

Modelação da Integração em Architecturas de Sistemas de Informação

André Vasconcelos

Instituto Superior Técnico, Lisboa, Portugal

andre.vasconcelos@ceo.inesc.pt

Miguel Mira da Silva

Instituto Superior Técnico, Lisboa, Portugal

mms@dei.ist.utl.pt

António Fernandes

Instituto Superior Técnico, Lisboa, Portugal

antonio.fernandes@gsi.inesc.pt

José Tribolet

Instituto Superior Técnico, Lisboa, Portugal

jose.tribolet@ceo.inesc.pt

Modelação da Integração em Arquitecturas de Sistemas de Informação

Resumo

No passado, na definição de arquitecturas de sistemas de informação (ASI) a integração era considerada uma questão de segunda ordem, dado que a maioria dos sistemas de informação (SI) eram construídos de raiz. Actualmente, com o aumento na variedade e número de SI em médias e grandes empresas, incluindo sistemas ERP, CRM, B2B, B2C, B2E, entre outros, a necessidade de integração é maior do que nunca. Para agravar esta questão, hoje em dia, a maioria das organizações pretende integrar os seus SI com SI de outras organizações.

Neste artigo propomos uma extensão a uma investigação anterior de forma a permitir a representação de ASI considerando a diversidade de cenários de integração entre SI, incluindo intra e inter organizações. Em particular a representação proposta permite modelar integração manual e automática e síncrona e assíncrona. Apresentamos também um exemplo que ilustra a proposta apresentada com necessidades de integração entre SI num caso real.

Palavras-chave: Arquitectura dos Sistemas de Informação, Integração de Sistemas de Informação, framework CEO, Arquitectura Empresarial, Integração de Aplicações Empresariais.

1. Introdução

A integração entre componentes de software foi sempre uma parte fundamental de qualquer sistema de informação. Recentemente, a sua importância tem crescido devido à necessidade de integrar diversos sistemas de informação, quer dentro das organizações quer entre organizações.

O movimento ERP dos últimos 10 anos – largamente impulsionado pelo euro e pelo *bug* do ano 2000 – não reduziu a necessidade de integração. Na verdade, talvez tenha até aumentado essa necessidade devido à introdução de ERP nas organizações que já estavam integradas. Além disso, para tirar partido do negócio electrónico é necessário integrar os vários sistemas de informação dentro da organização [Kalakota et. al. 2000].

No entanto, a investigação sobre arquitecturas de sistemas de informação (ASI) não tem prestado atenção suficiente à integração de sistemas de informação porque assume geralmente que uma única base de dados para toda a organização elimina a necessidade de integração. Até recentemente este pressuposto fazia sentido porque a maioria das organizações utilizavam sistemas proprietários baseados numa única base de dados.

Mesmo quando nos últimos anos a integração se transformou num problema – devido principalmente à aquisição a terceiros de sistemas de informação específicos – assumiu-se que o problema poderia ser resolvido substituindo todos os sistemas de informação por um único ERP.

À medida que mais e mais organizações começaram a instalar sistemas de informação incompatíveis entre si (incluindo ERP) para resolver problemas específicos tornou-se evidente que a solução nunca poderia ser baseada numa única base de dados ou mesmo num único ERP. Desta forma, a necessidade de integrar sistemas de informação não podia mais continuar a ser evitada devendo ser possível o seu pensamento, representação, análise e discussão tendo por base as arquitecturas de sistemas de informação.

Este artigo é baseado no trabalho anterior realizado pelo nosso grupo de investigação (Centro de Engenharia Organizacional, ou CEO) nesta área que complementamos com novas propostas

para incluir aspectos de integração numa arquitectura de sistemas de informação, incluindo um exemplo concreto para validar as nossas propostas.

O artigo começa com uma revisão das arquitecturas de sistemas de informação e depois apresenta a nossa própria *Framework CEO* que já tinha proposto alguns conceitos de alto nível para representar a integração. Em particular, um conceito chamado *IS Service* já podia ser usado para representar a integração entre dois blocos do mesmo sistema de informação. No entanto, ainda não tinha sido proposto como representar a integração quer ao nível tecnológico quer ao nível aplicacional – que são precisamente as principais contribuições deste artigo.

Após esta revisão apresentamos uma breve introdução aos conceitos mais importantes da integração de sistemas de informação, principalmente para mostrar que a integração é muito mais complexa e interessante que o famoso RPC. Sendo apenas uma chamada síncrona a um procedimento remoto, o conceito RPC tem mudado de nome (e tecnologia de base) ao longo dos tempos e actualmente designa-se *Web Services* [W3C 2001]. No entanto, embora as chamadas síncronas possam ser utilizadas para integrar módulos de software, são claramente inapropriadas para integrar sistemas de informação – especialmente aqueles pertencentes a organizações distintas.

Nesta altura do artigo, assumindo que são claras as limitações das actuais notações para representar arquitecturas de sistemas de informação, propomos então um conjunto de conceitos novos para descrever uma variedade de cenários de integração, nomeadamente *IT Integration Block* e *IT Integration Service*.

A principal novidade neste artigo é que o conceito de *Service* introduzido por nós em artigos anteriores deixa de estar limitado a serviços síncronos. Propomos ainda que a integração seja classificada de acordo com o nível de automatização (manual ou automático) e de acordo com o papel nessa integração (fonte ou alvo) para além de caracterizarmos os serviços de acordo com seu nível tecnológico, aplicacional e organizacional.

Finalmente, apresentamos e discutimos um exemplo concreto inspirado num projecto sobre segurança alimentar em que participamos actualmente. Este exemplo ilustra como os conceitos propostos ao longo do artigo podem ser utilizados na prática para representar a integração entre sistemas de informação, tanto ao nível dos sistemas de informação como ao nível das tecnologias de informação.

