workflow no seu artigo, "Workflow na integração de aplicativos".

E, para encerrar o grupo de artigos sobre workflow desta edição, Andrew Needleman mostrará uma técnica que ele usa chamada "pontos e linhas", para simplificar o processo de comunicação de fluxos de trabalho complexos com especialistas de negócio.

Depois dos artigos sobre workflow, teremos a sorte de ler o artigo de William Oellermann sobre o ESOMM (Modelo de Maturidade Empresarial de Orientação a Serviços), um modelo de maturidade que analisa a gestão de serviços na empresa.

Por fim, para fechar com chave de ouro esta edição, decidimos não deixar Arvindra inteiramente em paz, e ele volta na companhia de Beat Schwegler, para mostrar a primeira parte de uma excelente análise da "Modelagem orientada a serviços para sistemas conectados".

A segunda parte estará nas páginas da 8ª edição do *Jornal*.

No geral, foi uma grande jornada que trilhamos para montar a minha primeira edição, e espero que você considere úteis e provocantes os artigos que apresentamos. Na qualidade de novo editor, gostaria de lhes dar novamente as boas-vindas e de dizer que estou ansioso para receber notícias de vocês em um futuro próximo.



Simon Guest



Desenvolva aplicativos em uma plataforma de workflow

por David Green

Resumo

O workflow pode ser algo útil para resolver problemas de negócios. Neste artigo vamos conhecer e investigar a idéia do desenvolvimento de aplicativos em uma plataforma de workflow. Esta plataforma apóia os principais conceitos do workflow, fornecendo a base para desenvolver aplicativos estruturados, incluindo os produtos de workflow da forma como até agora compreendemos. Faremos uma pesquisa em uma série de aplicativos para explorar as características necessárias, que leva a uma discussão dos benefícios em potencial do desenvolvimento de aplicativos em uma plataforma de workflow. Também vamos analisar o Windows Workflow Foundation como meio de perceber esses benefícios na prática.

A idéia do workflow é conhecida há muito tempo e tem sido consistentemente atrativa como forma de atacar os problemas de negócio. Os problemas aos quais se aplicou a abordagem do workflow geralmente apresentam três características: O principal valor fornecido para os negócios é a coordenação, por exemplo, ao organizar várias contribuições para a preparação de um preço ou ao realizar uma revisão de documentos. Cada instância do processo de negócios em questão é de longa duração, estimado geralmente em dias, semanas ou meses, em vez de minutos. O processo de negócios tem participantes que geralmente contribuem com a maior parte do produto final.

No entanto, apenas uma pequena parcela dos problemas de negócios, que apresentam essas características é resolvida com uma abordagem de workflow. O que é mais comum é que o processo não seja registrado como informações legíveis por máquina. Em vez disso, cada uma das pessoas participantes interage com os sistemas de negócios não estão cientes da semântica do processo como um todo, por exemplo, o sistema de informações do cliente, e com outros participantes por meio de canais de comunicação neutros em termos de conteúdo, como o e-mail. Cada participante usa um modelo mental de sua parte em todo o processo de negócios para determinar o seu comportamento.

Não é difícil perceber o que atrai na criação do modelo legível por máquina no processo de negócios, ou seja, um workflow. Três benefícios importantes que um modelo de workflow pode fornecer são: Conhecimento, monitoramento e otimização. Pode-se usar um conjunto de modelos de workflow relacionados para se obter conhecimento sobre o fluxo de trabalho em uma organização. No monitoramento, saber quais

peças estão contribuindo com trabalho para este ou aquele processo de negócios é algo muito útil quando se tenta entender custos e cargas de trabalho. Na otimização, ter um modelo do trabalho que está sendo realizado e conseguir usá-lo para interpretar o comportamento são atividades que possibilitam raciocinar sobre como otimizar o processo de negócios.

Um modelo de workflow

Dados esses benefícios convincentes, por que os modelos de workflow não são usados mais amplamente? A resposta mais provável é que o investimento para utilizá-los é alto demais, que incluem custos de produto, ou seja, a compra direta de um produto de workflow; custos de integração, na qual os processos modelados como fluxos de trabalho precisam ser integrados como parte de um sistema de negócio maior; e custos de padronização, em que é difícil uma grande organização padronizar uma única tecnologia de workflow. As variações nos produtos de workflow também indicam que as habilidades e a portabilidade do modelo são problemas.

Daremos uma olhada na possibilidade de resolver essas questões obstrutivas desenvolvendo aplicativos em uma plataforma de workflow que tem baixo custo, está presente em todos os lugares, é uniforme e facilmente integrada aos aplicativos. Para ser claro, a idéia é não substituir os produtos de workflow. Em vez disso, a hipótese é que é útil decompor em fatores o apoio em alguns conceitos principais em uma plataforma na qual tanto os produtos de workflow quanto outros aplicativos podem ser desenvolvidos (consulte a Figura 1).

O workflow é um modelo, o que significa que é uma descrição, legível por máquina, do comportamento do negócio que não é um código. O significado e os benefícios deste conceito no contexto do valor de uma plataforma de workflow serão discutidos posteriormente.

O modelo descreve uma organização de unidades de trabalho. Por exemplo, suponhamos que um processo de revisão de documentos especifique que Joe escreve o documento e depois Fred o revisa. Aqui as unidades de trabalho estão, em primeiro lugar, escrevendo e, em segundo, revisando o documento, e a organização significa que uma tarefa deve seguir a outra. Esse conceito não é uma idéia radical. O código que faz chamadas sucessivas a duas sub-rotinas é um exemplo válido do conceito. O interesse reside, em vez disso, nas formas que essa organização assume.

Para testar a hipótese da plataforma, consideremos uma variedade de aplicativos reais e explorar suas características para se mostrar útil.

O processo de revisão de documentos toma como parâmetro de entrada um conjunto de pares [revisor, função] que descrevem quais pessoas estão envolvidas no fluxo e em quais funções. Os possíveis valores para a função são obrigatório, opcional, aprovador final e proprietário.

O processo de revisão continua até que todos os revisores realizem suas funções designadas e depois avisa o proprietário sobre o resultado.

Aqui os itens de trabalho são as revisões de documento organizadas pelo processo de revisão. Há três características interessantes que devem ser mencionadas, que são: Múltiplos pontos de interação, atividade manual e automatizada, e a necessidade de lidar com mudanças dinâmicas.

Contratos de workflow

O workflow tem vários pontos de interação ou contratos. Em primeiro lugar, há um contrato com o revisor. Ele envolve solicitar ao revisor revisar o documento, aceitar o veredicto e quaisquer comentários de revisão e também dizer para um outro revisor que seus dados não são mais necessários (se a revisão for cancelada, ou talvez se um número suficiente de revisores disser sim). O contrato também permite que o revisor delegue a tarefa de revisão. Então há um segundo contrato com o aprovador final, que é uma especialização do contrato do revisor. Em terceiro lugar, há um contrato com o proprietário da revisão, que permite a ele cancelar a revisão e ser notificado sobre o resultado dela. Por fim, há um contrato com o iniciador do processo de revisão, que instancia a revisão e fornece os parâmetros necessários.

Mostra-se algo típico dos fluxos de trabalho que eles conectam várias partes por meio de uma variedade de contratos (consulte a Figura 2). O workflow da revisão de documentos é essencialmente um coordenador, iniciado por um contrato, que está coordenando uma série de participantes por meio de um ou mais contratos adicionais.

O workflow da revisão de documentos orienta a atividade manual, e é responsável também pelas atividades automatizadas, como armazenamento de versões do documento em um repositório à medida que a revisão avança. Do ponto de vista do workflow, não há diferença essencial. Neste caso, pode ser encarado como algo que se comunica, em geral, com os serviços por meio de contratos. Um caso especial de serviço é outro workflow. Outro caso especial é o indivíduo. De várias formas, um indivíduo é o serviço assíncrono original: nunca se sabe quando ou se ele vai responder. Uma característica desse tipo de workflow é que os participantes solicitarão mudanças no fluxo à medida que é executado. Por exemplo, o revisor pode delegar a tarefa de revisão a um colega ou compartilhar o trabalho envolvido em uma tarefa de revisão com um subordinado.

Há duas maneiras de atender a essa exigência. Uma é compreender todas as mudanças possíveis no workflow. Dessa forma, a solicitação de delegação torna-se apenas outra função do contrato entre o workflow e o revisor. A outra possibilidade é enxergar a mudança como algo separado do workflow, em que a mudança é implementada como função externa que muda o modelo existente. Nessa abordagem, o resultado da delegação é um novo modelo de workflow idêntico àquele em que a tarefa de revisão foi atribuída à pessoa designada desde o início.

Solicitar uma etapa adicional de aprovação acrescentaria uma nova tarefa de aprovação ao modelo do workflow, que não poderia, de forma alguma, ter contido etapas de aprovação em sua forma original. Com isso não é mais necessário antecipar todas as modificações possíveis; no máximo, ele estará preocupado com a restrição das áreas do modelo que estão sujeitas à mudança.

As duas abordagens são úteis. Escrever um fluxo em um workflow é algo simples de modelar e entender. A generalização das operações é algo mais complexo para o modelo, por outro lado, mais poderoso e ágil.

Em um caso extremo, mas interessante da última abordagem, o workflow inicia a execução com pouco ou nenhum conteúdo, e o comportamento necessário é acrescentado dinamicamente pelos participantes no fluxo. Aqui as operações disponíveis para modificar o workflow tornam-se um vocabulário que o usuário pode usar para construir o comportamento desejado à medida que o fluxo avança.

Colaboração na resolução de problemas

Para dar uma olhada em um exemplo específico de um aplicativo de colaboração na resolução de problemas, pense em um problema de disponibilidade no estoque. Uma linha de montagem está construindo um dispositivo, e o computador indica que há peças suficientes no estoque para aquela finalidade. No entanto, quando o gerente de estoque vai buscar as peças para entregar à linha de montagem, descobre-se que estão faltando 10 peças.

Para resolver o problema, é necessária a colaboração entre o gerente de estoque, o gerente de conta do cliente, o departamento de suprimentos e o gerente de produção. Cada função no processo de colaboração pode realizar ações características. O departamento de suprimentos poderia solicitar mais peças, talvez usando um fornecedor diferente ou pagando mais para o fornecedor atual entregar o produto com mais rapidez. O gerente de contas poderia se dirigir ao cliente e pedir permissão para atrasar a entrega ou dividir a entrega em duas partes, arcando com o custo

“DE MUITAS FORMAS, UM INDIVÍDUO É O SERVIÇO ASSÍNCRONO ORIGINAL: NUNCA SE SABE QUANDO OU SE ELE VAI RESPONDER”

extra de envio. O gerente de produção poderia desviar os dispositivos montados de um pedido de outro cliente. O gerente de estoque poderia pesquisar o estoque físico na tentativa de achar as peças que faltam. Qualquer uma dessas ações poderia ser realizada várias vezes.

Uma restrição óbvia é que a colaboração não é concluída até que o déficit seja resolvido por alguma combinação das ações anteriores. Em geral, também haverá restrições de negócios. Por exemplo, pode haver uma regra que afirme que o adiamento da entrega para clientes especiais nunca é permitido. Além disso, as ações afetarão umas às outras. Por exemplo, pode haver uma política que determine que o custo total adicionado da ação corretiva não pode ultrapassar 5% do custo original de fabricação. Assim, solicitar um pedido urgente de suprimentos, por um preço superior, pode impedir que uma remessa seja dividida.

Nesse caso, os itens de trabalho são as ações que os vários participantes podem realizar à medida que buscam resolver o problema de déficit no estoque. A organização, no entanto, não é igual àquela exigida na revisão de documentos. Os participantes não são forçados a tanto; em vez disso, eles escolhem quais ações devem ser realizadas e quando realizá-las. No

Figura 1 Um workflow monolítico para a pilha

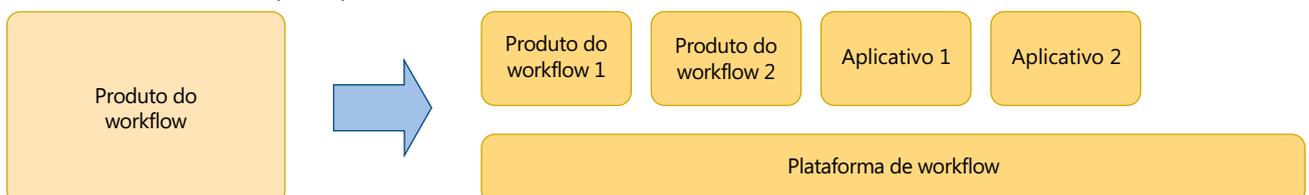
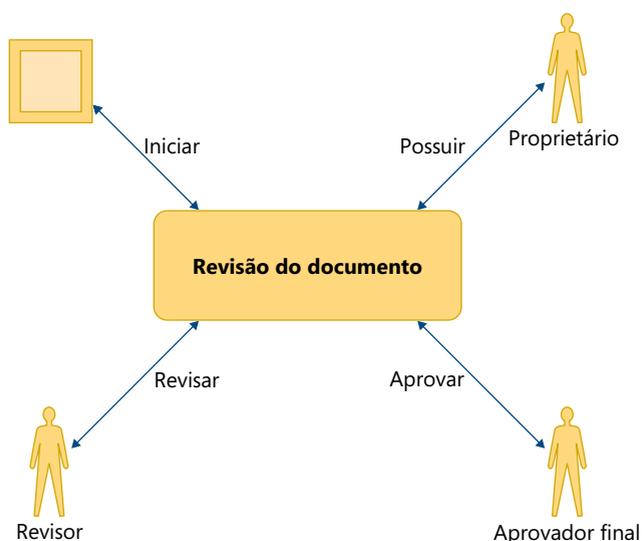


Figura 2 Um diagrama de contrato para o aplicativo de revisão de documentos



entanto, tais escolhas são restringidas pela organização do workflow, que tem dois aspectos: 1) As ações focam no cumprimento de um objetivo, nesse caso, resolver o problema de déficit no estoque. É criado um espaço de colaboração delimitado quando se inicia a resolução do problema, não sendo fechado até que o objetivo seja cumprido. 2) Os participantes não estão livres para realizar ações arbitrárias. Em vez disso, as ações disponíveis são determinadas pela função que o participante está desempenhando e pelo estado da colaboração.

O conjunto de ações disponíveis é determinado pelas políticas relacionadas ao objetivo e as globais, como restrição no não-atendimento de clientes especiais. As ações disponíveis variam à medida que a colaboração avança. A experiência do participante não é mais a de realizar as tarefas designadas. Em vez disso, o participante solicita as ações atualmente disponíveis a ele, realiza uma dessas ações ou nenhuma delas e depois repete o ciclo.

A principal exigência nova aqui, portanto, é uma forma de organização dos itens de trabalho que seja orientada essencialmente ao estado dos dados e ao objetivo. Também há a exigência de fornecer suporte a um estilo de consulta/ação de contrato com um participante do workflow.

Operações com script

As operações com script são simplesmente compostas de um conjunto de operações com utilização de um roteiro. Como exemplo, podemos citar uma ferramenta de desktop que permite que um usuário defina e execute uma série de tarefas comuns, como copiar arquivos e comentá-los.

Seria incomum pensar em usar um produto típico de workflow para essa finalidade. No entanto, ele se ajusta ao padrão da plataforma para um conjunto de unidades de trabalho organizadas por um modelo. Nesse caso, o modelo é uma seqüência, talvez com suporte para looping e execução condicional. Portanto, se uma plataforma de workflow tiver um custo suficientemente baixo e estiver em todos os lugares, será possível pensar em aplicá-la a esse tipo de problema. Será que agrega algum valor?

Um dos recursos das operações com script que atualmente não é tratado pelas implementações típicas é a questão do fluxo de dados. É comum que os dados exigidos por uma operação sejam o resultado de alguma operação anterior, mas essa informação, em geral, não é modelada no script. Assim, o usuário que organiza tarefas com uma ferramenta

de desktop pode não ser informado, ao criar um script, que os dados necessários para uma tarefa ainda não foram fornecidos, e só descobriria o erro ao executar. O modelo de workflow que pode descrever essas dependências de dados agregaria um valor claro para os autores do script.

Uma das abordagens é simplesmente incluir estruturas de fluxo de dados no modelo de workflow. Pode argumentar facilmente que o modelo básico de workflow precisa incluir recursos estruturais básicos como seqüências, condições e loops; mas não fica claro que o fluxo de dados é suficientemente universal para ser representado por elementos de primeira classe do modelo. Uma abordagem alternativa é dispor em camadas o suporte ao fluxo de dados sobre um workflow básico e extensível. O modelo de workflow que pode ser enriquecido com abstrações apropriadas a uma variedade de domínios de problema se ajusta bem à noção de uma plataforma de workflow. Essa abordagem evita a complexidade criada ao se incluir no modelo-base uma grande variedade de estruturas semânticas, especializadas para problemas diferentes, e as limitações impostas pela restrição do modelo de workflow ao um conjunto fixo de estruturas.

Agora vamos dar uma olhada em um aplicativo que oriente o usuário. Um exemplo é um sistema de URA (Unidade de Resposta Automática) e o outro é o de call center que orienta os operadores de telefone nos roteiros de suporte ou de vendas.

A essência desse aplicativo é orientar os usuários pela série de operações necessárias para cumprir um objetivo. A organização dessas operações é geralmente utilizada para nortear a apresentação ao usuário, seja um discurso gerado ou um conjunto de botões ativados e desativados em um formulário.

Uma característica desse tipo de aplicativo é que o workflow é a parte alterada com mais freqüência. Além disso, os patrocinadores do sistema

“UM MODELO DE WORKFLOW QUE EXPRESSA O PRINCIPAL OBJETIVO DE NEGÓCIOS DO APLICATIVO, DESPOJADO DE QUALQUER MATERIAL TÉCNICO IRRELEVANTE, É UMA MANEIRA EFICAZ DE CONSEGUIR A COMUNICAÇÃO ENTRE A EQUIPE DE TI E O PESSOAL DA ÁREA DE NEGÓCIOS”

geralmente estão bastante envolvidos em especificar as mudanças, tornando importante fornecer uma maneira de a equipe de TI e a equipe de negócios se comunicarem de forma clara e eficiente sobre as mudanças. Um modelo de workflow que expressa o principal objetivo de negócios do aplicativo, despojado de qualquer material técnico irrelevante, é uma maneira eficaz de conseguir essa comunicação.

Esses aplicativos também exigem flexibilidade na estrutura do workflow. Em um aplicativo de URA, o usuário geralmente terá muitas restrições, movendo-se em um conjunto hierarquicamente estruturado de menus. No entanto, também haverá comandos de escape – por exemplo, “voltar ao menu principal” ou “voltar ao menu anterior”.

O aplicativo de call center terá mais flexibilidade do que um aplicativo de URA, alterando as opções oferecidas ao usuário em resposta ao estado de um pedido ou ao retorno do cliente, por exemplo, pulando as informações de venda se o cliente começar a reagir negativamente.

Esse tipo de aplicativo exige suporte para um misto de organizações de itens de trabalho, combinando seqüências, loops e condições com saltos de um estado para outro, e também o tipo de comportamento orientado a dados visto na colaboração da resolução do problema.

Regra e política

Como discutido anteriormente, uma maneira pela qual a abordagem do workflow pode agregar valor é isolando o foco da mudança em um aplicativo. Geralmente, encontra-se na forma segundo a qual os itens de trabalho são estruturados, mas, em alguns aplicativos, o foco da mudança está nas expressões vinculadas a uma estrutura que muda de forma relativamente lenta.

Um exemplo, é um sistema de cotação de apólices de seguro, no qual utiliza um conjunto de cálculos que mudam com frequência para orientar a tomada de decisão no processo de cotação. O requisito é que o workflow modele essas expressões, oferecendo dois benefícios principais: Em primeiro lugar, os custos com testes e implantação são muito inferiores àqueles que normalmente seriam incorridos se as expressões fossem escritas como código, já que fornece um forte isolamento que restringe o escopo das possíveis mudanças. Em segundo lugar, as mudanças podem ser feitas pelo pessoal que entende a importância das expressões para os negócios, mas não tem as habilidades de entendimento do código técnico nas quais as expressões escritas inevitavelmente precisariam ser incorporadas.

O padrão MVC (Model View Controller) geralmente é usado para vincular uma interface com o usuário (UI - User Interface) a um modelo de objeto subjacente (consulte a Figura 3). O modelo representa o comportamento do sistema, independentemente de qualquer representação particular da UI. O controlador faz parte da camada de UI que é usada para mapear os eventos gerados por ela nas chamadas de método necessárias para orientar o modelo. A própria UI não é contaminada por nenhuma pressuposição sobre o modelo subjacente.

Os workflows considerados até agora, vistos deste ponto de vista, enquadram-se todos na categoria de Modelos no sentido MVC. No entanto, o controlador também pode ser visto como um workflow. Os itens de trabalho que ele organiza são os métodos fornecidos pelos objetos do Modelo. O controlador também interage com a UI e com o modelo por meio de contratos bem definidos. Um modelo desse tipo geralmente é chamado de fluxo de página.

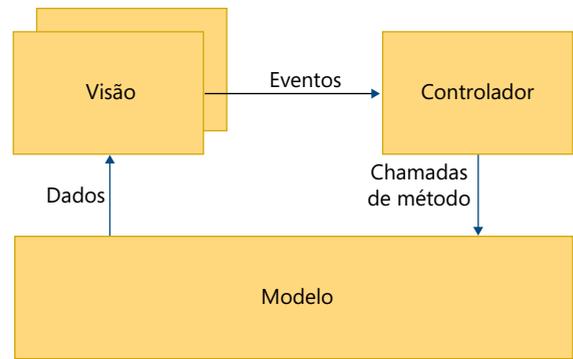
Como ocorre nas operações com script, o fluxo de página não seria implementado hoje com um produto típico do workflow. Há duas razões a serem consideradas ao se desenvolver um fluxo de página com uma plataforma de workflow. Em primeiro lugar, pode-se representar prontamente um modelo em termos visuais, ajudando os desenvolvedores e analistas a expressar e comunicar o comportamento exigido. Em segundo lugar, se o fluxo de página muda com frequência, então a abstração do fluxo de página como modelo aumenta a agilidade.

Há duas exigências principais caso esse problema precise ser resolvido com uma plataforma de workflow. O tempo de execução do workflow deve ser leve, já que um fluxo de página pode estar sendo executado em um pequeno aplicativo no computador desktop, e os contratos aceitos devem incluir o contrato baseado em eventos que é característico das UIs, bem como os contratos de método de objeto expostos pelo Modelo.

Agora vamos dar uma olhada em um exemplo de aplicativo de gravação/reprodução de teste. O objetivo deste exemplo final é verificar os limites da capacidade de aplicação da hipótese da plataforma de workflow.

O aplicativo utilizado testa os aplicativos desenvolvidos como conjunto de serviços. A ferramenta usa um mecanismo de interceptação para gravar toda a interação entre os serviços que ocorrem durante a realização manual de um caso de testes para o aplicativo. Essa gravação pode ser então reproduzida novamente. Durante o processo, são geradas mensagens de fonte externa sem a intervenção manual, e as mensagens entre o conjunto de serviços que compõe o aplicativo são verificadas quanto à seqüência e ao conteúdo em relação à gravação original.

Figura 3 Um aplicativo MVC



O workflow é o caso, organizando as unidades de trabalho que são os serviços participantes. O workflow é tanto ativo, na medida em que estimula o comportamento de mensagens de fonte externa, quanto passivo, na medida em que monitora as interações entre os serviços.

Um recurso exclusivo desse aplicativo é que o workflow é escrito não por um desenvolvedor ou por um usuário, mas por um programa, como parte do ato de gravação do caso. A criação do modelo de workflow deve ser totalmente programável. Também há exigências em termos de capacidade de extensão e atualização dinâmica.

A capacidade de extensão faz-se necessária porque a semântica estrutural é rica. Por exemplo, só porque duas mensagens chegaram a um serviço uma após a outra na gravação, não há implicação necessária de que essa ordem precise ser preservada em uma reprodução. Se não houver dependência causal entre as mensagens, uma reprodução que inverte a ordem das mensagens estará correta. Assim, a semântica da seqüência no modelo usado para gravar os casos de teste precisa incluir uma noção de causalidade, que provavelmente não é um recurso do modelo central de seqüência do workflow.

A atualização dinâmica é necessária porque a interação do indivíduo com o modelo ocorrerá durante a reprodução. As discrepâncias descobertas durante a reprodução, entre o comportamento gravado e observado, surgem imediatamente para um testador. Se a discrepância ocorrer porque uma mensagem inclui um campo data/hora que pode variar entre execuções, o testador atualizará o modelo para marcar o campo "não importa". Se a discrepância ocorrer em um teste de regressão porque o software mudou, o testador poderá aprovar a alteração e atualizar o teste para esperar o novo comportamento em todas as execuções subsequentes.

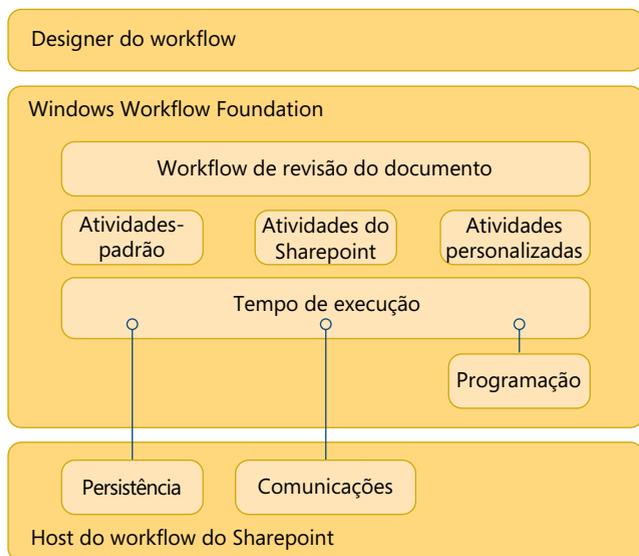
Valor da plataforma de workflow

Por definição, a plataforma de workflow não tem o conjunto completo de recursos oferecidos pelos produtos típicos atuais. Em vez disso, a plataforma de workflow considerada aqui se foca em fornecer suporte ao conceito de um workflow como modelo da organização de itens de trabalho. Já vimos que essa idéia de uma organização de itens de trabalho é de fato aplicável em uma ampla variedade de aplicativos, mas o fato de que se pode usar uma plataforma de workflow não significa que ela deva ser usada.

Dois questões devem ser respondidas: Que valor adicional é gerado da abordagem da plataforma de workflow? E essa abordagem é prática? Esse valor da abordagem da plataforma de workflow deve vir da expressão da organização do trabalho como modelo, que vamos discutir depois. Vamos resumir as características que uma plataforma prática e eficaz de workflow deve apresentar.

Para demonstrar como um modelo difere do código, o trecho de código a seguir é um workflow válido para a definição usada aqui, ou seja, uma organização de unidades de trabalho:

Figura 4 O esquema de implementação da revisão de documento



```
public void HandleLoanRequest (
    string customerID,
    Application app)
{
    if (CheckCredit(
        customerID, app.Amount))
    {
        MakeOffer (customerID, app);
    }
}
```

E, sob certo sentido, é um modelo. É possível analisar esse código e desenvolver uma árvore CodeDOM que o represente.

No entanto, a semântica do modelo resultante é tão geral quanto opaca. É possível dizer que o código contém chamadas de função, mas não é fácil distinguir uma função que represente a chamada de um item de trabalho de uma função que converte inteiros em texto. Um modelo de workflow faz uma distinção explícita dessas idéias. Em geral, utiliza um elemento especializado do modelo que representa a chamada de um item de trabalho, e as funções de conversão não podem ser expressas diretamente no modelo. Assim, um modelo de workflow é aquele no qual seu gráfico é desenvolvido a partir de elementos que são significativos no domínio do workflow. A riqueza semântica de tal modelo pode ser explorada de várias formas.

Visualização. A representação visual do modelo – em geral, em forma gráfica – é útil para o desenvolvedor, durante o desenvolvimento e a manutenção, e também para os usuários, que desejam saber por que lhes foi atribuída uma determinada tarefa, ou para o profissional de operações de TI que desejam entender o que um aplicativo que se comporta de forma incorreta deve estar fazendo.

Conhecimento. O modelo de workflow é receptivo a desenvolvimento de código para uma série de finalidades. Um exemplo é uma análise estática para determinar as dependências e o fluxo de trabalho em um conjunto de workflows auxiliares, ou o uso do modelo para orientar uma simulação que prevê as cargas de trabalho que serão geradas por uma nova versão do processo.

Expressividade. A especialização do modelo de workflow para o domínio do workflow indica que os problemas característicos podem ser expressos

de forma mais rápida e compacta. É uma DSL – Domain-Specific Language (Linguagem de Domínio Específico), especializada em fornecer suporte a problemas característicos. Pensemos em um processo de revisão de documentos no qual três votos positivos de cinco revisões indicam que o documento é bom, e que se podem cancelar quaisquer outras revisões. Esse processo é bastante difícil de codificar, mas um modelo de workflow pode fornecer estruturas prontas que resolvem esses problemas.

Mais Exploração semântica

Como vimos na discussão sobre aplicativos de operações com scripts, ampliar o modelo de workflow para especializar ainda mais a linguagem de modelos pronta é uma técnica muito poderosa fornecida para adicionar valor. Um exemplo disso é a criação de uma linguagem destinada a usuários finais, como na revisão de documentos realizada com uma definição improvisada do processo de revisão que foi discutida anteriormente.

Execução. A especialização do modelo possibilita adicionar suporte de tempo de execução para problemas comuns. Um bom exemplo é o estado de longa duração. Dos aplicativos discutidos aqui, o gerenciamento do estado de longa duração é necessário para o processo de revisão de documentos, para a colaboração da resolução de problemas e nos aplicativos voltados ao usuário. O tempo de execução da plataforma de workflow pode resolver esses problemas complexos uma vez, usando elementos simples e expressivos de modelo para controlar uma funcionalidade comum e liberar o desenvolvedor para focar nos problemas de negócio.

