

Arquitetura de Software para Colaboração e Integração de Sistemas de Informação Através da Troca de Mensagens: Uma Proposta

Carlos Constantino Moreira Nassur

Analista Legislativo da Câmara Federal dos Deputados, estagiário do Curso de Altos Estudos de Política e Estratégia da Escola Superior de Guerra em 2007.

Resumo

Este trabalho aborda, preliminarmente, a problemática resultante do baixo volume de intercâmbio de dados e informações entre os órgãos da Administração Pública e aperfeiçoa-se mediante a apresentação de uma sugestão de arquitetura de *software* capaz de viabilizar a construção de uma rede de colaboração utilizável naquela realidade. Espera-se que a construção da rede, ora proposta, dê abertura ao aumento do trabalho colaborativo e ao incremento do compartilhamento de informações entre as unidades administrativas de todos os níveis do governo. É esperado que a adoção da rede de colaboração em tela intensifique a velocidade de resposta da Administração às questões e desafios que surgem. Para tanto, o texto desenvolve-se com uma introdução ao problema, seguida das motivações que justificam o estudo do assunto e de uma revisão teórica dos conceitos principais. Após, apresenta-se a arquitetura de *software* destinada ao aumento do intercâmbio de dados e informações; e, nessa, incluem-se alguns cenários ilustrativos da aplicação da solução e uma estratégia para sua implantação real. Por fim, a conclusão apresenta alguns pontos que podem ser explorados e aperfeiçoados, a partir do modelo proposto.

Palavras-chave: Arquitetura de Software. Integração de Sistemas de Informação. Troca Eletrônica de Mensagens. Colaboração Eletrônica.

Abstract

This study deals, initially, with the problems resulting from the low-volume of information and data exchanges between agencies of the federal and local governments. Subsequently, it proposes a software architecture to build a collaborative network designed to improve the communications and sharing of information among its members. Further, the study suggests that the network evolving from the proposed architecture will revert the current trend, and will increase collaborative work within the public sector. It also predicts that the proposed network

will also enhance the manner in which governmental agencies coordinate their responses to the day-to-day challenges they face. To achieve those objectives, the study introduces the problem, reveals the motivations behind this proposal, and provides a technical review of concepts essential to its understanding. After that, it shows the proposed software architecture design, followed by practical examples of its application. Finally, the study shows the advantages that the software architecture has.

Keywords: Software Architecture. Information Systems Integration. Electronic Message Exchange. Electronic Colaboration.

I - INTRODUÇÃO

Nas diversas esferas de governo, há uma grande variedade de organizações destinadas às diversas atividades, serviços e políticas institucionalmente atribuídos à administração pública brasileira. Muitas delas, porém, por mera incipiência de informação, na realidade gastam, sistematicamente, significativas frações de suas energias na realização de trabalhos de mesmos volumes e naturezas daqueles concomitantemente realizados por outras unidades governamentais, de mesmo nível estrutural ou não e, ocasionalmente, até da mesma área de atuação institucional.

Para exemplificar essa redundância de esforços, basta lembrar alguns que, com diferentes interesses, envolvem uma mesma atividade singular e típica de qualquer estado: a criação, a execução das ações e o desenvolvimento dos trabalhos de inteligência. Neste exemplo específico, temos (I) a Agência Brasileira de Inteligência (ABIN), que entre outras atividades acompanha e analisa conjunturas de questões nacionais e internacionais, com o objetivo de identificar possíveis obstáculos à integração, à implementação e à consecução de objetivos governamentais; (II) as agências vinculadas ao Ministério da Defesa, i.e., o Departamento de Inteligência do Ministério da Defesa, a Chefia de Estratégia do Estado-Maior da Armada, o Estado-Maior do Exército e a Secretaria de Inteligência da Aeronáutica, que levantam informações de interesse militar, com vistas em auxiliar o planejamento estratégico e a tomada de decisões do Ministro da Defesa e dos comandantes das armas; (III) o Departamento de Combate a Ilícitos Cambiais e Financeiros do Banco Central do Brasil, que levanta e analisa dados sobre as operações que envolvam o sistema financeiro nacional na consecução de operações ilegais; (IV) a Diretoria de Inteligência Policial do Departamento de Polícia Federal, que executa atividades de inteligência para a prevenção e o combate ao crime organizado, assim como (V) os diversos outros órgãos vinculados às secretarias estaduais de segurança (NASSUR, 2002). Portanto, a listagem de todas as atividades executadas pela

administração pública e dos órgãos, que as executam, produziria uma lista expressivamente maior.

Apesar da clara institucionalização das finalidades de cada um dos órgãos da administração, o rol de atividades e serviços desempenhados pelo estado não aparenta, necessariamente, fazer o melhor uso dos recursos públicos, pois, tal como na área de inteligência, ocorre freqüentemente a falta de recursos, de uniformidade de métodos e processos, de rotinas na execução dos trabalhos e, especialmente, de troca sistemática de dados e informações entre os órgãos, além de diversos outros problemas.

Por outro lado, o precário compartilhamento de informações entre os diversos órgãos da administração pública enseja uma oportunidade de melhoria no uso dos recursos públicos: a criação de uma arquitetura de *software* comum a maioria, para a troca e compartilhamento de informações, acordada e aceita por todos os participantes, de maneira a viabilizar o tráfego de dados de interesse amplo e a tornar efetivo o desempenho das atividades dos vários órgãos. Esta ferramenta auxiliar deve ser concebida como um sistema de informações capaz de conhecer e processar, de forma precisa e tempestiva, todas as informações sobre atividades realizadas e captadas, sem necessariamente nelas intervir.

O desafio de criar um mecanismo comum de troca de informações não é inédito no Brasil; *v.g.*, já foi constituído pelo Banco Central do Brasil, que engenhosamente criou uma solução para viabilizar a comunicação entre as diversas instituições financeiras mediante o novo Sistema de Pagamentos Brasileiro (SPB) (NASSUR, 2002). Apesar de possuírem diferentes ambientes computacionais, processos internos diversos e sistemas contábeis distintos, as instituições financeiras brasileiras são capazes de, a qualquer hora e dia, trocar dados e informações sobre as suas diversas operações através do SPB.

O sistema bancário fornece também outro mecanismo que pode ser usado como parâmetro na concepção do modelo ora proposto: o Sistema Nacional de Compensação, onde um agente central e imparcial realiza as trocas dos papéis recebidos pelas diversas instituições financeiras participantes.

Esses dois exemplos, o Sistema de Pagamentos Brasileiro e o Sistema Nacional de Compensação, podem servir, portanto, para a presente concepção da arquitetura de integração e colaboração. Um modelo compartilhado com administração central, imparcial, receptor e emissor de mensagens, encaminháveis através de uma infra-estrutura de comunicação comum, produzidas pelos diversos órgãos e entidades participantes, que delas possam livremente derivar as informações que interessem ao desempenho de suas atribuições específicas.

Isto posto, o presente trabalho pretende apresentar uma proposta para a concepção de uma arquitetura de *software* que viabilize, incentive e incremente a colaboração entre os diversos órgãos da administração pública,

dos poderes da União e nas três esferas de governo, mediante a integração de seus sistemas de informação e da livre troca de mensagens.

Para tanto, este texto apresenta os elementos motivadores do desenvolvimento do tema. Em seguida, é apresentada uma revisão teórica, que contempla os principais conceitos necessários para o entendimento da proposta de arquitetura que se segue. Também são incluídos alguns cenários ilustrativos para aplicação da arquitetura proposta, as estratégias necessárias para sua implementação e uma conclusão final.

II - MOTIVAÇÃO

O relatório da comissão parlamentar norte-americana, que avaliou os eventos ocorridos no território dos Estados Unidos da América (EUA) em 11 de setembro de 2001 (The National Commission on Terrorist Attacks, 2004), informa como foi possível aos terroristas desencadear os catastróficos ataques. Além das medidas para prevenção de novos atentados, o relatório aponta como uma grande vulnerabilidade na segurança daquele país, à época, a resistência, humana ou sistêmica ao compartilhamento de informações entre os órgãos da administração.

A comissão apurou que, apesar de o governo dos Estados Unidos possuir uma vasta quantidade de dados, os órgãos de inteligência responsáveis pela sua análise e conseqüente produção de informações não contavam com um adequado sistema para seu processamento e uso. Assim, apesar de os órgãos norte-americanos de defesa e inteligência produzirem informações relevantes, que, em alguns casos, poderiam frustrar ou comprometer os atentados, elas ou não alcançaram os níveis decisórios adequados ou não o conseguiram em tempo hábil.

O relatório cita exemplos como o da informação não distribuída pela *National Security Agency* (NSA), em janeiro de 2000, que poderia facilitar na identificação de Nawaf al Hazmi, um dos terroristas que dominou o voo 77 da *American Airlines* que se espatifou no Pentágono. Aquela informação, apesar de disponível, não foi compartilhada porque nenhum órgão a solicitou!

Tal situação, a existência de uma informação extremamente importante mas não compartilhada simplesmente pela falta de solicitação específica, revela uma característica vital de um sistema dessa natureza: a necessidade de conhecimento prévio e disseminado da existência de determinada informação.

Esse enfoque, porém, expõe certos pontos nevrálgicos:

- a) pressupõe o conhecimento prévio de qual órgão ou entidade necessitará de dada informação; e
- b) assume que o risco da divulgação institucional ou inadvertida de

determinada informação pode ser significativamente maior que o benefício de seu compartilhamento amplo.

Essa postura perante à troca de informações, fruto do período da Guerra fria, não é atualmente a mais vantajosa. Para ilustrar o precário modelo adotado pelos EUA até setembro de 2001, o relatório de início citado mostra que cada uma das agências possuía seu próprio procedimento para classificação e divulgação das informações. O mesmo documento também aponta que o modelo defensivo existente até aquela data preocupava-se fundamentalmente com o resguardo das informações e não incentivava seu compartilhamento. A cultura para a guarda da informação estava tão arraigada nos órgãos que resultava em normas que puniam os agentes pela divulgação inapropriada de informações; mas, por outro lado, não havia, à época, punição aos agentes públicos pelo não compartilhamento de informações, mesmo quando assim devessem proceder.

Em síntese, o relatório oferece recomendações ao governo norte-americano sobre a maneira como se deveria promover uma verdadeira mudança no paradigma cultural do tratamento de informações e sobre a necessidade de se criarem mecanismos práticos de incentivo ao compartilhamento de dados entre os órgãos governamentais dos EUA.

O mesmo documento, também, apresenta uma possível solução para o problema do compartilhamento de informações. Mediante um modelo de rede descentralizada, os órgãos poderiam adequadamente trocar as informações que possuíssem e que poderiam ser úteis a outros agentes em prol do bem comum e do melhor uso dos recursos públicos.

Apesar das questões de segurança retromencionadas, elas não seriam as únicas justificativas para se resolver o precário intercâmbio de informações entre os órgãos da administração de qualquer país. A vida do século XXI não admite processamentos demorados e/ou pouco eficazes de comunicação. Tampouco permite o isolamento, o trabalho não colaborativo, a restrição ao acesso aos dados e informações, a falta de transparência dos atos e a ineficiência dos serviços públicos.

Quando os órgãos da administração de um país qualquer utilizam os canais tradicionais para comunicarem-se ou não colaboram entre si, *v.g.*, na forma de troca e intercâmbio de dados e informações para uma adequada prestação de serviços públicos, tornam-se teoricamente cobráveis por uma sociedade que, em geral e em termos igualmente teóricos, exigiria de seu governo melhores serviços.

Assim, a proposição de um mecanismo que favoreça a colaboração entre os diversos órgãos da administração de qualquer país, que vise especificamente à concepção de uma infra-estrutura adequada à troca de informações entre os participantes ensejará:

- a) a diminuição do nível de ignorância do estado (ARMOUR, 2000) nas questões da realidade da sociedade nacional;
- b) o incremento da velocidade de reação dos agentes públicos no enfrentamento dos desafios para cujo gerenciamento foram criados; e
- c) a minimização dos prejuízos gerados pela ineficiência ou pela duplicação de ações ou mesmo pela demora de atendimento.

III - REVISÃO TEÓRICA

1 - Arquitetura de Software

A *arquitetura de software*, seja de um programa ou de um aplicativo computacional, define-se como o conjunto de elementos significativos que descrevem suas forma e estrutura. Um elemento é uma parte, item ou unidade de determinado sistema que se deseja demonstrar abstratamente. No caso da representação da arquitetura de *software*, um elemento pode ser um componente de *software*, como uma implementação, um de *hardware*, como um circuito, ou um enlace de rede. É a apresentação dos componentes internos que constitui o *software*, seus relacionamentos e suas propriedades que são externamente visíveis (PRESSMAN, 2006).

As propriedades, externamente visíveis de um determinado elemento, são aquelas que podem ser conhecidas por programas ou sistemas externos. Entre tais propriedades, incluem-se os serviços que podem ser prestados, as características de desempenho, o tratamento de erros e as necessidades para uso de recursos compartilhados.

A arquitetura de *software* apresenta a informação de como relacionam-se os elementos constituintes do aplicativo. Desta forma, omite informações daqueles elementos que não compõem o aplicativo (SEI, 2007). Assim, a arquitetura de *software* é uma abstração do sistema que suprime detalhes e informações dos elementos que não participam ou não afetam seu uso.

Pode-se utilizar a descrição arquitetônica (LEFFINGWELL & WIDRIG, 2000) para:

- a) entender aquilo que um sistema faz e como funciona;
- b) entender e trabalhar com suas partes componentes;
- c) estender o sistema e suas funcionalidades; e
- d) reutilizar partes do sistema para a construção de outro diferente.

Em todos os sistemas modernos, os elementos constituintes de um aplicativo interagem entre si mediante interfaces que compartilham, pública ou privadamente, os seus detalhes, mecanismos e informações. A arquitetura de *software* interessa-se somente pelos seus detalhes públicos. As características internas, as privativas de cada elemento, estão sob o

domínio da sua construção e não interessam à arquitetura.

Além de descrever a estrutura interna de um *software*, a arquitetura também deve apresentar as propriedades externamente visíveis ao próprio aplicativo, tais como os mecanismos de comunicação e os detalhes do ambiente operacional.

A proposição da arquitetura do *software* é uma etapa extremamente importante na elaboração dos produtos. Mais que importante, é fundamental (MENDES, 2002), sobretudo quando se pretendem criar linhas e famílias de produtos que compartilham uma mesma base de código-fonte. Neste caso, deve-se inicialmente elaborar uma arquitetura-base para a linha de produtos e, então, construir cada um dos distintos produtos que comporão a família respectiva.

1.1 - Projeto de Arquitetura

O projeto da arquitetura de um *software* deve definir os componentes fundamentais e estruturais do aplicativo. Assim, deve ser concebido de maneira a satisfazer os requisitos definidos (MENDES, 2002) para o aplicativo e a superar as restrições ambientais e tecnológicas que, normalmente, se interpõem ao desenvolvimento do *software*.

O projeto da arquitetura de um *software*, também, é útil para reduzir a complexidade da construção, pois este pode ser, via projeto, representado mediante visões que apresentam a estrutura do produto a partir de enfoques específicos; é este o recurso que visa a representação do desenho arquitetônico a serviço do projetista para acesso às visões funcional, estrutural, física, de distribuição, de código, etc..

Os requisitos do aplicativo são, usualmente, identificados nas etapas iniciais do seu processo de desenvolvimento. Devem ser descritos juntamente com as informações do domínio do problema que o produto de *software* será projetado para solucionar. Assim, em conjunto com os estilos arquitetônicos existentes, os requisitos, previamente, identificados determinarão a adequação entre o perfil final do projeto e a solução pretendida do problema.

Sumariando, a arquitetura proposta neste trabalho parte dos requisitos (funcionais e não-funcionais) mais significativos que devem ser atendidos e visa a integração sistêmica entre os órgãos participantes da rede de colaboração. Será representada conforme o modelo “4+1” proposto por Kruchten (KRUCHTEN, 1995) (KONTIO, 2005).

O modelo “4+1” utiliza a sintaxe e a semântica da *Unified Modelling Language* para demonstrar os elementos e relacionamentos da arquitetura. Permite a separação da arquitetura de *software* em quatro visões essenciais:

a lógica, a de processos, a de implementação e a de implantação (LEFFINGWELL & WIDRIG, 2000) (KONTIO, 2005).

Esta abstração e esta divisão em muito facilitam o entendimento do problema (MENDES, 2002) que será solucionado com o desenvolvimento do aplicativo, pois permite ao seu arquiteto concentrar-se em cada visão isoladamente e, portanto, lidar separadamente com os diversos aspectos de um mesmo problema.

O modelo, também, inclui uma visão funcional, apresentada através de diagramas de casos de uso, que são utilizados para descrever os aspectos funcionais do sistema como um todo.

Cada uma das visões do modelo destaca determinados elementos do sistema enquanto intencionalmente suprime outros. Assim, o modelo apresenta uma forma para que o arquiteto entenda corretamente o problema e elabore uma solução adequada para o sistema. O modelo ainda é utilizado para documentar, comunicar e representar a arquitetura do *software*, especialmente para os distintos membros da equipe de desenvolvimento.

1.1.1 - Visão Funcional

A visão funcional apresenta um subconjunto dos casos de uso do sistema. Os casos de uso, que o compõem, são selecionados entre os representativos das funcionalidades mais importantes que devem integrar a estrutura.

Esta visão descreve, de maneira geral, os aspectos funcionais do sistema e é a responsável por dirigir a concepção de toda a sua arquitetura.

1.1.2 - Visão Lógica

A visão lógica representa os requisitos comportamentais e mostra como o sistema é decomposto em um conjunto de abstrações. Nesta visão, as classes e os objetos são os principais elementos estudados.

Geralmente, os diagramas de classe ou de seqüência, entre outros, são utilizados para ilustrar o relacionamento entre os elementos que constituem a visão lógica do sistema.

1.1.3 - Visão de Implementação

A visão de implementação descreve os elementos que são relevantes e diretamente relacionados com a própria construção da estrutura, tais como

o código fonte, as bibliotecas, os componentes reutilizados, entre outros. Esta visão expõe a perspectiva estática dos componentes do sistema e não se presta ao entendimento de como seus componentes interagem e se relacionam.

1.1.4 - Visão de Processo

A visão do mecanismo permite a descrição e o estudo dos processos internos e externos do sistema, sobretudo da maneira de como se dá a comunicação entre eles. Uma visão geral dos processos e de sua comunicação permite que sejam tratados erros não intencionais, além de possibilitar que o arquiteto do *software* distribua adequadamente seu componente de forma a elaborar uma solução que produza um melhor desempenho global para o *software* resultante.

Esta visão pode ser bastante útil quando se deseja representar sistemas que possuem múltiplos processos, processos simultâneos ou tarefas concorrentes no *software*.

1.1.5 - Visão de Implantação

A visão de implantação mostra como o sistema é instalado, ou seja, como são distribuídos os componentes do sistema na infra-estrutura de *hardware* e de *software* que proverão o ambiente computacional para operacionalizar seus serviços. Esta visão deve lidar com os detalhes do sistema operacional, da plataforma física e das principais restrições que são impostas, uma vez que a escolha da sua distribuição nos distintos equipamentos de *hardware* é diretamente afetada pelos requisitos não-funcionais levantados.

1.2 - Requisitos

Os requisitos de *software* são descrições das funções e das restrições que um determinado produto deve atender. Define-se requisito de um aplicativo como uma condição ou capacidade que deve possuir para que seja capaz de prover um serviço, solucionar um problema, produzir um resultado ou alcançar um objetivo. Usualmente classificam-se os requisitos em duas modalidades distintas: os funcionais e os não-funcionais.

Os funcionais são as declarações das funções que o produto deve desempenhar. O termo funcionalidade é genericamente utilizado para representar a capacidade de realização de operações pelo sistema, seja por meio de comandos dos seus usuários ou pela ocorrência de eventos internos ou externos ao sistema.

Os requisitos não-funcionais descrevem as restrições impostas ou enfrentadas pelo produto de *software* (KRUCHTEN, 1995) (LEFFINGWELL & WIDRIG, 2000). Incluem-se entre os exemplos as restrições de tempo, restrições no processo de desenvolvimento, nos padrões e qualidades globais de um produto como manutenibilidade, confiabilidade, escalabilidade, usabilidade, desempenho, portabilidade etc.. Os requisitos não-funcionais são aqueles que não estão diretamente relacionados com as funcionalidades que o produto de *software* deve possuir.

Os requisitos, especialmente os não-funcionais, desempenham papel de extrema importância no desenvolvimento de sistemas de informação (MENDES, 2002). Como a arquitetura de uma solução emerge naturalmente:

a) dos principais requisitos funcionais que a aplicação deve atender; e
b) das restrições ambientais que serão impostas ao produto de *software* resultante, o imperfeito entendimento dos requisitos não-funcionais postos para o problema em estudo pode levar ao fracasso produto que se deseja elaborar.

Assim, a correta identificação, consideração, descrição e tratamento dos requisitos não-funcionais deverão servir de base para a seleção e proposição do estilo arquitetônico mais adequado para o problema em estudo.

Para tornar-se eficaz, a identificação dos requisitos não-funcionais deve servir-se de uma organização que facilite seu tratamento e que, adequadamente, os classifique e descreva.

Das diversas propostas de organização dos requisitos não-funcionais, uma bastante adequada e eficaz é aquela que divide estes em classes definidas. Uma tipificação adequada a esse estudo (PRESSMAN, 2006) pode ser obtida utilizando-se as seguintes classes de requisitos:

a) Usabilidade: Os requisitos de usabilidade aplicam-se especialmente aos sistemas interativos, nos quais a interação entre o produto e seus usuários requer a definição de interfaces de fácil uso, de fácil aprendizagem e de alta inteligibilidade.

b) Manutenibilidade: A manutenibilidade diz respeito à facilidade de aplicar modificações no produto de *software*. Tais modificações podem ser advindas de defeitos descobertos, alterações nas funcionalidades ou mesmo da necessidade de inclusão de novos requisitos e serviços no produto.

Como a arquitetura do *software* define seus componentes e as relações entre eles, o seu projeto desempenha papel fundamental na facilidade de aplicação das modificações, ou seja, na manutenibilidade que se deseja alcançar.

A manutenibilidade de uma arquitetura é, especialmente, afetada quanto às decisões adotadas na divisão, separação e agregação dos componentes do produto em módulos.

c) Confiabilidade: A confiabilidade de um produto de *software* refere-se à probabilidade que este elemento cause ou não falhas no sistema do qual faz parte, sob condições previamente determinadas. A execução das funções, serviços e operações atribuídas ao *software*, em conformidade com os requisitos previamente estabelecidos, também determina sua confiabilidade.

Os atributos, que configuram a confiabilidade de um aplicativo, podem ser obtidos a partir das restrições de sua disponibilidade; da taxa de ocorrências de falhas; da probabilidade de falha durante sua operação; e do tempo médio até a ocorrência de falhas.

d) Desempenho: As necessidades de desempenho impostas aos produtos de *software* estão entre os tipos de requisitos não-funcionais que mais influenciam na proposta e no desenho da arquitetura. Entre outros aspectos, em sistemas onde os tempos de resposta ou de processamento como, por exemplo, os sistemas de controle de tráfego aéreo ou de cartões de crédito, os rígidos requisitos sobre os tempos para processamento das informações e transações são determinantes na solução arquitetônica desses sistemas.

Os requisitos de desempenho (MENDES, 2002) podem ser também classificados conforme o tipo de restrição aplicada ao produto. Assim, as restrições quanto aos tempos de resposta que os produtos de *software* devem respeitar; a temporização do seu processamento; o *throughput*; e de espaço (em disco, memória etc). devem ser bem avaliadas e consideradas no projeto da arquitetura.

e) Portabilidade: A portabilidade de um produto de *software* refere-se à facilidade de levá-lo a operar em um ambiente distinto daquele que originalmente foi proposto. Portanto, um produto possui alta portabilidade quando é relativamente fácil fazê-lo operar em um ambiente, de *software* ou de *hardware*, diferente de seu originário sem que, para isso, haja necessidade de grandes modificações em sua implementação ou em seus componentes.

Em termos de arquitetura de *software*, diz-se que é portátil quando seu desenho geral pode ser realizado em plataformas distintas sem grandes modificações.

f) Segurança: A segurança de um aplicativo de *software* diz respeito aos seus atributos de integridade, confidencialidade, disponibilidade dos serviços, de resistência a ataques e de proteção operacional, especialmente quanto ao acesso a dados e serviços restritos.

2 - Integração de Sistemas

Define-se integração de sistemas como o uso de produtos de *software* ou de princípios de arquitetura de sistemas para a interconexão de um determinado conjunto de aplicações computacionais em um único aplicativo.

A integração de sistemas faz-se necessária para a construção de soluções complexas que combinem em um mesmo produto de *software* as funcionalidades ou o acesso aos dados de mais de um aplicativo. Também é fundamental para a viabilização de negócios através de soluções sistêmicas como, por exemplo, o comércio eletrônico, a compensação bancária de documentos ou mesmo em sistemas corriqueiros como aqueles que provêm os serviços de reserva conjunta de passagens aéreas e de hospedagem. Tais serviços certamente teriam elevados custos de implementação caso não fosse possível a integração de sistemas distintos.

É requisito para a perfeita composição de sistemas que o uso das funcionalidades, ou dos dados tomados emprestados dos outros aplicativos, torne-se fluído ao usuário da aplicação, ou seja, que ao utilizar o produto, o usuário não consiga perceber que determinada funcionalidade é provida por outro produto e não por aquele que está utilizando.

A interconexão de sistemas, resultante direta dos ataques terroristas de 11 de setembro, imposta aos serviços de emissão de vistos de entrada nos EUA (CUELLO, 2007), é útil para exemplificar a aplicação dessa integração. Anteriormente aos ataques, os sistemas judiciais estaduais norte-americanos não estavam conectados ao sistema judicial federal nem ao de emissão de vistos. Assim, era perfeitamente possível a qualquer pessoa que atendesse aos requisitos tradicionais obter um visto de entrada nos EUA, mesmo que procurada por crime cometido em algum dos estados norte-americanos.

Esta situação foi revertida após a integração dos diversos sistemas estaduais com os federais. Atualmente, a concessão de novos vistos para os EUA, bem como a entrada de pessoas nos postos de fronteira, utilizam sistemas de informação interconectados com os sistemas judiciais estaduais. A integração quase universal dessas bases de dados permite que os oficiais consulares norte-americanos possam decidir de maneira mais fundamentada. Assim, devidamente informados de delitos menores e até crimes mais sérios que antes não estavam acessíveis, é perfeitamente possível que os agentes neguem novos vistos de entrada àqueles que cometeram crimes nos EUA. Também, tornou-se possível revogar os vistos já expedidos.

2.1 - Abordagens para Integração

Tipicamente, pode-se classificar a integração de sistemas conforme a abordagem adotada na sua implementação. Neste caso, podemos ter integração de sistemas baseada em processos e em dados (ENDREI, MARTIN, & SADTLER, 2004).

A primeira, a integração baseada em processos, é aquela na qual a integração entre as aplicações se dá através da inclusão de determinado processamento na seqüência de execução das operações. A integração objetiva produzir uma visão única e consolidada de determinada entidade ou negócio que trata a aplicação como, por exemplo, um cliente. Esta visão única acontece através da integração de múltiplos serviços, dispersos em um ou mais produtos de *software*.

A integração, baseada em processos, pode ser construída utilizando-se:

- a) os mecanismos baseados nos gerenciamento de filas de mensagens;
- b) o encadeamento de transações; e
- c) a interconexão direta de aplicações, ponto-a-ponto ou ponto-a-*hub*.

A interconexão direta de aplicações ponto-a-ponto é o estilo mais simplório de promover a integração. Fundamentada numa topologia uma-a-uma, esta modalidade habilita a conexão direta entre aplicações, que tanto pode utilizar soluções baseadas nos sistemas gerenciadores de mensagens quanto aquelas que requerem a inteira codificação dos mecanismos de conexão, as chamadas diretas entre as aplicações. Neste último caso, todas as sufocantes questões inerentes à conexão, tais como as dependências de protocolo, necessidades de conversão de dados, a garantia de entrega e recebimento de mensagens, temporização, etc., deverão ser tratadas diretamente pela própria aplicação.

A abordagem ponto-a-*hub* adota uma topologia um-a-muitos para a integração de aplicações. Este tratamento separa as regras de distribuição da própria aplicação e permite a interação de uma única fonte com diversos destinatários, concorrentemente. Esta topologia requer que um terceiro componente, o roteador, encaminhe uma mesma mensagem para diversas aplicações distintas, conforme regras adequadas de distribuição.

A integração, baseada em dados (SWITHINBANK, et al., 2007), obtém-se quando duas ou mais aplicações compartilham a mesma base de dados ou de informações. Esta abordagem de integração pode ser classificada através do mecanismo selecionado para o compartilhamento dos dados: *Extract Transform Load* (ETL) ou *Enterprise Information Integration* (EII).

Este trabalho centrará os estudos sobre a integração de sistemas fundamentada em processos e mediante o uso de gerenciadores de filas de mensagens, uma vez que tal abordagem permite:

a) o compartilhamento de informações entre os participantes da rede de colaboração independentemente de se conhecer, previamente, qual órgão ou entidade necessitará de dada informação;

b) a utilização dos atuais sistemas de informação como emissores automáticos de mensagens;

c) que os órgãos receptores das mensagens possam desenvolver seus próprios modelos de interpretação das mensagens recebidas e, assim, adequadamente responder às situações postas;

d) que cada participante avalie a conveniência e sigilo das informações publicadas;

e) a reutilização de qualquer infra-estrutura de *hardware* e *software* comercialmente disponível e utilizada pela administração pública brasileira;

f) uma flexível temporização na interconexão entre os sistemas, possibilitando tanto conexões síncronas quanto assíncronas;

g) que seja implementada sobre tecnologias maduras (McCOY, et al., 2003), disponíveis no mercado desde o final dos anos 80, contando com uma quantidade grande de fornecedores de produtos e em variedade considerável de ambientes computacionais; e

h) que os sistemas de informações dos órgãos interessados sejam conectados à rede de colaboração sem grandes modificações em suas próprias implementações.

2.2 - Gerenciamento de Filas de Mensagens

O gerenciamento de filas de mensagens é uma tecnologia de *middleware* que simplifica a comunicação entre serviços, sistemas e nós de sistemas. Esta tecnologia permite que os serviços comuniquem-se de maneira fácil e flexível. Também torna possível a construção de laços de comunicação sem, necessariamente, conhecer os detalhes de ambiente, da plataforma operacional, da implementação ou do funcionamento dos serviços que participarão da comunicação. O gerenciamento de filas de mensagens possibilita também a comunicação entre sistemas, independentemente da eventual indisponibilidade dos meios de comunicação e dos próprios participantes, o que aumenta significativamente a confiabilidade e diminui substancialmente a complexidade na implementação (DAVIES & BROADHURST, 2005) dos laços de comunicação ou das próprias aplicações.

Este é um mecanismo de comunicação entre programas através do qual uma determinada aplicação comunica-se com outra mediante a escrita e

a leitura de dados em mensagens. As mensagens são, então, encaminhadas e recebidas através de um *middleware* especial, o gerenciador de filas, que é especialmente concebido para solucionar os problemas técnicos inerentes à troca das mensagens via rede de comunicação.

Esta tecnologia é concebida a partir de dois conceitos fundamentais: mensagens e filas.

a) Mensagens

Um nó em um sistema geralmente necessita de comunicar informação para, ou de requerer determinado serviço de, outro nó qualquer no mesmo sistema ou em outro sistema interconectado. Esta peça de informação, ou solicitação de serviço, pode ser considerada uma mensagem.

A mensagem pode conter simples caracteres, dados numéricos, dados binários complexos, uma requisição de informação, um comando ou mesmo um conjunto destes elementos. É fundamental que qualquer nó, alvo que receba a mensagem, entenda perfeitamente seu formato e conteúdo. Entretanto, a infra-estrutura, responsável pela transferência da mensagem, por mais complexa que seja, não deve ser necessariamente conhecida pelos participantes da comunicação, mas, somente, garantir a integridade da informação contida na mensagem e sua correta entrega ao destinatário.

b) Filas

As filas são os contêineres das mensagens. Novas mensagens são inseridas no final das filas e, geralmente, retiradas de seu início. A Figura 1 ilustra a comunicação entre aplicações via fluxos de mensagens através de filas.

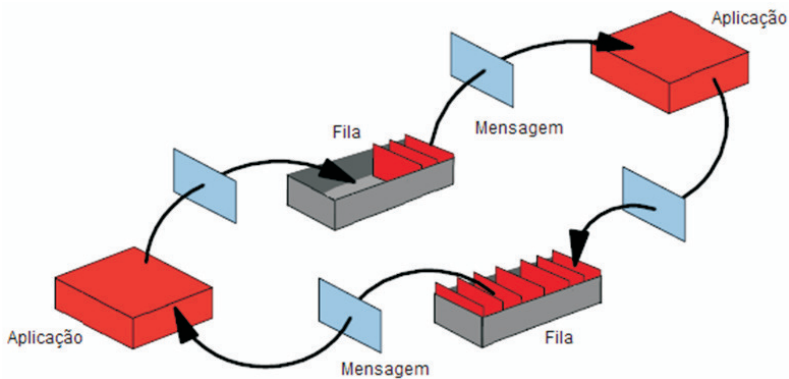


Figura 1: Mensagens fluindo através de filas

O ponto central desta solução de comunicação é o gerenciador de filas. Sua atribuição é gerenciar as filas e as mensagens destinadas às aplicações.

O gerenciador provê uma interface *Application Programming Interface* (API) através da qual as aplicações inserem e retiram as mensagens nas filas e o gerenciador responsabiliza-se por entregar corretamente as mensagens aos respectivos destinatários.

O gerenciador de filas efetua seu processamento conforme as necessidades impostas pelo desenho geral da solução (WACKEROW, 1999) ou de necessidades específicas das aplicações que o utilizam para sua interconexão. Assim, é possível atribuir níveis de prioridade distintos às mensagens inseridas nas filas; controlar a ordem de seu processamento; a data de validade das mensagens; a garantia ou não de entrega; a obtenção de recibo da mensagem por parte de seu destinatário, entre outras diversas possibilidades que podem ser exploradas.

A comunicação entre as aplicações, através desta arquitetura pode ser estabelecida independentemente da infra-estrutura de rede de comunicação de dados intermediária. O gerenciador de filas é o responsável por estabelecer a comunicação com o outro par do laço, provendo todos os serviços que garantam a correta entrega da mensagem.

Uma aplicação pode enviar mensagens para outra aplicação, executada no mesmo equipamento, onde se processa o gerenciador de filas, ou para outra, que é efetuada em uma máquina remota, como no caso de um servidor ou hospedeiro de aplicações. Neste último caso, o sistema remoto deverá também possuir seu próprio gerenciador de filas, que se responsabilizará por entregar a mensagem recebida ao destinatário adequado.

O gerenciador de filas transfere as mensagens para o outro gerenciador de filas através de canais de comunicação que utilizam os serviços de rede disponíveis, independentemente dos protocolos de rede utilizados.

Múltiplos gerenciadores de filas podem residir em um mesmo equipamento e podem gerenciar uma grande quantidade de filas.

Os programadores das aplicações não necessitam conhecer onde se encontram as aplicações com as quais desejam comunicar-se nem mesmo os detalhes dos canais de comunicação. Para estabelecer uma comunicação com outra aplicação, basta entregar a mensagem ao gerenciador de filas que aquele elemento entregará a mensagem ao destinatário. O gerenciador, então, será o responsável por resolver todos os detalhes para a comunicação e entrega da mensagem ao seu destinatário, seja ela uma aplicação local ou remota, que está em execução, não disponível ou ocupada em seu ambiente operacional.

Outro atributo fundamental desta arquitetura de comunicação, que necessita ser ressaltado, reside no fato de que a implementação dos gerenciadores de filas pode ser realizada para plataformas e ambientes computacionais diferentes.

Um aplicativo gerenciador de filas pode ser construído para ser executado em plataformas de *hardware* e *software* muito distintas (McCOY, et al., 2003), como os ambientes computacionais de grande porte da IBM, além das plataformas de menor porte como os da *Microsoft* ou das diversas implementações de *Unix*, entre outros.

Assim, esta solução permite uma grande variedade e abrangência para o estabelecimento de comunicação entre aplicações. É possível manter, *v.g.*, a comunicação entre sistemas transacionais de grande porte, como os usualmente encontrados nos órgãos do governo ou estabelecimentos bancários, e sistemas em plataforma menores, como os desenvolvidos para o ambiente *Windows*.

Ademais, a comunicação pode ser estabelecida sem grandes modificações nos ambientes que executam os participantes, bem como nas próprias aplicações que necessitam ser integradas. Basta criar, nos pontos adequados da aplicação, as chamadas para os serviços do gerenciador de filas do ambiente, entregar-lhe as mensagens que deseja encaminhar e recuperar aquelas a si endereçadas.

2.3 - Extensible Markup Language (XML)

O XML, ou *Extensible Markup Language*, é um subconjunto simplificado da *Standard Generalized Markup Language* (SGML), originalmente desenvolvida pelo consórcio W3C (W3C, 2007). XML é uma metalinguagem para a definição de dados estruturados. Originalmente desenvolvida para viabilizar a publicação eletrônica em larga escala, é um protocolo padronizado e aberto que contempla os mecanismos para a definição e a interpretação de estruturas de dados. Os dados estruturados são largamente utilizados na troca de informações entre sistemas e serviços, especialmente aqueles que utilizam a *Internet* como meio de interconexão.

O XML incorpora características importantes que facilitam o desenvolvimento de soluções para integração de aplicações, tais como: a separação entre o conteúdo (os dados) e sua formatação (apresentação); a simplicidade e legibilidade, tanto para humanos quanto para computadores; a possibilidade de criação de marcadores (*tags*) sem limitação; a criação de arquivos para validação das estruturas de dados, os *Document Type Definition* (DTD); a reutilização de dados; e a interface padrão entre sistemas heterogêneos.

As informações são escritas em XML no formato de texto simples, o que permite sua compatibilidade nos mais diversos ambientes e plataformas computacionais. XML também possibilita que os desenvolvedores de sistemas criem seus próprios marcadores (elementos) de forma que possam representar a estrutura ou a natureza dos dados da maneira mais adequada possível (INJEY, DONG, KAPLAN, & KRISHNAMURTHI, 2001).

Assim, quando se define determinado documento em XML o que se está realmente criando é uma descrição de uma estrutura de dados que melhor se adapta às necessidades da aplicação ou da própria transferência desses e, em tal estrutura, praticamente quaisquer tipos podem ser representados e, portanto, transmitidos.

A linguagem XML é altamente capaz de prover as necessidades de integração; supera muito dos mecanismos de descrição de dados já que pode ser utilizada para compartilhar dados e informações de praticamente quaisquer ambientes e plataformas computacionais disponíveis. Tal portabilidade se dá uma vez que as estruturas de dados criadas em XML são auto definidas, ou seja, possuem todas as informações necessárias e suficientes para sua correta interpretação. Tais características estão alçando XML como uma séria opção para a integração de soluções, negócios e sistemas, em plataformas homogêneas ou não.

XML reúne uma família de tecnologias que provêem mecanismos uniformes para a descrição, a transformação, a ligação, o acesso e a pesquisa de informações em documentos escritos em XML. Os padrões do protocolo definem a sintaxe dessa linguagem e o comportamento que os aplicativos devem ter no processamento dos documentos.

2.3.1 - *Document Type Definition (DTD)*

O DTD especifica a estrutura de um documento escrito em XML permitindo que seja analisado e seu conteúdo corretamente interpretado pelo destinatário do dado. O DTD contém a lista de marcadores, e seus respectivos atributos, que são utilizados em determinado documento. Especificamente, define como os elementos que formam a estrutura de dados de um documento XML se relacionam, além de também especificar quais tipos de dados podem ser inseridos nos elementos. Assim, o DTD define tanto as estruturas de dados um documento pode conter bem como os insumos para verificar a conformidade dos dados reais com o documento definido. Um documento XML é dito válido quando está em conformidade com o DTD que o define.

O DTD pode ser armazenado em um arquivo em separado ou incluído no mesmo documento XML que contém os dados em trânsito.

A colocação de DTDs de documentos XML em arquivos separados dos dados é uma boa abordagem, pois permite seu compartilhamento entre aplicações e organizações. Permite também a pesquisa por algum DTD que melhor defina a estrutura de dados recebida.

Apesar de não ser necessário que todo documento XML possua um DTD que o defina, os analisadores terão maior capacidade de identificar inconsistências naqueles que o possuem, pois a tais aplicativos serão

repassadas informações explícitas sobre o que deve ser verificado quando da determinação da validade do documento ou da interpretação dos dados.

IV - A ARQUITETURA PROPOSTA

A arquitetura proposta para promover a integração entre os órgãos da administração ora concebida emerge a partir de um modelo de rede de colaboração descentralizada. Neste modelo, os participantes deverão manter seus próprios sistemas de informação e bases de dados, trocando, entretanto, mensagens bem formadas, encaminhadas e recebidas através de sistemas gerenciadores de filas de mensagens, responsáveis pelo fornecimento da infra-estrutura de comunicação.

A troca de mensagens não deverá comprometer nem afetar os sistemas de informação ou aplicativos dos participantes da rede. Para integrar-se à rede de colaboração, cada órgão participante deverá implementar, nos seus próprios aplicativos e sistemas de informação, os mecanismos de conexão com a rede descritos na arquitetura de integração.

Assim, esta arquitetura visa descrever, de modo abstrato, uma infra-estrutura de troca horizontal de mensagens no qual não se faz necessário obedecer a determinada hierarquia ou subordinação funcional.

1 - Requisitos Não-Funcionais

1.1 - Usabilidade

A arquitetura proposta promove a colaboração dos órgãos participantes mediante a troca sistêmica de mensagens, ou seja, sem interferência humana. Assim, os requisitos de usabilidade não aplicam restrições significativas na sua concepção. Ademais, o uso de padrões abertos como XML e *Java Message Service (JMS)* permitem amplo acesso às definições dos padrões adotados e das informações a eles relacionados, características que favorecem a compreensão e o pleno entendimento da arquitetura.

1.2 - Manutenibilidade

Este é um requisito imperativo na integração de sistemas. Portanto, essa arquitetura deverá possuir alta manutenibilidade, ou seja, facilitar ao máximo a correção de defeitos, a inclusão de novos serviços ou a modificação daqueles já existentes.

O amplo uso da modularização de componentes garante imprimir alta manutenibilidade na solução aqui proposta, uma vez que tal

abordagem facilita a descoberta, a análise e o reparo de erros; permite o confinamento de defeitos em módulos específicos; e facilita a própria evolução do *software* na medida em que possibilita a inclusão de novos módulos e de novos serviços nos módulos existentes.

1.3 - Confiabilidade

O desenho geral da arquitetura aqui apresentado favorece o desenvolvimento de um sistema bastante confiável. Tal assertiva pode ser verificada a partir das características intrínsecas dos elementos que compõem a arquitetura, conforme segue:

a) Disponibilidade – os componentes responsáveis pelo gerenciamento das filas e pelo recebimento e encaminhamento das mensagens podem prover os seus serviços com altos índices de disponibilidade. Em muitos casos, o índice de disponibilidade geral do *software* básico é muito superior ao do *hardware* utilizado. Assim, a disponibilidade geral da arquitetura pode ficar condicionada aos índices de disponibilidade da infra-estrutura física do parque computacional do órgão participante;

b) Taxa de ocorrência de falhas – assim como o índice de disponibilidade, a taxa de ocorrência de falhas também é dependente do ambiente computacional como um todo. Neste caso, a contribuição do desenho da arquitetura não deverá aumentar a taxa de ocorrência de falhas, pois não contém características próprias em seu desenho responsáveis pelo incremento de risco;

c) Probabilidade de falha durante a operação – o sistema que emergirá a partir da arquitetura proposta deve tratar um conjunto muito pequeno de eventos distintos. Assim, a probabilidade de falhas durante a operação deverá ser significativamente baixa; e

d) Tempo médio até a ocorrência de falhas – esta é uma medida que deverá ser obtida após a operação cotidiana do sistema e utilizada como um dos indicadores da qualidade geral do sistema.

1.4 - Desempenho

Um requisito importante que a arquitetura deve atender refere-se ao uso geral dos recursos computacionais dos órgãos que aderirem à rede de colaboração. Como não há dados gerais e significativos sobre a capacidade das instalações dos órgãos, a arquitetura deve ser bastante flexível quanto às suas próprias necessidades de processamento, de memória e de temporização.

Uma solução centrada nos sistemas gerenciadores de filas de mensagens possibilita grande flexibilidade quanto às exigências de desempenho, pois permite o processamento *on-line* ou *batch* das filas de mensagens, além de também viabilizar a conexão síncrona ou assíncrona entre os sistemas conectados.

Dois outros aspectos também são importantes para que o desenho arquitetônico proposto atenda aos mais rigorosos requisitos de desempenho.

A definição das mensagens, as metamensagens, serão armazenadas e publicadas exclusivamente nas instalações do gestor do sistema. Desta forma, cada um dos órgãos participantes decidirá quais mensagens necessita armazenar e, em casos de necessidades mais estritas quanto às capacidades de armazenamento, poderá compactá-las, uma vez materializadas em arquivos XML, que são, em sua essência, textos simples altamente compactáveis.

1.5 - Portabilidade

Aqui reside um desafio para a arquitetura proposta: torná-la portátil entre os diversos ambientes computacionais encontrados usualmente nos centros de computação dos potenciais órgãos participantes da rede de colaboração.

A opção pela infra-estrutura de comunicação, fornecida pelos sistemas gerenciadores de filas de mensagem, garante grande portabilidade, pois esta arquitetura encontra-se atualmente disponível em diversos produtos e em muitas plataformas distintas.

Para exemplificar a alta disponibilidade de produtos que fornecem a infra-estrutura dos serviços de gerenciamento de filas de mensagens, basta citar a especificação JMS. Esta API é implementada pelos fabricantes de servidores de aplicação aderentes à especificação da plataforma *Java 2 Platform, Enterprise Edition* (J2EE) e que oferecem seus produtos ao mercado. Assim, fornecedores de grande porte como a *Sun Microsystems*, a *IBM*, a *Oracle* ou a *BEA Systems* tornam disponível ao mercado produtos que podem ser utilizados em distintas plataformas de *hardware* ou de *software*. Dos fornecedores acima citados, o produto *WebSphere MQ*, da *IBM*, implementa as especificações JMS e *Message Queue Interface* (MQI) e está disponível em 80 combinações de plataformas de *hardware* e *software* distintas.

Outro fator que atesta a alta portabilidade dos serviços de gerenciamento de filas de mensagens é a sua implementação nativa em alguns sistemas operacionais. Por exemplo, o protocolo *Microsoft Message Queue* (MSMQ), disponível na linha de produtos dos sistemas operacionais para servidores da Microsoft desde 1998, habilita a comunicação via troca de mensagens entre aplicações que operam neste ambiente com outras quaisquer, na mesma plataforma ou em ambientes distintos. Por fim, existe um número considerável de aplicativos livres e/ou de código aberto, como o *Apache ActiveMQ*, que viabilizam a integração através desta plataforma.

1.6 - Segurança

Uma vez que a arquitetura proposta não visa substituir os sistemas de informação dos órgãos participantes da rede de colaboração, os aspectos fundamentais de segurança que a solução deverá atender referem-se, exclusivamente, às questões de integridade das mensagens trafegadas, da não-repudição e, eventualmente, do sigilo dos dados de algumas delas.

Para garantir a integridade, o sigilo e a não-repudição das mensagens trafegadas, a arquitetura deverá prever os mecanismos de criptografia e de certificação digital.

Os outros aspectos inerentes à segurança da informação, tais como os de segurança física e lógica, guarda e recuperação de dados, entre outros, deverão ser tratados diretamente pelos sistemas dos próprios órgãos que farão uso da arquitetura para o tráfego das mensagens.

2 - Visões

A seguir, serão descritas visões do modelo “4+1” que apresentam a arquitetura proposta.

2.1 - Visão funcional

Os aspectos funcionais da arquitetura de integração proposta serão apresentados em dois diagramas de casos de uso distintos.

O primeiro deles (Figura 2) apresenta as funcionalidades inerentes aos serviços gerenciais da arquitetura.

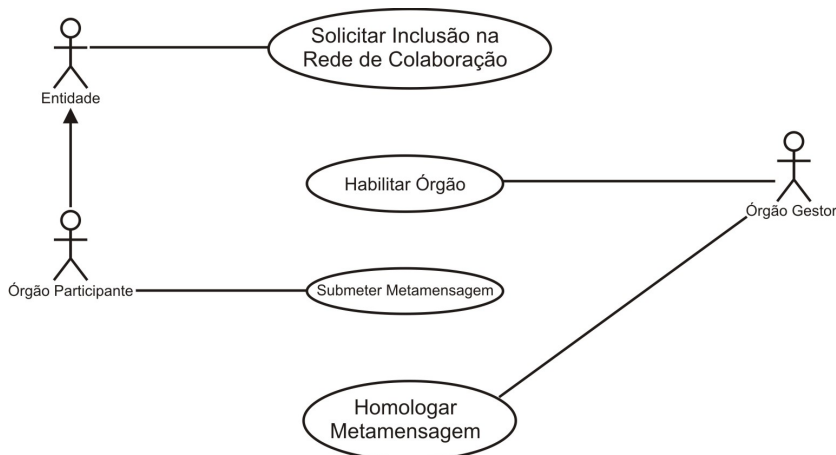


Figura 2: Funcionalidades gerenciais da arquitetura

Segue-se o detalhamento dos atores e dos casos de uso representados no diagrama da Figura 2.

Atores

- a) **Entidade:** representa qualquer órgão que ainda não é participante da rede colaborativa.
- b) **Órgão participante:** assim como apresentado no diagrama, é uma especialização de um órgão qualquer. Entretanto, possui uma característica especial, está habilitado para participar da rede de colaboração.
- c) **Gestor:** representa um órgão que é participante do sistema e possui atribuições especiais: é o gestor da rede de colaboração. Sugere-se que este papel seja desempenhado, na arquitetura ora proposta, pelo Gabinete de Segurança Institucional (GSI) da Presidência da República.

A atribuição da gestão da rede de colaboração ao GSI é bastante natural, pois é de sua competência, entre diversas outras atribuições, a prevenção da ocorrência e a articulação do gerenciamento de crises. Esta tarefa poderá ser significativamente facilitada quando da plena operação desta rede de colaboração, especialmente pelos serviços representados pelos aspectos funcionais operacionais (Figura 3).

Casos de uso

a) Solicitar inclusão na rede de colaboração

Este caso de uso congrega os serviços prestados pelo sistema para que um órgão qualquer, que não participa ainda da rede de colaboração, solicite sua inclusão.

b) Habilitar órgão

Sob a responsabilidade do gestor do sistema. Este caso de uso é o que provê os serviços administrativos para habilitar determinado órgão a participar da rede de colaboração.

c) Submeter metamensagem

Conforme será melhor detalhado na visão lógica da arquitetura, cada mensagem que trafega na rede de colaboração deverá ser bem formada e inteligível por qualquer um dos demais órgãos participantes. Assim, este caso de uso fornece os serviços para que o determinado órgão participante descreva adequadamente as mensagens que poderá compartilhar com os outros participantes através da rede de colaboração. A metamensagem deverá ser criada pelo órgão na forma de um arquivo DTD que possua todos os campos e informações necessárias ao pleno entendimento da mensagem.

d) Homologar metamensagem

De posse de uma nova metamensagem, o gestor da rede de colaboração deverá verificar se o arquivo DTD realmente possui todos os elementos necessários ao pleno entendimento da mensagem e, em caso afirmativo, homologar a metamensagem.

Os aspectos funcionais operacionais da arquitetura são apresentados na Figura 3.

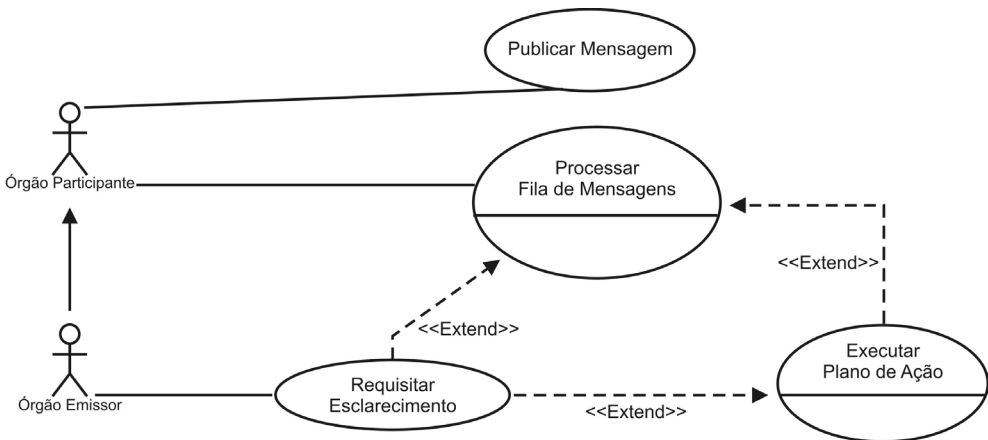


Figura 3: Funcionalidades operacionais da arquitetura

Atores

Órgão emissor: este ator representa um órgão participante da rede de colaboração e que emitiu uma mensagem que despertou interesse em outro órgão qualquer. O órgão interessado, então, poderá solicitar

maiores informações sobre a mensagem ao seu emissor.

Casos de uso

- a) **Publicar mensagem:** este caso de uso representa o conjunto de funcionalidades central da arquitetura. Através dele, cada um dos órgãos participantes poderá compartilhar os dados e as informações que possui com os demais da rede de colaboração. A mensagem deverá ser publicada em formato XML válido, ou seja, possuir um arquivo DTD descrevendo-a, o que, para esta arquitetura, necessariamente, deverá ter sido previamente homologado pelo gestor.
- b) **Processar fila de mensagens:** cada órgão participante deverá efetuar uma análise semântica das mensagens recebidas através da infra-estrutura de comunicação e conceber estados de atenção.
- c) **Executar planos de ação:** determinados estados de atenção, quando atingidos, poderão iniciar a execução de planos de ação no órgão participante.
- d) **Requisitar esclarecimento:** quando do processamento das mensagens recebidas, ou da execução dos planos de ação, se fizer necessário obter mais informações sobre determinada mensagem recebida, o órgão participante poderá solicitar ao emissor tais esclarecimentos.
É importante ressaltar que, caso trafegue pela rede de colaboração este maior detalhamento de mensagens, deverá seguir todos os mecanismos formais de definição e homologação de metamensagem.

2.2 - Visão lógica

A Figura 4 apresenta o diagrama de classes que contempla os principais conceitos que constituem a arquitetura proposta.

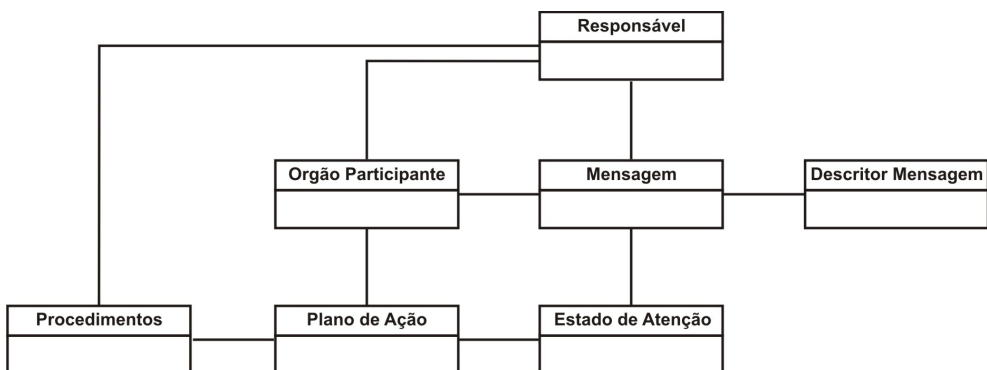


Figura 4: Diagrama de classes significativas

Classe órgão participante:

Descreve os órgãos que participam da rede de colaboração.

Classe mensagem:

Descreve as mensagens válidas que podem ser emitidas pelos órgãos participantes.

Classe responsável:

Descreve o agente público responsável pela emissão da mensagem, que pode ser tanto um sistema automático quanto uma entrada manual. Também descreve o agente que é responsável pela execução de determinado procedimento previsto em um plano de ação.

Classe descritor mensagem:

Representa a metamensagem. Classe que descreve completamente as mensagens que trafegam na rede de colaboração.

Classe estado de atenção:

Classe que descreve os estados de atenção criados pelos órgãos participantes. Os estados de atenção são alcançados quando determinado conjunto de mensagens é formado.

Classe planos de ação:

Descreve os planos de ação que deverão ser executados quando determinado estado de atenção é alcançado.

Classe procedimentos:

Descreve os procedimentos que compõem os planos de ação definidos pelos órgãos participantes.

2.3 - Visão de Implementação

Como este trabalho não pretende efetivamente construir uma solução que implemente a arquitetura proposta, não cabe aqui compor detalhadas considerações sobre sua visão de implementação.

Entretanto, algumas recomendações quanto à implementação da rede de colaboração podem ser feitas, especialmente quanto à plataforma a ser utilizada no seu desenvolvimento.

Sugere-se que seja adotada a plataforma Java para a sua implementação, especialmente por tal tecnologia possuir dois atributos significativos e que a diferenciam das demais:

a) Multiplataforma: a plataforma Java foi concebida, desde seu projeto inicial, para ser executada em plataformas de *hardware* distintas. Tal característica viabiliza, por si só, a fácil interconexão de aplicativos que operam em plataformas de *hardware* diversas; e

b) Padrão aberto: a plataforma Java é concebida sobre padrões e especificações abertas, o que lhe garante um grande número de fornecedores de implementações, inclusive de código aberto. Assim, a construção da rede de colaboração pode contar com um grande número de fornecedores e de produtos, inclusive gratuitos, o que pode diminuir significativamente os custos de sua construção.

2.4 - Visão de Processo

Os processos e relacionamentos preconizados pela arquitetura, ora proposta, seguem um modelo de comportamento bastante padronizado. Assim, basta apresentar o comportamento da postagem de mensagens na rede de colaboração para que se possa criar um entendimento geral de como se dá a operacionalização da arquitetura como um todo.

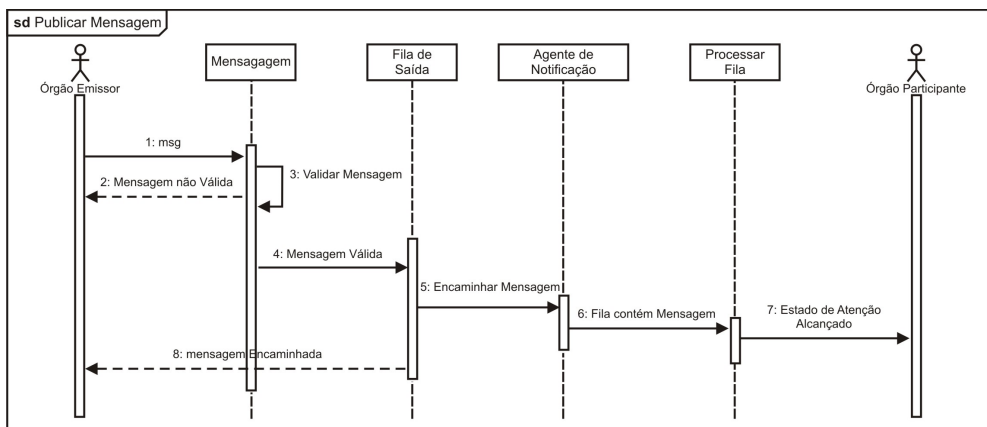


Figura 5: Publicação de mensagens

2.5 - Visão de Implantação

Assim como a visão de implementação, a visão de implantação é diretamente afetada pela construção da solução. Portanto, restringe-se a reiterar que as decisões, quanto à distribuição dos componentes do sistema, devem ser tomadas considerando as melhores alternativas para o atendimento dos requisitos não-funcionais, especialmente quanto às questões de desempenho, portabilidade e segurança anteriormente esboçadas.

V - CENÁRIOS DE APLICAÇÃO

A Cidade do Rio de Janeiro foi sede, em 2007, dos XV Jogos Pan-Americanos. Participaram da organização e preparação da cidade para os jogos diversos órgãos das três esferas de governo. Os esforços foram profícuos e o evento esportivo hemisférico transcorreu de maneira bastante satisfatória.

Entre as diversas ações preparatórias, foi desenvolvida pela Companhia de Água e Esgotos (CEDAE) do Rio de Janeiro uma vultuosa obra de manutenção nas instalações de tratamento de água do Guandu (MENDONÇA, 2007). Tal manutenção paralisou, por um longo período, o abastecimento de água para 80% da região metropolitana. Este acontecimento, por sua amplitude e possíveis desdobramentos, certamente deve ter sido precedido de um minucioso planejamento por parte das autoridades locais. Mesmo assim, pode ser utilizado para ilustrar a aplicação da rede de colaboração ora proposta. Supondo que a CEDAE publicasse eletronicamente sua intenção de paralisação das instalações do Guandu, diversos órgãos seriam capazes de desenvolver ações internas preventivas, tais como: os sistemas de informação dos hospitais poderiam automaticamente providenciar estoques adicionais de água; os bombeiros-militares estariam aptos a cancelar férias ou a reprogramar as escalas de serviço nos quartéis das áreas afetadas; e a defesa civil ativar seus mecanismos de gerenciamento de crises. Assim, devidamente informados e atuando de maneira pró-ativa, os diversos órgãos envolvidos estariam mais bem preparados para o enfrentamento dos óbices resultantes da interrupção no abastecimento de água.

Em outro exemplo, é público que são freqüentes as descobertas de tentativas de obtenção de aposentadorias fraudulentas junto ao Instituto Nacional de Seguro Social (INSS); algumas delas, inclusive, após longo período de tempo vigência e conseqüente prejuízo para os cofres da União. Entre as modalidades de fraudes, encontram-se aquelas nas quais são concedidos benefícios de aposentadoria por falecimento fictício (PREVNotícias, FRAUDE: Casal é preso por fraudar pensão de marido vivo, 2006) e por concessão indevida de benefícios (PREVNotícias, FORÇA-TAREFA: Operação investiga 236 benefícios suspeitos de fraude, 2006). Os sucessos dessas duas modalidades de fraudes seriam significativamente prevenidos se os diversos órgãos nelas envolvidos trocassem, em tempo hábil, informações relevantes entre si. Assim, caso os hospitais divulgassem eletronicamente as mortes ocorridas, os cartórios informassem as certidões de óbito registradas, ou os órgãos públicos publicassem também, de forma eletrônica, a posse de seus servidores, o INSS poderia utilizar tais informações para verificar sua própria base de dados em busca de operações fraudulentas, agindo, assim, de forma pró-ativa em benefício de toda sociedade.

Por fim, a seção de economia da versão eletrônica do jornal “O Estado de São Paulo”, em sua edição de 17.09.2007, apresentava uma matéria com o seguinte título: “Registro em hotel revela paradeiro de Cacciola em Mônaco” (OESP, 2007). Procurado pela Interpol por crime cometido contra o Sistema Financeiro Nacional, o ex-banqueiro brasileiro Salvatore Cacciola foi identificado pela polícia do principado através do *check-in* no hotel onde se encontrava hospedado, quando as informações por ele preenchidas no formulário de registro foram repassadas para a polícia local. Neste exemplo, caso implementada a rede de colaboração proposta, o sistema de informações do próprio estabelecimento hoteleiro poderia postar eletronicamente uma mensagem na rede de colaboração informando todos os detalhes dos hóspedes que se hospedaram.

VI - ESTRATÉGIA

A construção da rede de colaboração que instrumentaliza a arquitetura proposta viabilizará o intercâmbio eletrônico de informações e poderá promover e incrementar a colaboração entre as entidades.

O primeiro passo nesse sentido deve ser a nomeação de seu órgão gestor. Sugere-se que tal papel seja desempenhado pelo Gabinete de Segurança Institucional da Presidência da República uma vez que, conforme anteriormente visto, é o órgão responsável pela prevenção da ocorrência e pela articulação do gerenciamento de crises no Governo Federal.

Feito isso, deve-se constituir uma pequena equipe inicial, composta por representantes de órgãos estratégicos e hábeis profissionais de tecnologia da informação, com a missão de efetuar levantamentos preliminares junto a alguns órgãos, entidades e sistemas envolvidos e definir os detalhes que viabilizem a implementação de um protótipo-piloto para o sistema.

Com um conhecimento inicial da situação dos sistemas de informação, dos parques computacionais representativos, dos métodos, processos e procedimentos dos órgãos e entidades envolvidos, será possível criar a implementação de referência para este sistema. Neste momento, conforme anteriormente discorrido, sugere-se que a construção do modelo de referência utilize produtos de *software* livre e de código aberto, especialmente os *softwares* básicos responsáveis pela sua infra-estrutura de comunicação, os gerenciadores de filas.

Quando estiver disponível, o sistema deverá entrar em operação piloto. Nesta etapa, grande ênfase deverá ser dada aos comentários dos usuários selecionados e na elaboração dos manuais de operação, especialmente quanto aos procedimentos para implementação da conexão entre os sistemas de informação dos órgãos participantes e a rede de

colaboração. Também devem ser documentados os procedimentos para definição e formatação das mensagens (metamensagens) que poderão trafegar na rede de colaboração. Além disso, é importante que se crie um modelo ilustrativo de como um conjunto qualquer de mensagens pode configurar um estado de atenção. Este exemplo deverá ser concebido de maneira a facilitar o entendimento geral da operação da rede e facilitar, aos demais órgãos, o entendimento de como se criam e se operacionalizam os estados de atenção.

Encerrada a etapa de operação piloto, o órgão gestor do sistema de colaboração poderá planejar a inclusão dos demais órgãos intensificando, portanto, a colaboração, a troca de informações e o comportamento pró-ativo da administração.

VII - CONCLUSÃO

Este trabalho apresentou uma solução sistêmica para um problema real e impactante, diuturnamente enfrentado pela população brasileira: o da baixa troca de informações e de colaboração entre os diversos órgãos da administração pública.

Com este propósito, foram inicialmente delineadas as motivações para a elaboração deste trabalho, seguidas de uma revisão teórica que esclarece os conceitos necessários ao desenvolvimento da solução.

Seguiu-se, então, a proposta de uma arquitetura de *software* para a construção de uma rede de colaboração; apresentaram-se alguns exemplos de aplicação e, por fim, uma estratégia para sua implementação.

A proposta, demonstrada neste trabalho pode, efetivamente, incrementar o intercâmbio de informações entre os órgãos da administração pública. Sua arquitetura permite que seja elaborada sem a necessidade da criação de um órgão especialmente dedicado ao serviço de promoção da integração. Espera-se, também, que sua construção não necessite de vultosos recursos, pois;

a) poderão ser utilizados produtos de *software* livres e de código aberto;

b) não requer a substituição dos sistemas de informação já implementados nos diversos órgãos; e

c) reutiliza a infra-estrutura de tecnologia da informação atualmente disponível. Os riscos tecnológicos envolvidos no seu projeto também são mitigados, uma vez que utiliza soluções de *software* maduras e amplamente utilizadas nos mais diversos ramos da indústria.

A implementação da rede de colaboração abre ainda um amplo leque de oportunidades. Novos trabalhos podem explorar a análise semântica

das mensagens trocadas entre os órgãos mediante a rede de colaboração e, portanto, criar mecanismos eficazes para a determinação de cenários, especialmente daqueles que comprometam a estabilidade institucional do país, e neles atuar.

Assim, apesar de igualmente verificado em outros países, como foi apresentado no relatório *The 9/11 Commission Report*, o problema da baixa troca de informações produz efeitos de maior magnitude no Brasil, visto que aqui encontramos uma maior demanda por serviços públicos, oferecida por uma população mais carente, frente a uma disponibilidade de recursos significativamente menor. Logo, torna-se imperioso incrementar o uso eficiente e eficaz dos recursos de que o estado dispõe para atender às demandas sociais. Portanto, a construção da rede de colaboração, nos moldes da arquitetura aqui proposta, deverá viabilizar um cotidiano mais colaborativo para os órgãos, resultando em um melhor aproveitamento dos recursos disponíveis, e intensificar a velocidade de resposta e pró-atividade do estado perante os problemas, as necessidades e os desafios enfrentados pela população brasileira.

Referências:

Agência de Notícias do Ministério da Previdência Social. *FORÇA-TAREFA: Operação investiga 236 benefícios suspeitos de fraude*. Disponível em: <<http://www.previdenciasocial.gov.br/agprev/MostraNoticia.asp?Id=23019&ATVD=1&xBotao=1>>. Acesso em: 27 set. 2007.

Agência de Notícias do Ministério da Previdência Social. *FRAUDE: Casal é preso por fraudar pensão de marido vivo*. Disponível em: <http://www.previdencia.gov.br/AgPREV/agprev_mostraNoticia.asp?Id=23626&ATVD=1&DN1=10/05/2006&H1=17:11&xBotao=0>. Acesso em: 27 set. 2007.

ARMOUR, P. G.. *The Five Orders of Ignorance*. Communications of the ACM, 43.

CUELLO, K. A.. Integração do sistema de emissão de vistos dos Estados Unidos da América. Rio de Janeiro: 2007. (Entrevista ao Autor em 2 de outubro.)

DAVIES, S., & BROADHURST, P. *Websphere MQ V6 Fundamentals*. Disponível em: <<http://www.ibm.com/redbooks>>. Acesso em 18: de agosto de 2007.

ENDREI, M., MARTIN, K., & SADTLER, C. (2004). *Application Integration Patterns*. Disponível em: <<http://www.ibm.com/redbooks>>. Acesso em 18: de agosto de 2007.

INJEY, F., DONG, E., KAPLAN, R., & KRISHNAMURTHI, J. *Using XML on z/ OS and OS/390 for Application Integration*. New York: IBM Corp., 2001.

KONTIO, M. (2 de fevereiro de 2005). *Architectural Manifesto: Designing Software Architectures, Part 5*. Disponível em: <<http://www.ibm.com/developerworks/wireless/library/wiarch11/#resources>>. Acesso em: 17 set. 2007.

KRUCHTEN, P. *Architectural Blueprints: The “4+1” View Model of Software Architecture*. Disponível em: <<http://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf>>.

LEFFINGWELL, D., & WIDRIG, D. *Managing Software Requirements: A Unified Approach*. 1 ed. Boston: Addison-Wesley, 2000.

McCOY, D. W.; PEZZINNI, M.; SCHULTE, R. W.; THOMPSON, J.; LHEUREUX, B. J.; SINUR, J.; et al. *Hype Cycle for Application Integration and Platform*. Stamford: Gartner, 2003.

MENDES, A. *Arquitetura de Software: desenvolvimento orientado para arquitetura*. Rio de Janeiro: Campus, 2002.

MENDONÇA, A. V. *Ações da Ceda e para o Pan vão paralisar Estação do Guandu*. Disponível em G1: <<http://g1.globo.com/Noticias/Pan2007/0,,MUL30110-8610,00.html>>. Acesso em: 27 de agosto de 2007.

NASSUR, C. C. *Sistema de Informação para a CCAI*. Disponível em: <<https://extranet2.camara.gov.br/servidor/servicoseinf/camaraideias/bancoideias/ideiasapresentadas2002.html>>. Acesso em: 20 jul. 2007.

O Estado de São Paulo Digital. Disponível em: <http://www.estadao.com.br/economia/not_eco53090,0.htm>. Acesso em: 17 set. 2007.

PRESSMAN, R. S. *Engenharia de Software*. Trad. G. Costa, Ed., & R. D. Penteadó. São Paulo: McGraw-Hill, 2006.

SEI, S. E.. *Software Engineering Institute*. Disponível em: <http://www.sei.cmu.edu/architecture/published_definitions.html#Modern>. Acesso em: 13 de agosto de 2007.

SWITHINBANK, P., BANDARU, S., CROOKS, G., FERRIER, A., KRISHNASWAMY, R., HE, J., et al. *Connecting Enterprise Applications to WebSphere Enterprise Service Bus*. 1 ed.. Armonk, NY: IBM Corp, 2007.

The National Commission on Terrorist Attacks. *The 9/11 Commission Report*. Washington: 2004.

W3C, W. W. *Extensible Markup Language (XML)*. Disponível em: <<http://www.w3c.org/xml>>. Acesso em: 28 ago. 2007.

WACKEROW, D. *MQSeries Primer*, 34. New York: IBM Corp., 1999.