2. Arquitectura dos Sistemas de Informação

A Arquitectura dos Sistemas de Informação (ASI) representa a estrutura dos componentes dos sistemas de informação, suas relações, princípios e directrizes [Garlan et al. 1995], tendo por principal objectivo o suporte do negócio [Maes et al. 2000].

Nos anos 80, arquitectura de software (ASW) e ASI eram considerados sinónimos. Apenas nos anos 90 emergiu a necessidade para manipular conceitos que excediam a descrição de como um sistema é internamente constituído. A framework de Zachman (1987) pode ser considerado o primeiro sinal importante de que as ASW não eram suficientes.

Enquanto que as ASW representam os detalhes internos dos sistemas (utilizando, por exemplo, diagramas E-R e DFD), o foco da ASI é no suporte aos processos de negócio da organização [IEEE 1998, Zijden et al. 2000]. Recorrendo à noção metafórica de “cidade”, pode-se usar o conceito de “Urbanização dos SI” de forma a realçar a necessidade de modelos que guiem a evolução dos SI independentemente da tendências tecnológicas actuais [Sassoon 1998].

Uma ASI pode ser dividida em três níveis [Spewak et al. 1992]:

- Arquitectura Informacional (ou de Dados) – representa os principais tipos de dados que suportam o negócio;

- **Arquitetura Aplicacional** – define as aplicações necessárias à manipulação dos dados e suporte ao negócio;
- **Arquitetura Tecnológica** – representa as principais tecnologias usadas na implementação das aplicações e nas infra-estruturas que fornecem um ambiente para o funcionamento dos SI.

2.1. *Arquitetura Informacional*

O principal propósito da arquitetura informacional é a identificação e definição dos principais tipos de dados que suportam o desenvolvimento do negócio [Spewak et. al. 1992, DeBoever 1997]. Por exemplo, os dados (o suporte da arquitetura informacional) pode ser categorizada de acordo com diferentes dimensões, incluindo: primitiva vs. derivada, privada vs. pública, histórica vs. operacional vs. provisional (para mais detalhes ver [Inmon 1999]).

2.2. *Arquitetura Aplicacional*

O segundo nível arquitetural, a arquitetura aplicacional, define as principais aplicações necessárias para a manipulação dos dados e suporte ao negócio [DeBoever 1997]. Esta arquitetura define os principais componentes funcionais da arquitetura que asseguram acesso aos dados em tempo, formato e custo aceitáveis [Spewak et al. 1992], não devendo ter preocupações em relação à definição do software usado para a sua implementação.

Spewak propõe uma metodologia – *Enterprise Architecture Planning* (EAP) – para a definição da arquitetura aplicacional a partir da arquitetura informacional e dos requisitos de negócio [Spewak et al. 1992]. Mais recentemente, vários outros autores e organizações adaptaram a framework de Zachman e o EAP de Spewak (entre outras metodologias) para melhor responderem às suas necessidades, nomeadamente as propostas do Governo Federal Americano [Concil 1999], o Modelo Técnico de Referência DoD [Dod 2002], e a Framework de Arquitetura Empresarial do Departamento de Tesouro Americano [TEAF 2002].

2.3. *Arquitetura Tecnológica*

Esta arquitetura define as tecnologias que fornecem um ambiente para a construção das aplicações e para o seu funcionamento. A este nível, os principais conceitos tecnológicos são identificados, tais como as tecnologias a usar na implementação das aplicações, a comunicação entre componentes tecnológicos, as tecnologias de gestão dos dados, entre outros [Spewak et. al. 1992].

A nível tecnológico os EAB (*Enterprise IT Architecture Blueprints*) são uma referência base [Boar 1999]. Boar verificou que genericamente as arquiteturas tecnológicas não apresentavam uma representação repetível, coerente, não ambígua e facilmente perceptível, propondo um conjunto de representações (*blueprints*) de novas noções e símbolos, não suportados em quaisquer regras ou *standards*. Consequentemente os potenciais utilizadores sempre demonstraram relutância em usar estas propostas dado verem-se na obrigação de desenvolver um conhecimento e experiência elevados, mesmo antes de definirem qualquer arquitetura tecnológica.

2.4. *Comparação com as Arquiteturas de Software*

Nos anos 90 a arquitetura de software (ASW) apresentava preocupações similares. Em particular, não existia um consenso em relação aos conceitos da ASW. Em resposta, o IEEE criou uma *taskforce* que definiu um *standard* denominado “*Recommended Practice for Architectural Description of Software-Intensive Systems*” que forneceu a framework conceptual para a ASW [IEEE 1998].

Suportado neste *standard*, o *Open Group* propôs a framework TOGAF para o desenho e avaliação da ASI [TOGAF 2001]. Esta framework fornece não só uma metodologia para o desenvolvimento da ASI mas também uma taxonomia, princípios arquitecturais e *standards* para a ASI, sobretudo a nível tecnológico.

Adicionalmente, a TOGAF propõe um modelo técnico de referência que define uma taxonomia para uma descrição coerente, consistente e hierárquica dos serviços disponibilizados pela plataforma aplicacional tais como gestão dos dados, rede, sistema operativo, processamento de transacções e administração de sistema. A TOGAF apresenta ainda várias qualidades arquitecturais inerentes à definição da arquitectura tais como performance, disponibilidade, usabilidade, adaptabilidade e portabilidade.

Se considerarmos como objectivos a representação da ASI (aos diversos níveis) e conceitos de integração, a TOGAF apresenta várias limitações. A limitação mais relevante consiste no facto do foco desta framework ser sobretudo tecnológico, não endereçando as arquitecturas aplicacionais ou informacionais. Outra limitação reside no facto desta framework apenas apresentar princípios e noções, não representações concretas para a modelação da ASI. Isto torna a TOGAF interessante para reflectir em relação à ASI sobre um ponto de vista tecnológico, mas claramente inadequada para a modelação da ASI numa forma global e coerente.

2.5. A Framework CEO

De forma a endereçar a modelar todos os conceitos relevantes no seio da organização o Centro de Engenharia Organizacional (CEO) propôs a framework CEO [Vasconcelos et al. 2001], possibilitando a modelação da empresa usando um conjunto restrito de objectos de negócio. A framework CEO foi definida através de um perfil UML (para mais detalhes ver [UML 1997]) e está suportada em investigação recente de outros autores nomeadamente [Malone et. al. 1999, Eriksson et. al. 2000].

Apesar da framework CEO (2000), não podia ser usada para a modelação de todos os conceitos relevante na ASI, apresentando algumas extensões importantes que permitem representar as dependências entre o negócio e as tecnologias de informação. Os objectos de negócio definidos na framework são os «*goal*» para modelação da estratégia; «*process*» para modelação de processos de negócio, «*resource*» para modelação de recursos de negócio, e «*block*» para modelação de sistemas de informação. A framework CEO também assegura consistência, facilidade de uso e mecanismo de integridade com o objectivo de minimizar as discrepâncias entre as arquitecturas de negócio e de tecnologias e sistemas de informação.

Recentemente, os conceitos da framework CEO a nível dos SI foram mais explorados em [Vasconcelos et al. 2003a] e foi proposto um perfil UML [Vasconcelos et al. 2003b] que fornece as ferramentas conceptuais e notação para modelação da ASI a nível informacional, aplicacional e tecnológico. A Figura 1 apresenta os actuais conceitos nucleares da framework CEO (a nível da ASI).

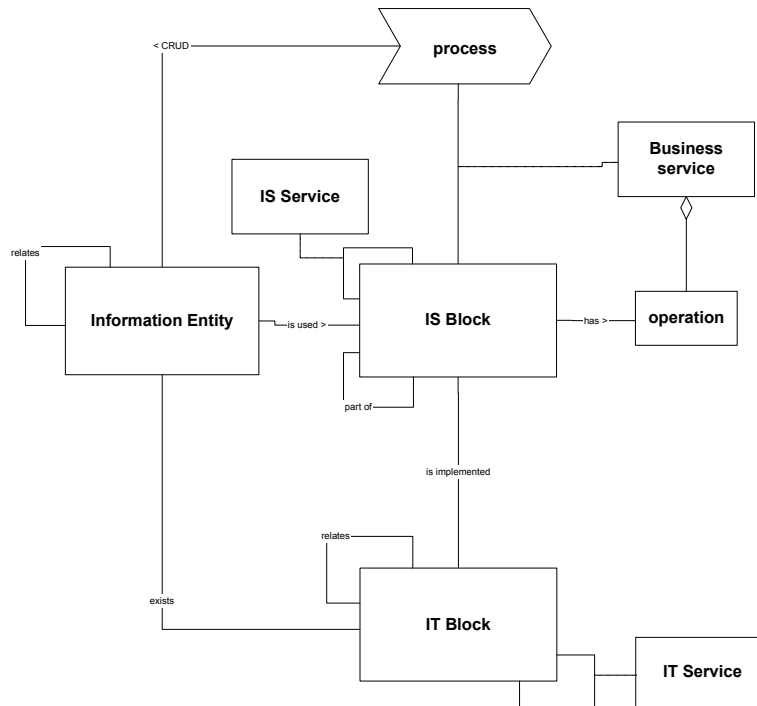


Figura 1. Extensões CEO ao Meta-modelo do UML para modelação da ASI [Vasconcelos et al. 2003b]

Os conceitos base no perfil da framework CEO são:

- Processo de Negócio – uma colecção de actividades que produzem valor para um cliente;
- Entidade Informacional – pessoa, lugar, coisa física ou conceito relevante no contexto do negócio e sobre o qual é possível e relevante (para a organização) guardar informação;
- Bloco Aplicacional («*IS Block*») – colecção de mecanismos e operações organizados de forma a manipular os dados;
- Bloco Tecnológico («*IT Block*») – infra-estrutura, plataforma aplicacional e/ou componente tecnológico/software que realiza (ou implementa) um (ou vários) blocos aplicacionais.

Do ponto de vista tecnológico os conceitos são (ver Figura 2):

- «*IT Infrastructure Block*» – representa os conceitos físicos e infra-estruturais tais como os nós computacionais (servidores, computadores pessoais, dispositivos móveis, etc.) e os nós não computacionais (por exemplo impressoras, redes) que suportam as plataformas aplicacionais.
- «*IT Platform Block*» – representa a colecção de serviços necessários para a implementação e funcionamento das aplicações.
- «*IT Application Block*» – representa a implementação tecnológica de um bloco aplicacional («*IS Block*»). A este nível é relevante considerar o tipo de Bloco tecnológico aplicacional (nomeadamente de apresentação, lógica, dado e coordenação), e os seus princípios tecnológicos (por exemplo, se é implementado usando componentes, módulos, ou objectos), entre outras características.

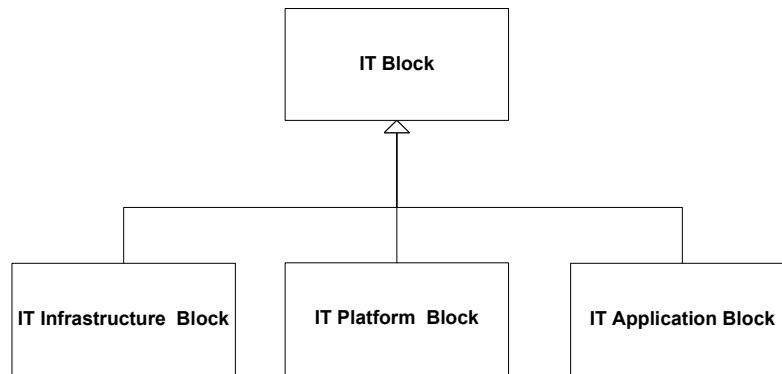


Figura 2. Metamodelo do bloco tecnológico («*IT Block*»)

É interessante notar que em [Vasconcelos et al. 2003b] já estão patentes alguns conceitos de integração. Particularmente o conceito de serviço definido enquanto uma agregação de operações fornecidas por um bloco arquitetural, organizado em três diferentes categorias:

- «*Business Service*» - coleção de operações fornecidas por blocos aplicativos («*IS Block*») que suportam um (ou vários) processo(s) de negócio;
- «*IS Service*» - conjunto de operações fornecidas por blocos aplicativos («*IS Block*») a outros blocos aplicativos;
- «*IT Service*» - representa os serviços tecnológicos disponibilizados por plataformas aplicativos (de acordo com as propostas [TOGAF 2001]).

O Serviço é um conceito basilar na representação da integração numa ASI e servirá de base às propostas apresentadas na secção 4.

2.6. Conclusão

A curta revisão apresentada aponta a inexistência de uma forma sistemática para representação dos conceitos de integração (apresentados na secção seguinte) numa ASI aos diferentes níveis (informacional, aplicativo e tecnológico), de forma a desenvolver subsequente inspeção e/ou simulação de vários cenários de negócio e/ou tecnológicos. Algumas recentes aproximações, das quais destacamos a framework CEO, fornecem um ponto de partida sólido para a modelação da ASI, apesar de não detalharem os conceitos de integração.

3. Integração de Sistemas de Informação

A integração de sistemas de informação – também designado EAI, de *Enterprise Application Integration* – esteve sempre presente no desenvolvimento de sistemas de informação. A popularidade dos ERP nos anos 90 foi em parte alimentada pela expectativa de eliminar a necessidade de integração, quando na realidade talvez tenha aumentado ainda mais o problema. Uma vez que um ERP nunca consegue substituir todos os sistemas de informação da organização, em particular os sistemas de informação específicos que suportam o negócio como a facturação, todos esses sistemas de informação devem ser agora integrados com o ERP. Aliás, a recente aposta da SAP na tecnologia NetWeaver [SAP 2003] mostra como os seus clientes exigem integrar facilmente o SAP com outros sistemas.

Embora integrar os vários sistemas de informação seja hoje em dia fundamental para qualquer organização, o grande desafio actualmente é integrar os sistemas de informação internos com os sistemas de informação externos dos seus clientes e fornecedores. Esta integração entre organizações (em Inglês normalmente designado *B2B Integration*) apenas estende a integração

para a cadeia de valor e tecnicamente é perfeitamente acessível – como temos visto nos diversos projectos de EDI na Internet, inclusive em Portugal. No entanto, em relação às arquitecturas de sistemas de informação levanta-se uma nova questão porque torna-se pela primeira vez necessário especificar onde termina a nossa organização mesmo que essas fronteiras sejam cada vez mais difíceis de definir com clareza [Linthicum 2001].

Por outro lado, embora existam diversos tipos de integração – ao nível das interfaces, dos métodos, dos dados, etc. – todos são baseados na troca de dados entre aplicações [Vernadat et.al. 1996]. As diferenças residem na forma como essa troca de dados é realizada, nos tipos dos dados que são trocados, nas garantias que são oferecidas, e assim por diante. Por exemplo, ao nível dos métodos são trocados dados entre duas aplicações enquanto ao nível dos dados são trocados dados entre duas bases de dados. Por isso, muito mais importante que especificar o “tipo de integração” como fazem certos autores, consideramos mais útil especificar as três ou quatro características mais importantes dessa integração.

Uma outra fonte de confusão nesta área é a enorme quantidade de *buzzwords* que foram criadas ao longo do tempo, particularmente nos últimos anos, sem corresponderem a verdadeiras mudanças tecnológicas. Por exemplo, nem para um especialista na área é fácil identificar as diferenças entre DDE, OLE, VBX, OCX, ActiveX, COM, COM+ e DCOM para citar apenas as tecnologias da Microsoft. Pelo contrário, embora a norma CORBA seja normalmente entendida como uma tecnologia na verdade trata-se apenas de uma arquitectura que deu origem a diversos produtos incompatíveis entre si.

Recentemente, os *Web Services* [W3C 2001] foram propostos com a promessa de revolucionar a integração entre sistemas de informação (tanto dentro como entre organizações) quando na realidade não passam do velhinho RPC agora de cara lavada.

O próprio XML, no qual os *Web Services* são baseados, é simplesmente uma linguagem normalizada para formatar dados à semelhança do ASN.1 [ASN.1 2003]. Por isso o XML resolve apenas uma pequena parte do problema da integração. Sem transacções, segurança, robustez ou desempenho, os *Web Services* podem ser usados para integrar aplicações pertencentes ao mesmo sistema de informação mas obviamente não são apropriados para integrar sistemas de informação, principalmente entre organizações [Mira da Silva 2003b].

Os *Web Services* utilizam uma norma chamada SOAP [Newcomer 2002] para trocar XML entre duas aplicações utilizando normalmente HTTP como protocolo de transporte. Infelizmente, o SOAP é basicamente síncrono e não suporta características importantes, tais como políticas de segurança, tipos normalizados de documentos, qualidade de serviço, definição de fluxos de trabalho, etc.

Apesar desta aparente simplicidade dos *Web Services* tem sido difícil colocar produtos que suportam a norma SOAP a trabalharem uns com os outros sem alterações de baixo nível. Tal como aconteceu com o CORBA na década passada, é provável que as novas normas para acrescentar mais funcionalidades ao SOAP apenas venham dificultar ainda mais a sua interoperabilidade.

Finalmente, sendo basicamente uma tecnologia síncrona de comunicação entre aplicações (equivalente conceptualmente ao RPC, DCOM e CORBA) é natural que venha a sofrer exactamente dos mesmos problemas [Mira da Silva 2000]. Por exemplo, é estranho que a própria Microsoft tenha desvalorizado os *Web Services* na próxima versão do Windows quando é um dos seus principais proponentes.

Apesar destes problemas, que fizemos questão de salientar como contraponto à euforia que se vive actualmente nesta área, a integração entre sistemas de informação tem um conjunto vastíssimo de características que vão muito para além do que é actualmente oferecido pelos *Web Services*. Por isso, a base deste artigo reside na constatação que as características mais importantes da integração devem ser especificadas na arquitectura de sistemas de informação.

Alguns exemplos das características da integração, sem nenhuma ordem em particular, são:

- A integração pode ocorrer ao nível dos dados, dos métodos, das interfaces, dos portais e dos processos – esta variedade representa basicamente a forma como a aplicação interage com a integração, embora idealmente a aplicação não tenha de ser alterada (nem sequer acedida) para ser integrada com outra aplicação;
- A integração pode ocorrer dentro de um computador, dentro de uma Intranet (rede interna de uma organização), dentro de uma Extranet (rede virtual na Internet com acesso restrito a certas organizações) ou na Internet pública – cada zona terá as suas próprias garantias de segurança, largura de banda, fiabilidade e robustez;
- A integração pode ocorrer dentro de um departamento, dentro de uma organização, dentro de um grupo económico, ao longo de uma cadeia de valor ou entre duas (ou mais) organizações completamente autónomas – dentro de uma organização é relativamente fácil fixar regras e tecnologias, mas torna-se mais difícil num grupo económico e normalmente impossível nos clientes e fornecedores.
- A integração pode ocorrer dentro do mesmo país, dentro da mesma comunidade de estados (como a UE) ou entre dois países com culturas muito diferentes – por exemplo, os certificados digitais emitidos nos Estados Unidos não podem ser utilizados para assinar facturas digitais na Europa;
- A integração pode ser síncrona ou assíncrona – a integração assíncrona não tem resposta imediata mas tem um desempenho mais elevado e escala muito melhor;
- A integração pode ser transaccional – garantindo que todas as etapas da integração (ou nenhuma) ocorrem e eventualmente assim estender o conceito de integração entre sistemas de informação;
- A integração pode oferecer vários níveis da segurança desde nenhuma até ao não repúdio de recepção – as garantias de segurança em geral são muito úteis mas complicam a integração, aumentam os custos e reduzem o desempenho, por isso devem ser usadas com moderação;
- A integração pode ser usada para trocar *bytes* (por exemplo, com TCP/IP), estruturas de dados mais ou menos complexas (em XML) ou documentos de negócio como encomendas e facturas de acordo com determinados formatos normalizados (em EDIFACT ou UBL) segundo determinados fluxos de negócio (ebXML) ou mesmo processos do negócio – a maioria dos projectos da integração actualmente são baseados na troca de XML entre aplicações, mas essa abordagem de baixo nível apenas pode ser usada nos documentos, fluxos e processos de negócio mais simples;
- A integração pode ser executada directamente entre dois (ou mais, através de *multicast*) sistemas de informação ou indirectamente entre vários sistemas de informação através de um intermediário – enquanto a maioria dos produtos para trocar mensagens assíncronas entre aplicações utilizam um intermediário para armazenar as mensagens ao nível da implementação, um *message broker* oferece inúmeros serviços de valor acrescentado como por exemplo conversão de dados entre formatos diferentes, definição e execução de fluxos e processos, adaptadores para os sistemas mais conhecidos (como SAP) e várias formas de trocar dados com outros sistemas de informação.

Obviamente algumas destas características são mais importantes para certos níveis das arquitecturas de sistemas de informação que outras, como por exemplo:

- **Arquitectura Informacional** – define que tipos de entidades são trocados entre sistemas de informação. Embora actualmente o XML tenha grande protagonismo, na realidade a linguagem de formatação dos dados é completamente irrelevante neste tipo de arquitecturas.

Em vez disso estamos interessados em definir a estrutura de entidades tais como clientes, encomendas e produtos.

- **Arquitetura Aplicacional** – define que aplicações estão integradas, que tipo de integração utilizam e como é que estão integradas. Por exemplo, a integração pode ser síncrona ou assíncrona e manual ou automática.
- **Arquitetura Tecnológica** – define que tecnologias são usadas na integração, tais como HTTP como protocolo de comunicação, XML para formatar documentos, SOAP para encapsular esses documentos e certificados digitais X.509 para garantir a segurança. A maioria dos técnicos informáticos sente-se confortável neste nível, embora represente uma pequena parte da integração e sirva apenas para aqueles que vão implementar a solução.

Neste artigo focamos a nossa atenção nas arquiteturas aplicacionais e tecnológicas utilizando tanto conceitos já existentes como novos que serão propostos mais à frente neste artigo:

- Os conceitos existentes de *IS Block* e *IS Service* [Vasconcelos et. al.2003b] podem ser usados para representar os serviços fornecidos por outro sistema de informação. (Note-se que este “serviço” não tem de ser um serviço síncrono do tipo Web Service.)
- Os novos conceitos de *IT Integration Block* (proposto como especialização de IT Block) e *IT Integration Service* serão usados para representar que aplicações estão integradas e como estão integradas.

4. Modelação da integração em ASI

As secções anteriores sublinharam a inexistência de uma praxis, mecanismo ou linguagem que permitisse a modelação dos conceitos de integração numa ASI.

Esta secção propõe um conjunto de primitivas (incluindo a sua representação gráfica) que possibilita a manipulação semântica dos conceitos de integração numa ASI.

4.1. *Integração a nível aplicacional (IS level)*

Neste artigo, em alinhamento com a framework CEO introduzida na secção 2.5, propomos um conjunto de extensões à linguagem standard de modelação UML [UML 1997] de forma a acomodar os conceitos de integração que devem ser representados numa ASI. De facto, a framework CEO não define os conceitos (e correspondente estereótipos UML) para a modelação da integração numa ASI.

Propõe-se que o conceito de «*IS service*» seja usado enquanto um conceito base para modelação da integração a nível aplicacional. Assim, o *IS Service* descreve como as operações, pertencentes a blocos aplicacionais são agregadas e disponibilizadas a outros blocos aplicacionais. Neste caso não se introduz qualquer novo estereótipo para modelação da integração a nível aplicacional, recorrendo-se aos conceitos já definidos na framework CEO (podendo-se posteriormente especializar este conceito, se necessário).

4.2. *Integração a nível tecnológico (IT level)*

O processo de integração pode ser dividido em três partes (representado em baixo na Figura 3): a fonte (o sistema que chama um serviço ou envia uma mensagem), o alvo e o porto de integração representando a relação entre a fonte e o alvo. Assim, a nível tecnológico propõe-se a separação das características associadas exclusivamente à fonte ou ao alvo e à relação.

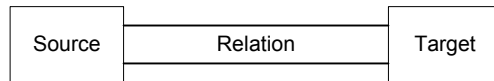


Figura 3. Processo de Integração

Considerando exclusivamente as características associadas ao sistema (fonte ou alvo), a integração pode ser descrita segundo duas dimensões:

- Nível de automação– representado se os serviços de integração são executados no sistema fonte ou alvo automaticamente (sem interferência humana) ou manualmente (implicando interação humana);
- Tipo de Papel – o sistema pode ser a fonte ou alvo dos dados. Por exemplo, num *web service* a fonte é o cliente, num produto de mensagens a fonte é o sistema a enviar a mensagem.

Tendo por base os conceitos tecnológicos apresentados na Figura 2, propomos que a integração seja um novo conceito que encapsule simultaneamente a plataforma (e.g., *J2EE*, *.Net*, *CORBA*, etc.) e/ou a aplicação do ponto de vista tecnológico. A Figura 4 apresenta a nossa proposta do «*IT Integration Block*» no âmbito da Figura 2.

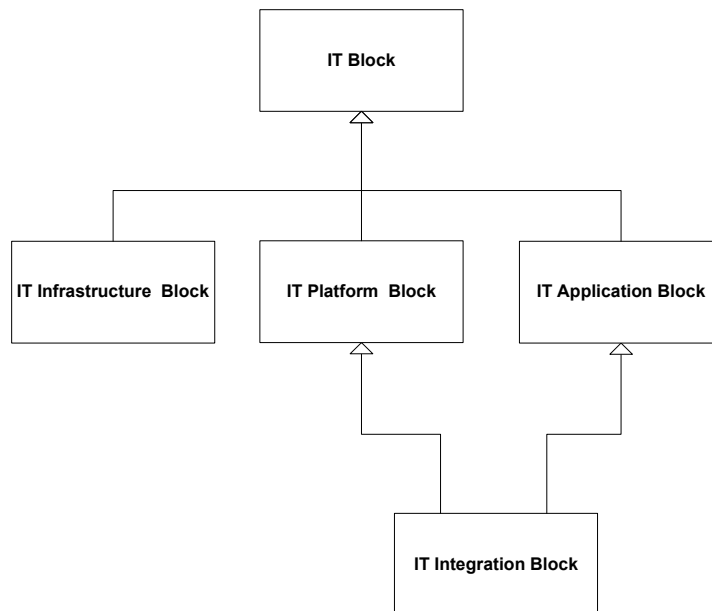


Figura 4. Extensões UML propostas para modelação da integração na ASI

A Figura 5 apresenta os atributos do estereótipo UML do «*IT Integration Block*» UML supra-proposto.

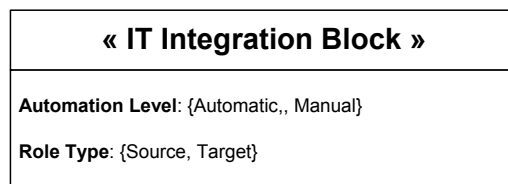


Figura 5. «*IT Integration Block*» em detalhe

Optou-se por não especializar mais o «*IT Integration Block*» de forma a acomodar a diversidade de conceitos e contínuos avanços nesta área. No entanto, dependendo dos objectivos e da audiência alvo, o «*IT Integration Block*» pode ser especializado para modelar conceitos

específicos de integração como *message broker*, *WebServices*, entre outros. O caso de estudo, apresentado na secção seguinte, exemplifica algumas destas possibilidades.

O «*IT Integration Service*» (proposto em [Vasconcelos et al. 2003b]) pode ser usado para modelar a relação entre sistemas, tal como descrito na Figura 3. Propõe-se que este componente seja caracterizado em termos de:

- Nível tecnológico – indicando se a integração ocorre dentro de um computador, numa *Intranet*, numa *Extranet* ou na Internet.
- Nível de Sincronismo – indicando se a integração entre «*IT Blocks*» é síncrona (tal como nos RPC, por exemplo), ou assíncrona (usualmente sem resposta, escalável e com elevada performance).
- Nível Organizacional – distingue a integração a nível de um departamento, numa empresa, numa *holding*, numa cadeia de valor, ou entre duas (ou mais) organizações sem relações.

A Figura 6 apresenta a extensão UML proposta em detalhe.

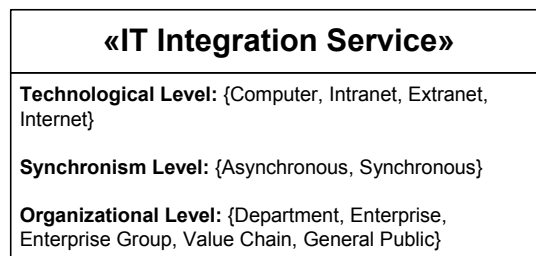


Figura 6. «*IT Integration Service*»

A próxima secção aplica estes conceitos a um caso real e concreto de forma a demonstrar e validar as propostas.

5. Caso de Estudo: SafeFood

O objectivo do projecto SafeFood é criar um Sistema de Informação que suporte o controle de qualidade efectuado pelas companhias de distribuição através da troca de informação relacionada com os produtos (quase) em tempo-real.

O projecto SafeFood não só envolve uma empresa de distribuição de produtos perecíveis, mas também muitas outras organizações externas, nomeadamente fornecedores. Todas estas entidades têm o seu sistema de informação que tem de ser integrado. Por exemplo, o departamento de Controle de Qualidade é responsável pela aprovação ou rejeição dos produtos. O armazenamento e a distribuição dos produtos para as lojas é efectuado pela Logística. As lojas são responsáveis pela venda dos produtos aos clientes e o Departamento Agrícola é responsável pela gestão de contractos com Organizações de Produtores (denominado de OP), estando estas comprometidas a entregar os produtos perecíveis nas datas negociadas.

Na Figura 7 a Arquitectura de Sistemas de Informação ao nível Aplicacional é apresentada. As dependências entre os «*IS Blocks*» são definidas usando os «*IS Service*». As setas indicam as dependências entre os «*IS Blocks*». Por exemplo, o «*IS Block* “SafeFood System”» depende do serviço fornecido pelo «*IS Service* “Control Quality API”».

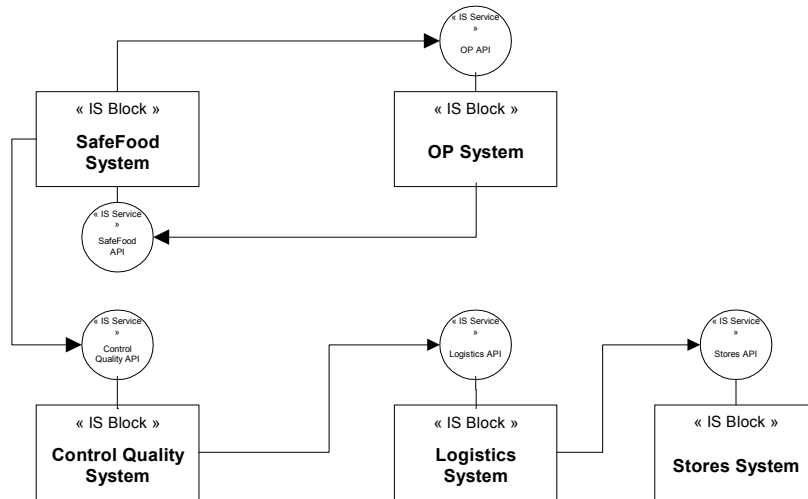


Figura 7. ASI ao nível Aplicacional

As dependências entre IS Blocks representam pontos de integração entre esses sistemas. Na Figura 8 a dependência entre dois IS Blocks reais é representada em detalhe.

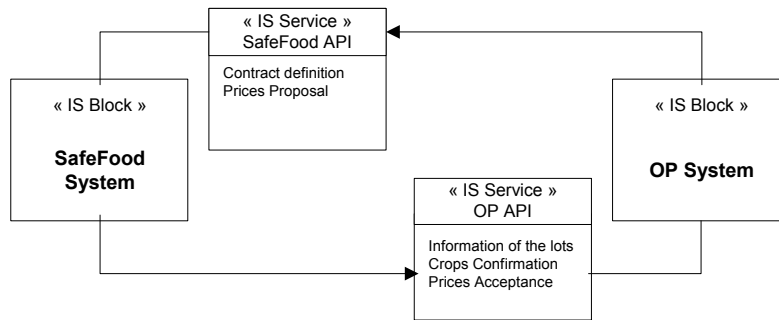


Figura 8. Dependências entre o Sistema SafeFood e o Sistema da OP

O IS Block “SafeFood System” poderia ser decomposta em três sistemas de informação (Sistema de Gestão Agrícola, Sistema Gestão Comercial, Sistema de Aproveitamento) como representado na Figura 9. Cada um destes IS Blocks é implementado por um IT Block.

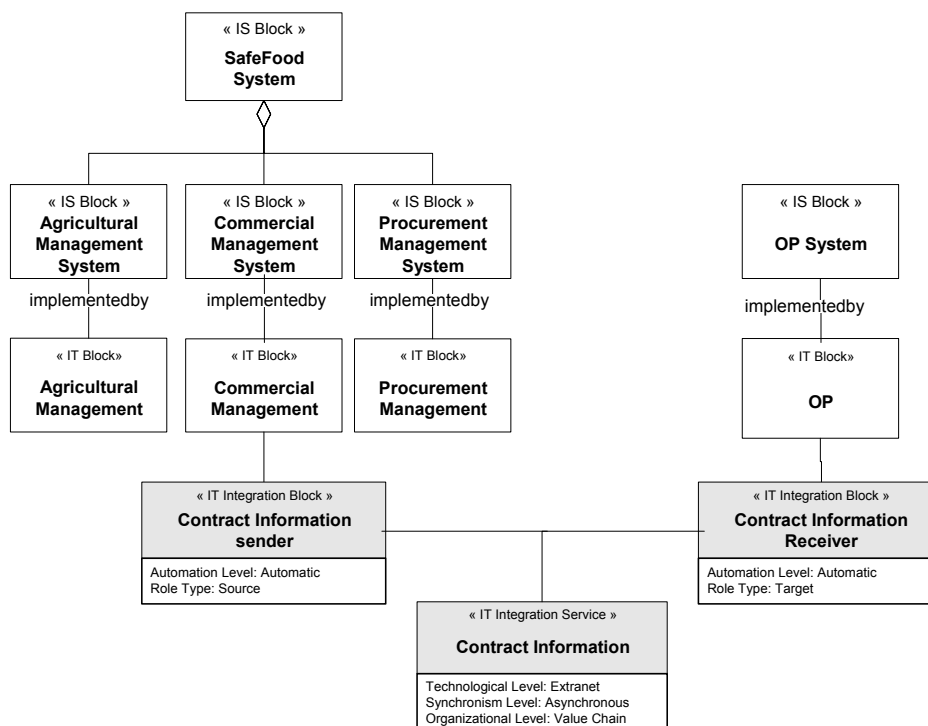


Figura 9. Detalhe da integração entre IT Blocks

A integração entre o IT Block “Commercial Management” e o IT Block “OP” é efectuada através de dois IT Integration Block e um IT Integration Service. Neste exemplo, o “Contract Information” é um conjunto de informação assíncrona trocada entre duas organizações pertencentes à mesma Cadeia de Valor e ocorre dentro de uma extranet (por exemplo, uma VPN na internet).

6. Conclusão

Neste artigo começa-se por apresentar uma revisão da área das arquitecturas dos sistemas de informação e da inexistência de formas de representação apropriadas para os conceitos de integração inerentes à ASI. Seguidamente procedemos a uma breve introdução à variedade de modelos de integração existentes, de forma a desmistificar a ideia que todos os problemas de integração podem, ou serão, resolvidos pelos “Web Services”.

A principal contribuição deste artigo é a extensão à nossa investigação prévia, no âmbito da representação de ASI, de forma a incluir um conjunto de primitivas específicas de integração ao nível aplicacional e tecnológico. Em particular, propõe-se que a representação de integração deve ser feita através de um conjunto de características (e.g., manual ou automática) e não limitada a serviços síncronos.

O artigo também apresenta um caso de estudo real (retirado de um projecto em que estamos envolvidos) de forma a ilustrar as propostas com problemas concretos de integração entre SI.

No futuro pretendemos detalhar a integração a nível tecnológico, em particular na forma de mapear os vários tipos de tecnologias de integração (dos últimos 40 anos) num conjunto limitado de conceitos. Estamos particularmente interessados nos conceitos de *Web Services*, *Message Brokers* e integração entre organizações em que a fiabilidade e segurança são preocupações extra.

7. Referências

- ASN.1 Information Site. <http://asn1.elibel.tm.fr/en/index.htm>, 2003.
- Boar, Bernard, *Constructing Blueprints for Enterprise IT Architecture*, John Wiley & Sons, 1999.
- Britton, C., *IT Architectures and Middleware*. Addison-Wesley, 2000.
- Chappell, D. et al., *Professional EbXML Foundations*. Wrox Press. 2001.
- DeBoever, L., *Enterprise Architecture Boot Camp & Best Practices: A Workshop*, Meta Group, 1997.
- Department of Defense Joint Technical Architecture, Julho 2002.
- Eriksson, Hans-Erik, and Magnus Penker, *Business Modeling with UML: Business Patterns at Work*, John Wiley & Sons, ISBN 0-471-29551-5, 2000.
- FEAPMO - The Business Reference Model - A Foundation for Government-wide Improvement, 2002.
- Federal Enterprise Architecture Framework, Version 1.1., Setembro 1999.
- Goodyear, Mark (Ed), *Enterprise system Architectures, Building Client/Server and Web-based Systems*, Accenture, Auerbach publications, 2000.
- Ferstl, Otto K., Elmar J. Sinz, *SOM*, *Wirtschaftsinformatik* 32 (6), pp.566-581, 1990.
- Garlan, D. et al., *Architectural Mismatch (Why It's Hard to Build Systems Out of Existing Parts)*, Proceedings 17th International Conference on Software Engineering, Seattle, WA, Abril 23-30 1995, pp.170-185.
- Hammer, M., Champy, J., *Reengineering the Corporation: A Manifesto for Business Revolution*, N. Brealey Publishing, London, 1993.
- IEEE Architecture Working Group, *Recommended Practice for Architecture Description – Draft IEEE standard P1471/D4.1*, IEEE, Dezembro 1998.
- Inmon, W. H., *Data Architecture – The Information Paradigm*, QED Technical Publishing Group, 1999.
- Kalakota, Ravi and Marcia Robinson, *E-Business 2.0*, Addison-Wesley Longman, Incorporated, 2000.
- Linthicum, D., *B2B Application Integration*, Addison-Wesley, 2001.
- Maes, Rik, Daan Rijsenbrij, Onno Truijens, and Hans Goedvolk, *Redefining Business – IT Alignment Through a Unified Framework*, White Paper, Maio 2000.
<http://www.cs.vu.nl/~daan/>
- T. W. Malone et al., *Tools for inventing organizations: Towards a handbook of organizational processes*, Management Science, Março 1999.
- M. Mira da Silva, *Information Systems Integration*, FCA, 2003.
- M. Mira da Silva. *Challenges for EDI Adoption by Small and Medium-size Enterprises (SME)*. Accepted to the IADIS International Conference e-Society, Lisboa, Portugal, 2003.
- Newcomer, Eric, *Understanding Web Services: XML, WSDL, SOAP, and UDDI*, Addison Wesley Professional, (ISBN: 0201750813), 2002.
- How do You Define Software Architecture?*, *Software Engineering Institute*, