

# Arquitetura Orientada a Serviços para o Suporte a Interoperabilidade de Aplicações de C2

Breno Oliveira de Barros, brenobarros@gmail.com, Eduardo Martins Guerra, guerraem@gmail.com, Alisson Sandes Palmeira, alisson@cds.eb.mil.br

Stefanini IT Solutions – Belo Horizonte/MG, Instituto Tecnológico de Aeronáutica - ITA – São José dos Campos/SP, Centro de Desenvolvimento de Sistemas – Departamento de Ciência e Tecnologia – Exército Brasileiro – Brasília/DF

**Resumo** — Este trabalho tem por objetivo apresentar uma solução baseada em arquitetura orientada a serviços (SOA) para incrementar a interoperabilidade no contexto do Sistema Militar de Comando e Controle (SISMC<sup>2</sup>). Para tal, o artigo procura demonstrar como a arquitetura proposta respeita a autonomia das Forças Armadas, no tocante aos seus sistemas de Comando e Controle, ao passo que contempla as necessidades do SISMC<sup>2</sup> no nível estratégico. Por fim, o documento relata as principais validações realizadas, apontando para trabalhos futuros considerações para sua utilização nos níveis operacional e tática.

**Palavras-Chave** — SOA, Interoperabilidade, Comando e Controle.

## I. INTRODUÇÃO

Para que seja possível a condução de operações militares conjuntas com eficiência e eficácia, é necessário que as informações provenientes de mar, terra e ar fluam entre os participantes com oportunidade, segurança e coerência semântica, provendo aos comandantes a consciência situacional o mais próximo possível do dito tempo real. Nesse cenário dinâmico, complexo e heterogêneo, cresce de importância a interoperabilidade entre os sistemas de Comando e Controle (C2).

O Ministério da Defesa, ciente dessa necessidade, vem investindo em diversos programas para aumentar o nível de interoperabilidade do Sistema Militar de Comando e Controle (SISMC<sup>2</sup>). O presente trabalho é fruto de um projeto de pesquisa e desenvolvimento entre a Stefanini e FINEP, que tem por objetivo de estabelecer uma solução de interoperabilidade para a integração dos sistemas de C2 através de um conjunto de provas de conceito que atendam os requisitos propostos pelo Ministério da Defesa e pelas Forças Armadas.

A solução prévia ao presente trabalho limitava-se a enviar informações das aplicações a um sistema central em formatos proprietários. Além disso, essas comunicações se davam em um único sentido, informações das Forças para o Ministério da Defesa, e também não eram totalmente automatizadas. Como consequência, a integração das aplicações era deficitária e a colaboração bidirecional para a gestão das operações militares não era possível, impedindo o Comando Conjunto de exercer sua autoridade de uma forma mais eficiente.

Uma das propostas apresentadas para solução desse problema é a adoção de uma arquitetura orientada a serviços (SOA) [3] que visa transformar a camada heterogênea atual em uma camada homogênea, tanto em termos de formato quanto em termos tecnológicos. Essa arquitetura possibilita a exposição de funcionalidades na forma de serviços interoperáveis baseados em padrões abertos de mercado, e tem como boa prática o estabelecimento de um modelo comum de dados para troca de informações. Devido a requisitos da área de defesa, somente a adoção de SOA não é suficiente, gerando a necessidade de uma especialização desse paradigma arquitetural.

O objetivo do presente trabalho é apresentar uma arquitetura de integração baseada ao SOA, de forma a respeitar a autonomia de cada força perante a suas aplicações, representando um incremento de interoperabilidade no SISMC<sup>2</sup>. Em outras palavras, a solução proposta é não intrusiva, de forma a permitir que cada Força possa implementar os seus serviços de forma independente e respeitando as redes privadas e as políticas de segurança existentes [1]. Isso permite que cada serviço seja disponibilizado apenas internamente ou para reuso por aplicações de outras Forças.

O presente trabalho está estruturado da seguinte forma: a seção 2 descreve com maior detalhe os requisitos considerados; a seção 3 oferece uma visão geral da arquitetura proposta; a seção 4 descreve com mais detalhes as estratégias de implementação dos componentes arquiteturais; a seção 5 descreve as abordagens empregadas para identificação dos serviços a serem implementados, enquanto a seção 6 relata os testes já realizados. Finalmente, a seção 7 conclui o trabalho e aponta os passos futuros.

## II. REQUISITOS

Os requisitos a serem contemplados pela solução arquitetural baseada em SOA foram concebidos através da fusão dos requisitos normalmente encontrados em projetos de integração de sistemas [6] com requisitos específicos do segmento de Defesa. Isso diferencia o problema a ser resolvido dos já existentes no mercado e em outros segmentos públicos e privados.

A arquitetura de integração está apoiada em uma série de funcionalidades que são implementadas em componentes localizados entre as aplicações, dando robustez à solução.

Devido a essa característica, esse grupo de componentes é chamado de Infraestrutura Intermediária, e reside aí a principal diferença entre uma arquitetura SOA tradicional, como fica evidenciado nos requisitos não-funcionais. As próximas subseções apresentam os requisitos funcionais e não funcionais dessa infraestrutura. Esses requisitos foram definidos com o objetivo não só de atender as necessidades de interoperabilidade, mas também de viabilizar uma fácil adoção do mesmo pelos sistemas das Forças.

1) Requisitos Funcionais

i) A Infraestrutura Intermediária deve ser capaz de abstrair a **Comunicação Síncrona e Assíncrona** entre consumidores e provedores de serviços. Por exemplo, considere um dado serviço exposto por um sistema de C2 configurado como síncrono que está retardando a resposta e influenciando a experiência do usuário no sistema consumidor. Este serviço deverá poder ser configurado para responder de forma assíncrona em tempo de execução, sem a necessidade de modificações no sistema que o provê.

ii) A Infraestrutura Intermediária deve permitir que um único consumidor de serviço interaja com um único provedor de serviço (**Integração Um-para-Um**) ou com vários (**Integração Um-para-Muitos**), de forma transparente.

iii) A Infraestrutura Intermediária deve possibilitar a criação de novos serviços a partir da **composição de serviços** já existentes, oferecendo aos sistemas C2 um comportamento adaptativo com diminuição do tempo de resposta à mudanças.

iv) A Infraestrutura Intermediária deve ser capaz de permitir o **agendamento de processos**, de forma a atender aos cenários em que seja necessária a programação de processamentos em intervalos de tempo pré-definidos, ainda que, *a priori*, as informações sejam processadas em tempo real.

v) A Infraestrutura Intermediária deve oferecer total **transparência de localização** dos provedores de serviços, permitindo que os sistemas de C2 móveis também sejam integrados nas operações conjuntas.

vi) A Infraestrutura Intermediária deve estabelecer um **modelo único de representação dos dados** a serem tramitados, provendo uma semântica de linguagem franca entre todos os sistemas de C2. Este requisito está baseado em um padrão para integração de aplicações denominado Esquema Canônico [7] ou Modelo Canônico de Dados [6]. Com isto, se uma nova aplicação necessita ser incorporada à Infraestrutura Intermediária, apenas uma transformação do seu modelo de dados específico para o modelo único será necessário, conforme pode ser observado na Fig. 1.

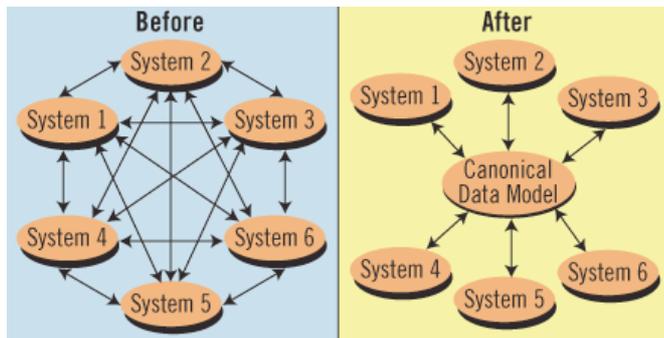


Fig. 1: Exemplo simplificado de uso de modelo canônico, antes e depois [18].

2) Requisitos Não-Funcionais

i) A Infraestrutura Intermediária deve ser capaz de suportar as políticas de **segurança** de cada Força, de forma não intrusiva, mantendo a autonomia de decisão do controle de acesso de cada subsistema envolvido, proporcionando maior descentralização e facilidade de manutenção.

ii) A Infraestrutura Intermediária deve possuir **alta disponibilidade e robustez** de forma a manter seu funcionamento independente de falhas que possam acontecer nas aplicações que estão participando da integração. Deve ainda continuar processos interrompidos em sua reinicialização caso seu funcionamento seja interrompido por questões fora de seu controle, como quedas de energia.

iii) A Infraestrutura Intermediária deve prover **alta capacidade e desempenho** para suportar o processamento de grandes volumes de dados e requisições, sem afetar a experiência de usabilidade por parte de seus usuários.

iv) A Infraestrutura Intermediária deve ser **escalável** com o intuito de manipular uma porção crescente de trabalho de forma uniforme, ou estar preparado para o crescimento da mesma.

v) A Infraestrutura Intermediária deve ser facilmente instalada e configurada, pelos indivíduos responsáveis, em qualquer plataforma de software base, sem transtornos, atendendo assim os requisitos de **instalabilidade, configurabilidade e portabilidade**.

vi) A Infraestrutura Intermediária deve oferecer um satisfatório nível de **manutenibilidade**, de forma que sua manutenção, evolutiva e/ou corretiva, seja realizada sem gerar impactos nos artefatos e ambientes já desenvolvidos e testados.

vii) A Infraestrutura Intermediária deve ser **extensível** de forma a receber incrementos de novos componentes tanto de hardware como de software. Os componentes da infraestrutura intermediária devem possuir uma interface bem definida, de forma a permitir que sejam intercambiáveis por outros similares.

viii) A Infraestrutura Intermediária deve ser desenvolvida com a ampla utilização de **Softwares Livres**, aderindo aos **padrões abertos de mercado** compatíveis com os **Padrões de Interoperabilidade de Governo Eletrônico (e-PING)** [16], estabelecido pelo Governo Federal.

III. DESCRIÇÃO DA ARQUITETURA

A arquitetura proposta para atendimento dos requisitos técnicos e não-funcionais é segregada em quatro macro camadas, como pode ser visto na Fig. 2: Camada de Integração; Camada de Serviços; Camada de Conectividade; e Camada de Coreografia dos Processos de Negócios.

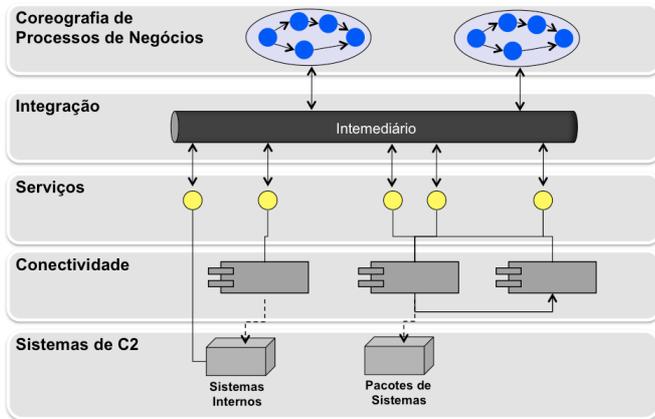


Fig. 2: Visão Geral da Arquitetura

A arquitetura a partir daqui descrita, já se encontra disponível nas dependências do Ministério da Defesa.

### 1) Camada de Conectividade

Esta camada tem como objetivo traduzir os protocolos das linguagens de programação específicas, e em alguns casos proprietárias, dos sistemas de C2 para os protocolos de padrão aberto. Além disso, os componentes internos desta camada visam a traduzir as representações de informação internas de cada linguagem (por exemplo: instância de objetos em memória nos casos de linguagens orientadas a objetos) em uma representação estruturada que possa ser facilmente portátil em diversas plataformas, e que esteja aderente a um Modelo Canônico de Dados [7].

Os componentes internos desta camada são comumente denominados de adaptadores (*adapters*) ou conectores (*connectors*).

### 2) Camada de Serviços

Esta camada representa as funcionalidades dos sistemas C2 na forma de serviços interoperáveis, que estão disponíveis para serem consumidos nas operações conjuntas. Esses serviços podem ser expostos através dos componentes internos da Camada de Conectividade ou serem desenvolvidos diretamente nesta camada.

### 3) Camada de Integração

A camada de Integração contempla boa parte dos requisitos funcionais e não-funcionais da Infraestrutura Intermediária, oferecendo as seguintes capacidades: roteamento inteligente; validação e transformação de mensagens; configuração do estilo de comunicação entre os serviços; declaração das políticas de segurança; gerência de

transações; ferramentas de monitoramento dos requisitos de qualidade dos serviços (Quality of Service - QoS).

### 4) Camada de Coreografia de Processos de Negócios

Esta camada tem por objetivo orquestrar os serviços expostos em uma sequência lógica (fluxo) para realização de um conjunto de ações, oferecendo estruturas para: controles de decisões, execuções paralelas; iterações, notificações e intervenções humanas.

É nesta camada que o requisito de Composição de Serviços é atendido, e onde os processos das operações conjuntas são implementados.

## IV. IMPLEMENTAÇÃO DA ARQUITETURA

A arquitetura da Infraestrutura Intermediária foi projetada de acordo com a especificação JSR-208 Java Business Integration (JBI) [11], que define um framework para construção e hospedagem de componentes plugáveis para integrações de sistemas, atendendo principalmente os requisitos de **instabilidade, configurabilidade, manutenibilidade, extensibilidade e portabilidade**.

Basicamente, a arquitetura JBI é constituída por três componentes, que podem ser visualizados na Fig. 3. São eles:

1) Binding Components (BC): Componentes que provêm um transporte de mensagens independentes de protocolos. Em outras palavras, sua função é transformar a camada heterogênea de protocolos dos sistemas alvos da integração em uma camada homogênea.

2) Service Engines (SE): São componentes que podem tanto atuar como Proxy Services [7] para serviços expostos pelos sistemas alvos da integração, quanto implementar funcionalidades específicas, tais como: orquestração de serviços, transformação de mensagens, motor de regras de negócios, dentre outros.

3) Normalized Message Router (NMR): Componente central para roteamento das mensagens a serem intercambiadas entre consumidores e provedores de serviços.

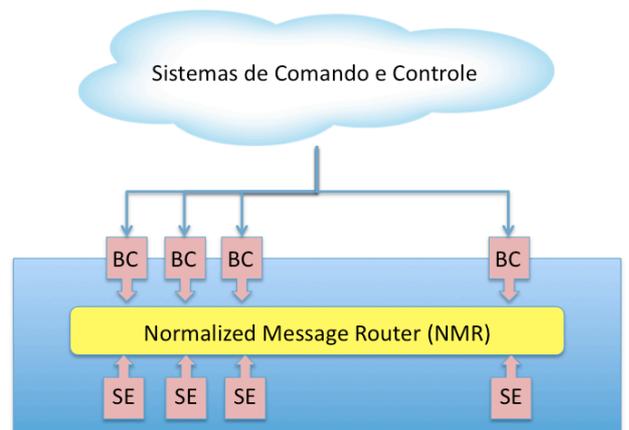


Fig. 3: Arquitetura JBI

Com isso, visando a acelerar a implementação da arquitetura e seguir o requisito de utilização de **Softwares Livres** o desenvolvimento da Infraestrutura Intermediária adotou como base a ferramenta OpenESB, com seus componentes JBI sendo distribuídos nas camadas descritas na seção III:

#### 1) Camada de Conectividade

É nesta camada que os componentes de *binding* (BC) da especificação JBI foram largamente utilizados, principalmente para os sistemas de C2 que não possuíam suas funcionalidades expostas na forma de *web services*.

Pelas características atuais dos sistemas de C2, os principais componentes de *binding* utilizados foram:

i) HTTP BC: provê conectividade para consumo de serviços publicados em protocolos HTTP ou HTTPS utilizando mensagens SOAP/XML.

ii) File BC: fornece conectividade com os sistemas de arquivos tanto para leitura quanto escritas de arquivos de dados. Esse componente foi muito importante para integração de sistemas mais antigos, cuja única interface de comunicação disponível é via arquivos texto.

iii) JDBC BC: provê conectividade com os mais variados Sistemas Gerenciadores de Banco de Dados, transformando tanto as operações SQL quanto Stored Procedures em *web services*.

Por serem de código aberto, esses componentes de *binding* foram estendidos para, além de executarem suas funções básicas, também transformarem os dados oriundos dos sistemas C2 em mensagens, no formato XML, seguindo a sintaxe e semântica do **modelo único de representação dos dados** definido, o JC3IEDM.

O JC3IEDM consiste em um modelo comum destinado ao intercâmbio de informações entre sistemas de comando e controle, em todos os níveis, mantido pelo organismo denominado *Multilateral Interoperability Programme* (MIP) [9]. Embora o modelo tenha sido concebido para interoperabilidade baseada em bancos de dados, há uma linha de pesquisa em aberto voltada justamente para a troca de informações baseada em *web services*. O esquema XML utilizado foi o WS-OO proposto pelo MIP, com pequenas adaptações [10].

#### 2) Camada de Serviços

Os serviços expostos nesta camada devem estar aderentes às diretrizes *Web Services Interoperability Organization* (WS-I) [14] e ao padrão e-PING.

Com isso, esta camada foi implementada como um repositório de serviços, com a utilização da ferramenta Apache JUDDI, que consiste em uma implementação de código aberto do padrão UDDI [12] provendo um modelo para descrição e publicação de serviços.

Além disso, é nesta camada que as implementações para suportar o requisito de **transparência de localização** estão abstraídas, com os recursos de descoberta de serviços providos pelo padrão UDDI.

#### 3) Camada de Integração

Para facilitar a implementação desta camada, a mesma foi subdividida de acordo com o padrão estrutural VETRO [15], que organiza os componentes de acordo com as etapas de tratamento das mensagens trafegadas em um cenário de integração. Essas mensagens são relativas a Validação, Enriquecimento, Transformação, Roteamento e Operação das mensagens trafegadas.

Vale destacar que as etapas de Enriquecimento e Transformação são opcionais dentro da arquitetura proposta. Pois como está sendo adotado um modelo canônico de dados baseado no JC3IEDM [9], as mensagens já tramitam pela Infraestrutura Intermediária com a estrutura e conteúdos mínimos desejados.

Para a etapa de Validação, foi implementado um componente para validar se a estrutura das mensagens estão de acordo com a estrutura do modelo canônico JC3IEDM (Validação Estrutural). Adicionalmente, foi implementado um componente que valida as credenciais de autenticação e autorização para garantir que todas as mensagens possuem permissão de trafegar pela Infraestrutura Intermediária (Validação de **Segurança**).

Para as etapas de Enriquecimento e Transformação, quando necessárias, será utilizado o componente *JBI XSLT Service Engine*, de código aberto, que provê o suporte tanto para enriquecimento quanto de transformação dos conteúdos das mensagens.

A etapa de Roteamento foi implementada com a extensão do componente *JBI Camel Service Engine*, de código aberto, que além de dar suporte aos padrões clássicos de integração [6], foi estendido para também implementar as **integrações Um-para-Um e Um-para-Muitos, tanto síncrona quanto assíncrona**.

Por fim, a etapa de Operação constitui na execução dos serviços providos pelos sistemas C2.

#### 4) Camada de Coreografia de Processos de Negócios

Para a implementação desta camada, visando a suportar a execução dos processos das operações conjuntas através da **composição de serviços**, foi utilizado o componente de código aberto denominado *JBI BPEL Service Engine*, da ferramenta OpenESB.

Esse componente foi escolhido pois oferece o suporte à linguagem BPEL (*Business Process Execution Language*) [13], que permite descrever um processo de negócio com o auxílio de recursos visuais, capaz de interagir com os serviços expostos pela Camada de Serviços (*web services*).

### V. ABORDAGENS ADOTADAS

Para habilitar o paradigma SOA com mais facilidade neste projeto, foi adotada uma combinação das abordagens top-down e bottom-up.

Enquanto uma equipe de analistas de processos de negócios segue a abordagem *top-down*, com foco no levantamento dos processos que dão suporte à condução de operações conjuntas, uma equipe composta de arquitetos,

analistas de sistemas e desenvolvedores realiza a abordagem *bottom-up*, priorizando a interoperabilidade, baseada em serviços, dos sistemas de C2 que já participam do SISMC<sup>2</sup>.

### 1) Abordagem Top-down

Somente a definição de uma arquitetura adequada não é suficiente [5]. Um mapeamento dos processos das operações conjuntas, direcionado pelas práticas da disciplina Business Process Management (BPM) [4] faz-se também necessário.

A abordagem top-down, direcionada pela disciplina de BPM, visa aplicar à arquitetura orientada a serviço um alinhamento maior com as expectativas e metas organizacionais para as operações conjuntas.

Basicamente, essa abordagem é constituída pelos seguintes passos [17]:

i) Identificar as Metas Organizacionais: visa a mapear quais são as ambições do Ministério da Defesa em termos de C2. Por exemplo, incrementar o nível de Consciência Situacional em algum grau e em um período de tempo específico é uma meta;

ii) Mapear as restrições atuais para atendimento das metas identificadas: seguindo uma análise causa-efeito, este passo tem por objetivo descobrir quais são as principais restrições que impedem que as metas sejam alcançadas;

iii) Identificar os Processos que devem atender as Metas: visando a mapear e modelar os processos atuais das operações conjuntas, guiados pela Doutrina Militar de C2, que deverão satisfazer as metas;

iv) Decompor os Processos: objetiva o estabelecimento de granularidade um para um entre os processos e metas. Ou seja, caso um processo atenda duas ou mais metas, o mesmo será decomposto em sub-processos até que cada um atenda a uma meta específica, evitando assim que uma mudança em um dado processo interfira em mais de uma meta;

v) Adequar/Melhorar os Processos: com base nas restrições identificadas, propor soluções aos processos de forma que tais restrições sejam eliminadas para o completo alcance das metas. Os processos somente estarão alinhados com a estratégia organizacional se os mesmos executarem atividades capazes de atingir as metas no período de tempo estimado.

vi) Definir métricas para monitoramento e controle dos processos: visa a estabelecer as formas de medição de cada processo, sub-processo e atividade;

vii) Habilitar indicadores de desempenho: com base nas métricas estabelecidas, implementar indicadores para a avaliação do desempenho dos processos, de forma que seja possível saber se as metas estão sendo alcançadas, municiando os esforços de melhoria contínua nos processos e atividades;

viii) Mapear os serviços que automatizarão os processos: para cada atividade dos processos e sub-processos, verificar se a mesma pode ser automatizada com algum serviço já existente identificado na abordagem Bottom-up. Alternativamente pode ser definido se um dado serviço já existente precisa ser modificado ou não para oferecer

máximo grau de reuso pelos processos das operações conjuntas;

ix) Otimizar os processos: à medida que os serviços estiverem executando os processos mapeados, são realizadas medições nos processos, com base nos indicadores identificados. Nesse momento, é possível diagnosticar possíveis desvios nos processos, os quais poderão ser minimizados através de adequações simples, sem que seja necessário alterações bruscas nos mesmos.

### 2) Abordagem Bottom-Up

A abordagem bottom-up visa avaliar os sistemas internos de C2 do Ministério da Defesa e das Forças Armadas que participam das operações conjuntas, para identificar os potenciais para intercâmbio de dados.

Basicamente, essa abordagem tem como foco mapear as funcionalidades existentes nos diversos sistemas que podem ser reutilizadas, reduzir o número de sobreposições ou redundância entre elas, remover as interfaces de integração do modelo tradicional ponto-a-ponto e realizar a adequação ao modelo canônico de dados JC3IEDM.

Alem disso, essa abordagem visa identificar as restrições de infraestrutura e propor melhorias, se necessário, tais como largura de banda e intermitência das conexões de redes.

Com isso, o objetivo desta abordagem é mover as funcionalidades relevantes dos sistemas importantes para uma camada na qual possam ser facilmente reusados, compostos e integrados aos processos de C2, melhorando assim o grau de interoperabilidade.

## VI. TESTES DA ARQUITETURA

Para validar a arquitetura proposta, alguns testes contínuos vêm sendo realizados. Essa seção tem como objetivo apresentar as validações que já foram concluídas.

Nos primeiros seis meses do projeto, os testes foram realizados com a utilização de serviços simulados para verificar a comunicação síncrona e assíncrona. Foram simulados os cenários um para um, onde um consumidor comunicava-se com um único provedor, e um para muitos, onde um consumidor comunicava-se com vários serviços simultâneos de origens distintas.

No cenário um para muitos, também foram provadas algumas variações: o consumidor acionava os vários serviços provedores, cujas mensagens de respostas eram consolidadas em um única mensagens e todas as informações eram utilizadas; o consumidor acionava os vários serviços provedores definindo os critérios de seleção das mensagens de respostas, por prioridade ou por conteúdo relevante; o consumidor apenas acionava os serviços provedores e não aguardava nenhum retorno, para que estes executassem, cada um, suas ações específicas. Este último cenário foi denominado de *Fire-and-Forget*.

Entretanto, com o passar dos testes, foi percebida a necessidade de antecipar a adoção da arquitetura em cenários mais reais. Para tal, dois sistemas que atualmente participam das operações conjuntas foram escolhidos para iniciarem a adoção da arquitetura proposta: o Programa C2 em Combate

(C2Cmb), do Exército, e o Sistema de Planejamento Operacional Militar (SIPLM), do Ministério da Defesa.

Como esses dois sistemas já se comunicam hoje de forma unidirecional (sentido C2Cmb – SIPLM) baseado no envio de arquivos XML via FTP, ficaria mais fácil coletar os indicadores de performance para comparar a situação atual de integração com a nova situação provida pela arquitetura proposta.

Diante disso, foram selecionadas duas funcionalidades essenciais que precisavam ser expostas como serviços para tramitar informações entre os dois sistemas em tempo real: a consulta às informações da Situação das Forças Amigas, ou Nossas Forças, em dado Teatro de Operações; e a consulta às informações da Situação das Forças Inimigas, também em um dado Teatro de Operações.

Sendo assim, foram construídas na arquitetura dos dois sistemas uma camada de serviços, responsáveis por abstrair os protocolos das linguagens de desenvolvimento e modelos de dados internos de cada aplicação com os protocolos abertos utilizados na arquitetura proposta, SOAP/HTTP e extensões XML como WS-Addressing para comunicação assíncrona, e o modelo canônico JC3IEDM.

Até o momento, os testes em laboratório foram executados com sucesso e em julho deste ano a solução foi colocada em prática durante 16 dias consecutivos na operação conjunta Atlântico II, no Rio de Janeiro/RJ e Brasília/DF, com os seguintes resultados obtidos:

i) Tamanho médio trafegado por dia das mensagens XML do modelo canônico: 216.800,00 linhas, totalizando 42 mega bytes;

ii) Total de requisições para a Infraestrutura Intermediária, durante os 16 dias de operação conjunta: 281 requisições, sendo 211 com sucesso e 70 falhas;

iv) Tempo médio de resposta/por requisição: nos primeiros quatro dias da operação, o tempo médio foi de 1 minuto e 10 segundos. Já nos demais dias, o tempo médio foi de 2 minutos e 3 segundos;

iii) Motivo da falhas: instabilidade da rede física (64%); e problemas de resolução nos endereçamentos de rede (NAT) pelos sistemas participantes (36%).

Após esses resultados, algumas melhorias foram realizadas na Infraestrutura Intermediária para compactar as mensagens XML, com base na especificação WS-Encrypt, e implementar de um mecanismo de cache em segundo nível.

Em outubro deste ano, a solução passará por um novo teste.

## VII. CONCLUSÃO

Neste artigo foram apresentadas as experiências de adoção de uma solução baseada em arquitetura orientada a serviços (SOA) para atender aos requisitos de interoperabilidade do Sistema Militar de Comando e Controle, de modo a permitir o intercâmbio de informações entre os sistemas de C2 na condução de operações conjuntas.

O trabalho procurou demonstrar como a arquitetura proposta contempla os requisitos do SISM2, respeitando a autonomia das Forças tanto na gerência quanto na manutenção de seus sistemas de C2. Ainda procurou-se

destacar as principais decisões de projeto visando uma implementação aderente a padrões abertos, fortemente baseados em software livre e com a adoção de um modelo canônico para intercâmbio de informações.

Os testes realizados até o momento indicam que a solução está no caminho certo para se atingir o grau desejado de interoperabilidade dos sistemas de Comando e Controle, assunto este prioritário no âmbito do Ministério da Defesa. Ainda em fase de prova de conceito, a maturidade alcançada pela solução já permite uma primeira experimentação em campo, que ocorrerá durante a Operação Atlântico II.

Os próximos passos apontam para uma ampliação da prova de conceito, incorporando os sistemas de Comando e Controle da Força Aérea Brasileira (FAB). Além disso, é fundamental institucionalizar a abordagem de Business Process Management (BPM) para o alinhamento dos processos de condução de operações conjuntas à arquitetura proposta. Por último, mas não menos importante, há que se definir e habilitar as políticas e diretrizes de governança para gestão do ciclo de vida dos ativos de software envolvidos nos cenários de integração.

Como trabalho futuro, recomenda-se estudos aprofundados da solução no sentido de adaptá-la aos níveis operacional e tático, onde a largura de banda e a instabilidade dos enlaces são fatores extremamente limitadores e críticos.

## REFERÊNCIAS

- [1] M. Varchavsky, “Solução em Controle de Acesso para a Integração de Aplicativos de C2”. ITA, 2007.
- [2] Seleção Pública MCT/FINEP/Subvenção Econômica à Inovação 01/2007. Disponível em <[http://www.finep.gov.br/fundos\\_setoriais/subvencao\\_economica/editais/Selecao\\_Publica\\_Subvencao\\_2007\\_versao\\_final.pdf](http://www.finep.gov.br/fundos_setoriais/subvencao_economica/editais/Selecao_Publica_Subvencao_2007_versao_final.pdf)>
- [3] T. Erl, “Service-Oriented Architecture (SOA): Concepts, Technology, and Design”. Prentice Hall, 2005.
- [4] M. Weske, “Business Process Management: Concepts, Languages, Architectures”. Springer, 2007.
- [5] K. Mashruwala, “SOA Strategy and Execution is failing in many companies”. The Financial Express, 2008.
- [6] M. Fowler, “Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions”. Addison Wesley, 2003.
- [7] Thomas Erl, “SOA Design Patterns”, Prentice Hall, 2009.
- [8] W3C, “Extensible Markup Language (XML) 1.0 (Fifth Edition)”, W3C, 2008.
- [9] Multilateral Interoperability Programme (MIP), “The Joint Information Exchange Model”, MIP, 2009. Disponível em <[http://www.mip-site.org/040\\_Public\\_Documents.htm](http://www.mip-site.org/040_Public_Documents.htm)>, Acesso em: 14/05/2010.
- [10] Multilateral Interoperability Programme (MIP), “MIP XML Reference Material Download”, MIP, 2008. Disponível em <[http://www.mip-site.org/043\\_Public\\_Documents\\_XML\\_02.htm](http://www.mip-site.org/043_Public_Documents_XML_02.htm)>, Acesso em: 14/05/2010.
- [11] Java Community Process (JCP), “Java Business Integration (JBI) Final Release”, JCP, 2005. Disponível em <<http://jcp.org/aboutJava/communityprocess/final/jsr208/index.html>>
- [12] OASIS, “UDDI Spec Technical Committee Draft Version 3.0.2”, OASIS, 2004. Disponível em <<http://uddi.org/pubs/uddi-v3.0.2-20041019.htm>>
- [13] OASIS, “Web Services Business Process Execution Language Version 2.0”, OASIS, 2007. Disponível em <<http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>>
- [14] Web Service Interoperability Organization (WS-I), “Basic Profile Draft 1.1”, WS-I, 2006. Disponível em <<http://www.ws-i.org/Profiles/BasicProfile-1.1-2006-04-10.html>>

- [15] D. Chappell, “ESB Integration Patterns: An insider’s look into SOA’s implementation backbone”, SOA World Magazine, 2004. Disponível em <<http://soa.sys-con.com/node/46170>>
- [16] Ministério do Planejamento, Orçamento e Gestão, “Padrões de Interoperabilidade de Governo Eletrônico (E-PING)”, E-PING, 2006. Disponível em <<http://www.governoeletronico.gov.br/acoes-e-projetos/e-ping-padros-de-interoperabilidade/versoes-do-documento-da-e-ping>>
- [17] Y. Lederer, “Business Process Management Common Body Of Knowledge (BPM CBOK) v2.0”, ABPMP, 2009.
- [18] S. Hoberman, “Canonical Data Model“, Information Management, 2008. Disponível em <[http://www.information-management.com/issues/2007\\_50/10001733-1.html](http://www.information-management.com/issues/2007_50/10001733-1.html)>