Monitoramento. A existência de um modelo possibilita produzir um fluxo de eventos com uma semântica significativa sem nenhum esforço adicional do desenvolvedor. Dos aplicativos aqui descritos, esse fluxo de eventos é útil na revisão de documentos, na colaboração de resolução de problemas, na gravação/reprodução de testes e nos aplicativos voltados ao usuário. O fluxo de eventos pode ser usado para monitorar instâncias de workflow ou desenvolver visões agregadas do estado de um grande número dessas instâncias. A padronização do fluxo de eventos facilita o desenvolvimento de tais visões agregadas nos workflows que foram desenvolvidos independentemente um do outro.

Outra idéia poderosa é a apresentação de erros usando uma semântica de negócios. Em geral, uma falha técnica, como a não entrega de uma mensagem, leva a contatar um especialista técnico, pois a importância da falha não fica clara sem uma investigação especializada. Se o erro pode ser

“O WF IMPLEMENTA A IDÉIA DO WORKFLOW COMO ORGANIZAÇÃO DOS ITENS DE TRABALHO, ABSTRAÍDA DAS IDÉIAS COM AS QUAIS FOI VINCULADA NOS PRODUTOS TRADICIONAIS DE WORKFLOW”

mapeado para um modelo de workflow – de forma que fique claro que o erro tem a ver com uma notificação não crítica de mudança, por exemplo – então o contato pode ser restrito a casos em que é necessário.

Composição. Se um aplicativo é dividido em unidades com interfaces de trabalho bem compreendidas, podem ser reutilizadas por outros fluxos de trabalho. Os próprios fluxos de trabalho também definem unidades que podem ser igualmente usadas por outros fluxos.

Personalização. Suponhamos que uma empresa de desenvolvimento (ISV) envie um workflow que é personalizado por um revendedor de valor agregado (VAR) e depois novamente personalizado por um cliente. A reaplicação dessas personalizações quando a ISV envia uma nova versão

básica é um problema de manutenção que apresenta desafios. A utilização de um modelo compartilhado e bem compreendido de workflow faz com que as conseqüentes fusões de três versões fiquem mais manejáveis. A personalização e a composição permitem que existam ecossistemas onde definições de trabalho e fluxo se tornam artefatos compartilhados ou comercializados.

Manipulação. Como vimos nas discussões da revisão do documento e nos aplicativos de gravação/reprodução de testes, em geral há exigências para inventar ou modificar os fluxos de trabalho às pressas. Essa modificação não pode ser realizada com segurança caso necessite de uma mudança no código. Usar um modelo possibilita a manipulação dinâmica que é controlável e abrangente.

Esses benefícios fazem uma lista convincente que demonstra claramente que a descrição de uma organização de itens de trabalho como modelo tem muito a oferecer.

Características da plataforma

Deve existir suporte para conceitos estruturais básicos como seqüências, condições e loops, e também o suporte às abordagens dirigidas a dados para que seja possível lidar com as estruturas pouco organizadas que aparecem em aplicativos que orientam o usuário de colaboração na resolução de problemas.

É importante permitir que novos elementos semânticos sejam adicionados para criação de linguagens ricas e especializadas, como a composição de operações com scripts preparados para fluxo de dados. Adicionar novos elementos semânticos pode ir longe o suficiente para exigir a redefinição de idéias fundamentais como seqüências – por exemplo, no aplicativo de gravação/reprodução de testes.

O workflow também deve conseguir se comunicar de formas variadas. Os fluxos de trabalho respondem a eventos de UI, orientam tipos diferentes de serviços (manuais, desenvolvidos e outros fluxos de trabalho) e dão suporte à pesquisas sobre o estado atual dos contratos – por exemplo, na hora de determinar as ações disponíveis a um ator em um aplicativo de colaboração de resolução de problemas.

Se a plataforma de workflow deve ser usada em todos os aplicativos em que acrescenta valor, como MVC, então ela deve ser leve. Da mesma forma, precisa resolver as exigências de escalabilidade e desempenho indicados pelos aplicativos, como a revisão de documentos.

Além disso, o modelo de workflow em si deve ser totalmente programável, incluindo criação de modelos (como no aplicativo de gravação/reprodução de testes) e a atualização dinâmica do modelo para aceitar a mudança não-antecipada (como nos aplicativos de gravação/reprodução de testes e de revisão de documentos).

Agora vamos dar uma olhada na aplicação dessas características necessárias na Windows Workflow Foundation (WF). Até agora recapitulamos de forma eficaz o raciocínio que fomentou o desenvolvimento da WF, como aplicação desses conceitos, é o meio pelo qual esse valor pode ser traduzido em soluções entregues.

A WF implementa a idéia do workflow como organização de itens de trabalho, abstraída das idéias com as quais foi vinculada nos produtos tradicionais de workflow. As abstrações enquadram-se em três categorias principais: projeto e visualização, hospedagem e semântica.

Projeto e visualização. O workflow na WF é uma árvore de itens de trabalho (chamados de atividades). Essa árvore pode ser manipulada diretamente como modelo de objeto. É fornecido um designer, mas sua utilização não é obrigatória. É possível criar novos designers especializados em comunidades particulares de usuários ou em organizações particulares de itens de trabalho. Também é possível especializar o designer fornecido,

que pode ser usado não apenas no Visual Studio, mas também em qualquer outro aplicativo de hospedagem.

Hospedagem. O tempo de execução da WF é suficientemente leve para ser hospedado em um contexto de cliente, como um controlador em uma aplicação “rich-client”. Também apresenta um desempenho suficiente para escalar quando incorporado em um host de servidor, como no SharePoint

“O FOCO DO RUNTIME DO WF É FORNECER OS RECURSOS EXIGIDOS POR QUALQUER WORKFLOW, EVITANDO A NECESSIDADE DE IMPLEMENTÁ-LOS NOVAMENTE EM APLICATIVOS DIFERENTES, MAS SEM COMPROMETER A FLEXIBILIDADE DE UMA ABSTRAÇÃO DE WORKFLOW”

Server fornecido com o Office 2007. As expectativas do runtime da WF quanto ao seu host são abstraídas como interfaces para serviços como threading, transações, persistência e comunicações. Implementações úteis dessas interfaces estão disponíveis no tempo de execução, mas podem ser substituídas conforme necessário.

Semântica. Problemas diferentes respondem à semânticas de modelo diferentes. A WF fornece suporte a três estilos principais de workflow: fluxo, máquina de estado e orientação a dados, sendo adequado para os aplicativos em que o workflow está no controle, como ocorre no exemplo de operações com script. A máquina de estado é mais adequada quando o workflow é orientado por eventos externos, como no MVC ou em aplicativos que guiam o usuário. Por fim, a abordagem dirigida a dados é adequada para os aplicativos em que as ações dependem do estado, como na colaboração de resolução de problemas.

Essa semântica pode ser estendida desenvolvendo atividades personalizadas para criar um vocabulário específico de domínio para utilização em qualquer um desses estilos. No entanto, como a estrutura de um workflow é por si expressa como conjunto de atividades pode-se usar a mesma abordagem para definir estilos e semânticas inteiramente novas, se necessário.

Um tempo de execução comum de workflow

O foco do tempo de execução da WF é fornecer os recursos exigidos por qualquer workflow, evitando a necessidade de implementá-los novamente em aplicativos diferentes, mas sem comprometer a flexibilidade de uma abstração. Esses recursos comuns enquadram-se em quatro categorias principais: programação de atividades, transações e estado de longa duração, exceções e compensação, e comunicações. Vamos dar uma olhada em cada um deles em mais detalhes.

Programação de atividades. Define um protocolo de atividades que todos os itens de trabalho implementam. Esse protocolo define o ciclo de vida básico da atividade (iniciado, em execução e fechado) e os estados adicionais necessários para lidar com exceções (com defeito, cancelando e compensando). Essa definição permite que se forneça a programação de atividades para todos os fluxos de trabalho.

Transações e estado de longa duração. Fornece suporte à execução de transações ACID. Essas transações são particularmente úteis para manter a consistência no estado do workflow e no estado externo, como no estado de aplicativos e mensagens. No entanto, as transações ACID não são adequadas para gerenciar o estado de longa duração devido às suas implicações de recursos e bloqueios. Implementa um mecanismo mais

amplo de ponto de verificação e recuperação para lidar com o estado de longa duração. Desse ponto de vista, as transações ACID tornam unidades da execução em uma framework maior. O desenvolvedor não precisa fazer nenhum trabalho para obter o benefício do suporte da WF ao estado de longa duração, pois é o comportamento-padrão. No entanto, se for necessário um controle mais detalhado, é fornecido um conjunto de elementos simples para essa finalidade.

Exceções e compensação. A idéia familiar de exceções throw-try-catch é aceita e representada no modelo de workflow padrão. No entanto é aceita uma visão mais ampla do tratamento de falhas que inclui a idéia de compensação para as unidades transacionais concluídas com êxito.

Comunicações. Como vimos, os fluxos de trabalho precisam se comunicar de várias formas, e isso se reflete na WF, que fornece suporte à comunicação por meio de métodos NET, das interfaces de eventos e das interfaces de Web Services. No futuro, também será disponibilizado o

“AS TRANSAÇÕES ACID SÃO PARTICULARMENTE ÚTEIS PARA MANTER A CONSISTÊNCIA NO ESTADO DO WORKFLOW E NO ESTADO EXTERNO, COMO NO ESTADO DE APLICATIVOS E MENSAGENS”

suporte à Windows Communication Foundation. Assim, a WF aplica de fato a abordagem de plataforma de workflow aqui proposta.

A Figura 4 ilustra o esquema de implementação de alto nível do aplicativo de revisão de documentos e como tudo o que vimos se interliga. Uma implementação usa o SharePoint como host do workflow, que usa o serviço de programação padrão fornecido atualmente com a WF. No entanto, os serviços padronizados de persistência e comunicação são substituídos por implementações especializadas para o host do SharePoint. O serviço de persistência armazena o estado de longa duração do workflow no banco de dados do SharePoint, e o serviço de comunicação disponibiliza os recursos avançados de interação com o usuário do SharePoint ao workflow. Esses dois serviços são, na verdade, fornecidos com o Microsoft Office 2007.

Três tipos de atividade são usados para definir o próprio workflow de revisão de documentos. Em primeiro lugar, as atividades prontas da WF são usadas para fornecer elementos estruturais como If-Else e While. Em segundo lugar, as atividades oferecidas como parte do Office 2007 são usadas para acessar os serviços de comunicação com o usuário do SharePoint. Em terceiro lugar, são usadas atividades personalizadas para implementar a semântica específica da organização para encaminhamento e delegação de uma forma padrão e reutilizável. O WF designer é usado como meio de definir o workflow e também fornecer ao proprietário representações visuais do estado de uma instância do workflow de revisão de documentos.

Atacando os problemas

Em resumo, a plataforma de workflow fornece suporte a uma abstração de idéias que tornou os produtos de workflow uma abordagem atraente aos problemas de negócio. Ela não substitui os produtos atuais de workflow, no entanto. Em vez disso, divide-os em plataforma e superestrutura.

A plataforma de workflow incorpora duas idéias principais: uma é uma organização de unidades de trabalho, e a outra é um modelo, ou seja, uma descrição legível por máquina que não seja código. Essas idéias são valiosas em uma ampla variedade de aplicativos, dentro e além do domínio de problemas solucionados por produtos típicos do workflow.

Tal plataforma de workflow é mais útil se for de baixo custo e estiver em todos os lugares.

Os principais benefícios oferecidos surgem de uma expressão da organização de itens de trabalho como modelo, que tem várias vantagens em relação a uma representação em código

- **Transparência.** Os objetivos de negócios do sistema são claros, permitindo que os usuários e a equipe de TI se comuniquem de forma eficaz sobre o comportamento desejado e sobre a forma como serão abordados o projeto para agilizar todo o processo.
- **Isolamento da mudança.** As áreas do aplicativo que mudam com mais probabilidade são expressas como workflow, e não como código. Isolando as partes que se movem com rapidez no aplicativo, é possível realizar as mudanças com mais confiança.
- **Agilidade.** O resultado de todos esses benefícios é a agilidade nos negócios. Se os usuários corporativos podem entender o sistema, os desenvolvedores podem acelerar o processo, e os riscos associados à mudança são minimizados. Assim, o sistema pode ser classificado como ágil.

Uma plataforma amplamente útil de workflow deve ter estas características: definir um modelo central de workflow como padrão, que seja extensível e totalmente programável no momento do projeto e no tempo de execução; conseguir se comunicar de diversas formas; ser leve e incorporável; e conseguir ser escalável e ter um bom desempenho em ambientes de altos volumes. A WF é um produto que exibe todas essas características. Como componente do WinFx e parte da plataforma Windows, a WF também custa pouco e está em todos os lugares.

O conceito de plataforma de workflow, como descrito aqui, é um modelo apropriado e personalizável com vários benefícios para uma ampla variedade de aplicativos, e é totalmente aplicável na WF. •

Sobre o autor

David Green entrou na IBM como desenvolvedor no laboratório Hursley em 1977. Desde então, subiu e desceu na cadeia de suprimentos de software, trabalhando em um jornal, na American Express, na Siemens Nixdorf e em um grande banco do Reino Unido, em várias funções técnicas, de pré e pós-venda. O tema que percorre toda a sua experiência é o desenvolvimento de aplicativos para o mundo dos negócios, achando que era muito mais difícil do que deveria ser criar grandes soluções de negócios e tentando criar abordagens e ferramentas para fazer alguma coisa sobre isso. David passou os últimos dois anos na Microsoft trabalhando no Windows Workflow Foundation, que ele acredita ser um acréscimo significativo ao arsenal do desenvolvedor de aplicativos.



A metáfora Amazing Race

por Vignesh Swaminathan

Resumo

O BPM – Business Process Management (Gerenciamento de Processo de Negócio) está chegando aos ouvidos de todo mundo atualmente, e muitas empresas já estão percebendo o potencial da automatização dos processos de negócio. À medida que o falatório se estabiliza e o BPM se aproxima do campo da produtividade, há alguns desdobramentos práticos. Um deles é a necessidade de um melhor gerenciamento dos processos de negócio de alto nível, que são compostos por processos de negócio simples e reutilizáveis. O interessante é que a resposta não se dá inteiramente por meio da modelagem gráfica, mas também através do gerenciamento de processos de negócio de alto nível por meio de regras isoladas de definição de processo. Vamos dar uma olhada na definição, nos benefícios e na implementação das regras de definição de processo.

A Amazing Race da CBS é um programa de televisão que vai ao ar nos EUA no qual os participantes competem em uma corrida ao redor do mundo, indo de cidade em cidade. Para chegar a uma cidade, eles utilizam meios comuns de transporte, como carro, trem ou avião. As equipes participantes precisam encontrar uma pista em cada cidade para descobrir a próxima que devem visitar. As pistas controlam a escolha do próximo mapa de rota a ser seguido; os participantes não conhecem essas pistas nem o próprio mapa as contém. Ou seja, as pistas são divulgadas quando os participantes vão de uma cidade para outra.

Usando essa analogia, realiza-se todo o processo de viajar pelo mundo (vamos chamar isso de “processo global”) combinando processos menores de ir de uma cidade para outra (vamos chamar essa etapa de “processos de cidade”), criando o jogo na televisão. Por sua vez, os processos de cidade que formam determinado processo global são determinados pelas pistas. Os participantes também decidem, por exemplo, pegar um avião caso não possa chegar à cidade de carro ou de trem. Essas são as regras internas simples que são aplicadas no processo de cidade. As pistas, por outro lado, são inteiramente externas ao processo de cidade e controlam o processo global.

Uma idéia interessante para o jogo seria se as pistas determinassem o próximo processo de cidade, com base nas características dos jogadores, a cidade na qual estavam e outros parâmetros relevantes. Esse processo levaria a uma variedade maior de cidades para as quais os participantes poderiam viajar em seguida, todas as vezes que viajassem com êxito para uma cidade. Embora esse cenário fique interessante para os participantes,

ele não é mais um cenário atraente para os criadores do jogo. Agora eles têm uma gama enorme de possíveis processos de cidade que podem ser combinados no processo global. Os processos de cidade combinados são dinâmicos e não são predeterminados (consulte a Figura 1). Já que apenas as opções são predeterminadas, e não o processo de cidade final, quanto mais parâmetros houver para se criar uma pista, maior será a complexidade de opções nas mãos deles para criar o processo global.

A Amazing Race do BPM empresarial

Agora, o que a Amazing Race tem a ver com o BPM na empresa? As empresas têm uma abordagem muito semelhante dos processos de negócio em seus domínios. Cada empresa tem um repositório valioso de processos de negócio que são aplicáveis em um subdomínio específico na empresa. Esses processos de negócio de subdomínio são modelados para se realizar uma tarefa ou workflow em particular no contexto desse domínio. Na maioria dos casos práticos, as empresas começam a perceber que os processos de negócio aplicáveis no subdomínio são reutilizáveis em um contexto maior. As empresas então acabam criando processos de negócio válidos para toda a empresa (alto nível) reutilizando os processos de negócio de subdomínio (baixo nível).

A maturidade da abordagem de criar um processo de negócio de alto nível retoma a analogia da Amazing Race. As empresas percebem a eficiência de combinar seus processos de cidade para fazer um de seus processos globais (consulte a Figura 2). Então elas começam a combinar os processos de cidade de modos mais inovadores, usando regras baseadas em parâmetros. Nesse ponto, elas também enfrentam um dilema semelhante àquele enfrentado pelos criados do jogo, que é o de gerenciar a complexidade criada pelas ricas possibilidades de usar regras baseadas em parâmetros para controlar o processo global.

Outro dilema com a analogia do jogo é que os criadores dele têm um enorme labirinto de opções para visualizar um único processo global que forme o episódio desta temporada. Usando esse labirinto, eles conseguem manter o processo global para a atual temporada.

Imagine uma situação na qual, devido a certas circunstâncias imprevisíveis, como catástrofes naturais, o início de uma guerra, terrorismo ou qualquer coisa que saia do controle, os participantes do jogo não conseguem usar um processo de cidade em particular. Nesse caso, eles teriam que voltar e alterar o processo global para garantir que o processo de cidade problemático seja removido do processo global ou tenha uma alternativa. Como o processo global é controlado agora por vários parâmetros e regras baseadas nesses parâmetros, todo o modelo do processo global precisa ser verificado manualmente para garantir que não haja possibilidade de atingir o processo de cidade problemático. Além disso, o jogo já está no ar nesta temporada, e o processo global está em

andamento. Esse exemplo exige que, mesmo com o jogo no ar, nenhuma das equipes participantes acabe no meio de uma cidade em guerra!

O dilema é ainda mais crítico na empresa. O processo global da empresa (alto nível) será influenciado pelas mudanças no processo de cidade da empresa (baixo nível). As razões para a mudança podem ser igualmente imprevisíveis, como ocorre no jogo da televisão. O negócio pode exigir que a empresa consiga mudar dinamicamente em um processo de alto nível para atender às novas e imprevisíveis exigências dos negócios (consulte a

Figura 3). O processo global da empresa também está em andamento, e, mesmo com o processo em andamento, será obrigatória uma mudança para que a empresa tampouco acabe no meio de uma cidade em guerra. Ao contrário do jogo na televisão, partes da empresa não podem ser simplesmente canceladas. O show deve continuar!

Processos de negócio de alto nível

Como mencionado anteriormente, os processos globais são processos de alto nível, e como tais são gerenciados inteiramente por usuários não técnicos. Os criadores do jogo têm mais habilidades em gerenciar um jogo do que experiência em tecnologia de transmissão. Esse cenário se aplica também para a empresa na qual o processo de negócios de alto nível é gerenciado por gerentes executivos que estão familiarizados com a tarefa de dirigir a empresa, e não por especialistas na criação de modelos de computador da empresa. Assim, o primeiro aspecto do gerenciamento de um processo de negócio de alto nível é a necessidade de retirar do usuário as tecnicidades.

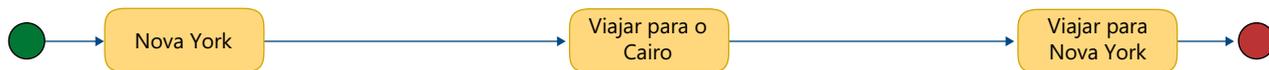
A segunda necessidade é garantir que um processo global em execução possa ser alterado dentro de determinados limites. Essa flexibilidade é uma exigência importante para os processos de negócio de alto nível, que estão sujeitos a um maior impacto resultante da mudança nos negócios do que os processos de baixo nível.

A terceira necessidade é fornecer uma interface clara, fácil e simples para o usuário gerenciar o processo global. Essa interface de gerenciamento deve garantir a criação e a manutenção de um processo de negócios de alto nível. A mesma interface deve permitir que o usuário altere convenientemente os processos globais em andamento, e deve permitir uma pesquisa facilitada das regras baseadas em parâmetro, as quais controlam o processo de negócios de alto nível. Essa pesquisa permitiria que o usuário localizasse rapidamente as regras que desejam alterar. A interface deve ser leve por natureza, permitindo que os usuários atualizem rapidamente o processo de negócio de alto nível sem esforço.

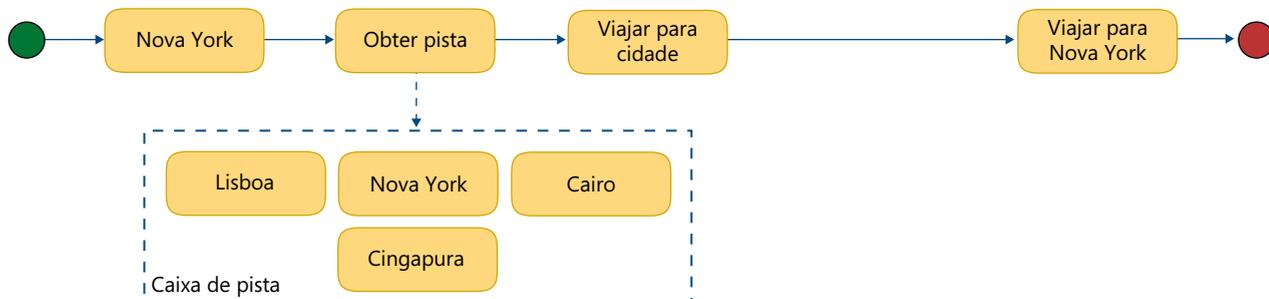
A exigência final é compreender que o processo global é determinado mediante a combinação de um repositório diversificado de processos de cidade fundamentado em regras baseadas em parâmetros. Na prática, o número de parâmetros não é limitado, mas pode passar de 30 em muitos cenários reais de empresas. Encontramos esse problema prático em uma dos projetos de nossos clientes. O cliente, um grande banco europeu com sede na Holanda, enfrentou esse problema no seu setor varejista. Havia uma média de cinco parâmetros definidos para tratar de cada etapa em um processo de solicitação de empréstimos. Cada parâmetro tinha entre 3 e 100 valores possíveis que levavam a um processo complexo de negócios, considerado impossível de ser gerenciado pelos meios convencionais.

Figura 1 Viajando pelo mundo

1. Viagem simples pelo mundo



2. A Amazing Race (pistas estáticas)



3. A Amazing Race com pistas inteligentes (pistas baseadas em parâmetros)

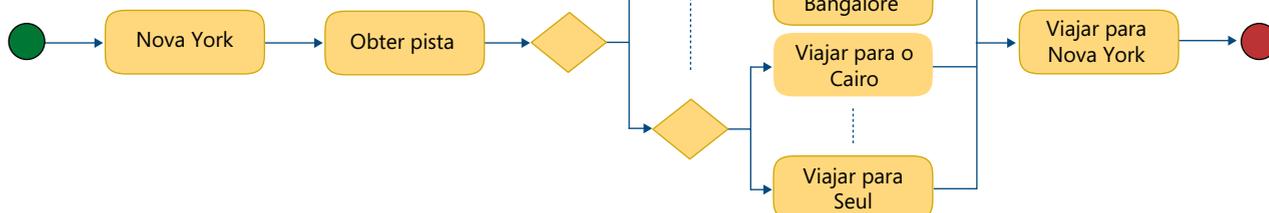
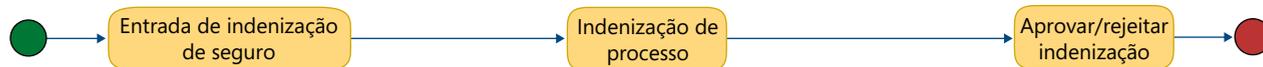
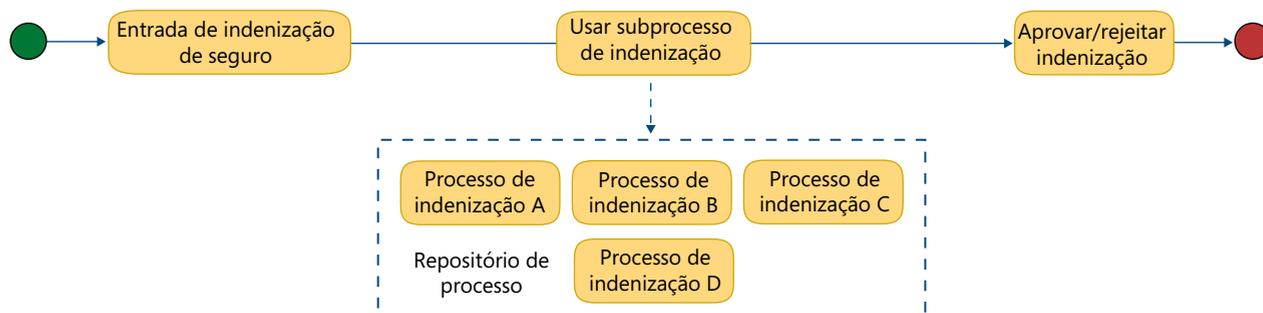
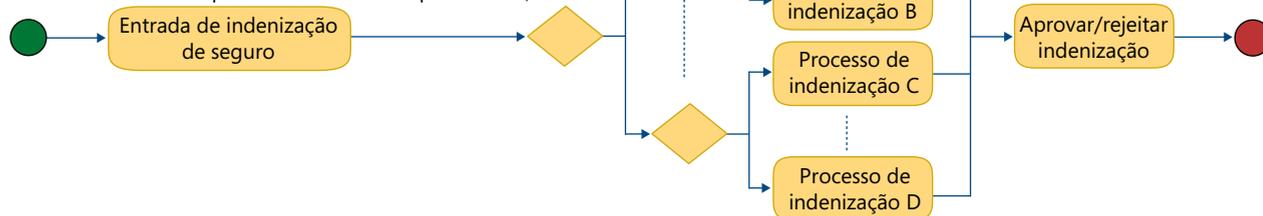


Figura 2 A empresa

1. Processo de negócios simples
(etapas redundantes)2. Processo de negócios de alto nível
(chamadas de subprocessos estáticos)3. Processo de negócios de alto nível
(encaminhamento de subprocesso baseado em parâmetros)

Para mostrar essa complexidade em ação, partamos do princípio de que a Amazing Race sempre tem cinco equipes competindo em cada nova corrida (processo global). Cada equipe em qualquer estágio do jogo teria viajado por uma lista de cidades antes de viajar para próxima cidade. Para este exemplo, há um total de dez cidades pelas quais as equipes podem passar, o que significa dez possibilidades diferentes para a cidade na qual a equipe está atualmente.

Elas também teriam chegado à cidade atual usando um dos três meios principais de transporte, ou seja, carro, trem ou avião. Usando essa analogia, o trabalho dos criadores do jogo seria montar o processo global com estes três parâmetros: número de equipes, lista de cidades anteriores e meio de transporte. Para uma combinação exclusiva desses três parâmetros, um processo de cidade particular deve ser combinado no processo global. O processo global deve garantir que os participantes não usem o mesmo meio de transporte mais do que uma vez continuamente, e também garantir que eles terminem a corrida na cidade em que a começaram.

Aplicando as regras

Os criadores do jogo também devem usar o número de equipes para determinar certas condições especiais aleatoriamente, a fim de garantir que o jogo tenha um conteúdo mais diversificado, certificando-se de que as equipes passem por cidades diferentes, em ordens diferentes e em meios de transporte diferentes. Assim, o próximo processo de cidade a ser usado no processo global em qualquer estágio deve ser determinado pela equipe envolvida, pela cidade atual e pelo meio de transporte que foi usado pela última vez.

Essa combinação se traduz em aproximadamente 150 caminhos diferentes de fluxo que podem determinar o próximo processo de cidade a ser usado no processo global. O número de caminhos de fluxo possíveis é determinado desta maneira: 5 equipes × 10 cidades atuais possíveis × 3 meios possíveis de transporte = 150 opções exclusivas para determinar o próximo processo de cidade (consulte a Figura 4). Por exemplo, a equipe número três poderia ter chegado a Nova York de trem e depois recebido o próximo processo como “para Londres de avião”.

“O PRIMEIRO ASPECTO DO GERENCIAMENTO DE UM PROCESSO DE NEGÓCIO DE ALTO NÍVEL É A NECESSIDADE DE RETIRAR DO USUÁRIO AS TÉCNICIDADES”

Essa complexidade se apresenta também para as empresas. Os domínios financeiros e de seguro fornecem vários cenários que necessitam de processos de negócio de alto nível controlados por regras baseadas em parâmetros. Por exemplo, uma indenização de seguro a ser investigada por uma empresa mundial de seguros pode precisar de um processo de alto nível de indenização que escolha um processo de investigação de indenização específico do país, baseado em outros parâmetros extras. Imaginemos que a indenização seja categorizada em cinco faixas etárias possíveis, de dez países diferentes de operação e pertença a três grupos definidos pela faixa de valor da indenização:

5 faixas etárias possíveis × 10 países possíveis de operação × 3 faixas possíveis de valor da indenização = 150 opções exclusivas para determinar qual processo de investigação de indenização deve ser usado.

Por exemplo, se a faixa etária for B (30–45), o país for a Índia e a faixa de valor da indenização for maior do que US\$ 150.000, use o “processo de investigação de indenização na Índia Grupo C, de alto valor”.

Se voltarmos para o exemplo financeiro anterior envolvendo o banco da Holanda, os cálculos funcionariam assim: 5 canais possíveis de interação com o cliente × 3 tipos possíveis de usuário × 4 segmentos possíveis de cliente × 100 ofertas possíveis de produto × 7 etapas possíveis de processo de alto nível = 42.000 implementações possíveis no repositório do processo.

Um item extra a ser considerado: na metáfora da Amazing Race, o número de processos de cidade disponíveis se reduz a cada iteração da etapa do processo de viagem à cidade. No entanto, nas empresas em tempo real, não é apenas uma etapa do processo que tem uma ampla gama de implementações possíveis de processo, mas todas as etapas do processo de negócios de alto nível podem ter várias implementações possíveis. A metáfora foi deliberadamente simplificada, e o aumento da complexidade em cada nova etapa que depende das regras baseadas em parâmetros é mostrado na Figura 5.

Como podemos gerenciar esses cenários? Ainda queremos usar modelos gráficos para gerenciar o processo de negócios de alto nível?

A noção inicial e imediata que atinge o usuário quando usamos um exemplo de processo global e de cidade é criar o processo global como subprocessos de cidade. Essa noção não chega a surpreender, uma vez que a

utilização de subprocessos é a maneira mais comum de combinar processos simples e processos mais complexos. Usar subprocessos é uma boa opção quando estamos tentando fragmentar um processo de negócios em submódulos para melhor reutilização. No entanto, eles não são aplicáveis a processos de negócio de alto nível controlados por regras baseadas em parâmetros, pois, quando são usados subprocessos, as decisões que controlam o fluxo dos processos de negócio de alto nível são incorporadas e ficam internas ao processo.

Essa característica viola a necessidade de haver flexibilidade, como declarado nas exigências de processos de negócios de alto nível. Também não resolve o problema de gerenciar as decisões, uma vez que o usuário fica com uma opção ineficiente para modelar 150 caminhos diferentes de fluxo nos processos de negócio de alto nível e carregar o enorme modelo gráfico resultante todas as vezes que surgir a necessidade de haver uma mudança no processo global. Além disso, essa abordagem não se encontra em sintonia com os princípios orientados a serviço de acoplamento flexível, já que os processos, no final, seriam expostos como serviços.

Qualquer esforço de otimização para dividir ainda mais os pontos de decisão em subprocessos também seria inútil, pois o dilema é a visibilidade do processo de negócios de alto nível, e o usuário agora tem um problema adicional que é a exigência do detalhamento para fazer as alterações. Mesmo que o usuário passe pelas alterações, essa abordagem não garante que possam ser feitas alterações em um processo global em andamento. Assim, usar subprocessos em todos os aspectos não resolve o problema do BPM de alto nível.

Figura 3 A complexidade

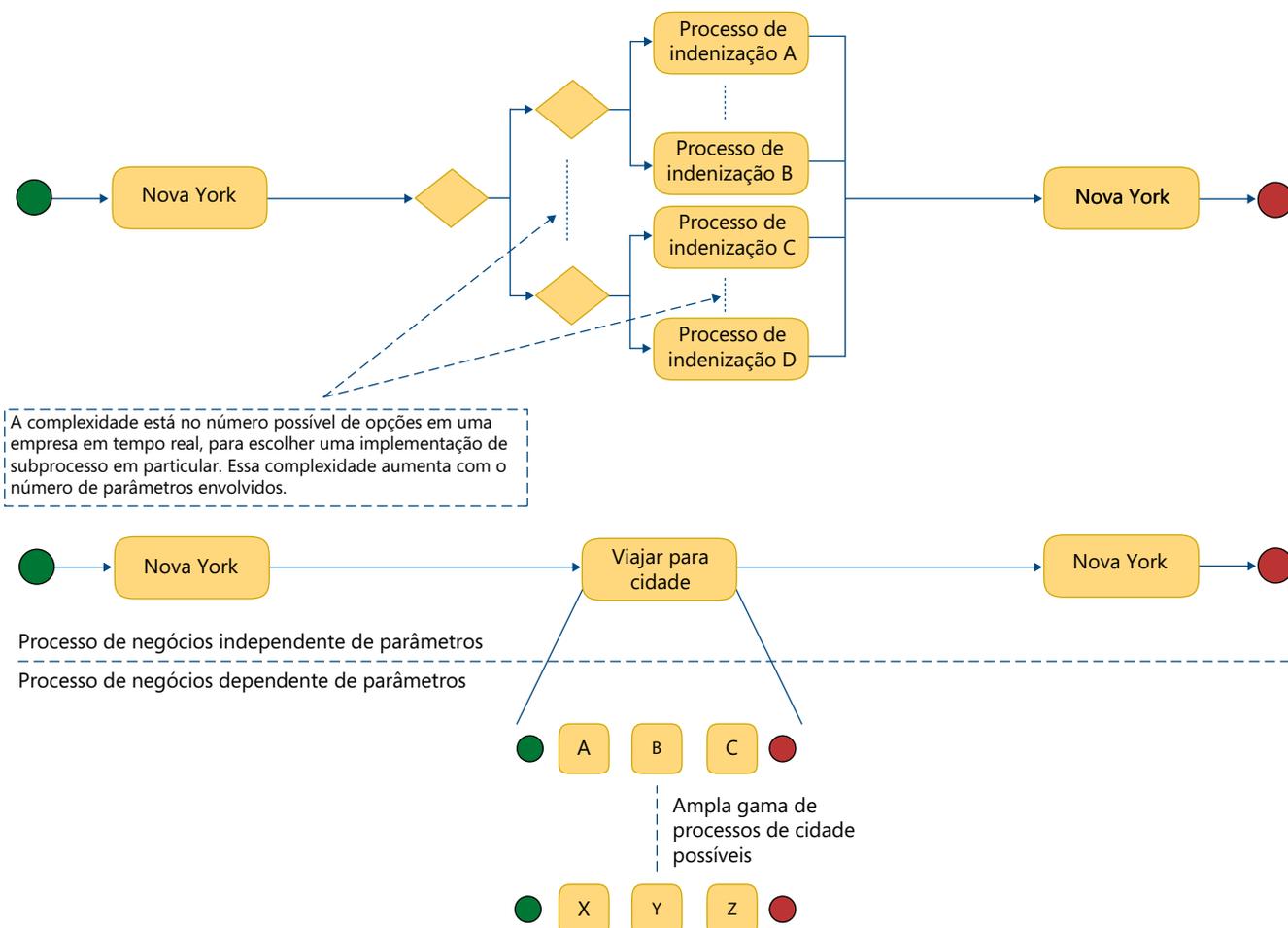
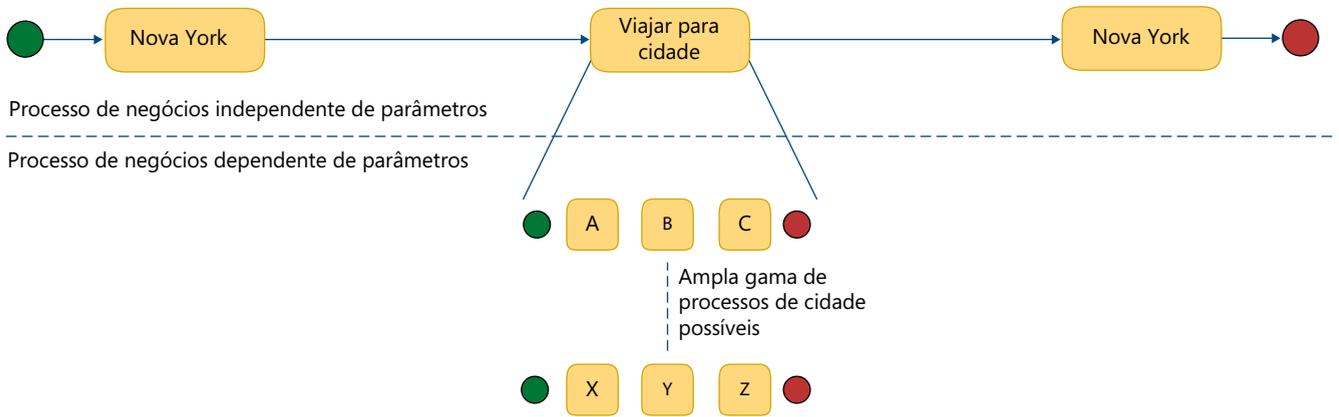


Figura 4 Determinando o número de caminhos de fluxo



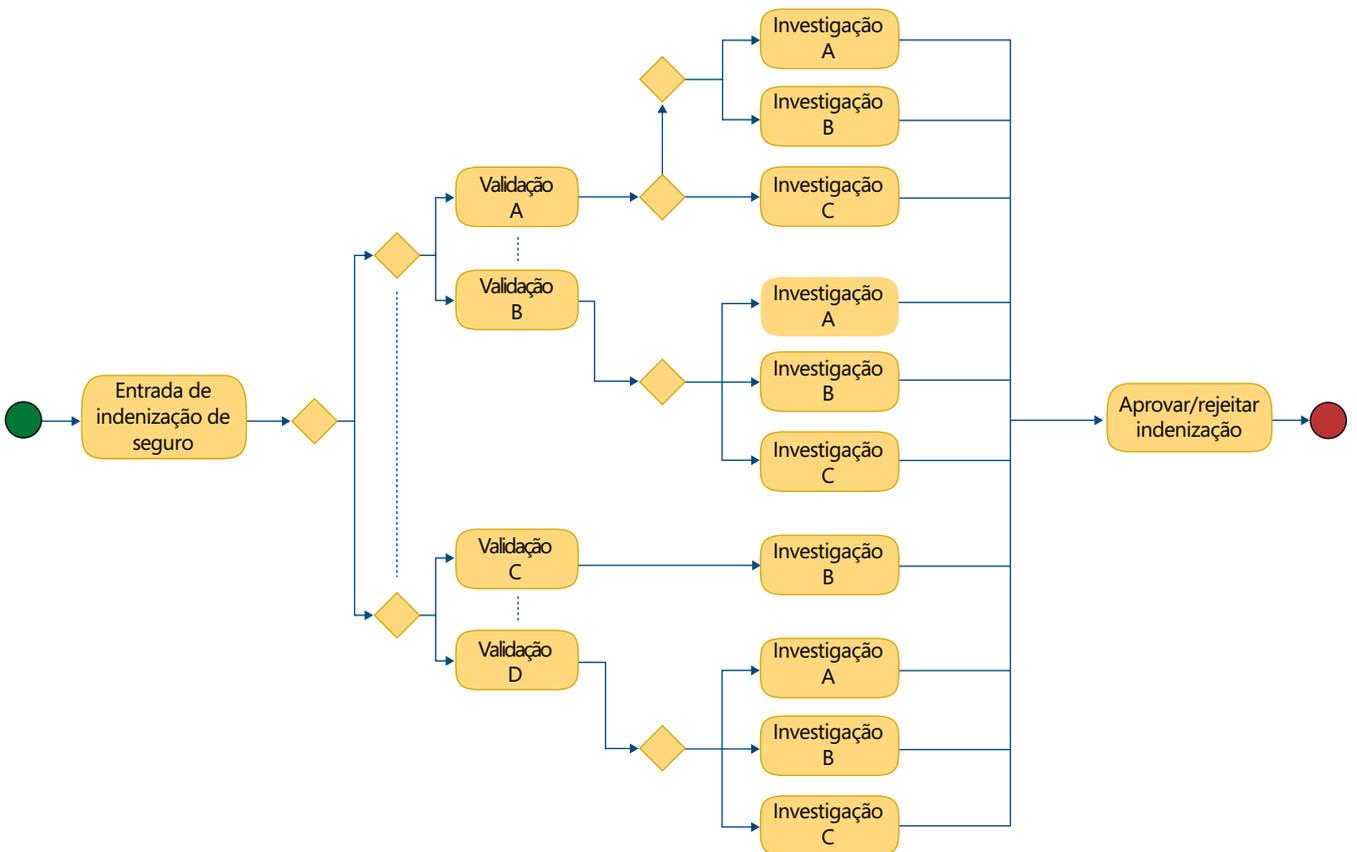
Exemplo: os três parâmetros para uma corrida internacional podem ser equipe, cidade e transporte.

| Equipe | | Cidade | | Transporte | Possíveis implementações exclusivas |
|--------|---|--------|---|------------|-------------------------------------|
| 5 | x | 10 | x | 3 | 150! |

Acrescentando um país mais de operação e duas faixas mais de valor da indenização:

| | | | | | |
|---|---|----|---|---|------|
| 5 | x | 11 | x | 5 | 275! |
|---|---|----|---|---|------|

Figura 5 Mais uma etapa



Condição externalizada

Uma das principais desvantagens da solução de subprocesso era que as decisões estavam incorporadas e estáticas no processo de negócio de alto nível. Algumas ferramentas inteligentes de BPM fornecem uma solução que resolve esse problema específico. Elas fornecem recursos para modelar os processos de negócio que trazem a regra de decisão de um repositório de pesquisa ou fazem referência a uma mensagem de entrada durante a execução da regra de decisão.

A abordagem aqui é ter um armazenamento de banco de dados das regras de negócios baseadas em parâmetros, as quais controlam o fluxo do processo de negócios de alto nível (consulte a Figura 6). No modelo gráfico do processo de negócios de alto nível, é feita uma simples chamada para um serviço sobre o armazenamento de regra, para avaliar os dados no processo de negócios e determinar a próxima etapa lógica. Essa abordagem reduz muito a complexidade do processo de negócios de alto nível, já que agora o modelo gráfico não tem os caminhos de fluxo, mas apenas o progresso linear do processo de negócios de alto nível de um estágio para outro. A solução, no entanto, ainda não é boa o suficiente para resolver o problema principal da capacidade de gerenciamento.

As desvantagens dessa solução são que o armazenamento de regras abstrai a lógica inteira do processo de negócios de alto nível, tornando o modelo menos transparente para os usuários corporativos. O armazenamento de regra e as possibilidades de interação sobre os dados não são padronizados e não garantem que todos os detalhes técnicos das condições seriam abstraídos do usuário. Ele não consegue aproveitar as ferramentas e os métodos padrão atuais disponíveis. Assim, permanece a questão de uma solução melhor.

Vamos dar uma olhada em outra solução. As PDR – Process Definition Rules (Regras de Definição de Processo) são gerenciadas externamente do modelo gráfico, semelhante à abordagem da condição isolada até o ponto de manter as regras fora do processo de negócios de alto nível. A diferença está na forma como essas regras externas são gerenciadas e na forma como o processo de negócios de alto nível é controlado. A solução de matriz PDR promove o mecanismo de regra como a camada de controle e muda o mecanismo de processo para uma camada de execução, tornando o fluxo do processo de negócios de alto nível mais visível nas regras, e não em um modelo gráfico.

“SÃO CRIADOS PROCESSOS DE NEGÓCIO DE ALTO NÍVEL COMBINANDO VÁRIOS PROCESSOS DE NEGÓCIO DE BAIXO NÍVEL, FORMANDO AS DUAS CAMADAS DO BPM”

Com essa solução, ainda há a falta de uma abordagem padrão para modelar e manter o repositório PDR. O ponto crítico aqui é que as abstrações técnicas fornecidas por uma ferramenta gráfica de modelagem do processo de negócios devem estar intactas na ferramenta para modelar e manter o repositório PDR. Já que o repositório PDR é um repositório de regra, precisamos olhar as ferramentas padronizadas disponíveis no mundo das regras de negócios. A que se mostra proeminente entre elas, que surge como uma boa opção para fornecer uma ferramenta leve, simples e clara, é a tabela de decisão.

Figura 6 Uma condição isolada

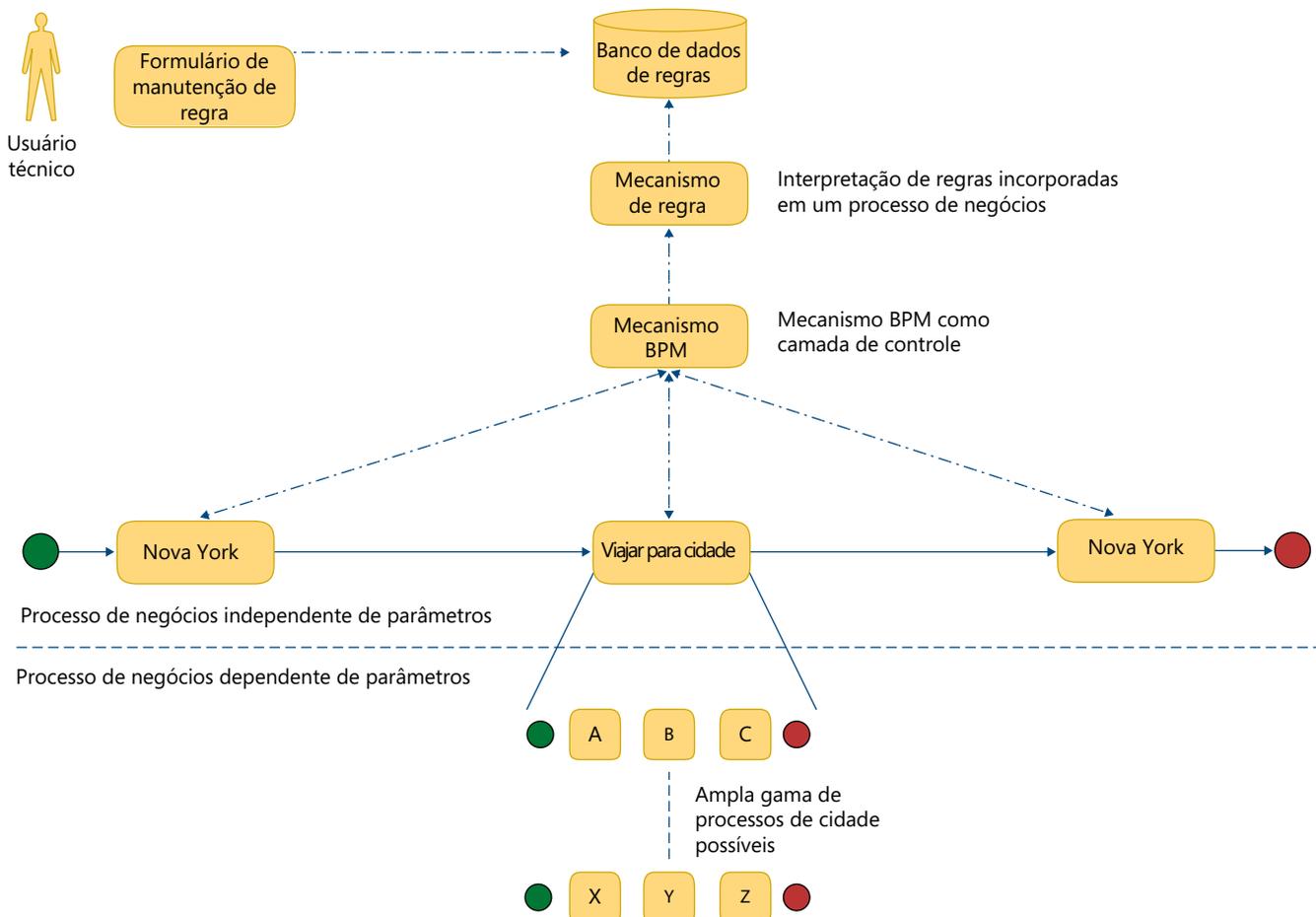
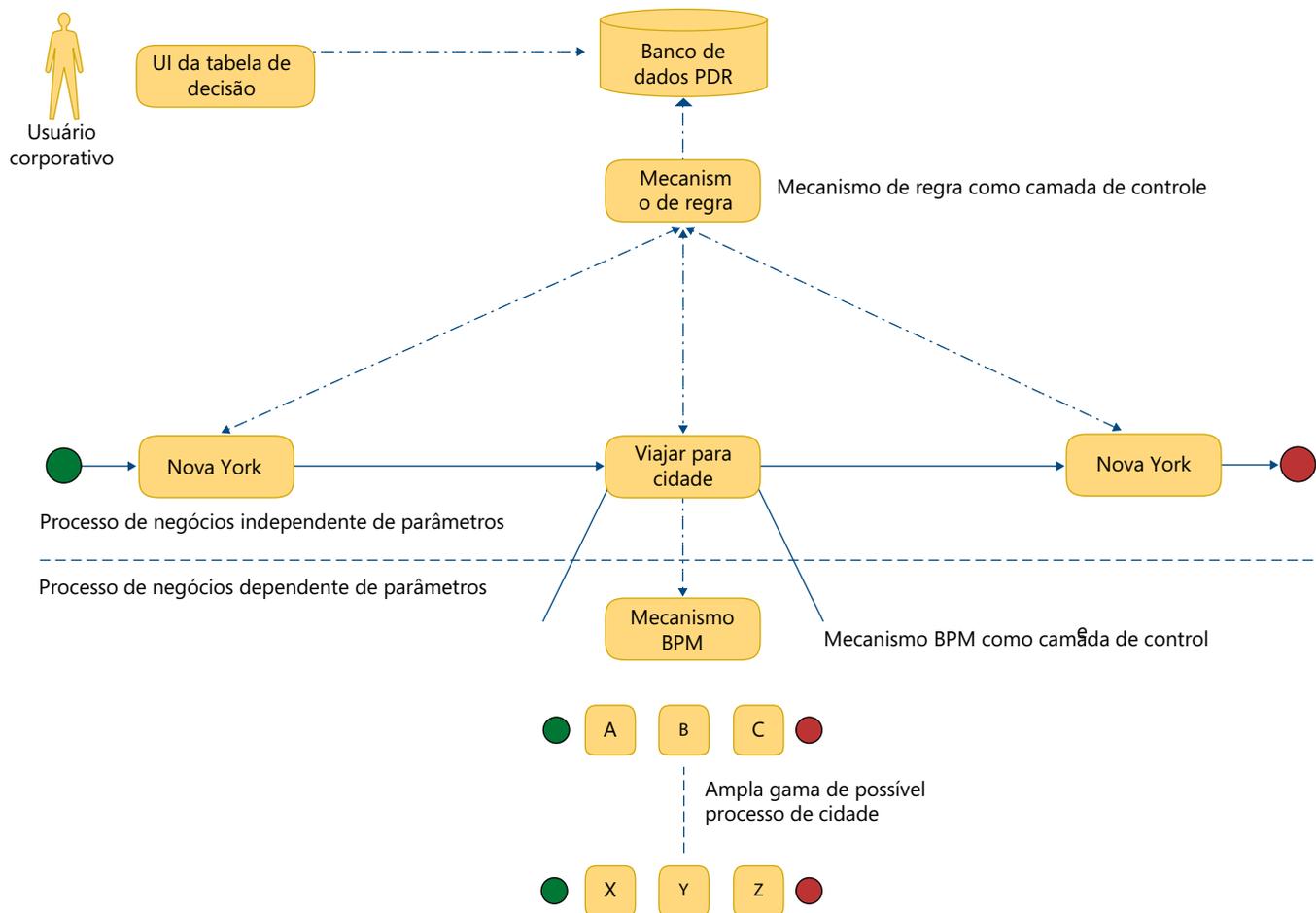


Figura 7 Uma solução de matriz PDR



A *tabela de decisão* é uma ferramenta padrão para modelar e gerenciar os repositórios de regra e é uma excelente abstração para os usuários corporativos, algo simples de usar, pois é baseada em texto. As tabelas de decisão baseadas em texto também são boas candidatas para pesquisar o repositório de regras e permitir que o usuário altere rapidamente as regras relevantes de definição de processo.

Usando essa ferramenta, podemos criar uma solução que tenha um processo gráfico de negócios de alto nível que seja simplificado ao se isolar as regras e fornecer uma camada de serviço na parte superior do repositório de regras PDR (consulte as Figuras 7 e 8). O repositório de regras PDR é então modelado e gerenciado por meio das tabelas de decisão. Chamamos essa solução de matriz PDR, e o conceito de processo global e de processo de cidade pode ser mostrado como duas camadas. A primeira camada é o processo de negócios de alto nível válido para toda a empresa, e a segunda camada é um processo de negócios reutilizado e de baixo nível.

O fim do jogo

Trabalhamos em um problema muito prático aqui, com o BPM nas empresas: a complexidade em definir e gerenciar processos de negócio de alto nível. São criados processos de negócio de alto nível combinando vários processos de negócio de baixo nível, formando as duas camadas do BPM. Os processos de negócio de alto nível são marcados pela necessidade

de regras baseadas em parâmetros para se escolher dinamicamente um processo de negócios de baixo nível.

Essas regras baseadas em parâmetros, quando empregadas para gerenciar um processo de negócios de alto nível, aumentam a complexidade da cópia do modelo gráfico com cada novo valor, para cada parâmetro. Esse aumento na complexidade torna o modelo gráfico de um processo de negócios de alto nível impossível de ser gerenciado e pouco prático. A solução para esse problema é usar uma abordagem baseada em regras para definir e gerenciar processos de negócios de alto nível.

Você pode definir e gerenciar essa abordagem usando uma matriz PDR baseada em tabela de decisão. As tabelas de decisão são ferramentas leves, não gráficas e amigáveis ao usuário para criar e gerenciar regras. A matriz PDR abstrai a complexidade de gerenciar um modelo gráfico de processo de negócios de alto nível. Portanto, utilize a matriz PDR para gerenciar o BPM de duas camadas e coloque o usuário corporativo de volta no assento de comando de sua empresa. •

Sobre o autor

Vignesh Swaminathan é gerente de produtos da Cordys R&D India (www.cordys.com), que fornece uma suíte de última geração de plataforma de aplicativos que vai além do EAI e do BPM básicos para dar conta de muitos aspectos práticos da empresa. Nos últimos cinco anos, Vignesh vem trabalhando com orquestração do processo de negócios, regras de negócios, transformação de dados e outras tecnologias relacionadas, tendo se especializado em dados, processos e integração humana. Entre em contato com Vignesh nos endereços vswamina@cordys.com e vigneshs@hotmail.com.

Fonte
CBS.com

www.cbs.com/primetime/amazing_race5



Explore as arquiteturas do workflow humano

por Jesus Rodriguez e Javier Mariscal

Resumo

Os sistemas de workflow humano e alguns dos principais padrões de desenho das interações indivíduo – processos de negócio se dividem em dois componentes. O primeiro são os sistemas de workflow humano e as interações entre eles à medida que são implementados em plataformas de integração. O segundo são padrões de desenho das interações de workflow humano e a forma como são implementados usando as interações entre os sistemas de workflow humano. Esta análise observará de perto esses processos.

A total automatização dos processos de negócios é algo praticamente impossível de ser realizada sem se considerar a interação humana, um fator que está semanticamente vinculado a muitos aspectos da automatização e da integração de processos. A interação humana está presente em alguns dos processos de negócio mais comuns, como aprovações de pedidos e gerenciamento de recursos humanos, sendo que podem variar de um processo simples de atribuições de tarefas a notificações muito complexas do processo de negócios e re-atribuições de tarefas.

As ferramentas para workflow humano estão presentes em uma série de servidores de integração conhecidos, de diferentes fornecedores. Vamos analisar conceitualmente alguns dos principais componentes que estão presentes nessas plataformas de workflow humano e as interações entre elas. Também analisaremos alguns dos padrões mais comuns de projeto de workflow humano e a forma como eles podem ser implementados com esses componentes.

Os sistemas de workflow humano precisam fornecer suporte a comunicações entre pessoas e sistemas. Para conseguir isso, todo o sistema de workflow de um indivíduo precisa fornecer funções básicas como atribuição de tarefas, gerenciamento de identidade, notificações, controle e interoperação com sistemas BPM – Business Process Management (Gerenciamento de Processo de Negócio) (Consulte a Figura 1).

Quatro são os serviços principais da arquitetura de workflow humano: Um de gerenciamento de tarefa, um de controle, um de notificação e outro de identidade. Antes de explorarmos esses serviços em profundidade, é importante entender o papel das tarefas nos sistemas de workflow humano.

As tarefas são as principais unidades de comunicação entre os processos de negócio e as pessoas. Em geral, é atribuída uma tarefa a um usuário que precisa realizar alguma ação relacionada. Por exemplo, o supervisor talvez precise aprovar a solicitação de compra de alguns itens. Normalmente, o usuário precisa realizar uma série de tarefas que são agrupadas semanticamente. Ou seja, nosso supervisor talvez precise agrupar todos os itens a serem aprovados a cada dia em um grupo chamado “aprovações de hoje”. Esses grupos são chamados de listas de tarefas.

Estados de tarefa

Durante o tempo de vida de um sistema de workflow humano, as tarefas nas listas de tarefas estão constantemente mudando de um estado para outro. Por exemplo, a tarefa do nosso supervisor primeiro entra em um estado de pendência; quando o supervisor aceita a tarefa, o estado muda para *solicitada*. Por fim, quando o supervisor aprova a solicitação, a tarefa entra no seu estado final de *concluída*. Os estados são usados conceitualmente para descrever o ciclo de vida da tarefa. Entre os estados mais comuns estão: pendente, quando a tarefa foi criada; solicitada, quando o usuário solicitou a tarefa e recebeu seus dados de entrada; *concluída*, quando o usuário concluiu a tarefa e forneceu os dados de saída; e com *falha*, quando o usuário concluiu a tarefa e forneceu uma mensagem com problemas.

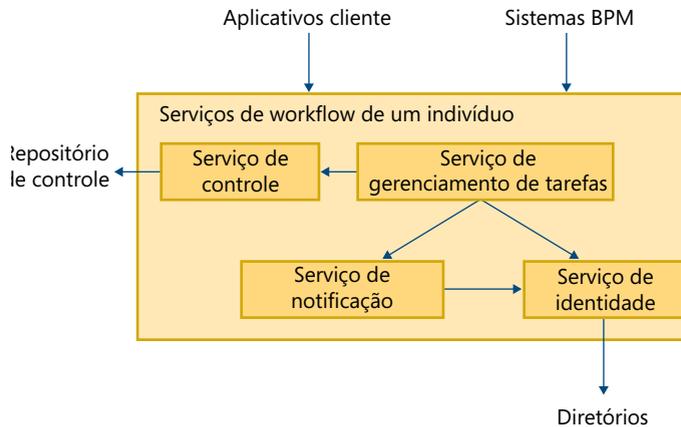
As tarefas estão normalmente associadas a cronogramas de tempo: expiração, escala, delegação e renovação. Em nosso exemplo, a tarefa de aprovação pode *expirar* caso o supervisor não tome ações apropriadas sobre ela em um período de tempo especificado, sendo que ela poderá ser escalada depois para outro roteiro de ação ou atribuição. Além disso, o supervisor pode decidir delegar a tarefa para outra pessoa (um gerente, por exemplo) para agir no seu lugar. O gerente também pode decidir que precisa de outro gerente para obter um maior entendimento da tarefa. Se esse segundo gerente não trabalhar na tarefa em um determinado cronograma de tempo, a tarefa será renovada por outro período de tempo.

Em outros cenários, as tarefas estão semanticamente relacionadas umas às outras. A questão “qual é a próxima tarefa?” nem sempre terá uma resposta simples. Em alguns casos, essa resposta precisa ser determinada em tempo de execução. As tarefas podem ser agrupadas sequencialmente em uma instância do processo de negócios, para que o usuário conheça a(s) próxima(s) tarefa(s) a ser (em) realizada(s) após a conclusão da tarefa atual.

Em nosso exemplo, vinte solicitações deveriam ser aprovadas para o supervisor após a conclusão do processo de negócios. Sempre que o supervisor concluir uma aprovação, o mecanismo deve conseguir identificar a próxima tarefa. As cadeias de tarefas representam uma abordagem orientada a metadados para descrever a relação entre um conjunto de tarefas no escopo de um processo de negócios. Essas cadeias podem agrupar as tarefas semanticamente para ajudar os usuários a obter funcionalidades como execução de seqüências e gerenciamento de falhas.

As tarefas que dependem de um processo de negócios específico são conhecidas como *tarefas acopladas*. As tarefas que são totalmente independentes de um processo de negócios em particular são tarefas autônomas. As tarefas acopladas geralmente têm acesso aos dados relacionados a um processo de negócios e são armazenadas em artefatos nativos do processo de negócios, como variáveis ou mensagens. Por outro lado, as tarefas autônomas interagem com os processos de negócio por meio de uma interface bem definida sem nenhuma dependência dos dados do processo em si.

Figura 1 Os principais componentes da arquitetura de workflow humano



Serviços de workflow

O componente de arquitetura que lida com tarefas relacionadas é o serviço de gerenciamento de tarefas. Em um cenário típico, o *serviço de gerenciamento de tarefa* recebe uma solicitação para criar uma tarefa, interage com o serviço de identidade para selecionar todas as pessoas qualificadas para ela, a adiciona nas listas de trabalho associadas aos usuários selecionados e atribui cronogramas e políticas específicas. No fim, um usuário decide trabalhar com a tarefa solicitando-a. O usuário pode então trabalhar com a tarefa ou solicitar informações adicionais.

Um aspecto importante dos sistemas de workflow humano é a capacidade de identificar um grupo de usuários que têm permissão de executar uma tarefa. Esse processo de identificação de usuário pode se basear em interações com as plataformas de gerenciamento de identidade. Em nosso exemplo, os serviços de workflow de um indivíduo precisam identificar quais usuários tem direitos para aprovar a solicitação – nesse caso, os supervisores. Para realizar essa identificação, uma plataforma de workflow humano precisa entender o conceito “supervisor” em relação a um grupo de usuários e papéis tradicionalmente armazenados no serviço de diretório de usuários.

Vários tipos de relações podem ser estabelecidos entre pessoas e processos. Um dos mais comuns é como as pessoas interagem com os processos (funções do indivíduo). As pessoas em uma organização podem ser agrupadas em funções que estão semanticamente relacionadas a alguma atividade de negócios, como administrador de processos ou proprietário de tarefas. Outra relação comum é como os processos identificam as pessoas com quem se pode interagir (links de pessoas e consultas). Em um processo de negócios, certos grupos de usuários são importantes do ponto de vista do negócio. Os links de pessoas são usados para representar os diferentes grupos de pessoas que participam da execução do processo. Uma consulta ao serviço de diretório da organização é utilizada para determinar as pessoas associadas a um link de pessoa e está vinculada ao link de pessoa. No exemplo, a função genérica, gerente financeiro, poderia ser qualificada pelo link de pessoa “supervisor”, que está vinculado à consulta “selecionar chefe de departamento, onde o nome do departamento é finanças”.

O serviço de identidade é responsável pelas funcionalidades relacionadas ao usuário, como autenticação, autorização e identificação de pessoas. As informações do usuário são frequentemente armazenadas em serviços de diretórios (por exemplo, Active Directory, diretório LDAP ou banco de dados relacional). O serviço de identidade pode funcionar de forma independente do serviço de diretório. Com base no modelo “adaptador”, é possível extrair o acesso do diretório utilizando um provedor, que consegue executar as consultas para obter as informações do diretório. Essa abordagem extrai as funcionalidades do workflow humano do repositório de usuários.

Em nosso exemplo, vamos assumir que um gerente gostaria de remontar a execução do workflow de aprovação da solicitação para verificar se há deficiências. O serviço de controle examina as mudanças de estado

relacionadas às tarefas e às cadeias de tarefas. Esse serviço deve fornecer a base da funcionalidade necessária para reconstruir as mudanças no histórico de tarefas e realizar a análise dessas tarefas.

Também precisamos de um serviço que notifique o supervisor por e-mail quando for criada a tarefa de aprovação da solicitação. O serviço de notificação trata dos mecanismos de notificação para o usuário relacionado às mudanças de estado da tarefa.

Os quatro serviços aqui discutidos fornecem uma boa visão de algumas das funcionalidades mais comuns necessárias para os sistemas de workflow humano. A combinação desses serviços trata de alguns dos cenários mais comuns desse tipo de workflow. Agora vamos explorar alguns dos padrões comuns de workflow humano.

Padrões de atribuição de tarefas

Os processos orientados a fluxos de trabalho estão presentes no mercado há anos. O conhecimento adquirido serve como base para aperfeiçoamentos no desenvolvimento de sistemas de workflow. Os padrões abstraem os sistemas de workflow em níveis diferentes, como aprovação de tarefa, criação de tarefa e gerenciamento do estado de tarefa. Não pretendemos definir aqui um catálogo de padrões de workflow humano. Em vez disso, vamos analisar alguns padrões comuns em sistemas de workflow humano da perspectiva da arquitetura previamente definida.

Vamos começar examinando um *exemplo de workflow de um único usuário*. Pode-se atribuir uma tarefa a um usuário, e apenas um usuário pode trabalhar nela. Por exemplo, um funcionário, por meio do portal dos funcionários, envia uma solicitação de férias. O portal inicia um processo de negócios que inclui uma tarefa de usuário modelada com um workflow simples. A tarefa é atribuída ao gerente do funcionário. Quando o gerente aprova ou rejeita a solicitação de férias, o funcionário é notificado por e-mail da decisão tomada.

Para essa solução, combine os componentes do workflow humano por meio de quatro interações. A tarefa é configurada usando aplicativos cliente que interagem com o serviço de gerenciamento de tarefa; seu ciclo de vida ou os estados que estão configurados. Como parte do processo de negócios, a tarefa é atribuída a um grupo de usuários que usam o serviço de gerenciamento de tarefa. Um dos usuários solicita a tarefa, e o serviço aplica a política correta para impedir que outros usuários trabalhem na mesma tarefa. O processo de negócios usa o serviço para atualizar o status.

Agora vamos considerar um *exemplo de workflow sequencial*. O workflow sequencial representa um cenário no qual uma tarefa deve ser aprovada sequencialmente por um conjunto de usuários. Por exemplo, quando um sistema de aprovação de pedido de compra processa uma ordem de compra usando um processo de negócios, um funcionário que pertence ao grupo “Supervisor” inicialmente avalia a ordem de compra. Depois que o usuário inicial aprova a ordem de compra, o gerente daquele usuário a aprova. Depois de aprovada, ela é encaminhada para os departamentos de cobrança e envio. Essa solução tem a ver com interações. Interaja com o serviço de gerenciamento de tarefas para configurar as tarefas e definir as políticas apropriadas. Defina a sequência de usuários que devem trabalhar na tarefa. Inicie a tarefa interagindo com o serviço de gerenciamento de tarefas. O primeiro usuário solicitará a tarefa para começar a trabalhar nela, e, após a conclusão, o serviço de gerenciamento de tarefa encaminhará a tarefa para o próximo usuário no grupo.

Um *padrão de workflow paralelo* representa o cenário no qual uma tarefa deve ser aprovada por usuários diferentes, ao mesmo tempo. Cada aprovador pode adicionar comentários e anexos que são independentes dos outros. Por exemplo, um processo de contratação é usado para contratar novos funcionários. Cada entrevistador vota a favor de um candidato ou contra ele. Se 75 por cento dos votos são favoráveis, o candidato é contratado, caso contrário, o mesmo é rejeitado. O processo é modelado com o workflow paralelo, no qual cada entrevistador pode votar independentemente de outros entrevistadores. Para a implementação, essa solução tem cinco interações. Interaja com o serviço de gerenciamento de tarefas para configurar as tarefas

e definir as políticas apropriadas. Defina a sequência de usuários que devem trabalhar na tarefa. Inicie a tarefa interagindo com o serviço de gerenciamento de tarefas. O serviço de gerenciamento de tarefas encaminha a tarefa para todos os usuários. O serviço de gerenciamento de tarefas concluirá a tarefa apenas quando todos os usuários tiverem terminado de trabalhar nela.

Atribuição de tarefas usando políticas

Agora vamos dar uma olhada na utilização de políticas para atribuição de tarefas ao usuário seguindo regras específicas. Por exemplo, o procedimento de ponte de safena é alocado ao cirurgião que tem o menor número de operações alocadas a ele. A implementação dessa solução combina os componentes do workflow humano por meio de cinco interações. Interaja com o serviço de gerenciamento de tarefas para configurar as tarefas e definir as políticas apropriadas. Configure a política de atribuição como parte da definição de tarefa – por exemplo, selecione dentre os usuários disponíveis aquele que tem o menor número de tarefas atribuídas. Como parte do processo de negócios, a tarefa é atribuída a um grupo de usuários que usam o serviço de gerenciamento de tarefa. Inicie a tarefa interagindo com o serviço de gerenciamento de tarefas, que executará as políticas de tarefas em relação aos usuários em potencial que podem trabalhar na tarefa e, por sua vez, selecionará um usuário que corresponda aos critérios de política e atribuirá a tarefa a esse usuário.

Em um exemplo de *workflow com um único usuário* e escala, pode-se atribuir uma tarefa a vários usuários, mas apenas um usuário pode trabalhar nela. Se a tarefa expirar, o gerente do usuário precisará trabalhar nela. Por exemplo, o processo de solicitação do serviço de help-desk permite que os usuários arquivem os tickets de solicitação do serviço. Se a pessoa que receber o ticket não realizar nenhuma ação em um determinado período o ticket será enviado automaticamente para o gerente daquela pessoa. O ticket será escalado automaticamente três vezes se ninguém realizar nenhuma ação relacionada a ele em um período predefinido de tempo, até chegar ao CEO da empresa. Se o CEO não tomar nenhuma ação, o bilhete expira.

Na implementação dessa solução, combine os componentes do workflow humano por meio de cinco interações. A tarefa é configurada usando aplicativos cliente que interagem com o serviço de gerenciamento de tarefa; propriedades, como o ciclo de vida da tarefa ou o estado das tarefas, são configuradas nessa etapa. Como parte do processo de negócios, a tarefa é atribuída a um grupo de usuários que usam o serviço de gerenciamento de tarefa. Um usuário solicita a tarefa, e o serviço de gerenciamento de tarefa aplica a política correta para impedir que outros usuários trabalhem na mesma tarefa. Se ela expirar, o serviço de gerenciamento de tarefa aplica as políticas de escala para repassar a tarefa ao usuário certo. Se novamente expirar, o serviço de gerenciamento de tarefas a cancela.

Em um exemplo de workflow de usuário único com delegação, o usuário que solicitou trabalhar em uma tarefa pode atribuí-la a outro usuário. Por exemplo, antes de sair, o contador-chefe passou todas as suas tarefas não resolvidas para o contador assistente. Na implementação dessa solução, combine os componentes do workflow humano por meio de quatro interações. A tarefa é configurada usando aplicativos cliente que interagem com o serviço de gerenciamento de tarefa; propriedades, como o ciclo de vida da tarefa, política de escala ou o estado das tarefas, são configuradas. Como parte do processo de negócios, a tarefa é atribuída a um grupo de usuários que usam o serviço de gerenciamento de tarefa. Um usuário solicita a tarefa, e o serviço de gerenciamento de tarefa aplica a política correta para impedir que outros usuários trabalhem na mesma tarefa. Utilizando os aplicativos cliente apropriados, o usuário inspeciona a lista de pessoas possíveis que podem trabalhar na tarefa. O usuário atribui a tarefa a uma dessas pessoas, ajustando as propriedades da tarefa.

Para o exemplo do padrão “*encadeamento de tarefa*”, a tarefa precisa começar com base no status de conclusão de outra tarefa. Por exemplo, começando imediatamente o(s) próximo(s) item(ns) de trabalho em um processo de coordenação de um resgate de emergência quando o anterior tiver terminado. Para implementar essa solução, combine os componentes do workflow humano por meio de quatro interações. A tarefa é configurada usando aplicativos cliente que interagem com o serviço de gerenciamento de tarefa; propriedades, como o ciclo de vida ou o estado da tarefa, são configuradas. Os metadados da cadeia de tarefas são configurados, indicando que a segunda tarefa será disparada mediante a conclusão da primeira. A cadeia de tarefas é iniciada por meio do serviço de gerenciamento de tarefas, causando o início da primeira tarefa. Quando a primeira tarefa é concluída, o serviço de gerenciamento de tarefas usará os metadados na cadeia de tarefas para iniciar a segunda tarefa.

Integração de processos de negócio

É possível projetar as soluções do workflow humano usando uma combinação de serviços, como gerenciamento de tarefas, gerenciamento de identidade, controle, notificações e aplicativos cliente. Exploramos alguns dos componentes mais comuns presentes nas arquiteturas do workflow humano e sua implementação em uma série de interações. Algumas das interações mais complexas entre indivíduos e sistema podem ser modeladas com um conjunto de padrões de interação humana que, por sua vez, são implementados com os componentes do workflow humano. Esses componentes e padrões básicos da arquitetura desse tipo de workflow representam aspectos importantes nas opções disponíveis para integração de processos de negócio, e, quando implementados, criam uma poderosa solução de workflow humano.

Os conceitos e cenários aqui apresentados têm a intenção de ajudá-lo a entender melhor os componentes encontrados em plataformas típicas de workflow humano. Para obter mais informações sobre os conceitos de workflow humano, consulte a lista de recursos desse artigo. Além disso, consulte as arquiteturas do mecanismo de workflow de algum dos principais fornecedores de servidores de integração.

Gostaríamos de agradecer Kirsti Elliott, Ben Elliott e a equipe de estratégia de arquitetura da Microsoft, pelo feedback que nos deram e pelas correções feitas neste artigo. •

Sobre o autor

Jesus Rodriguez é arquiteto-chefe de produtos de software na Two Connect Inc. (www.twoconnect.com), empresa Microsoft Gold Partner com sede em Miami, na Flórida. Também é MVP de Microsoft BizTalk Server. A ampla experiência de Jesus com a integração de processos de negócios e fluxos de trabalho humano veio de várias implementações de sistemas desacoplados e fundamentados nos princípios da SOA. Também contribui ativamente para as comunidades .NET e J2EE, focando-se nos aspectos de interoperabilidade entre essas duas plataformas. Suas contribuições incluem vários artigos para diferentes publicações, como MSDN, sessões em conferências Microsoft, como a Teched, e Web casts sobre tecnologias Microsoft. Jesus tem um prolífico blog e escreve sobre todos os assuntos relacionados à integração, sendo um verdadeiro apaixonado pela tecnologia. Entre em contato com Jesus pelo endereço jrodriguez@twoconnect.com ou pelo blog <http://weblogs.asp.net/gsusx>.

Javier Mariscal é presidente da Two Connect Inc. Javier dedicou boa parte dos seus 16 anos de carreira profissional a projetos e implantação de dados e soluções de integração de aplicativos, particularmente aquelas que envolvem a fusão de ambientes mainframe e AS/400 com plataforma WinTel. Tem uma verdadeira paixão por soluções de workflow e automatização de processos de negócio, as quais implementam a abordagem do SODA (Desenvolvimento Orientado a Serviços de Aplicativos), para o desenvolvimento e a manutenção de aplicativos distribuídos globalmente. Ele se mantém bastante ocupado em soluções como essas para a Fortune 1000 e grandes organizações de midmarket do mundo todo. Entre em contato com Javier no endereço Javier@twoconnect.com.

Recursos

A Workflow Management Coalition www.wfmc.org
OMG Business Process Initiative www.bpmi.org



Workflow na integração de aplicativos

por Kevin Francis

Resumo

Um dos maiores desafios que o arquiteto enfrenta hoje é a integração entre aplicativos. Iremos analisar um modelo de integração de aplicativos que se coloca além das abordagens comuns de integração isolada e em direção a uma estrutura coesa. Os requisitos para uma integração bem-sucedida serão descritos junto com a apresentação de uma abordagem arquitetônica para atender esses requisitos utilizando ferramentas como tecnologias de workflow.

A integração de aplicativos passou a ser mais comum, e a crescente disponibilidade de ferramentas e padrões (como os padrões WS-I para Web Services) e de SOA – Service Oriented Architecture (Arquitetura Orientada a Serviços) parece ser uma promessa de integração mais fácil. Existem muitos artigos que estimulam a simplicidade de vincular aplicativos utilizando Web Services ou mesmo SOA. Hoje, duas abordagens de integração de aplicativos são utilizadas comumente: integração ponto a ponto e integração por barramento de serviços (consulte a Figura 1).

No cenário ponto a ponto, links diretos são criados entre aplicativos realizando chamadas a uma API (Interface de Programas Aplicativos), FTP (Protocolo de Transferência de Arquivo) ou interfaces em lote. A transformação (conversão) de dados pode ocorrer à medida que os dados são transferidos no link. Geralmente as interfaces ponto a ponto são implementadas sem o uso de um produto de integração, com a conversão dos dados ocorrendo com a utilização de código no ponto de integração em uma ou nas duas extremidades.

A integração do barramento de serviço utiliza uma solução de tecnologia para fornecer um barramento, no qual os aplicativos podem

Figura 1 Integração ponto a ponto e integração de barramento de serviço

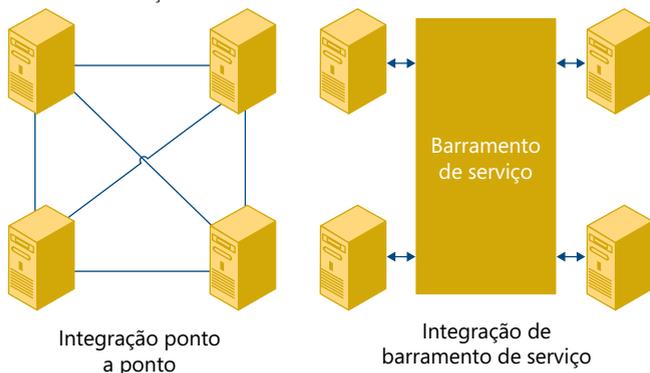
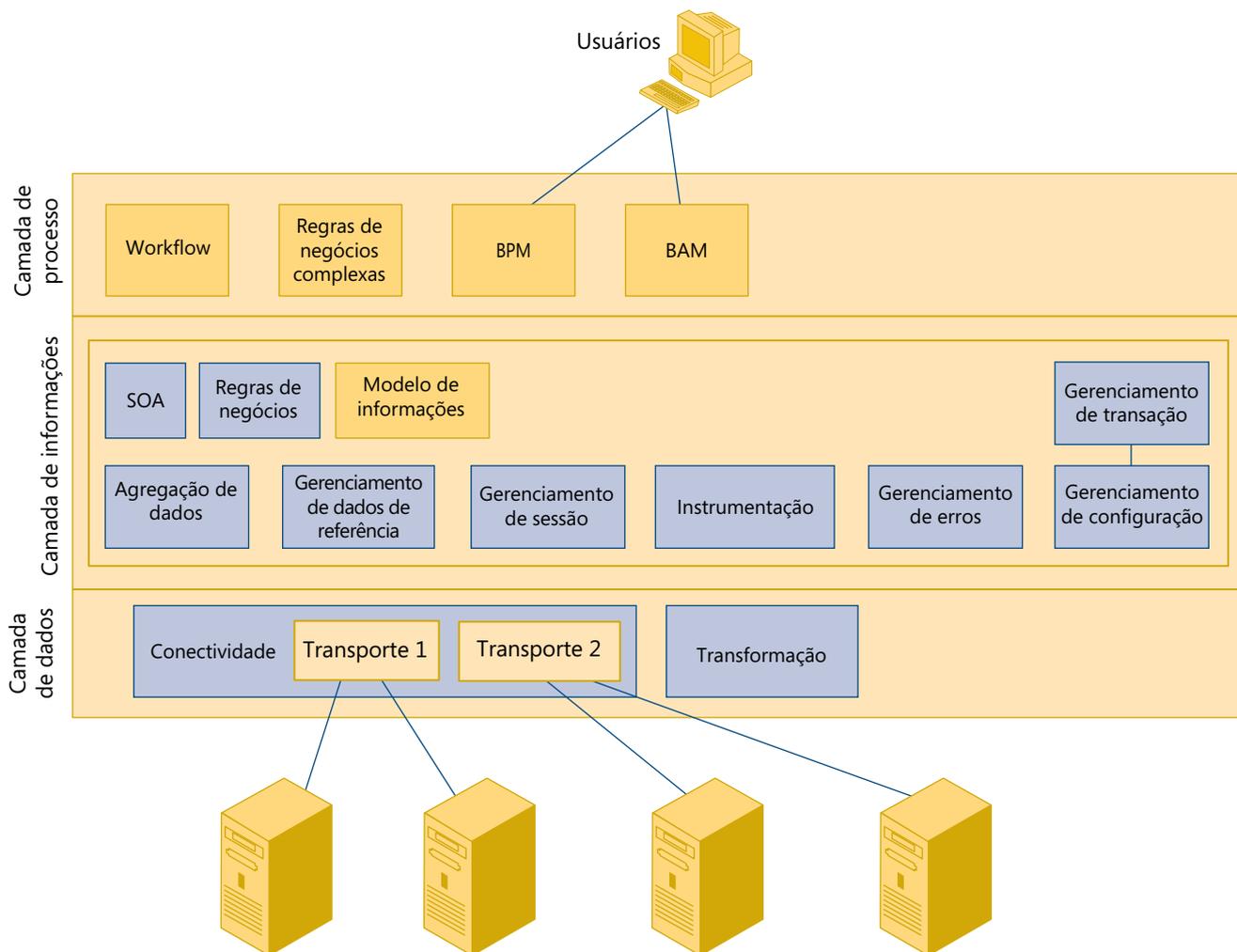


Tabela 1 Os requisitos de uma camada de integração

| Camada | Requisito | Descrição |
|--|---|--|
| Dados | Conectividade | A conectividade básica com a qual os aplicativos conseguem se comunicar uns com os outros |
| | Transformação | Conversão do formato de dados etc. entre aplicativos |
| Informações | Agregação de dados | Visão agregada dos dados em vários sistemas |
| | Regras de negócios | Desenvolvimento de regras de negócios em vários sistemas |
| | Gerenciamento de transação | Capacidade de realizar transações ACID em vários sistemas |
| | Modelo de informações | Um modelo de dados coeso em todos os sistemas no qual é obtido um entendimento comum de entidades e estruturas de dados |
| Processo | Gerenciamento de dados de referência | Gerenciamento em um único local de dados de referência de vários sistemas usados comumente |
| | Gerenciamento de sessão | Gerenciamento de informações da sessão em interações e sistemas |
| | Instrumentação | Ponto comum para o registro de informações operacionais |
| | Gerenciamento de erros | Abordagem consistente do gerenciamento de erros a partir de um conjunto de regras único |
| | Gerenciamento de configuração | Capacidade de configurar a operação do sistema inteiro em tempo de execução, configurar sua comunicação com os diversos sistemas que compõem o ambiente e implementar novas versões dos diversos componentes |
| | Workflow | Processos de workflow que cruzam vários sistemas |
| | Regras de negócios complexas | Regras de negócios reutilizáveis e compartilhadas que cruzam vários sistemas |
| Modelagem do processo de negócios | Modelagem dos processos de negócios em sistemas para otimização e integração | |
| Monitoramento da atividade de negócios | Monitoramento da velocidade e da eficiência dos processos completos de negócios para otimização e rastreamento de problemas | |

Figura 2 Três camadas de integração



colocar mensagens onde o próprio barramento gerencie o roteamento das mensagens entre os aplicativos. Ele geralmente gerenciará também a transformação dos formatos das mensagens entre os aplicativos.

À medida que as organizações empenham-se para trazer uma variedade maior de serviços on-line, procuram integrar as linhas de produtos e esforçam para agilizar a experiência dos call centers, uma solução de integração requer mais sistemas de retaguarda (backend). Não é nada extraordinário, por exemplo, os operadores de call center alternarem entre mais de 10 aplicativos para atender ligações dos clientes. A substituição desse tipo de cenário, em que o usuário está efetivamente integrando os aplicativos ao copiar e colar ou digitar novamente informações nas diferentes telas para concluir uma transação, é um impulsionador comum desta integração.

Base da arquitetura corporativa

Existem diversos métodos que podem ser usados para integrar aplicativos por trás de um site da Web ou aplicativo de call center, e a solução mais comum é construir uma série de interfaces ponto a ponto ou de barramento de serviço na primeira instância.

O que acontece, porém, quando uma solução do call center e um aplicativo da Web precisam acessar as mesmas informações? O cenário ideal é reutilizar as interfaces, certo? Bem, na maioria dos casos em que as equipes de desenvolvimento (e arquitetos) estão desconectadas essa não é a abordagem comum, devido às complexidades do uso de uma interface

existente com seu próprio conjunto complexo de códigos e permitir que ela seja chamada por alguma outra aplicação. Esse problema cresce de forma exponencial com o número de interfaces que precisam existir entre os sistemas e é proporcional ao tamanho da organização; as experiências das organizações maiores são obviamente piores do que as das menores.

A falha em implementar uma solução completa e bem arquitetada resulta em código duplicado em toda a empresa, abordagens de arquitetura

“A FALHA EM IMPLEMENTAR UMA SOLUÇÃO COMPLETA E BEM ARQUITETADA RESULTA EM CÓDIGO DUPLICADO EM TODA A EMPRESA”

inconsistentes em cada sistema e incapacidade de responder às necessidades dos negócios de forma adequada. Esses problemas são conseqüências de uma abordagem de arquitetura de solução centrada no projeto.

SOA é proclamada atualmente como a solução para os problemas de integração entre aplicativos, mas diversos recursos adicionais são necessários para uma solução de integração realmente eficiente. Tabela 1 traz uma lista desses recursos agrupados em três camadas de integração:

- *Integração de dados* – Sendo a camada mais básica, a integração de dados geralmente é alcançada mesmo nos cenários de integração

Tabela 2 Requisitos técnicos da camada de dados

| Requisito | Descrição |
|---------------|---|
| Conectividade | Conectividade é a capacidade de transferir informações utilizando uma variedade de protocolos e métodos e inclui o fornecimento de uma interface de Web Services. Também inclui protocolos específicos exigidos por cenários específicos, que irão mudar de uma organização para outra. Por exemplo, muitas organizações que conservam mainframes para sistemas de negócios centrais precisariam do IBM WebSphere MQ e/ou de conectividade SNA usando o Microsoft HIS (Host Integration Server). Outros sistemas podem precisar de interfaces COM+ ou mesmo HTTP, raw socket ou FTP. Todos os protocolos devem ser executados por meio de uma interface comum, com implementações específicas e conectáveis para os cenários específicos. Todos os requisitos técnicos específicos de determinados protocolos ou métodos devem ser manipulados pela implementação e não se deve exigir que sistemas externos contenham qualquer lógica para manipular casos específicos. |
| Transformação | Oransformação é a transformação de dados baseada em regra de uma estrutura para outra. Aqui também o mecanismo de transformação deve estar contido na camada de integração. |

Tabela 3 Requisitos técnicos da camada de informações

| Requisito | Descrição |
|--------------------------------------|---|
| Agregação de dados | O fornecimento de serviços que envolvem várias chamadas traz consigo a agregação de dados. É importante considerar a agregação de dados como um requisito separado, pois ela exige bastante cuidado na modelagem dos dados e o uso de estruturas de dados e regras de negócio bem projetados para permitir que os dados sejam agregados e não simplesmente acumulados. |
| Regras de negócios | É necessário ser cuidadoso no desenvolvimento de regras de negócios em um cenário de integração para garantir que somente as que fazem parte de cada aplicativo residam do aplicativo e que as que estão relacionadas à integração do aplicativo sejam encapsuladas na camada de integração. Essa abordagem oferece a maior oportunidade de reutilização, simplifica a manutenção dos aplicativos e fornece um ponto único e descentralizado para desenvolvimento e execução das regras. |
| Gerenciamento de transação | O gerenciamento de transação é um requisito complexo e necessário para a integração de aplicativos. É necessário porque existe a necessidade de confirmar ou reverter transações em todos os aplicativos que estão sendo acessados, orientado a partir do aplicativo inicial, o que é crítico. Isso é complexo, no entanto, porque a reversão em muitos sistemas pode envolver o fornecimento de entradas de reversão ou algum outro método baseado em código, que é um exemplo ideal da necessidade de centralizar essa lógica complexa. |
| Gerenciamento de dados de referência | Os dados de referência são usados comumente por interfaces do usuário de aplicativos para fornecer listas de opções ou para validação de dados. É normal os dados de referência serem comuns nos aplicativos (como listas de países, códigos postais, produtos, locais dos escritórios etc.). Acessar dados de referência de um local compartilhado na camada de integração fornece melhor desempenho, resultados consistentes e menos esforço de desenvolvimento. |
| Gerenciamento de sessão | O gerenciamento de sessão é usado para assegurar que todos os sistemas no cenário de integração tenham um entendimento atual de quem está acessando os dados e para garantir que as ações atuais estejam sincronizada nos sistemas. Um uso completo do gerenciamento de sessão pode ser utilizado para permitir que uma sessão seja capturada de um ponto, como um cliente acessando um portal da Internet, e reutilizada em outro, comopara permitir que um consultor repare e conclua o pedido internamente. |
| Instrumentação | De forma semelhante ao gerenciamento de dados de referência, a centralização de instrumentação (logging) em uma camada de integração é lógica já que ela é o ponto a partir do qual é feita a maioria das chamadas com outros sistemas, além de permitir também que o código de instrumentação exista em um local único. |
| Gerenciamento de erros | O gerenciamento de erros é semelhante ao gerenciamento de dados de referência já que o uso de uma camada de integração para gerenciamento de erros permite que o código de processamento de erros seja compartilhado e que um único conjunto de respostas a erros possa ser usado. |
| Gerenciamento de configuração | O gerenciamento de configuração do tipo de ambiente aqui escrito é uma questão complexa e merece seu próprio conteúdo. O gerenciamento de configuração, no entanto, deve fornecer um ponto central onde os endereços possam ser alterados, o acesso aos sistemas possa ser controlado (de preferência com funcionalidade cuidadosamente desativada) e onde as definições da configuração geral possam ser armazenadas e carregadas uma vez por um gerenciador de configuração compartilhado. |

mais básicos. Nessa camada os dados são movimentados entre os aplicativos, ocorrendo transformação para permitir que os dados sejam convertidos entre os aplicativos.

- **Integração de informações** – Nessa segunda camada, dados e chamadas para aplicativos são agregados para permitir que chamadas únicas acessem vários aplicativos, com as regras básicas de negócio visando permitir que chamadas únicas vinculem aplicativos. O uso dessas técnicas fornece agregação de serviços e atende os requisitos mínimos para alcançar uma implementação da SOA.
- **Integração de processos** – A terceira camada de integração desenvolve na integração de dados. Ao agregar e integrar os processos, os dados envolvidos na execução de um processo de negócios opera através dos limites dos aplicativos.

À medida que nos deslocamos por essas três camadas de integração, o foco muda de tecnologia para negócios. O resultado final, no entanto, permite uma maior capacidade de agregar valor aos negócios, com mais rapidez.

Muitas visualizações da SOA fornecem um modelo de conectividade comum e supõem fornecer transformação, mas na realidade, somente fornecem a integração de dados. Os desafios enfrentados por um arquiteto ao integrar aplicativos de maneira eficiente e reproduzível são muito mais complexos, particularmente quando o cenário de integração inclui aplicativos que já existem. Por que tudo se torna importante? Quando se constrói a primeira interface entre dois aplicativos, a simples integração de dados pode ser suficiente. Mas à medida que se constroem mais interfaces, o design torna-se mais importante. A falha em projetar e gerenciar um cenário de integração completo resulta em um aumento exponencial no custo de cada interface. Inversamente, à medida que cresce a funcionalidade na camada de integração, o código necessário em cada camada pode ser reduzido.

Componentes da camada de integração

Os componentes de uma camada de integração podem ser sub-divididos em três camadas: dados, informações e processo (consulte a Tabela 1).

Tabela 4 Requisitos de negócios para a camada de informações

| Requisito | Descrição |
|-----------------------|---|
| Modelo de informações | <p>A medida que cresce o número de aplicativos vinculados à camada de integração, também cresce a quantidade de dados disponíveis. Essa situação fornece a capacidade de começar a construir um modelo de informações nos sistemas que estão no escopo, o que pode ser o mais próximo de construir um modelo de informações para uma organização que a maioria seja capaz de alcançar.</p> <p>Quando implementado corretamente, o modelo de informações que se posiciona na camada de integração pode fornecer um ponto de verdade único para principais entidades de dados. Um exemplo de origem única da verdade é um entendimento de um cliente que cobre todos os sistemas com endereços, produtos, histórico do pedido e histórico do suporte.</p> <p>Dois métodos comuns para criar um ponto único de verdade são utilizar um sistema principal e tentar sincronizar em outros sistemas ou utilizar um banco de dados que esteja atualizado com alterações de vários sistemas.</p> <p>O uso de um modelo de informações compartilhado na camada de integração é particularmente poderoso, pois o resultado é um ponto de verdade único e compartilhado que fica disponível por meio da combinação de dados de várias origens, sem causar impacto nos dados ou no código dessas origens. Também, como não é exigida a sincronização de dados, os dados são muito provavelmente atuais.</p> |

Tabela 5 Requisitos de negócios para a camada de processo

| Requisito | Descrição |
|--|---|
| Workflow | Os sistemas de fluxo de dados podem ser implementados como orientados a usuários (humanos) ou orientados a sistemas. Ambos podem ser utilizados em uma camada de integração, embora seja importante separar os dois. O workflow orientado a sistemas inclui a execução das etapas envolvidas na interação com os diversos sistemas no ambiente. |
| Regras de negócios complexas | As regras de negócios complexas fornecem um valor adicional quando a camada de integração é concluída e quando se torna um lugar em que as regras de negócios começam a residir por seu próprio direito. Devido à função central que uma camada de integração começa a fornecer, ela pode tornar-se o repositório das regras de negócios que fornecem funcionalidade nova ou ampliada além da integração de aplicativos, que pode ser novos produtos, combinação de produtos, descontos adicionais, melhor verificação de crédito etc. |
| Modelagem do processo de negócios | Os sistemas de workflow também podem ser usados para a execução de workflows humanos, e essa razão é predominante para a sua existência. O fornecimento de uma camada de integração oferece o host ideal para um ambiente de modelagem do processo de negócios existir, permitindo que os fluxos de trabalho humano sejam automatizados. O uso de ferramentas de workflow permite que os processos sejam documentados e executados em formato gráfico. O uso da camada de integração por trás das ferramentas de workflow permite acesso a vários aplicativos, por meio da ferramenta de workflow, de uma forma mais fácil e rápida, particularmente quando os recursos descritos aqui, como o modelo de informações e suporte de transação, são implementados. Assim, a modelagem do processo de negócios é permitida para facilitar maiores aprimoramentos do que seria possível em outro modo. |
| BAM (Monitoramento das atividades de negócios) | BAM é o rastreamento de cada transação no sistema fim-a-fim para identificar bloqueios, desempenho lento, problemas com transações específicas e oportunidades de desempenho global. Baseados nos recursos de instrumentação fornecidos pela camada de informações, os recursos fornecidos pelo BAM completo são consideráveis. |

Além disso, podem ser separados em componentes que compõem uma *framework* técnica; eles aparecem em azul na Figura 2. A *framework* técnica fornece os requisitos de interligação e engenharia, já a *de negócios* fornece ativadores de negócios, mostrados em dourado na Figura 2. Vamos dar uma olhada mais de perto nos componentes que compõem as três camadas de uma camada de integração.

Camada de dados. Os dois componentes da camada de dados básica são conectividade e transformação, que formam a base da *framework* técnica (consulte a Tabela 2). Interfaces únicas ponto a ponto usando Web Services são utilizadas com frequência como exemplo de SOA, mas muitos princípios da SOA referem-se ao fornecimento de serviços que encapsulem várias chamadas entre sistemas. No entanto, como mencionado anteriormente, os Web Services existem na camada de dados, o que não possibilita esses princípios. Nesse contexto, os Web Services possibilitam a conectividade, mas precisamos explorar a camada de informações para obtermos integração relacionada a serviços.

Camada de informações. A camada de informações é formada por componentes que se desenvolvem na *framework* técnica (consulte a Tabela 3). A *framework* de negócios é construída sobre a base do modelo de informações (consulte a Tabela 4).

Camada de processo. À medida que o foco se desloca de requisitos técnicos para valor de negócio, o foco da funcionalidade que seria transmitido por meio da camada de processo muda para fornecer o valor real do conceito de camada de integração – processos de negócios rapidamente configuráveis (consulte a Tabela 5).

Ao reunir tudo isso, há uma série de maneiras de implementar uma camada de integração: desenvolvimento personalizado; uso de *frameworks*,

kits de ferramentas, código livre ou componentes fornecidos pelo sistema operacional; e uso de um pacote de integração mais completo. O desenvolvimento personalizado poderia ser aplicado de forma completa à camada de integração, mas não é recomendável devido à disponibilidade de kits de ferramentas e *frameworks*.

“SOA É PROCLAMADA COM FREQUÊNCIA HOJE COMO A SOLUÇÃO PARA OS PROBLEMAS DE INTEGRAÇÃO ENTRE APLICATIVOS, MAS DIVERSOS RECURSOS ADICIONAIS SÃO NECESSÁRIOS PARA UMA SOLUÇÃO DE INTEGRAÇÃO REALMENTE EFICIENTE”

Algumas *frameworks* disponíveis permitem integrar sistemas de forma rápida na camada de interface do usuário, como o Customer Care Framework da Microsoft. Esse tipo de solução de integração fornece uma solução excelente para a integração rápida de sistemas de retaguarda (backend) para obter uma interface do usuário consolidada, mas é importante entender os méritos relativos desse tipo de solução. As *frameworks* que permitem a integração de sistemas na camada de interface do usuário fornecem soluções rápidas para determinados problemas encontrados pelas empresas, como melhorar o tempo de atendimento para os operadores do call center. Uma camada de integração pode fornecer outros benefícios, a um custo inicial extra de implementação.

Estendendo o cenário de integração

Uma camada de integração permite que as funções principais sejam hospedadas centralmente e compartilhadas entre aplicativos, permitindo mais do que a integração das interfaces dos usuários. A variedade de dados, informações e funcionalidades de integração de processos pode ser usada nos aplicativos para diminuir o custo de cada desenvolvimento ou integração. A experiência mostra que o uso de uma camada de integração, como descrito aqui, produz uma economia crescente nos custos em uma série de aplicativos ao longo do tempo, com a possibilidade de criar novos aplicativos além do cenário de integração original.

Por exemplo, embora sistemas de faturamento, de atendimento, de gerenciamento de clientes e ERP possam estar integrados por trás de uma interface do usuário CRM na primeira instância, o fato de esses sistemas já estarem vinculados fornece soluções simples para a ajuda automática na Web e se estende para pedidos on-line e um canal B2B. Caso seja comprado um novo aplicativo, como um aplicativo de cadeia de suprimentos, a

“AS FRAMEWORKS QUE PERMITEM A INTEGRAÇÃO DE SISTEMAS NA CAMADA DA INTERFACE DO USUÁRIO FORNECEM SOLUÇÕES RÁPIDAS PARA DETERMINADOS PROBLEMAS ENCONTRADOS PELAS EMPRESAS”

integração dele no ambiente ainda não é um processo trivial. No entanto, a integração se torna muito mais fácil do que seria sem uma camada de integração porque a relação entre os dados pode ser conseguida mais facilmente fazendo a junção de um único conjunto de interfaces conhecidas na camada de integração e não em cada aplicativo. Além disso, o processo de integração pode ser melhorado pela customização dos roteiros de integração no motor de workflow, da mesma forma que a integração de monitoria de atividade de negócios (BAM), a agregação de dados e a modificações no modelo de informações etc. são obtidas com mais facilidade, simplesmente ampliando aquilo que já existe.

O componente de workflow é uma parte fundamental de qualquer solução de integração utilizada para fornecer processamento de fluxo de trabalho humano por meio de um recurso de gerenciamento de processo de negócio e para exercer a função central de executar os processos envolvidos na vinculação de sistemas. Essas duas tarefas são bem separadas e bastante distintas, mas ambas são perfeitamente adequadas para o uso das ferramentas de workflow. Um workflow de integração de sistemas pode incluir etapas como recuperar dados de cada sistema e agregá-los, validar entradas do usuário e atualizar os sistemas em conformidade com os dados inseridos ou atualizados.

As ferramentas de workflow fornecem o método ideal para realizar essas atividades, são amplamente superiores ao uso de código e podem, por isso, prover um benefício significativo aos negócios a partir de uma camada de integração, observando-se estes detalhes:

- Trabalhar através de vários sistemas é, de forma inerente, orientado a processos, executando tarefas passo a passo com pontos de decisão, ramificações e outras etapas básicas de workflow.
- Os fluxos de trabalho de integração de sistemas têm uma posição central nos aplicativos e, por isso, recebem com mais frequência os impactos causados por alterações no ambiente, uma vez que

cada alteração em cada sistema reflete em alterações na camada de integração. O uso de ferramentas de workflow permite que os processos sejam entendidos mais facilmente por um grupo de pessoas mais amplo que os desenvolvedores originais, devido à natureza gráfica e, de certo modo, de auto-documentação dos fluxos de trabalho; os processos podem ser alterados de forma mais rápida e fácil e depurados mais facilmente do que o código; e podem ser utilizados menos códigos, o que resulta em maior confiabilidade e depuração mais fácil.

- Como o formato em diagrama oferecido pelas ferramentas de workflow permite um entendimento mais fácil dos processos de negócios, a propriedade pode ser ampliada facilmente para além das equipes de desenvolvimento, para analistas de negócios e até mesmo usuários centrais na comunidade de negócios.

Uma postura de integração bem-sucedida

Possuir a ferramenta e utilizá-la corretamente são dois pontos bem diferentes, e esta discussão descreve uma *postura* necessária para se alcançar uma integração realmente bem-sucedida. Independente da abordagem utilizada lembre-se de que uma visão de integração no nível da empresa levará a benefícios muito maiores para os negócios; a integração não é trivial e deve ser considerada em um contexto corporativo; e, embora o este contexto e o projeto sejam considerados para obter o máximo de benefícios para os negócios, ainda é possível (e necessário) começar a implementar a solução com um projeto único.

A visão corporativa da integração deve ser desenvolvida o mais cedo possível. Embora haja claramente um lugar para integração ponto a ponto onde mais integração é improvável, um ambiente de integração mais capacitado, fornecendo o workflow e outros recursos aqui descritos, oferecerá vantagens à medida que crescer o número de sistemas e de integrações. Portanto, é necessária uma avaliação fora do escopo de cada projeto individual para descobrir as oportunidades que possam existir.

Como ocorre com todas as decisões de arquitetura, existem impulsionadores de negócios, custos, recursos etc. que são levados em consideração ao se tomar decisões de arquitetura. A abordagem de arquitetura de integração recomendada aqui é bem adequada às capacidades das ferramentas e aos conceitos de arquitetura atuais. Esperamos poder ajudá-lo a entender algumas das decisões que podem ser tomadas na área. •

Sobre o autor

Kevin Francis é um profissional de TI com 19 anos de experiência no mercado em uma série de cargos—de CIO em uma empresa de fabricação e consultor de segurança, a arquiteto global — e também administrou sua própria organização de desenvolvimento de software com grande êxito. Kevin está envolvido na arquitetura e no gerenciamento de projetos de ponta de e-commerce, desde seu nascimento. Como um dos principais arquitetos da Infosys, Kevin é responsável pelo êxito da implementação técnica da empresa para seus clientes. Ele faz consultoria para grandes organizações; trabalha com arquitetos, equipes de desenvolvimento e clientes da Infosys para garantir que cada solução técnica contribua para a qualidade e a direção estratégica global do cliente; e é membro do The Infosys Australia Technology Council. Kevin recebeu o prêmio MVP (Most Valuable Professional) da Microsoft em julho de 2005 pelo seu conhecimento e experiência na área Desenvolvedor visual – arquiteto de soluções.



Simplifique modelos de workflow complexos

por Andrew Needleman

Resumo

Vários fatores afetam a modelagem de um bom workflow. Ao simplificar o processo, é importante descrever por meio de diagrama as etapas necessárias de uma transação importante e específica, como um pedido de e-commerce ou uma consulta médica. Nesta conversa, analisaremos a representação do workflow em um novo tipo de diagrama chamado diagrama de “processo de pontos e linhas”. Começaremos diagramando as etapas possíveis que podem ser realizadas em cada ponto do workflow e indicaremos para onde esse diagrama nos leva no processo desse fluxo de trabalho. Também garantimos que não perderemos nenhuma das potenciais etapas ou pontos de estado durante o caminho.

Um modelo de workflow exige análise do processo de negócios, revisão do processo de negócios, análise de usabilidade e projeto de software. Essa lista assustadora de habilidades necessárias transforma o workflow em um desafio para os arquitetos de software mais experientes. Como decompor o problema em partes menores para começar na direção certa? Analisaremos a maneira de simplificar o processo de modelagem de sistemas complexos com vários tipos de usuários e centenas de estados potenciais. Ao simplificar o processo, pode-se pensar na solução do workflow e não no desenho do workflow.

Cada projeto de software possui um processo central que é a razão principal pela qual o sistema existe ou está sendo construído. Por exemplo, a meta principal de um site de e-commerce é tornar mais fácil para as pessoas comprar produtos. Esse processo principal é o que precisa da maior atenção durante seu projeto e implementação. Mostraremos a abordagem de modelagem de pontos e linhas que irá ajudá-lo a se comunicar sobre processos complexos com seus especialistas de negócio para determinar as ações e os estados disponíveis durante a criação do workflow.

Vamos começar com uma visão geral da abordagem de pontos e linhas para a criação de diagramas de workflow. Na abordagem de pontos e linhas, cada ponto representa um estado específico no processo e cada linha corresponde a uma ação que pode ser executada a partir desse ponto. Os pontos e as linhas são rotulados e uma chave é preparada à medida que o workflow é desenvolvido. Os pontos são rotulados com letras começando com A, e as linhas são rotuladas com números começando com 1. Dessa maneira, os pontos e as linhas não irão se misturar.

Os principais objetivos por trás da abordagem de pontos e linhas são: criar uma ferramenta de visualização simples em papel que possa ser utilizada com usuários corporativos; eliminar o preciosismo causado pela diagramação em um computador; tornar o processo o mais leve possível para permitir concentração na difícil tarefa de projetar o workflow; e separar as alterações na descrição de ações e estado das alterações no fluxo do processo (ou seja, semelhante à idéia de uma tabela de domínio).

Siga as regras

Além desses objetivos, há cinco regras que devem ser aplicadas a um diagrama que usa essa abordagem. Observemos os detalhes de cada regra.

O diagrama sempre segue somente em uma direção. Uma das idéias principais da abordagem de pontos e linhas é que o diagrama sempre segue em frente como uma linha do tempo. Pode-se terminar em um estado anterior, mas o diagrama não possui linhas indo para todas as quatro direções como um labirinto.

Os estados podem ser descobertos à medida que for construindo o diagrama. Em um workflow simples, se conhece todos os potenciais estados antes de iniciar a diagramação. No entanto, em um workflow mais complexo poderá ser necessário descobrir os estados possíveis construindo o diagrama. Determine as ações que podem ser realizadas em cada ponto do processo e veja para onde isso o leva. À medida que se descobre os estados, pode-se decidir como rotular cada um deles para o seu uso. Em seguida, após a conclusão do diagrama, é possível decidir como apresentar os estados a cada tipo de usuário e se deve combiná-los de maneira mais fáceis de entender.

“CADA PROJETO DE SOFTWARE TEM UM PROCESSO CENTRAL QUE É A RAZÃO PRINCIPAL PARA ESSE SISTEMA EXISTIR OU ESTAR SENDO CRIADO.”

Os rótulos dos estados e das ações são mantidos fora do diagrama. Quando temos as descrições de cada ação e estado, o diagrama fica confuso e difícil de entender. Colocar os rótulos dos estados e das ações como uma chave, facilita a visualização de um fluxo de processo maior em um espaço menor, o que é quase como normalizar o diagrama. É possível alterar os rótulos dos pontos sem precisar alterar o diagrama. Isso facilita visualizar e diagramar processos complexos.

O layout é simples e flexível, permitindo concentrar-se no conteúdo. O layout é um grande problema nos fluxos de processo tradicionais. Muito esforço é realizado para conectar e encaixar os diferentes itens no diagrama. Se for deixado de lado uma ação em um fluxo de processo, o diagrama

inteiro poderá precisar ser reorganizado. Na abordagem de pontos e linhas, apenas precisa encaixar uma linha para a ação e um ponto para o estado do resultado. Se for necessário alterar uma ação ou um estado, o seu respectivo número ou letra pode ser alterado. Se desejar alterar o nome de um estado ou uma ação, se pode simplesmente alterar a chave em vez de o diagrama. A simplicidade de adicionar, alterar e visualizar pontos e linhas do diagrama ajuda a ficar concentrado no processo.

“OS ESTADOS E AS AÇÕES DEVEM SER INCLUÍDOS SE FOREM NECESSÁRIOS PARA COMPLETAR O PROCESSO QUE ESTIVER SENDO ESTUDADO”

Os estados podem ser “retirados” do diagrama principal. Como se está sempre indo adiante com o diagrama, podem ser retiradas seções do diagrama sem se preocupar com elas conectam aos estados anteriores do workflow. Por exemplo, A conecta-se a B e desejamos retirar B para o seu próprio diagrama porque ele possui muitos filhos para encaixar no diagrama existente. Não precisamos nos preocupar se B tem uma ação que vai para C e se C tem uma ação que envia seu status de volta para A. Rotulando o estado com A, eles estão conectados, em vez de precisar ter uma linha de volta até o status A original. O nosso diagrama só avança e não retrocede, por isso não precisamos conectar de volta ao ponto A original. Essa regra permite uma grande flexibilidade quando estamos ficando sem espaço em um diagrama.

Ações e estados para fluxos de trabalho

Agora que analisamos algumas das vantagens da abordagem de pontos e linhas, vamos ver como determinar se deve haver uma ação ou estado no seu diagrama. Os estados e as ações devem ser incluídos somente se forem necessários para completar o processo que estiver sendo estudado. Por exemplo, embora efetuar logon ou registrar-se em um sistema de e-commerce não altere o estado de um entidade do pedido, esse procedimento é necessário para concluir uma compra. Portanto, ele deve ser incluído como uma ação no sistema, e os estados que essas ações criam devem ser estudados e diferenciados se estivermos examinando o processo de pedidos.

Estados e ações não precisam ser incluídos caso não sejam necessários para concluir o processo que estiver sendo estudado e não alterarem as etapas necessárias para concluir o processo que estiver sendo modelado. Em outras palavras, uma ação que faz regredir uma etapa do processo é algo que deve ser modelado. Por outro lado, uma ação que não estiver relacionada ao processo não precisa ser modelada.

Naturalmente, existe muito espaço para interpretação quanto ao que deve ser colocado em um diagrama de processo, porque ligar um computador é necessário para realizar qualquer processo de computador, mas provavelmente seria tolice incluir essa ação como parte de um diagrama de processo. Agora vamos usar um exemplo para tentar esclarecer o que deve e o que não deve estar no diagrama de pontos e linhas.

Observemos um exemplo familiar de workflow a ser examinado com esse processo — a compra de itens por meio de um site da Web de e-commerce. Examinaremos as etapas necessárias que devem ser seguidas para adquirir um produto.

Ao começarmos a projetar o workflow de uma compra, as etapas que não alteram o estado da transação de compra serão ignoradas. Por exemplo, o processo de pesquisar o produto não altera, de forma alguma, o

Tabela 1 Chaves para o diagrama de e-commerce

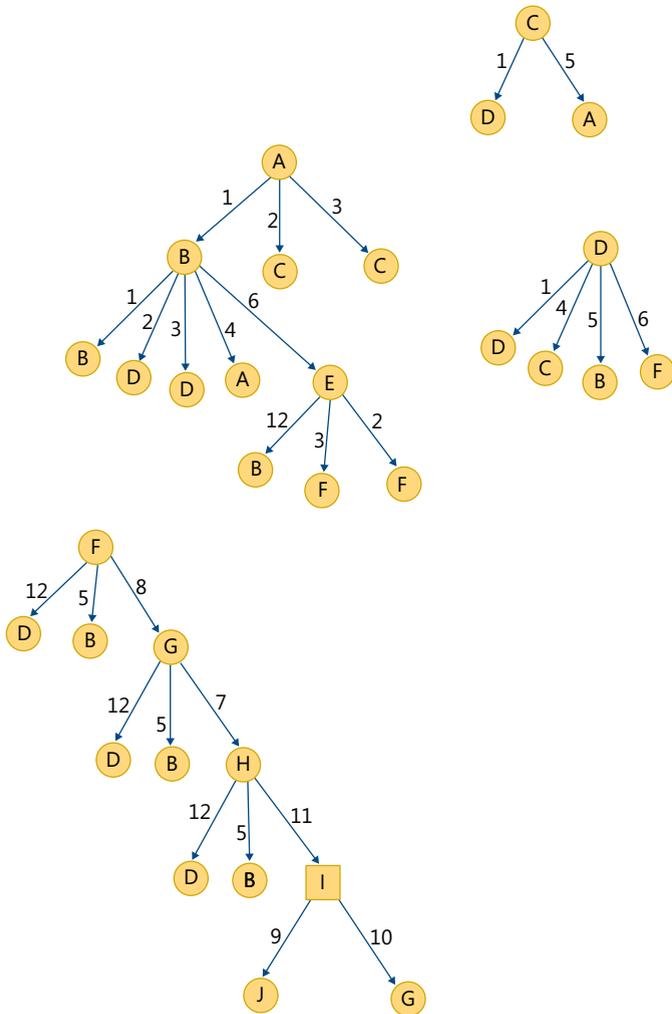
| Ações | |
|----------------------|--|
| 1 | Localizar e adicionar um produto ao carrinho de compras. |
| 2 | Registrar-se. |
| 3 | Efetuar logon. |
| 4 | Remover todos os produtos do carrinho de compras. |
| 5 | Efetuar logout. |
| 6 | Iniciar o processo de finalização do pedido. |
| 7 | Adicionar/selecionar informações de envio. |
| 8 | Adicionar/selecionar informações de pagamento. |
| 9 | Processamento da cobrança bem-sucedido. |
| 10 | Falha no processamento de cobrança. |
| 11 | Confirmar pedido. |
| 12 | Sair do processo de finalização do pedido. |
| Descrições do status | |
| A | Não há produtos no carrinho de compras, não logado. |
| B | O produto está no carrinho de compras, não logado. |
| C | Não há produtos no carrinho de compras, logado. |
| D | O produto está no carrinho de compras, logado. |
| E | Processo de finalização do pedido, não logado, sem dados de envio ou faturamento. |
| F | Processo de finalização do pedido, logado, sem dados de envio ou faturamento. |
| G | Processo de finalização de pedido, logado, com dados de envio, sem dados de faturamento. |
| H | Processo de finalização do pedido, logado, com dados de envio e faturamento. |
| I | Finalização do pedido confirmada, logado, estado temporário (quadrado). |
| J | Processamento do pedido bem-sucedido, logado. |

estado de um processo de pedido. Realmente não importa como o usuário circula pelo site a menos que realize uma das três ações que alteram seu estado atual. Essas três ações que podem modificar o estado da compra do produto são: adicionar um produto no carrinho de compras, registrar-se no site e efetuar logon no site.

Por exemplo, imagine que você vá para a seção de “livros” e clique de volta para a página principal. Essa ação alterou o seu estado de alguma maneira em termos da compra de um produto? Não. Imagine que você procure no site por “livros .NET” e irá receber uma lista deles. Em seguida você clica de volta para a página principal. Ainda não se avançou no processo de compra.

Mesmo se fizer de conta que possui um mecanismo que relaciona sugestões de produtos com o seu histórico de navegação, isso ainda não faz avançar o processo de compra até adicionar um produto ao carrinho de compras. É muito importante manter essa ação fora do diagrama porque irá complicá-lo de forma desnecessária. Podemos combinar a navegação e, em seguida, adicionar um produto ao carrinho de compras em uma ação no diagrama do processo de compra chamada “localizar e adicionar produto”. Se desejássemos diagramar o processo de localizar um produto, estaríamos interessados na maneira como o usuário selecionou os produtos para visualizar e adicioná-los ao carrinho de compras. No momento, iremos nos concentrar na diagramação do processo de compra.

Figura 1 Integração ponto a ponto e integração de barramento de serviço



cada transação que estiver ocorrendo no site, precisaremos agrupar vários estados sob uma única descrição. Por exemplo, os estados E–G podem ser agrupados como um estado “incompleto” para o administrador. É improvável que o administrador precise saber mais do que isso, exceto para montar estatísticas que seriam exibidas em um relatório e não na tela de trabalho diária. Além disso, se o administrador entrar em um desses pedidos,

“AO COMEÇARMOS A PROJETAR O WORKFLOW DE UMA COMPRA, IREMOS IGNORAR AS ETAPAS QUE NÃO ALTERAM O ESTADO DA TRANSAÇÃO DE COMPRA”

ele poderá fornecer os detalhes do estado desse ponto.

Uma das maneiras mais fáceis de atribuir os estados para cada usuário é usar o método “com quem está a bola?”. Em outras palavras, qual usuário deve fazer algo para avançar o workflow? O usuário que se encaixar nessa descrição deverá ver esse item agrupado com outros que estão aguardando sua ação. Em seguida se pode agrupar os outros itens em aberto em outra categoria e, finalmente, os itens concluídos/cancelados em uma terceira categoria.

Uma última recomendação em relação à abordagem de pontos e linhas é usar um papel bem grande e uma caneta para criar os diagramas. Outro pedaço de papel grande deve conter os nomes dos estados e outro ainda deve conter os nomes das ações. O uso de folhas de grandes e separadas dessa maneira permite uma boa visão panorâmica do processo inteiro em vez de vê-lo decomposto em várias telas.

O uso de papel também concentra a atenção de todos no problema imediato, em vez de fazê-lo parecer bonito. Se estiver usando papel, não existe uma maneira fácil de colocar um efeito em cada ponto ou reorganizar os itens para encaixá-los melhor. Você achará a abordagem de pontos e linhas tão útil quanto eu achei ao descobrir todos os possíveis estados e ações com os seus especialistas de negócios. •

Papéis no processo de compra

Primeiro, começaremos com alguém que acabou de entrar na página principal do nosso site e-commerce. Observamos o que vem a seguir no nosso processo. As três possibilidades em termos de comportamento que faz avançar o processo de compra são adicionar um produto no carrinho de compras, registrar um usuário e efetuar logon no site. O diagrama de pontos e linhas desse processo inteiro pode ser observado na Figura 1 e a suas chaves na Tabela 1.

É importante entender o impacto que vários papéis no site exercem nos nossos status e permissões visíveis. Por exemplo, suponha que nosso site de e-commerce tenha papéis para compradores e um administrador. Os compradores são as pessoas que adquirem o produto, enquanto o administrador assegura que elas recebam o produto. O nosso diagrama de pontos e linhas ajuda a determinar o que mostrar ao administrador, fornecendo uma lista abrangente dos estados em que um pedido está durante o processo de compra. Podemos analisar esses estados e percorrer a lista para descobrir quais pedidos devem ser vistos pelo administrador e quais não devem. Por exemplo, uma regra de negócios poderia ser aquela que o administrador pode ver transações de pedidos que fossem abandonadas na etapa de cobrança, para permitir que seja feito o acompanhamento desses clientes.

Em sites mais complexos com centenas de possíveis estados para

Sobre o autor

Andrew Needleman é o parceiro executivo da Claricode (www.claricode.com), desenvolvedores de soluções personalizadas exclusivamente para o setor de assistência médica. Ele tem escrito artigos para muitas publicações sobre o desenvolvimento de software nas indústrias de TI e de assistência médica. Sua experiência em projetos de workflow inclui desenvolver a arquitetura de um aplicativo com centenas de estados distintos, várias ações em cada estado e nove tipos de usuários distintos. Seu aplicativo foi reconhecido como um Intel Solution Blueprint para práticas recomendadas em assistência médica com tecnologias Microsoft. Entre em contato com Andrew no endereço andrew@claricode.com.



Prepare a empresa orientada a serviços

por William Oellermann

Resumo

Criar Web Services é muito fácil atualmente. Criar vários Web Services é um pouco mais difícil e gerenciar esses Web Services é realmente difícil. À medida que o número de serviços e de consumidores aumenta, os benefícios fundamentais da orientação a serviços diminuem se determinadas disciplinas de TI não forem seguidas. Vamos observar um modelo que pode facilitar a identificação e a priorização dessas disciplinas para uma empresa orientada a serviços.

Ao conversar com clientes e colegas sobre Web Services, descobrirá que grande parte do entusiasmo e do interesse concentra nas especificações que ajudaram essa tecnologia a tornar-se rapidamente um padrão de fato: SOAP, WSDL, XML, WS-Security etc. Embora essa padronização seja importante, ela geralmente nos distrai do nível de discussões seguinte sobre como aproveitar totalmente essas tecnologias. Esses protocolos permitem maior alcance e reutilização do que conseguíamos anteriormente, mas o seu uso sozinho não irá alcançar esse potencial.

Todo o trabalho investido em padrões e consórcios permite desenvolver serviços que podem interoperar com outros aplicativos e sistemas, mas não nos permite necessariamente a reutilização. Mais importante ainda, não fornece capacidade de gerenciamento. Se por acaso trabalhar em ou com uma empresa que alcançou êxito com serviços da Web, sem dúvida encontrou a necessidade de tratar, se não a dificuldade de não poder tratar, de testes de serviços, implantação, versionamento e gerenciamento. Essas áreas tornam-se muito problemáticas quando não são identificadas antes da construção de um grupo de serviços.

Essa condição já tornou algumas empresas vítimas do seu próprio sucesso. As abordagens de trabalho com alguns serviços em certa frequência não funcionam quando aumenta o número de serviços e sua utilização. Se não houver princípios e práticas em vigor para promover a reutilização e fornecer capacidade de gerenciamento à medida que crescer o uso dos serviços, se atingirá o “teto de escalabilidade”. (Consulte o artigo na barra lateral, “Uma palavra sobre modelos de maturidade”.)

Discutir esse problema é muito difícil, mesmo entre os arquitetos mais sábios e experientes, quanto mais com outros participantes menos técnicos que desempenham uma função no ecossistema de serviços. Parte dessa dificuldade deriva do uso de termos sobrecarregados como serviço, mas grande parte resulta da total complexidade envolvida em uma arquitetura massivamente distribuída que abrange muitas áreas de uma empresa. Se um problema é difícil de discutir, imagine quão difícil pode ser definir uma estratégia ou um plano para ele.

Esses desafios nos levam a considerar o uso de um modelo para ajudar a facilitar a discussão da ativação de serviços e o planejamento de uma empresa preparada para serviços. Com as contribuições e o retorno de muitas empresas e colaboradores atenciosos durante os últimos dois anos, desenvolvemos o ESOMM - Enterprise Service Orientation Maturity Model (Modelo de Maturidade Empresarial de Orientação a Serviços).

Um modelo de maturidade dirigido a disciplinas

Todas as 4 camadas, 3 perspectivas e 27 disciplinas definidos no ESOMM foram criados como um roteiro para oferecer suporte a serviços, não a qualquer específico com qualquer uso, implementação ou aplicativo específico, mas a qualquer serviço ou, mais especificamente, qualquer conjunto de serviços. Se a intenção for desenvolver somente um serviço, o valor oferecido por esse modelo será limitado. O objetivo do ESOMM é ajudá-lo a superar os problemas de dimensionamento em um nível de grupo, departamento, divisão ou empresa. Esse objetivo significa que o ESOMM pode ser aplicado como parte de uma iniciativa de base ou de uma estratégia corporativa. A capacidade de fornecer determinadas disciplinas poderá ser limitada pela extensão do seu alinhamento organizacional total, mas esse fator não deveria impedir a sua capacidade de começar a criar e implementar um plano aproveitando o ESOMM.

O desenvolvimento de uma estratégia de uso de serviços não é uma realização trivial e não deve ser tratado como um esforço único ou de curto prazo. No seu ápice, a orientação a serviços pretende fornecer um nível de agilidade mais elevado para uma empresa, de forma a poder oferecer uma resposta adequada às necessidades da empresa e de seus clientes. Alcançar essa agilidade com sucesso significa fornecer um plano que seja extensível e durável. Ao se concentrar no serviço e não na mensagem, implementação ou utilização, o ESOMM pode ajudá-lo a criar um plano que possa ser aplicado

“TODO O TRABALHO INVESTIDO EM PADRÕES E CONSÓRCIOS PERMITE DESENVOLVER SERVIÇOS QUE PODEM INTEROPERAR COM OUTROS APLICATIVOS E SISTEMAS, MAS NÃO NOS PERMITE NECESSARIAMENTE A REUTILIZAÇÃO”

de forma ampla em qualquer empresa. À medida que as implementações de serviço, opções de tecnologia e padrões de utilização preferidos mudam, o modelo deve fornecer um roteiro durável que possa resistir a essas mudanças.

O ESOMM consiste em quatro camadas, cada uma contendo um conjunto de disciplinas que, como um todo, estabelece um nível sólido de realização e valor para a empresa em determinada camada (observe a

Figura 1). Esse valor deverá ser percebido em sua totalidade se todas as disciplinas relacionadas forem cumpridas. É possível, e certamente provável que uma empresa possa tratar algumas partes de várias camadas; o objetivo não é tanto atender a uma camada e sim identificar o nível de valor que uma empresa deseja.

A primeira camada de maturidade do ESOMM, chamada utilizável, consiste na utilização apropriada de padrões e protocolos para projetar e desenvolver serviços que sejam utilizáveis nas plataformas e tecnologias de uma empresa. Criar serviços corretamente é absolutamente importante antes de podermos estabelecer com sucesso muitos dos outros recursos da SOA. O teste real para ver se os seus serviços estão bem construídos é observar a semelhança e a consistência presentes em todos os seus serviços, independente das ferramentas e tecnologias utilizadas.

A camada *reproduzível* trata muitas das disciplinas que tornam os serviços eficientes para desenvolver, implantar e manter na empresa. Sem tratar as disciplinas principais nessa área, será muito difícil dimensionar com sucesso o uso dos Web Services.

A camada *gerenciável* encapsula muitas das disciplinas necessárias para fornecer serviços com confiabilidade para aplicativos de missão crítica, incluindo o suporte operacional de serviços e, igualmente importante, os recursos de auto-serviço fornecidos aos clientes. Auto-serviço é uma ferramenta importante para permitir que os seus serviços escalem sem que os desenvolvedores e arquitetos precisem gastar uma quantidade de tempo imensa proporcionalmente aos clientes potenciais dos seus serviços. Essa camada é muito difícil de abordar com sucesso sem certo êxito em tornar os serviços reproduzíveis na sua implantação e no seu gerenciamento.

A camada *extensível* representa o ápice da percepção da agilidade nos negócios prometida pelos serviços e a reutilização dos mesmos, como os blocos de construção de outros serviços, ou a agregação de serviços. Essa noção geralmente envolve fornecer recursos diretos aos clientes e/ou parceiros por meio de serviços que geram novos canais de receitas, se não novos modelos de negócios. Existem diversas disciplinas obrigatórias para tornar essa realização possível, os quais compõem a maior parte do enfoque nessa camada extensível.

Perspectivas do roteiro

Como resultado dos relacionamentos e das interações com clientes nos últimos dois anos, é possível reconhecer a necessidade de tratar diferentes perspectivas por meio desse modelo. À medida que discutimos as disciplinas em cada camada do modelo, tornou-se aparente a aparição de níveis variados de apreciação ou consideração por eles, dependendo da função e dos desafios do participante. Por exemplo, uma empresa teve dificuldades em avaliar a necessidade de projetar uma interface orientada a contratos. Eles estavam bastante satisfeitos utilizando ferramentas para gerar dinamicamente contratos de serviço e consideravam essa abordagem como de grande sucesso.

No entanto, ao conversar com os usuários dos seus serviços, tivemos uma perspectiva inteiramente diferente ao saber que a utilização de tipos de dados e padrões de elementos específicos causava muita dificuldade durante a implementação. Ao reconhecer que nem todas as disciplinas são orientadas pelos proprietários ou provedores do serviço, os acionistas podem fazer uma avaliação melhor e tomar decisões mais informadas sobre como abordá-las e priorizá-las.

As disciplinas em cada camada do ESOMM são categorizadas em uma das três perspectivas: implementação, consumo e administração (consulte a Figura 2). As disciplinas de *implementação* visam o desenvolvimento e a implantação de Web Services da perspectiva do provedor, sendo essa que a maioria das empresas considera por padrão e que precisa de menos esclarecimento. As disciplinas de consumo são as que mais agradam aos clientes dos nossos serviços, tornando-os mais fáceis de utilizar e, por isso,

Figura 1 As quatro camadas de maturidade do ESOMM

| | |
|---------------------|--|
| Extensível | Uma empresa é capaz de agregar serviços e estender seu uso para além dos seus próprios limites. |
| Gerenciável | Uma empresa pode gerenciar de forma efetiva quantidades crescentes de serviços para SLAs garantidos. |
| Reproduzível | Uma empresa implementa, consome e reutiliza serviços de forma eficiente e consistente. |
| Utilizável | Uma empresa é capaz de preparar e consumir serviços em conformidade com padrões com alcance excelente. |

adotados com sucesso e de forma mais ampla. Acreditamos que essa é a perspectiva mais negligenciada, no entanto ela pode ser a mais importante para determinar o seu sucesso com a orientação a serviços. Se for difícil aproveitar seus próprios serviços, seu sucesso será limitado, não importa as outras áreas que se tenha otimizado. As disciplinas de *administração* utilizam os aspectos operacionais e de controle dos Web Services por meio da empresa. Essa perspectiva está se tornando mais reconhecida e apreciada, mas existe muito mais ainda que precise ser entendido sobre seu valor e impacto.

O valor adicional das perspectivas é o auxílio que elas fornecem na priorização da implementação de disciplinas específicas no modelo. Embora exista uma capacidade para ignorar determinadas disciplinas ao avançar no modelo, é um tanto arriscado ignorá-las em uma camada inferior dentro da mesma perspectiva. Esses pré-requisitos destinam não somente a ajudar o sistema a crescer de forma lógica, mas também a ajudá-lo a aprender

“A CAMADA EXTENSÍVEL REPRESENTA O ÁPICE DA PERCEPÇÃO DA AGILIDADE NOS NEGÓCIOS PROMETIDA PELOS SERVIÇOS E A REUTILIZAÇÃO DE SERVIÇOS, COMO OS BLOCOS DE CONSTRUÇÃO DE OUTROS SERVIÇOS, OU A AGREGAÇÃO DE SERVIÇOS”

como os serviços são utilizados e aceitos em toda a empresa porque essas informações irão causar impacto em determinadas disciplinas nas camadas mais elevadas.

Também é importante compreender que uma empresa pode escolher concentrar-se em uma ou duas das perspectivas, mas a maturidade total e, portanto, o valor total dos seus serviços, depende do nível de atenção apropriado a todas as três. Negligenciá-las tratará conseqüências — algumas aparentes mais rapidamente do que outras. As conseqüências seriam semelhantes a ignorar um dos grupos de alimentos; é possível, mas limitaria algumas das refeições que poderia desfrutar e não seria ideal para a sua saúde.

Disciplinas do ESOMM

Há 27 disciplinas descritas no modelo ESOMM e cada uma delas exigiria facilmente um artigo inteiro para ser explicado em detalhes (consulte a Figura 3). Aqui teremos uma breve visão sobre elas para ajudar a articular a premissa de cada disciplina e o seu alinhamento à perspectiva e à camada associadas.

A perspectiva de implementação inclui as disciplinas de processos de desenvolvimento e de padrões de projeto na camada utilizável (1): *Processos de desenvolvimento*. Do mesmo modo que os padrões de projeto, os processos de desenvolvimento abordam um conceito com o qual as equipes de desenvolvimento estão bastante familiarizadas, mas enfocam especificamente a implementação de serviços. Em um nível básico, esse conceito consistiria geralmente no uso de padrões comuns aceitos em plataformas e ferramentas mais comuns, incluindo SOAP, WSDL, protocolos WSA e os Perfis WS-I. Em um nível mais avançado, a disciplina incluiria a definição de processos reproduzíveis para o projeto de interfaces de serviço, a implantação e a publicação de serviços. Aqui também o objetivo é tornar esses processos amplamente disponíveis nos limites das plataformas.

Padrões de projeto. Toda vez que for desenvolver um software, é aconselhável aproveitar padrões de projeto comprovados que sejam apropriados para essa implementação. Esse recurso específico aborda os padrões de projeto que são apropriados e necessários, especificamente para uso com serviços. Temos boas informações disponíveis nessa área, começando com os quatro princípios da orientação a serviços (consulte Recursos). Outros exemplos de padrões relevantes incluem execução assíncrona, suporte a retorno de chamadas (callback), tratamento de exceções e utilização de dados de cabeçalho SOAP personalizado. A chave para obter essa disciplina é definir padrões que possam ser utilizados por desenvolvedores em todas as ferramentas e plataformas apropriadas.

Da perspectiva de consumo do nível utilizável, existem os contratos explícitos e os recursos de capacidade de teste:

Contratos explícitos. As informações mais básicas que se deve fornecer aos clientes dos seus serviços são contratos explícitos, o que inclui não somente esquemas ricos em detalhes (aproveitando a potência do XSD), mas dados de política completos que compartilham informações de contexto apropriadas. Essas informações permitem que eles conheçam não somente a composição do corpo da mensagem, mas os elementos de cabeçalho que são aceitos e necessários. A dificuldade desse recurso é que existe pouco suporte para alguns dos padrões que surgem em torno do compartilhamento de políticas, de modo que ele exige certa reflexão e empenho para o tratamento adequado.

Capacidade de teste. Essa disciplina fornece aos clientes um mecanismo para testar a chamada de um serviço para dar suporte à implementação e à solução de problemas do lado do cliente. Um dos aspectos mais desafiadores da implementação de sistemas distribuídos é testar a integração. Com tantas dependências, ela pode exigir muita coordenação para simplesmente completar uma chamada através da cadeia inteira, mesmo para uma operação de teste ou uma solicitação inválida. Os Web Services estão em uma posição privilegiada em separar a interface de aplicativos dos sistemas de retaguarda que estão realizando

o processamento; um ambiente de teste, físico ou virtual, pode ser uma ferramenta estratégica para aprimorar a eficiência da implementação e reduzir a necessidade de suporte manual.

A perspectiva de administração da camada utilizável contém disciplinas de monitoramento básico e de modelo de segurança:

Monitoramento básico. O suporte a Web Services começa com o suporte da infra-estrutura básica que hospeda e fornece esses serviços. Ele inclui a base da excelência operacional, que abrange o monitoramento de contadores e registros principais que poderiam indicar problemas de processamento e fornecer alertas sobre os acionadores que indicam interrupções de serviços básicos. Essa disciplina não inclui visibilidade ou monitoramento no nível de serviço, pois não é um empreendimento trivial e virão mais acima nas camadas do modelo.

“UM DOS ASPECTOS MAIS DESAFIADORES DA IMPLEMENTAÇÃO DE SISTEMAS DISTRIBUÍDOS É TESTAR A INTEGRAÇÃO”

Modelo de segurança. Quando ocorre a adoção de serviços, as empresas verão que eles são usados em uma variedade de aplicativos, tanto externos quanto internos. É importante possuir um modelo de segurança padrão que possa dar suporte a todos os cenários principais para atender os requisitos dos negócios. Esse modelo envolve criar um ou mais armazenamentos de credenciais que podem ser gerenciados e aceitos como parte do seu ecossistema de serviços. Os exemplos desses armazenamentos de credenciais incluem autoridades certificadoras, um diretório LDAP, serviços de diretório Active Directory e bancos de dados. Embora as credenciais constituam um componente importante do seu modelo de segurança de serviço, tratar esse recurso de forma completa também significa tratar os requisitos referentes a acesso, autenticação, autorização e criptografia. Essas abordagens não precisarão ser implementadas necessariamente como funcionalidades caso não forem exigidas no momento, mas um modelo de segurança completo precisa ser definido para quando chegar a hora. Essa implementação provavelmente será a primeira etapa na definição da sua política de serviços.

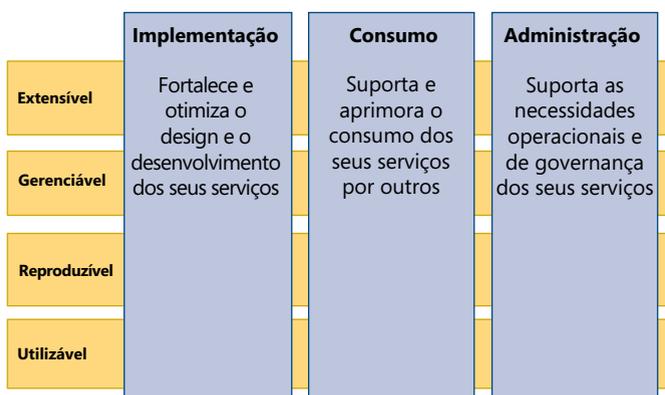
Disciplinas da camada reproduzível

A perspectiva de implementação inclui as disciplinas de blocos de serviço e esquema comum na camada reproduzível (2): Blocos de serviço. Uma solução para reduzir a quantidade de lógica de “rotina” imposta a cada um dos serviços é o uso de um conceito chamado blocos de serviço. Esse recurso é um modelo de arquitetura que fornece rotinas comuns que deveriam ser aproveitadas de forma ampla, como transformações, consultas de dados em um determinado ponto e processos utilitários. Um exemplo de bloco de serviço seria um serviço de logging, e de fato o nosso bloco de logging utilizado em outras aplicações poderia ser orientado a serviços para atender essa necessidade. Para esse modelo ter sucesso, esses blocos deveriam estar disponíveis a todos os serviços em toda a empresa.

Esquema comum. Ao fornecer serviços que trabalham com entidades comuns, é muito útil ter uma definição única para essas entidades para minimizar o esforço necessário para fornecer e consumir esses serviços. Devido aos vários sistemas normalmente envolvidos no fornecimento da lógica de negócios e dos dados de serviços, essa definição geralmente é um esforço contínuo que nunca necessariamente será considerado completo. No entanto, mesmo o estabelecimento de definições comuns para um punhado de entidades centrais pode resultar em imensos benefícios.

A perspectiva de consumo da camada repetível inclui as disciplinas de capacidade de descoberta de serviços e fornecimento automático: Capacidade de descoberta de serviços. Conforme começa a fornecer mais

Figura 2 Três perspectivas das camadas de maturidade



serviços, a capacidade de descoberta torna-se um recurso importante para ajudar a orientar de forma eficiente os clientes na identificação do serviço apropriado a usar. Essa orientação pode ser realizada manualmente ou programaticamente com ferramentas (por exemplo, por meio do UDDI). Infelizmente, apenas fornecer uma lista de serviços em uma página da Web é suficiente nesse nível, de modo que uma das chaves para um bom mecanismo de descoberta é a categorização inteligente. Uma ou mais taxonomias bem elaboradas pode ajudar os potenciais clientes a procurar intuitivamente nos lugares corretos os serviços baseados em um domínio ou uma implementação específica.

Fornecimento automático. À medida que desenvolve mais serviços, torna-se cada desafio para os clientes solicitar acesso a eles. Embora, como provedor, possa estar interessado em automatizar a outra ponta do processo de fornecimento do usuário, é importante bem no início permitir aos clientes identificar rapidamente e aproveitar as ferramentas disponíveis para solicitar acesso. Se esse processo for muito difícil ou problemático, a sua taxa de adoção provavelmente será prejudicada, limitando o seu êxito geral. Essa capacidade pode ser obtida com uma grande variedade de soluções que podem variar de muito simples a muito complexas.

A perspectiva de administração do nível reproduzível inclui as disciplinas de políticas corporativas e gerenciamento de implantação:

Políticas corporativas. As políticas de serviço são importantes para definir, dar suporte e colocar em vigor o uso de serviços da sua empresa à medida que ela cresce. O ponto inicial desse esforço era desenvolver um modelo de segurança, mas neste nível é necessário ampliar as políticas para dar suporte a um conjunto de opções possíveis e disponíveis dependendo dos diferentes critérios de um determinado cenário. Por exemplo, um serviço usado em um departamento provavelmente terá um conjunto de políticas diferente que o mesmo serviço usado pelos parceiros. Neste nível, a definição de categorização interna torna-se importante porque pode ajudá-lo a definir modelos de básicos alinhados com alguns dos cenários mais comuns para promover consistência e eficiência. O sucesso nessa área não é conduzido tanto pelas escolhas feitas e sim pelo estabelecimento, utilização e distribuição de políticas aos elementos que sofrem o impacto.

Gerenciamento de implantação. A melhor maneira de promover e colocar em vigor um modelo de controle consistente para a sua empresa é definir um processo de implantação comum para serviços que forneça valor real. Por exemplo, se o seu processo de implantação fornecer automaticamente serviços com suporte operacional, ele rapidamente

será adotado de forma ampla. Para fortalecer mais ainda a sua função, é importante também que esse processo seja flexível e fácil de usar, de forma que os desenvolvedores, arquitetos, testadores e operadores em toda a empresa aceitem e participem do modelo estabelecido.

Recursos da camada gerenciável

A perspectiva de implementação do nível gerenciável (3) contém as disciplinas de banco de esquema, política executável e versionamento. Vamos examiná-los em mais detalhe:

Banco de esquemas. Criar e dar suporte a um banco de esquemas é a etapa seguinte no aproveitamento de esquemas comuns. Isso permite aos desenvolvedores aproveitar e reutilizar definições de entidade existentes para criar mensagens que consigam adaptar automaticamente às extensões do esquema por meio de inclusões e importações. Esse aproveitamento é uma área de grande desafio que bem poucos clientes estão abordando com sucesso atualmente, mas irá rapidamente tornar uma ferramenta valiosa à medida que os serviços começarem a evoluir.

Política executável. Embora uma mensagem de solicitação inicie a chamada de um serviço, normalmente desejará aproveitar as informações da política para determinar exatamente o que deve ser executado em condições específicas. As políticas são geralmente consideradas como uma ferramenta de uso externo, mas também podem ser igualmente uma ferramenta poderosa para fornecer contexto internamente. Se necessitar alterar o processo de execução ou o comportamento com base em quem está fazendo a solicitação, no formato que os dados assumem ou na hora do dia, a política executável desempenhará uma função importante na ativação do serviço.

“PARA TORNAR OS CLIENTES DE SEUS SERVIÇOS MAIS AUTO-SUFICIENTES, PERMITA QUE TENHAM UMA VISIBILIDADE AMPLA AO USÁ-LOS”

Versionamento. O versionamento dos Web Services é algo que pode e deve ser tratado de forma diferente das abordagens normais para outros tipos de software devido à sua complexidade e extensão. Por exemplo, o versionamento de um serviço precisa ser tratado separadamente do versionamento das mensagens envolvidas no serviço. Felizmente, com a flexibilidade inerente do XML e dos esquemas XML, existe a capacidade de

Uma palavra sobre modelos de maturidade

Do mesmo modo que outros modelos de maturidade, o ESOMM deriva-se do CMM (Modelo de Maturidade de Disciplinas) de Carnegie Mellon (consulte Recursos). No entanto, tomamos emprestado a noção de modelo de maturidade dirigido a disciplinas e paramos ali. Em vez de aplicar a orientação a serviços ao CMM, tomamos esses princípios e os aplicamos ao paradigma de orientação a serviços, essencialmente criando o roteiro a partir do zero. Esse modelo não enfoca processos porque o foco está nos recursos de TI, não na adoção ou prontidão organizacional. Embora seja possível reconhecer algumas semelhanças conceituais com o CMM, também irá ver que o ESOMM é uma aplicação decididamente diferente do conceito de modelo de maturidade.

Embora possa haver valor agregado por meio de alguns dos outros modelos de maturidade relacionados a serviços que existem por aí, esteja ciente de quanto da implementação CMM isso inclui. O CMM

foi projetado para tratar dos processos de uma empresa relacionados à engenharia de software, o que é um problema decididamente diferente de definir um plano de TI para orientação a serviços. Se adotasse um modelo que misture esses dois, poderia ser esmagador e não produtivo para a empresa. Pior ainda, se adotar um modelo que simplesmente altere o nome de alguns dos níveis e disciplinas do CMM, poderia se estar caminhando na direção errada. Seria como pegar o mapa da Inglaterra com o nome de Londres substituído por Paris e tentar ir para a torre Eiffel — simplesmente não vai dar certo!

Aceitar o modelo de maturidade dirigido a disciplinas é uma decisão. Escolher o roteiro apropriado é outra. Assegure-se de estar adotando uma estratégia que se aplique aos seus objetivos. Nenhum modelo de maturidade se aplica a todos em todas as circunstâncias, tenha certeza de escolher um que o leve ao seu destino.

Figura 3 Recursos das camadas de maturidade

| | Implementação | Consumo | Administração |
|--------------|--|--|--|
| Extensível | Colaboração de serviço Orquestração de serviço | SDKs de serviço Polícia externa | Análítica de negócios Gerenciamento automatizado de políticas |
| Gerenciável | Versionamento Política executável Banco de esquema | SLAs explícitos Portal de serviço Visibilidade de execução | Auditoria Monitoramento Modelo de provisionamento |
| Reproduzível | Esquema comum Blocos de serviço | Autoprovisionamento Capacidade de descoberta do serviço | Gerenciamento de implementação Políticas corporativas |
| Utilizável | Padrões de design Processos de desenvolvimento | Capacidade de teste Contratos explícitos | Modelo de segurança Monitoramento básico |

dar suporte a essa complexidade por meio de métodos de versionamento implícitos e explícitos. No entanto, essa capacidade é um recurso que as ferramentas de desenvolvimento de serviços normais não aproveitam inicialmente e exige certo planejamento e suporte para ser executada de forma consistente.

A perspectiva de consumo da camada gerenciável é onde encontramos as disciplinas de visibilidade de execução, portal de serviços e SLAs explícitos:

Visibilidade de execução. Talvez o fator individual de maior peso na sua capacidade de escalar o uso dos Web Services seja a sua capacidade de dar suporte ao consumo crescente dos serviços, fornecendo recursos elaborados de auto-serviço a clientes existentes e potenciais. Se toda implementação de um serviço exigir uma série de chamadas telefônicas ou e-mails, uma matemática simples irá mostrar que a capacidade de sua equipe de desenvolver serviços será rapidamente consumida pelas necessidades de suporte. Um portal único que forneça acesso às informações certas pode ser um fator imenso para reduzir a demanda de suas equipes de desenvolvimento e suporte. Um portal único que forneça acesso às informações certas pode ser um fator imenso para reduzir a demanda de suas equipes de desenvolvimento e suporte.

Portal de serviços. Essa disciplina trata sobre a possibilidade de recursos serem acessados por portal de auto-serviço. Para tornar os seus clientes mais auto-suficientes, permita que tenham uma visibilidade ampla de como estão utilizando seus serviços. Mais visibilidade provém da capacidade de testar serviços fornecendo um nível de acesso particionado que permite que entendam o que está ocorrendo durante a execução, solucionar problemas e também deduzir se algum ponto não está atendendo suas expectativas. Esse recurso também oferece histórico de atividades para ajudar os clientes a acompanhar a frequência com que o serviço é consumido pelo(s) seu(s) aplicativo(s).

SLA explícitos. Para definir de forma efetiva as expectativas de seus serviços para os clientes, deve fornecer SLA - Service Level Agreement (Contratos de Nível de Serviço) explícitos. Fornecer SLA é uma capacidade que apresenta grandes desafios porque não existe um padrão disponível para comunicar essas informações para Web Services. No entanto, diversas formas de documentação ou comunicação podem ser usadas para compartilhar, de modo efetivo, informações importantes nessa área, incluindo duração do contrato, tempo de processamento previsto das solicitações e qualquer período de inatividade planejada. É importante observar que sem um nível suficiente de gerenciamento e visibilidade internos do serviço, será quase impossível dar suporte a SLA.

A perspectiva de administração do nível gerenciável contém as disciplinas de modelo de fornecimento, monitoramento e auditoria:

Modelo de fornecimento. A etapa seguinte para tornar os serviços auto-suficientes e compatíveis é um processo agilizado de fornecimento para novos usuários. Esse modelo seria usado para dar suporte a usuários externos e internos e incluiria o tratamento de solicitações de acesso, a criação de novas credenciais e o envio de alertas notificando os clientes sobre alterações nos serviços ou no status de suas conta. Esse recurso é construído com base no recurso de fornecimento automático definido na camada reproduzível (2) para fornecer um processo completo para clientes de serviços.

Monitoramento. O monitoramento e a visibilidade na execução dos serviços é um dos aspectos mais desafiadores do suporte a serviços em uma escala corporativa, devido à natureza distribuída dos Web Services. A única maneira de atender a essa funcionalidade de forma efetiva é adotar os seus serviços como um sistema, porque suporte na base de serviço por serviço é uma tarefa ineficiente e frustrante. Embora haja muitas maneiras de estabelecer esse sistema, acreditamos que uma arquitetura de execução comum é a mais eficiente.

Auditoria. O ponto central para realmente confiar nos serviços que fornece na sua empresa é um modelo de auditoria confiável que possa servir como um mecanismo de rastreamento de quem fez o quê e quando. Esse mecanismo não só ajuda a resolver qualquer disputa que possa surgir como também pode desempenhar uma função estratégica em um programa de conformidade da empresa.

Recursos da camada extensível

Vamos agora analisar as disciplinas da perspectiva de implementação da camada extensível (4), orquestração de serviços e colaboração de serviço:

Orquestração de serviços. Para aproveitar os Web Services em processos de negócios bem definidos, deve ter a capacidade de orquestrar. Essa capacidade envolve fluxos de trabalho que podem envolver processos manuais e automatizados, mas permite que defina uma utilização de serviços razoavelmente consistente e reproduzível que pode ser alterada ao longo do tempo.

Colaboração de serviços. Após estabelecer os serviços como funções autônomas que podem ser mantidas e receber suporte, se pode começar a agregá-los para fornecer serviços mais sofisticados e complexos. Chamamos essa agregação de colaboração de serviço. Por meio da colaboração pode definir padrões de execução síncronos e assíncronos que permitem dar suporte a processos inteiros que consistem em serviços existentes. Como exemplo, poderia adotar um serviço que envia um e-mail e um serviço que recupera todas as alterações no cronograma e oferecer um novo serviço que envia um e-mail aos clientes notificando sobre alterações no cronograma.

A perspectiva de consumo da camada extensível inclui as disciplinas de política externa e de SDKs de serviço:

“EMBORA SEJA IMPORTANTE SABER ONDE SE ESTÁ, OBTER O RUMO EXATO É MENOS IMPORTANTE DO QUE IDENTIFICAR AS DISCIPLINAS COM OS QUAIS PRECISA LIDAR PARA CONTINUAR AUMENTANDO O VALOR DA ATIVAÇÃO DE SERVIÇOS NA SUA EMPRESA”

Política externa. Para estabelecer uma estrutura segura de fornecimento de serviços a parceiros e clientes externos, devem ser estabelecidas políticas externas, que incluem não somente políticas de segurança aceitas externamente, mas as políticas que se definiu sobre como os seus serviços

podem ser agrupados com outros serviços de terceiros. Essas políticas incluem identificação de marca, pagamento e condições de uso que são importantes para proteger os interesses de seus negócios.

SDKs de serviço. Criar um canal ou modelo de negócios por meio de Web Services exige levar o nível de suporte aos clientes dos serviços para o nível seguinte. Esse avanço para o nível seguinte incluiria desenvolver e fornecer SDK - Software Development Kits (Kits de Desenvolvimento de Software) predefinidos que permitem aos desenvolvedores fornecer e implementar rapidamente os seus serviços. Esses SDK incluem potencialmente componentes da interface do usuário para uso direto no(s) aplicativo(s) do desenvolvedor e seriam adaptados para os ambientes de desenvolvimento de destino, uma vez que não existe atualmente um padrão para ferramentas de desenvolvimento entre plataformas.

Finalmente, a perspectiva de administração do nível extensível inclui as disciplinas de gerenciamento automatizado de políticas e análise de negócios:

Gerenciamento automatizado de políticas: Ao começar a expor os serviços a um público muito amplo e a usar políticas para definir o seu suporte de interoperabilidade, também começa a receber solicitações de alterações de políticas, por exemplo, alterações nos SLA ou nos requisitos de segurança, devem ser processadas de forma automatizada. Dessa maneira não criam sobrecarga significativa ao dimensionar sua interação com uma base ampla de clientes.

Análise de negócios. O ápice de visibilidade em um sistema de serviços corporativos é o que chamamos de análise de serviço. Essa visibilidade combina dados técnicos de metadados do serviço, logs de atividade e eventos e permite que faça uma referência cruzada com informações da política, dados sobre os negócios e dados sobre o usuário para fornecer visões repletas de recursos de cada dimensão dos seus serviços. Essas informações podem ser usadas para determinar quais serviços estão gerando receita e quais tendências do usuário estão presentes entre os clientes internos e externos dos seus serviços.

Aplicando o modelo

Agora que tivemos essa oportunidade de revisar resumidamente os componentes do ESOMM, espero que já esteja pensando em como ele pode ser aplicado na sua empresa. Para ser claro, a aplicação desse modelo não deve ser considerada como uma atividade para ser realizada de uma vez ou um processo de curto prazo. Pelo contrário, o modelo é mais bem aproveitado como um plano de trabalho que pode ser modificado ao longo do tempo, à medida que cresce o uso dos serviços e a sua experiência.

Infelizmente, a desvantagem de usar o termo maturidade com um modelo é que as pessoas irão querer saber imediatamente em que camada sua empresa está para obter uma noção de seu status ou identidade. Quando isso acontece, não existe uma maneira apropriada de responder a pergunta "em que nível de maturidade está a minha empresa?". Em vez de fazer uma graduação geral baseada em uma das camadas, optamos

Recursos

Carnegie Mellon Software Engineering Institute
Capability Maturity Model for Software (SW-CMM)
www.sei.cmu.edu/cmm/

Microsoft Developer's Network
"Architecting Industry Standards for Service Orientation," Josh Lee
(Microsoft Corporation, 2005)
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbd/html/aisforsoa.asp>

Tabela 1 Uma amostra de classificação ESOMM

| Camada de serviços | Nível de maturidade |
|--------------------|---------------------|
| 4 Extensível | 1 |
| 3 Gerenciável | 1 |
| 2 Reproduzível | 3,5 |
| 1 Utilizável | 3,5 |

pela abordagem de dar a cada camada o seu próprio nível de maturidade, variando de 1 a 5, com incrementos de meio ponto.

O nível de maturidade 1 significa que não teve a oportunidade de lidar com os recursos dessa camada, enquanto 5 significa que dominou todos os recursos definidos nessa camada. Um resultado comum é mostrado na Tabela 1. Essa classificação em cada camada é determinada pela força das disciplinas correspondentes, variando de 1 a 5, com força sendo classificada como 5 e fraqueza como 1. A média dessas qualidades fornecerá uma graduação geral da camada arredondada para o meio ponto mais próximo. Por exemplo, observando a camada 1, se tivermos 3 das 6 disciplinas classificadas como força, 2 classificadas como adequadas e 1 classificada como fraqueza, o nível de maturidade geral será 3,5.

O que não deve esquecer nessa discussão sobre classificação é o fato de que o ESOMM destina a ser aproveitado como um roteiro, mais do que uma ferramenta de avaliação. Embora seja importante saber onde se está, obter o rumo exato é menos importante do que identificar as disciplinas com as quais precisa lidar para continuar avançando o valor dos serviços na sua empresa. Se estiver disposto a fazer algumas perguntas difíceis para si mesmo de uma maneira objetiva em todos os grupos relevantes, poderá obter um bom entendimento dos seus desafios atuais. Se aplicar a estratégia e os objetivos da sua empresa, poderá conseguir identificar quais disciplinas precisará tratar a curto, médio e longo prazo. É tremendamente difícil ter uma conversa sucinta e construtiva sobre a empresa habilitada a serviços. O objetivo aqui é capacitá-lo, como arquiteto de software, com algumas ferramentas e informações necessárias para abordar as questões certas ao ajudar a sua empresa a alcançar o valor real por meio dessa tecnologia poderosa. Esperamos que essa visão geral resumida do ESOMM tenha trazido informações suficientes para começar a aplicar alguns dos conceitos nas suas atividades de ativação de serviços. Cada disciplina poderia facilmente ocupar um artigo inteiro, de forma que os seus comentários sobre qualquer área específica são bem-vindos.

Muitas pessoas contribuíram para o ESOMM e os conceitos aqui apresentados, mas gostaria de agradecer particularmente Wolf Gilbert, Scott Beaudreau, Michel Burger e Byron Howard pelas suas contribuições. •

Sobre o autor

William Oellermann é um arquiteto de soluções do Microsoft Consulting Services, que se dedica a orientar e dar suporte a oportunidades de ativação de serviços estratégicos no setor de comunicações. William juntou-se à Microsoft em fevereiro de 2004, tendo trabalhado anteriormente por mais de dez anos como arquiteto de software, exercendo funções corporativas e de consultoria na Avanade, na Ericsson e na Micrografx. Ele tem sido um líder importante na área de Web Services há mais de seis anos como orador freqüente em conferências, autor de vários artigos em revistas de comércio e autor de *Architecting Web Services* (APress, 2001). Quando não está se atualizando com as tendências e desenvolvimentos mais recentes da ativação de serviços, William se diverte ajudando sua esposa a cuidar das duas filhas muito ativas.



Modelagem orientada a serviços para sistemas conectados – Parte 1

por Arvindra Sehmi e Beat Schwegler

Resumo

Como arquitetos, podemos adotar uma nova maneira de pensar para essencialmente forçar uma consideração explícita sobre artefatos de modelo de serviços nos nossos processos de design, o que nos ajuda a identificar os artefatos corretamente e, no nível de abstração certo, atender e alinhar as necessidades de negócios de nossas empresas. Aqui oferecemos uma abordagem em três partes para modelagem de sistemas conectados e orientados a serviços de uma maneira que promove um alinhamento estreito entre a solução de TI e as necessidades de negócio. Começaremos examinando a perspectiva atual do pensamento orientado a serviços e explicaremos como o pensamento atual e a conceitualização equivocada da orientação a serviços resultou em muitas falhas e, de modo geral, em níveis insatisfatórios de retorno do investimento. Em seguida, examinaremos os benefícios de inserir um modelo de serviços entre os modelos de negócios e de tecnologia convencionais que são familiares à maioria dos arquitetos e discutiremos a metodologia Microsoft Motion e o mapeamento de habilidades para identificar habilidades de negócios que possam ser mapeadas para serviços. A Parte 2 desta discussão virá no Jornal 8 e mostrará como implementar esses serviços mapeados.

A orientação a serviços é prioritária na agenda de muitas empresas. O objetivo geralmente é adotar a orientação a serviços para permitir um alinhamento melhor entre os requisitos de negócios e os serviços de TI dentro da empresa e, no final, permitir e dar suporte a negócios mais ágeis. Hoje, no entanto, muitas empresas não estão conseguindo alinhar os serviços de TI com os requisitos de negócios de forma estreita o suficiente e, como resultado, não estão conseguindo ver o retorno do investimento que se esperava da orientação a serviços.

Ao tratar da orientação a serviços, as empresas estão tentando responder estas perguntas: Como fazemos para não cometer com as SOA - Service Oriented Architecture (Arquiteturas Orientadas a Serviços) os mesmos enganos cometidos por promissoras iniciativas anteriores? Como fazemos para garantir que a arquitetura de implementação escolhida relaciona-se ao estado real ou desejado dos negócios? Como fazemos

para garantir uma solução sustentável que possa reagir com a natureza de transformação dinâmica da empresa? Em outras palavras, como fazemos para ativar e sustentar uma empresa ágil? Como fazemos para migrar para esse novo modelo de uma maneira elegante e a um ritmo que possamos controlar? Como podemos fazer essa mudança com uma boa percepção de onde podemos agregar o maior valor para a empresa desde o início?

Uma premissa importante que propomos aqui é que para criar sistemas orientados a serviços bem-sucedidos é necessário modificar a maneira como se pensa sobre orientação a serviços e sobre SOA, e a maneira como se constroem os sistemas. Falhar na abordagem correta do problema e desenvolver a maneira errada de pensar desde o início são indiscutivelmente as causas de muitos problemas. Sem a maneira certa de pensar, é improvável obter os benefícios esperados da abordagem de orientação a serviços.

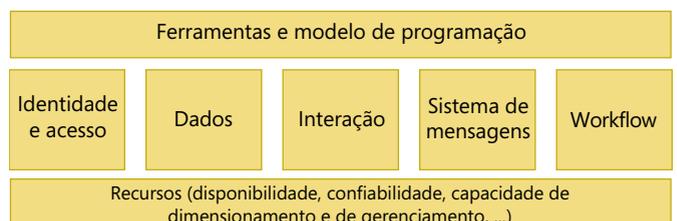
Uma parte essencial da maneira de pensar é ser capaz de conceituar a sua empresa da maneira correta e, ao fazê-lo, pode chegar a serviços sólidos e bem alinhados que mapeiam para as habilidades exigidas pelos seus processos de negócios. Tendo identificado os serviços certos, pode avançar para a implementação desses serviços utilizando as tecnologias de sua preferência, como as fornecidas pela plataforma de aplicativos Microsoft. Vamos agora analisar um modelo que pode ser aplicado a sistemas conectados e orientados a serviços para promover um alinhamento estreito entre as soluções de TI e as necessidades da empresa.

Cinco pilares dos sistemas conectados

O sistema de mensagens é importante para a orientação a serviços, mas não é o único aspecto necessário para modelar serviços, existindo também diversos outros pilares. Embora o sistema de mensagens permita a conexão com sistemas distintos e forneça a estrutura base de conexão, existe uma série de outros problemas importantes que precisam ser tratados. Vamos dar uma olhada mais de perto nos cinco pilares centrais dos sistemas conectados (consulte a Figura 1):

- Identidade e acesso. Esse pilar se relaciona à noção de identidade federada (sign-on único na Web) e autorização baseada em política.

Figura 1 Os cinco pilares dos sistemas conectados



Ele trata do gerenciamento do relacionamento de confiança e da forma como deve ser controlado o acesso aos sistemas conectados. Normas de conformidade e governança são outros fatores importantes que devem receber consideração adequada nesse pilar.

- **Dados.** Esse pilar trata da agregação de entidades e está relacionada à construção de uma fonte única e coerente de uma entidade de negócios específica, como um cliente, embora os dados do cliente possam estar duplicados muitas vezes em vários sistemas.
- **Interação.** Esse pilar é dedicado ao consumo humano de serviços – por exemplo, por meio do fornecimento de interfaces do usuário compostas e repletas de funcionalidades, com recursos on-line e off-line. Interação na extremidade das redes por meio da Web, mecanismos ponto a ponto e dispositivos móveis também são incluídos aqui.
- **Sistema de mensagens.** Esse pilar fornece a estrutura de base dos sistemas conectados e precisa dar suporte a sistemas de mensagens seguras, confiáveis e ordenadas.
- **Fluxo de trabalho.** Esse pilar trata do fluxo de trabalho ou da automação do processo de negócios, externo aos serviços. Ele é relacionado à orquestração dos processos de negócios e também a outros aspectos, como gerenciamento da interação com o usuário, processos especialistas e gerenciamento de exceções.

Embora o modelo de 3 partes discutido aqui esteja em grande parte relacionado ao pilar do sistema de mensagens, também iremos considerar determinados aspectos dos outros pilares. Antes de examinar uma nova abordagem para a modelagem de sistemas conectados, é importante reconhecer que muitos dos problemas derivam de uma maneira de pensar convencional que precisa mudar para compreender os benefícios reais da orientação a serviços.

A perspectiva antiga da orientação a serviços

A abordagem tradicional de arquitetura de soluções e construção é obter um conjunto de requisitos de negócios e a partir deles derivar um modelo de tecnologia que normalmente envolve orientação a objetos, tecnologia de componentes e, talvez, sistemas mainframe. Não trabalhar em uma proximidade suficiente com os negócios, pode normalmente ocasionar uma grande lacuna entre os negócios e as soluções de TI fornecidas. Em vez de voltado para dentro, o desenvolvimento precisa tornar-se uma disciplina de engenharia exteriorizada.

Outro grande problema é que embora o pessoal de negócio possam ser muito bons no que fazem, não é seu trabalho descrever sua atividade para pessoas que não estão na sua área, o que acontece quando conversam com a equipe de TI sobre os requisitos. Historicamente, a equipe de negócios não teve grandes ferramentas para ajudá-las a criar requisitos claros e objetivos. Por exemplo, mapas do processo são visualizações subjetivas que não conseguem expor conteúdo e métrica muito objetivos, e a TI precisa de visualizações objetivas.

Um outro problema ocorre com frequência devido à maneira como os departamentos de TI são organizados. Às vezes são de acordo com tecnologias específicas (por exemplo, mainframe, aplicativos da Internet, aplicativos da intranet, etc.) e, em outras, são organizados em torno de limites organizacionais rígidos e limites departamentais. Nas duas abordagens, essa organização conduz rapidamente a silos, e a maneira como as informações são transferidas é em grande parte por meio de troca de documentos.

Figura 2 Um modelo de sistema clássico



Essa estrutura organizacional aquém do ideal também torna muito difícil criar sistemas que abrangem várias tecnologias e é muito complicado determinar o custo real de uma solução, devido ao envolvimento de muitas pessoas de diversos departamentos diferentes.

Para ajudar a resolver esse problema, os departamentos de TI precisam tornar mais voltados para o exterior e seu relacionamento mais próximo aos negócios, sendo necessária uma nova maneira de pensar sobre orientação a serviços e de modelar sistemas conectados. Idealmente, o departamento de TI deve fornecer somente os aspectos de infra-estrutura da solução e o desenvolvimento da solução ser alinhado, com mais força, aos grupos relevantes da empresa. A orientação a serviços e o seu respectivo processo de desenvolvimento não ativarão sozinho empresas orientadas a serviços. Ela precisa ser parte de uma estratégia global que organiza a transição em direção a TI orientada a serviços.

Modelagem clássica de sistemas

As atuais abordagens de modelagem geralmente enfocam modelos de negócios e de tecnologia (consulte a Figura 2). Idealmente, existem correspondência e alinhamento precisos entre o modelo de negócios e o de tecnologia, mas na prática esse relacionamento geralmente não se realiza. Departamentos de TI auto-centrados, que não trabalham em proximidade suficiente com a empresa e de forma efetiva, constituem uma razão chave para isso. Embora os modelos de domínio de entidades sejam

“PARA CRIAR SISTEMAS ORIENTADOS A SERVIÇOS BEM-SUCEDIDOS É NECESSÁRIO MODIFICAR A MANEIRA COMO SE PENSA SOBRE ORIENTAÇÃO A SERVIÇOS”

usados com frequência em uma camada acima do modelo de tecnologia para ajudar a tratar esse problema, essa abordagem tende a não funcionar bem principalmente porque eles servem a duas finalidades conflitantes: Implementação interna e exposição funcional da empresa. Claramente o alinhamento real entre os modelos de negócios e de tecnologia raramente é alcançado porque a lacuna entre as duas perspectivas é simplesmente muito grande. Desse modo, o que podemos fazer para aproximar esse relacionamento? Considere estes quatro ingredientes essenciais:

Profissionais de TI voltados para fora. Os profissionais de TI precisam estar mais voltados para fora da área de TI e relacionarem-se de forma mais profunda com o lado de negócios de sua empresa. Embora não seja necessário que se tornem especialistas nos negócios, eles precisam de uma linguagem objetiva que lhes permita conversar com a área de negócios sobre negócios. Os arquitetos, em particular, constituem um canal de comunicação entre essa área e os departamentos de TI, e precisam assegurar que os requisitos de negócios e as soluções sejam os mais interdependentes possíveis. Pode ser conseguida trabalhando bem próximo da área de negócios para garantir que os contratos que o departamento de TI oferece à empresa estejam muito mais alinhados com os requisitos de negócios. Em nossa opinião, adotar essa perspectiva talvez seja a melhor (se não a única)

maneira de chegar à granularidade certa para os serviços que criará com uma abordagem orientada a serviços.

No nosso linguajar de hoje, falamos sobre “expor a arquitetura da empresa”. Como as funcionalidades de uma empresa em um mercado determinado são semelhantes ou até mesmo idênticos, o pessoal de negócios e de TI pode utilizar um conjunto de perguntas padrão para obter informações relevantes sobre a arquitetura da empresa para reunir requisitos. Ao utilizar um conjunto de perguntas padrão, até mesmo não especialistas em um domínio da empresa podem conduzir uma conversa muito útil sobre os requisitos de negócios e expor informações importantes sobre métrica, desempenho, maturidade, interligação, controle e conformidade. Como a pessoa do lado de negócios está respondendo perguntas de um arquiteto, ele na verdade ajuda a expor uma visão que a empresa pode ainda não ter.

Entendendo a mudança na empresa. Embora não seja nossa sugestão que toda a TI se torne “orientada a negócios”, pode ser de grande valor para a TI entender como a empresa evoluiu no passado. Esse conhecimento influencia as decisões de implementação no fornecimento de pontos de extensão, versionamento, recursos necessários e problemas de gerenciamento de projeto como quantidade e comprimento de iterações e gerenciamento de implantação, incluindo: Principal, secundária e delta esperadas em um determinado intervalo de tempo.

Uma infra-estrutura operacional comum. É necessária para dar suporte a aplicativos de negócios que fornecem práticas inter-empresariais e globais. Construir modelos íntegros de como operar e gerenciar a infra-estrutura e implantar aplicativos nela é fundamental para o sucesso. Aplicativos íntegros permitem que os responsáveis por decisões da empresa obtenham uma compreensão importante das informações que estão circulando por eles, o que finalmente permite aos indivíduos colaborar e tomar decisões para conduzir a empresa de uma forma mais efetiva.

Padrões de tecnologia de Web Services. Permitem que aplicativos sejam conectados. No final, o valor de conectar os sistemas leva as práticas de negócios mais eficientes e efetivas.

Uma nova perspectiva de orientação a serviços

Como arquitetos, podemos garantir que os contratos que o departamento de TI oferece à área de negócios estão bem alinhados com os requisitos da empresa? A introdução de um modelo de serviços entre o modelo de negócios e o modelo de tecnologia é um fator importante que pode ajudar a alcançar esse objetivo.

O modelo de orientação a serviços em três partes interpõe um modelo de serviços entre os modelos de negócios e de tecnologia de sistema clássico. O modelo de orientação a serviços em três partes interpõe um modelo de serviços entre os modelos de negócios e de tecnologia de

“É NO MODELO DE SERVIÇOS QUE PODE CAPTURAR A SEMÂNTICA NECESSÁRIA PARA EXPRESSAR OS SERVIÇOS QUE TORNAM A SUA SOLUÇÃO MAIS FLEXÍVEL E MAIS VOLTADA PARA FORA OU PARA OS NEGÓCIOS”

sistema clássico. A introdução do modelo de serviços apresenta várias desvantagens. É no modelo de serviços que pode capturar a semântica necessária para expressar os serviços que tornam a sua solução mais flexível e mais voltada para os negócios. O modelo de serviços fornece um lugar lógico para definir os contratos que garantem que o lado de negócios da sua empresa esteja alinhado com o lado da TI, de uma perspectiva de requisitos. Ao introduzir o modelo de serviços, os arquitetos são forçados a considerar explicitamente artefatos de modelo de serviços no processo de design.

O modelo de serviços ajuda os arquitetos a descobrir artefatos no nível certo de abstração para atender e alinhar as necessidades da empresa. Ele permite, além disso, que o analista de negócios tenha propriedade parcial do processo de design e alcance melhor rastreabilidade dos requisitos de negócios. Por último, as pessoas de negócio já conhecem a palavra serviço. Por exemplo, no contexto de terceirização, um SLA - Service Level Agreement (Contrato de Nível de Serviço) é bem entendido. Funções internas (não terceirizadas) são entendidas de forma semelhante, embora (geralmente) estejam associadas a uma SLE - Service Level Expectation (Expectativa de Nível de Serviço) menos formal e menos contratual. Em

Figura 3 O modelo de orientação a serviços em três partes

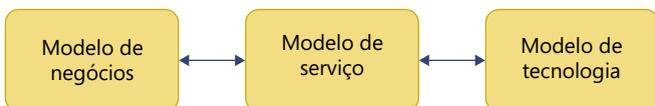


Figura 4

1) Representação da função de negócios, 2) Modelando níveis de serviços e 3) Implementando os modelos em diferentes níveis de abstração

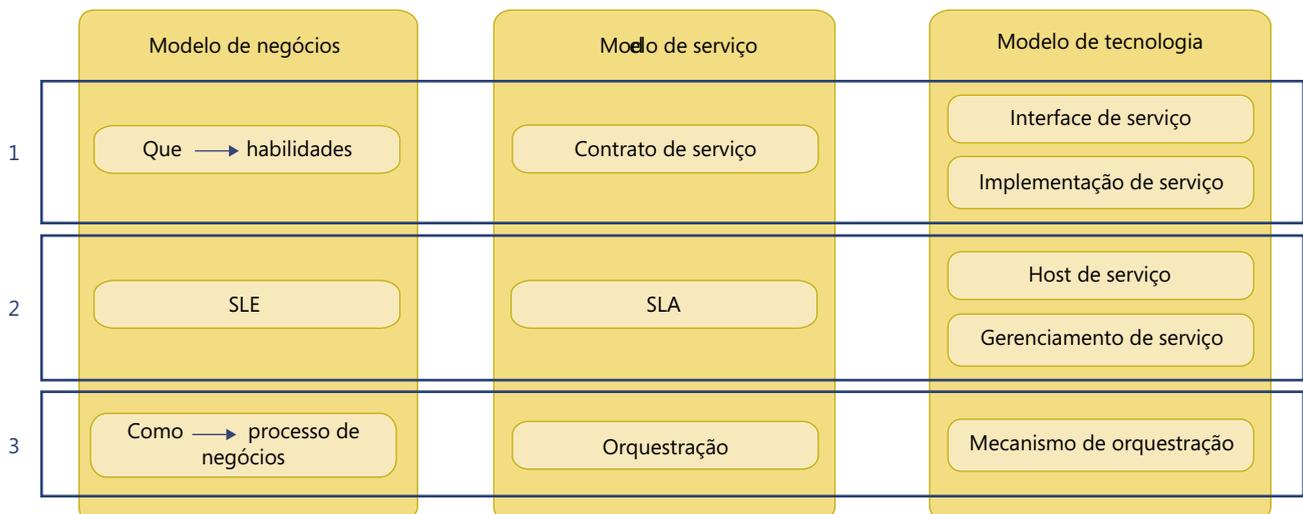


Tabela 1 Os requisitos dos quatro princípios e artefatos do modelo

| Requisito inicial | Modelo de negócios | Modelo de serviço | Modelo de tecnologia |
|-------------------|--|--|---|
| Limites | Recursos funcionais claramente demarcados (mapa de habilidades na arquitetura de negócios) | Interfaces de serviço consumíveis externamente (definição de ponto de conectividade) | Implementação de extremidade de serviço explícita (ponto de extremidade de serviço hospedado fisicamente) |
| Autonomia | Habilidades externas e internas (definições de recursos centrais, operacionais e ambientais) | Serviços flexíveis e intercambiáveis (considerações espaciais, temporais, técnicas e de design operacional) | Independência de implementação (independência de implementações de dados e de comportamento) |
| Contrato | Descrições de tarefa e processo (unidades lógicas de definição de trabalho e de fluxo de trabalho) | Definições de dados, interface e orquestração (XSD de dados e mensagens, WSDL e protocolos de interação de serviços) | Implementações de dados compartilháveis, interface e orquestração (WSDL, XSD e BPEL) |
| Política | Definições de governança, SLE e SLA (necessidades, preferências, expectativas e contratos) | SLA consumíveis externamente (definições de compatibilidade: afirmações “eu posso,” “eu preciso” e “eu prefiro”) | Dreocupações operacionais separadas (As principais preocupações incluem segurança, confiabilidade, transações e Perfil WSI) |

vez de conversar sobre serviços e orientação a serviços, é muito melhor conversar sobre SLE porque as pessoas da empresa entenderão seu valor.

Um aspecto interessante do modelo em três partes é que muitos aspectos dos quatro princípios da orientação a serviços, desenvolvidos originalmente no nível do modelo de tecnologia, também se aplicam aos níveis de negócios e de serviço.

O desenvolvimento orientado a serviços baseia-se em quatro princípios fundamentais que foram produzidos originalmente pela equipe da WCF - Windows Communication Foundation, (conhecido anteriormente como “Indigo”) na Microsoft. A equipe desenvolveu originalmente os princípios para descrever como usar o modelo de programação WCF para criar Web Services mais refinados que fossem mais relevantes para o modelo de tecnologia do que para os outros dois modelos. No entanto, percebemos que eles poderiam emprestar uma perspectiva importante ao modelo em três partes quando considerados como princípios de serviços (e negócios) aplicáveis nos respectivos níveis do modelo.

Princípios da orientação a serviços

Os quatro princípios são: os limites são explícitos; os serviços são autônomos; os serviços compartilham esquema e contrato, não classes; e a compatibilidade de serviços é determinada com base em política. Vamos analisar cada um deles detalhadamente.

Os limites são explícitos. A orientação a serviços baseia-se em um modelo de transmissão de mensagens explícitas e cruzar um limite é considerado um ato explícito. Para usar a analogia de entrar em um país estrangeiro, o ato de passar de um país para outro é um ato explícito. De uma perspectiva da empresa, é igualmente importante entender e definir

“UM BENEFÍCIO IMPORTANTE DA CONCENTRAÇÃO NOS RECURSOS DE NEGÓCIOS É QUE ENQUANTO AS ESTRUTURAS ORGANIZACIONAIS E OS FLUXOS DE PROCESSO SE ALTERAM, AS HABILIDADES ESSENCIAIS DAS EMPRESAS TENDEM A PERMANECER CONSTANTES AO LONGO DO TEMPO”

os limites organizacionais e os limites das funções dentro da sua empresa para identificar recursos claramente demarcados. Da perspectiva de um modelo de serviços, esse princípio exige interfaces de serviço consumíveis externamente.

Os serviços são autônomos. As alterações feitas em um serviço não devem causar nenhum impacto em outro serviço. Um serviço deve poder ser compilado sem causar impacto negativo nos clientes do serviço. Para continuar a analogia com país estrangeiro, um país pode introduzir um novo sistema de impostos independente de um outro país porque eles são autônomos. Do mesmo modo, no nível da empresa, as alterações que ocorrem internamente em um processo de negócios específico não devem causar impacto em outras partes que não dependem desse processo ou serviço. Autonomia no nível da empresa também significa que os serviços individuais podem ser hospedados de forma terceirizada sem qualquer impacto global no processo de negócios. Autonomia no nível do modelo de serviços exige serviços flexíveis e intercambiáveis. Autonomia no nível do modelo de tecnologia exige independência de implementação.

Os serviços compartilham esquema e contrato, não classes. Esse princípio está relacionado em grande parte a serviços que não expõem suas partes internas. Por exemplo, um país publica as regras exigidas para preencher os formulários de visto, mas não fornece os recursos internos para preenchê-los. Como outro exemplo, considere a interoperabilidade entre J2EE e .NET. Ela não será possível se tentar intercambiar tipos específicos da plataforma, para isso é necessário um esquema ou metamodelo intermediário.

A compatibilidade do serviço é determinada com base em política. No nível do modelo de negócios, esse princípio refere-se essencialmente a controle e SLA e suas definições. A política definida no metanível descreve o recurso de semântica de um serviço com base em um conjunto de instruções explícitas das suas funcionalidades.

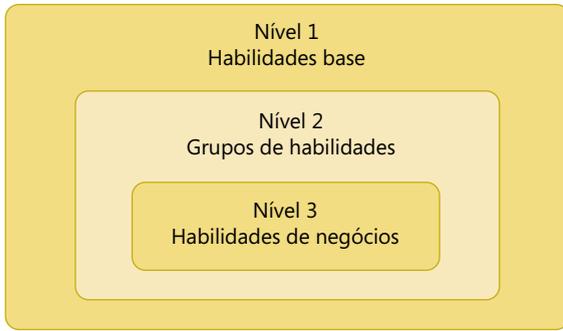
Cada um dos princípios desempenha uma função importante no modelo em três partes e influencia de maneiras diferentes os modelos de negócios, de serviço e de tecnologia. A Tabela 1 resume as necessidades e os artefatos de cada modelo em relação aos quatro princípios. Vamos agora ver o que cada modelo define e como se deve abordar sua criação.

Definição de modelos

O modelo de orientação a serviços em três partes deve poder representar aquilo que uma empresa específica é capaz (seus recursos funcionais principais) e não como a função é executada. Discutiremos brevemente o que é um recurso de negócios. A maneira como a função de negócios em uma empresa é representada nos diferentes níveis de abstração pelos três modelos está mostrada na Figura 4.

Funções de negócios. Da perspectiva do modelo de negócios, a habilidade de uma empresa é expressa identificando as funções de negócios. Identificá-las é uma tarefa importante ao desenvolver o modelo

Figura 5 A framework de base para construir um modelo de habilidades



de negócios. Os recursos e as abordagens para determinar essas habilidades serão discutidos mais adiante.

Os recursos funcionais apresentam um relacionamento próximo com a descrição de contrato de serviço definida no modelo de serviços. Descendo o nível de abstração, esse relacionamento se converte em uma descrição das interfaces de serviço e, no final, das implementações de serviço no modelo de tecnologia. (Observe que embora exista uma forte correlação entre um recurso e um serviço, não é necessariamente um para um.)

Níveis de serviço. Para garantir compatibilidade e consistência entre os modelos, foi introduzida a noção de níveis de serviço. No nível do modelo de negócios, os SLE definem os níveis de serviço esperados. Essas são expectativas de negócios. Depois de construído um sistema de software para fornecer o serviço, o SLE é convertido em SLA. No nível do modelo de tecnologia, os SLA causam impacto na maneira como se hospeda e se gerencia o serviço.

Por exemplo, se as suas expectativas de negócios ou SLA forem muito exigentes, será inadequado hospedar o seu serviço em um servidor com processador único que não ofereça nenhuma redundância. As expectativas e os contratos orientam os requisitos de disponibilidade, confiabilidade, segurança e capacidade de gerenciamento da sua solução. De modo semelhante, se a empresa considerar o sistema como de missão crítica, esse status acrescenta um ônus adicional nos recursos de gerenciamento que estabeleceu para a sua solução. A maneira como cada nível de serviço é representado em cada um dos três modelos está mostrada na Figura 4.

Na perspectiva do modelo de negócios a expectativa pode ser fornecida com informações qualitativas e quantitativas. A fidelidade das informações qualitativas para quantitativas torna-se mais específica à medida que se desloca em direção ao modelo de tecnologia. Efetivamente perde informações como parte das suas descrições. No entanto, o benefício de ter um conjunto de modelos conectados, particularmente se eles puderem ser descritos no metanível, significa que ainda se tem a capacidade de entender qual é a origem dos requisitos e o que está relacionado a quê.

Implementação. Precisa ser capaz de descrever como irá implementar os modelos e expressá-los em detalhes. No modelo de negócios, precisa expressar os processos de negócio externos que identificou e, em seguida, precisa de um mecanismo de orquestração para descrever os processos de negócio no modelo de serviços. Quando implementa as orquestrações, precisa de alguma tecnologia de workflow ou mecanismo de orquestração. Por exemplo, na plataforma Microsoft pode utilizar os serviços de orquestração fornecidos pelo Microsoft BizTalk Server (consulte a Figura 4). Embora possamos insinuar fluxos de trabalho automatizados ao falar sobre orquestração, não excluímos fluxos de trabalho manuais ou individuais

como forma de implementação no modelo de tecnologia, o que se torna óbvio durante a definição de processo do modelo de negócios.

Criando modelos de negócios

Para criar um modelo de negócios eficiente, precisa ser capaz de conceituar a empresa e identificar as suas funções de negócios principais. Isso permite que chegue a serviços bem alinhados que são mapeados de perto para os seus requisitos de negócios.

Mapeamento de habilidades. Essa técnica foi desenvolvida e refinada pela Microsoft para ajudar a conceituar a maneira como uma empresa funciona e ajudar a criar modelos de negócios. Uma habilidade de negócios modela o que uma função de negócios individual faz. Ela não está preocupada com a maneira como a função de negócios é conseguida, mas sim com o seu comportamento visível externamente e o seu nível de desempenho esperado (ou seja, seus resultados). Um benefício importante da concentração nos recursos de negócios é que, enquanto as estruturas organizacionais e os fluxos de processo se alteram, os recursos essenciais das empresas tendem a permanecer constantes ao longo do tempo. Um recurso de negócios abstrai e encapsula em um bloco de construção simples as pessoas, o processo, os procedimentos e a tecnologia associados a uma função de negócios determinada. A decomposição da empresa em habilidades fornece a separação em alto nível dos contratos de serviço base.

Pagar fornecedor, enviar produto e gerar fatura são exemplos de habilidades de negócios. Em um alto nível de abstração, uma habilidade de negócios é essencialmente uma caixa preta com determinados paralelos para serviços. Por exemplo, ambos possuem conexões que são importantes e relacionadas a seus trabalhos internos, porém separadas.

Desenvolvendo modelos de habilidades. É uma hierarquia aninhada de funções de negócios que permite modelar a empresa como uma rede estruturada, em oposição a uma rede integrada fisicamente. Um modelo de habilidades de negócios é um diagrama que descreve a rede de funções usada pela empresa. A estrutura usada para construir um modelo de recursos em níveis variados de abstração é mostrada na Figura 5.

“MOTION DESCREVE COMO EXTRAIR E DOCUMENTAR INFORMAÇÕES ESTRUTURAIS E ORIENTADAS A PROCESSO JUNTO COM HABILIDADES INDIVIDUAIS”

O nível de abstração com que pode visualizar uma habilidade de negócios varia. As de nível 1 são a base comum à maioria das empresas, independente do setor de mercado. Os níveis 2 e 3 (e além) fornecem níveis adicionais de detalhe de negócios. Antes de examinar cada nível com mais detalhe, observe que não é necessário decompor todas as habilidades para o mesmo nível de refinamento; em vez disso, é necessário concentrar-se nas mais relevantes para o problema ou a área ou a empresa que estiver sendo considerado.

Habilidades - nível 1. Tendo estudado empresas em muitos mercados, a Microsoft descobriu que em um alto nível de abstração, as empresas na maioria dos mercados exibem cinco habilidades principais junto com um conjunto de funções operacionais e ambientais, coletivamente denominados de habilidades de base (consulte a Figura 6).

As cinco habilidades base comuns à maioria das empresas são desenvolver produtos e serviços, gerar demanda, entregar produtos e serviços, planejar e gerenciar a empresa, e colaboração. O quinto recurso,

colaboração, é o processo que orquestra e coordena todas as demais funções da empresa. É chamada de colaboração porque o processo pode ser automático ou manual. O processo é essencialmente uma orquestração sobre funções.

As habilidades base também incluem funções operacionais, que são os itens dentro do limite físico da empresa, e recursos ambientais, que são todas as demais pessoas e organizações que interagem com a empresa e que estão fora do limite físico dela. Essas entidades poderão incluir clientes, parceiros,

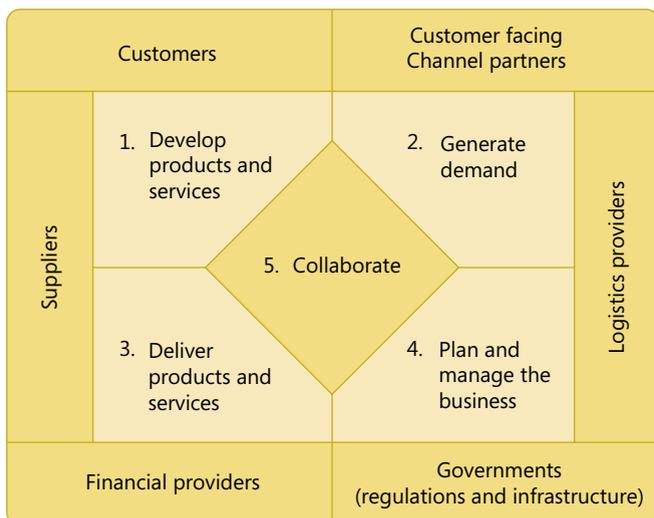
“AS CINCO HABILIDADES BASE, COMUNS À MAIORIA DAS EMPRESAS, SÃO DESENVOLVER PRODUTOS E SERVIÇOS, GERAR DEMANDA, ENTREGAR PRODUTOS E SERVIÇOS, PLANEJAR E GERENCIAR A EMPRESA, E COLABORAÇÃO”

provedores de serviços e autoridades de regulamentação. Elas são significativas porque todas influenciam a maneira como conduz a sua empresa.

Observe que o limite da empresa não é o mesmo que o limite físico da entidade corporativa. Por exemplo, uma entidade terceirizada contínua parte da arquitetura da sua empresa, embora o trabalho seja realizado por pessoas fora da sua empresa. Existem alguns exemplos interessantes de trabalho que está sendo mudado para fora do limite da empresa, como check-in no aeroporto, uma função que agora é realizada com frequência pelo cliente; é o mesmo recurso e tem o mesmo SLE. No entanto, agora existe tecnologia diferente e isso tem permitido diferentes pessoas realizar o trabalho, nesse caso o cliente. Essa visão holística do ecossistema de habilidades inteiro permite às organizações observar um leque de opções muito mais amplo que pode informar mudanças específicas que elas fazem em seus negócios.

Grupos de habilidade - nível 2. Os grupos de recursos fornecem o nível seguinte de detalhe no modelo de recursos. O nível 2 é onde começa a análise inicial porque é o que introduz SLE e impedimentos e restrições entre uma habilidade e outra e propriedade organizacional. Também identifica as entradas, saídas, funções de suporte e funções de controle.

Figura 6 Habilidades base, comuns à maioria dos negócios



Esse exemplo começa com a habilidade base 3, entregar produtos e serviços, e mostra uma hierarquia de funções aninhada no nível 3..n:

3. Entregar produtos e serviços
 - 3.1 Fornecer serviço
 - 3.2 Planejamento avançado
 - 3.3 Licitação
 - 3.3.1 Gerenciamento de contrato de terceirização e de fornecedor
 - 3.3.2 Compras
 - 3.3.2.1 Solicitar recursos
 - 3.3.2.2 Adquirir/comprar recursos
 - 3.3.2.3 Gerenciar fornecedores
 - 3.3.3 Recebimento de bens indiretos/de capital
 - 3.4 Produzir produto
 - 3.5 Logística

Observe como as habilidades estão sempre rotuladas dentro do seu grupo pai apropriado. Um grupo de funções geralmente é um nível inicial importante para fazer análise porque é no nível do grupo que SLE, impedimentos, restrições e propriedade organizacional podem ser abstraídos inicialmente e tornam-se acionáveis.

Habilidades de negócios - nível 3. Os grupos de habilidades são subsequentemente decompostos em habilidades de negócios. Ao mapear as funções individuais da empresa em um modelo de habilidades (consulte a Figura 6), acaba desenvolvendo uma hierarquia aninhada de funções de negócios. As que residem no nível 3 e abaixo são os blocos de construção do modelo.

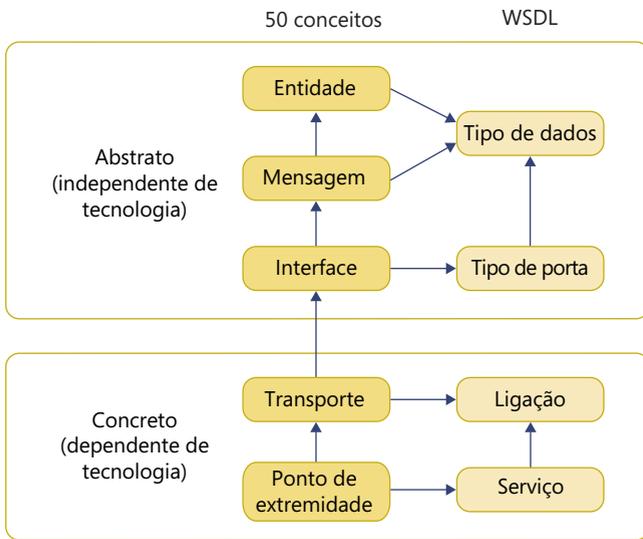
As habilidades de negócios podem ser decompostas de forma mais granular, por exemplo, quando atributos mais detalhados precisam ser definidos. Dentro da análise pode decompor-las para níveis bem detalhados (nível 4 e além) e agregar outras funções no nível 3. Não é necessário decompor todas as habilidades para o mesmo nível. Do mesmo modo, não é necessário modelar a empresa inteira. Pode selecionar áreas da sua empresa a serem modeladas com base nos objetivos de negócios e requisitos de projeto atuais.

Para cada habilidade que for identificado no final, define um conjunto de atributos para descrevê-la e documentá-la. Alguns dos atributos principais que precisa capturar para cada habilidade incluem: Quem é o proprietário? Quem é o seu cliente? Quais são as entradas e saídas? Quais são os mecanismos de tratamento de exceções e notificação de exceções necessárias? Quais são os requisitos de desempenho (passado, atual e futuro)? Existem implicações e governança e/ou conformidade? O desempenho da habilidade causa impacto direto no desempenho do seu recurso pai? No seu departamento? Na organização inteira? O que leva a função a ter tal desempenho? Pessoas? Processo? TI? Uma combinação dessas causas? E a habilidade é parte da razão pela quais os clientes, parceiros ou fornecedores fazem negócios com a empresa?

A captura de atributos como esses ajudam a definir o SLE no nível de negócios e os SLA no nível de serviços. Essa descrição elaborada pode ser encaminhada para as equipes de desenvolvimento que utilizarão as informações para ajudar a selecionar as tecnologias de implementação, as topologias de implantação e os hosts apropriados com base no SLE orientado a negócios.

Habilidades para serviços. Um dos principais benefícios da abordagem de modelagem de habilidades é que ela permite identificar os elementos estáveis da empresa em torno dos quais modela a arquitetura e fornece

Figura 7 Artefatos de contrato definidos pelo modelo de serviços



“AS HABILIDADES DE NEGÓCIOS PODEM SER DECOMPOSTOS EM FUNÇÕES DE NEGÓCIOS MAIS GRANULARES, POR EXEMPLO, QUANDO ATRIBUTOS MAIS DETALHADOS PRECISAM SER DEFINIDOS”

uma camada crítica que alinha de perto a definição de serviços de sistemas conectados. Pode traçar limites em torno dessas habilidades — por exemplo, no nível de grupo — e expressar cada uma delas com contratos. Em seguida, expô-las como serviços e criar uma orquestração por meio desses recursos utilizando os contratos de serviços.

Embora o mapeamento de habilidades seja uma maneira eficiente de analisar o estado atual da sua empresa, também pode construir um mapa para um estado futuro desejado da sua empresa e desenvolver uma maneira de transformar a sua empresa para aprimorar a agilidade. Como exemplo, considere a terceirização. Ao dividir as habilidade em aquelas que são centrais para a sua empresa e aquelas que não são, por meio de um processo de reengenharia pode decidir terceirizar as não centrais para outras empresas. O importante na perspectiva da empresa é que a arquitetura dê suporte a alterações no processo de negócios e não a alterações das habilidades, que devem permanecer intactas.

Metodologia Motion

A Microsoft desenvolveu um método simples orientado a projetos chamado Motion para expor a arquitetura da sua empresa por meio da construção de mapas de habilidades. Motion é uma metodologia para organizar, medir e avaliar as funções de negócios. Os quatro elementos são ferramentas e medidas, metodologia prescritiva, técnicas de mapeamento de habilidades de negócios e modelo com patente provisória.

A metodologia descreve o que a empresa faz na forma de recursos de habilidades em oposição a como a empresa faz na forma de pessoas, processo esta tecnologia. Motion ajuda a expor a arquitetura de negócios de uma organização e isola os elementos da empresa que definem o modelo de negócios e orientam o desempenho e a diferenciação. Modelo de negócios e arquitetura de negócios são separáveis e Motion ajuda

as organizações a expor os dois, permitindo que gerenciem melhor a mudança.

Por exemplo, considere duas empresas no mercado de varejo. Elas possuem as mesmas habilidades, mas uma poderá obter lucros da venda de produtos em grande quantidade e a outra obtém todo o seu dinheiro

“O BENEFÍCIO PRINCIPAL DE DOCUMENTAR AS HABILIDADES DE UMA MANEIRA DETALHADA E CONSISTENTE É QUE ISSO PERMITE UM ENTENDIMENTO SÓLIDO DO SLE QUE A EMPRESA TEM PARA CADA UMA DAS FUNÇÕES DE NEGÓCIOS”

Como o Motion funciona?

A metodologia Microsoft Motion não incentiva concentrar-se na decomposição da estrutura organizacional, das ferramentas e dos processos diretamente no modelo de habilidades. Em vez disso, ela fornece uma maneira efetiva de ajudá-lo a chegar às soluções de implementação apropriadas. A metodologia define estas quatro etapas de alto nível:

- *Estabelecer o contexto do projeto.* Você começa documentando os objetivos do projeto e, em seguida, gera um mapa de recursos de base e correlaciona os objetivos do seu projeto com as habilidades base.
- *Entender os negócios.* Existem dois esforços paralelos. Capturar as visões de negócios atuais, o que normalmente envolve interação entre as pessoas da empresa e as pessoas de TI. Capturar detalhes da arquitetura de negócios, que nesse estágio você começa a criar no mapa de habilidades de alto nível.
- *Completar a arquitetura de negócios “no estado em que se encontra”.* Nesse estágio, você adiciona mais detalhes para criar uma arquitetura de negócios.
- *Identificar as etapas seguintes recomendadas.* Você poderá ter concluído nesse ponto ou poderá ter identificado a necessidade de aplicar técnicas de aprimoramento de processo, a análise de terceirização de processo de negócios etc.

A metodologia às vezes é chamada de metodologia de “portão de fase” porque existem critérios distintos que você deve conseguir atender antes de prosseguir para a fase seguinte. Também é importante compreender que a metodologia estimula a iteração em cada uma dessas etapas. Tendo concluído as etapas, você pode utilizar as informações que descobriu para construir o modelo subsequente (consulte Recursos para obter mais informações).

de uma taxa de participação que cobra dos clientes. Embora o mercado seja o mesmo, as duas empresas possuem modelos de negócios bastante diferentes. A aplicação das práticas recomendadas do mercado varejista em uma delas criará valor em uma organização e o destruirá na outra. Por isso, é muito importante ter um entendimento claro e detalhado do modelo de negócios.

Motion descreve como extrair e documentar informações estruturais e orientadas a processos ao lado das habilidades individuais. Ele fornece a orientação para realizar a atividade de modelagem e inclui uma grande quantidade de modelos para ajudá-lo a documentar todas as informações necessárias sobre cada função (consulte a barra lateral, "Como o Motion funciona").

A metodologia também oferece orientação para mostrar como ela complementa uma série de estruturas de aprimoramento do processo de negócios existente, incluindo reengenharia do processo de negócios, Six Sigma, Lean e a estrutura Zachman. O Motion não apenas faz coisas que esses modelos não fazem, mas também ajuda com coisas que esses outros modelos nunca foram destinados a fazer.

O ponto central para entender por que Motion é diferente está em entender o mapeamento de habilidades e como ele se diferencia do mapeamento de processos. A arquitetura de recursos, e não a arquitetura de processo está no núcleo da metodologia Motion. Ao abstrair (e até mesmo ignorar) processos e analisar a habilidade de negócios inicialmente, pode obter uma visão inerentemente mais estável da sua empresa, particularmente valiosa do ponto de vista de versionamento. As habilidades tornam subseqüentemente os blocos de construção dos processos.

Olhando além das práticas atuais, muitas das quais não oferecem suporte ao ritmo rápido de mudança inerente aos negócios de hoje, a metodologia Motion foi desenvolvida como uma metodologia de arquitetura de negócios que pudesse tratar os desafios atuais e futuros da economia conectada.

A metodologia Motion define aproximadamente 80 atributos para ajudar a documentar habilidades. Alguns dos atributos que devem ser documentados para cada uma foram citados anteriormente na discussão dos habilidades de negócios de nível 3.

O benefício principal de documentar as habilidades de uma maneira detalhada e consistente é que isso permite um entendimento sólido do SLE que a empresa possui para cada um dos recursos de negócios. Esse entendimento permite obter SLA que por sua vez ajudam a determinar a tecnologia de implementação e a topologia de implantação mais adequadas para a implementação dos serviços. Tendo a descrição completa e detalhada de cada função, as suas equipes técnicas poderão implementar a solução da maneira correta.

Criando modelos de serviço

Considerando um modelo de negócios, a pergunta agora é como esse modelo de negócios pode ser convertido em um modelo que possa ser implementado no final? Antes de examinar uma abordagem que permita fazer essa conversão, consideremos que é necessário definir a criação de um modelo de serviços (consulte a Tabela 1). Os principais artefatos de contrato das saídas da análise e do design orientados a serviços e capturados pelo modelo de serviços são mostrados na Figura 7. Observe que somente os artefatos abstratos (independentes de tecnologia) precisam ser capturados para o modelo de serviços. O transporte e o ponto de conectividade são definidos no modelo de tecnologia.

Em um nível abstrato é necessário entender e identificar entidades, mensagens e interfaces. Quais são as entidades contidas nos dados transmitidos como mensagens durante a interação entre habilidades de negócios? Quais são as mensagens que precisam circular entre os diferentes sistemas? Quais são as interfaces que as habilidades de negócios (e, no final, de serviços) expõem?

No nível concreto, o modelo de tecnologia define então quais são os meios de transporte e como deve expor os pontos de conectividade de uma habilidade específica, que no final se converte em um serviço. No modelo de serviços e de uma perspectiva de um contrato de linguagem de definição de serviços, também precisa entender e identificar os tipos de dados e os tipos de portas.

Para identificar e documentar os itens anteriores em um modelo de serviços, não é necessário utilizar técnicas de análise radicalmente novas. Podemos utilizar habilidades existentes, como OOAD - object-oriented analysis and design (Análise e Design Orientados a Objetos) convencional, aplicadas em um nível de abstração diferente para acomodar orientação a serviços em vez de orientação a objetos.

Com a SOAD - service-oriented analysis and design (Análise e Design Orientados a Serviços), podemos reutilizar habilidades existentes e particularmente as associadas ao UML e OOAD convencional. Podemos reutilizar habilidades existentes e particularmente as associadas ao UML e OOAD convencional. No entanto, é necessária uma mudança na ênfase. Devemos parar de pensar em interações de chamada de método de estilo RPC entre objetos e começar a pensar em interação entre serviços baseada em mensagens. Fazendo isso, podemos tomar o OOAD clássico baseado em UML e aplicá-lo no nível da abstração de negócios para criar os modelos de serviços. A diferença principal entre SOAD e OOAD é que SOAD separa claramente processo e entidade para desacoplar serviços. Os serviços desacoplados são mais ágeis e refletem a realidade das práticas de negócios reais.

Por exemplo, consideremos como os modelos UML convencionais podem ser aplicados à análise e ao design orientados a serviços:

- Casos de uso, incluindo diagramas de atividade, são derivados facilmente das descrições de atividade e tarefa fornecidas pelo modelo de habilidades.
- O modelo de colaboração permite um bom entendimento do processo que precisa ser estabelecido entre as tarefas e as atividades, o que nos ajuda a chegar aos requisitos de orquestração.
- O modelo de interação ou diagrama de seqüência fornece informações sobre os padrões de troca de mensagens (solicitação/resposta síncrona, duplex assíncrono etc.) que precisam ocorrer entre os serviços. Também nos ajuda a começar a derivar o modelo de domínio canônico externo, que define o esquema dos dados na transmissão, presente nas mensagens. Com essas informações, podemos começar a definir contratos de serviços.
- O modelo de componentes esboçado em um estágio primário ajuda a enfatizar os problemas e as necessidades operacionais, considerando a estrutura organizacional, as responsabilidades e os proprietários do recurso (serviço). Ele captura informações importantes sobre disponibilidade, confiabilidade e dimensionamento de uma perspectiva de negócios e ajuda a revisar SLE e SLA.

Utilizando uma abordagem pragmática de SOAD, podemos extrair todas as partes necessárias para construir o modelo de serviços. Essa extração inclui contratos de serviços, SLA derivado do SLE definido para cada habilidade de negócios e os requisitos de orquestração de serviços.

“MOTION FORNECE A ORIENTAÇÃO PARA REALIZAR A ATIVIDADE DE MODELAGEM E INCLUI UMA GRANDE QUANTIDADE DE MODELOS PARA AJUDAR A DOCUMENTAR TODAS AS INFORMAÇÕES NECESSÁRIAS EM RELAÇÃO A CADA HABILIDADE”

Com um modelo de serviços detalhado e bem alinhado com o modelo de negócios e dele derivado, agora deve estar bem situado para mapear o modelo de serviços para um modelo de tecnologia que identifique como cada serviço será implementado, hospedado e implantado.

A segunda parte desta discussão será publicada na próxima edição do *Jornal de Arquitetura* e irá demonstrar, usando a abordagem anterior e criando o modelo de serviços, como se pode entregar ao departamento de TI esquemas de dados, contratos de serviços e requisitos de SLA para informar a definição e, no final, a implementação do modelo de tecnologia.

Uma nova direção

A maneira antiga de pensar sobre orientação a serviços não está funcionando e uma nova maneira de pensar é necessária. Adotando essa nova maneira de pensar, como arquitetos, podemos forçar a consideração explícita de artefatos de modelo de serviços nos nossos processos de design, o que nos ajuda a identificar os artefatos corretamente e, no nível de abstração certo, atender e alinhar as necessidades de negócios.

De uma perspectiva de modelagem, a lacuna entre os modelos de negócios e de tecnologia convencionais é muito grande, o que é o fator principal de contribuição para o fracasso de muitas iniciativas de orientação a serviços. Apresentamos um modelo em três partes com a introdução de um modelo de serviços entre os modelos de negócios e de tecnologia para promover um alinhamento dos serviços mais próximo com as necessidades dos negócios. Com um modelo de serviços detalhado e bem alinhado de com o modelo de negócios e dele derivado, está bem situado para mapear o modelo de serviços para um modelo de tecnologia que identifique como cada serviço será implementado, hospedado e implantado. O mapeamento de habilidades e a metodologia Motion fornecem uma maneira efetiva de identificar funções de negócios e, no final, serviços. A decomposição dos negócios em habilidades fornece a separação em alto nível dos contratos de seus respectivos serviços e não o contrário, como geralmente é hoje.

Recursos

“Modeling and Messaging for Connected Systems,” Arvind Sehmi e Beat Schwegler, um Webcast de uma apresentação no Enterprise Architect Summit–Barcelona (FTPOline.com, 2005) www.ftponline.com/channels/arch/reports/easbarc/2005/video/

Receba um estudo de caso sobre a metodologia Microsoft Motion enviando uma solicitação para motion@microsoft.com.

Sistemas conectados são instâncias do modelo em três partes completo e respeitam os quatro princípios da orientação a serviços. Eles podem ser implementados de forma mais completa utilizando os cinco pilares das tecnologias de plataforma da Microsoft. Lembra o que perguntamos no início: Como fazemos para não cometer com as SOA - Service Oriented Architecture (Arquiteturas Orientadas a Serviços) os mesmos enganos cometidos por promissoras iniciativas anteriores? Como fazemos para garantir que a arquitetura de implementação escolhida relaciona-se ao estado real ou desejado dos negócios? Como fazemos para garantir uma solução sustentável que possa reagir à natureza em transformação dinâmica dos negócios — em outras palavras, como podemos ativar e sustentar uma empresa ágil? Como fazemos para migrar para esse novo modelo de uma maneira elegante e a um ritmo que possamos controlar? E como podemos fazer essa mudança com uma boa percepção de onde podemos agregar o maior valor para os negócios desde o início?

A orientação a serviços com Web Services é apenas a implementação de um modelo específico. A qualidade e a base do modelo é o que determina as respostas a essas perguntas.

Agradecimentos

Os autores desejam agradecer a Ric Merrifield (diretor, Microsoft Business Architecture Consulting Group), David Ing (arquiteto de software independente), Christian Weyer (arquiteto, Thinktecture), Andreas Erlacher (arquiteto, Microsoft Áustria) e Sam Chenaur (arquiteto, Microsoft Corporation) pelos seus comentários sobre as primeiras versões deste artigo. Também queremos deixar o nosso agradecimento a Alex Mackman (tecnologista principal, CM Group Ltd.), um excelente pesquisador e escritor que nos ajudou imensamente.

Sobre o autor

Arvind Sehmi é o responsável por arquitetura corporativa no Microsoft EMEA Developer and Platform Evangelism Group. Dedicou-se à adoção de práticas recomendadas de engenharia de software corporativo por meio da comunidade de desenvolvedores e arquitetos da EMEA e lidera o evangelismo de arquitetura na EMEA para o mercado de serviços financeiros. Arvind é editor emérito do *The Architecture Journal*. Ele é Doutor em Engenharia Biomédica e Mestre em Negócios.

Beat Schwegler é arquiteto do Microsoft EMEA Developer and Platform Evangelism Group. Ele realiza suporte e consultoria para empresas em arquitetura de software e tópicos relacionados e é orador frequente em eventos e conferências internacionais. Com mais de 13 anos de experiência em arquitetura e desenvolvimento de software profissional, esteve envolvido em uma ampla variedade de projetos, desde sistemas de controle de construção em tempo real e produtos prontos de grande vendagem até sistemas CRM e ERP de grande escala. Nos últimos quatro anos, o enfoque principal de Beat tem sido na área de orientação a serviços e Web Services.



Microsoft[®]

09981045309

**THE
ARCHITECTURE
JOURNAL**[™]
Idéias para melhores resultados



Subscribe at: www.architecturejournal.net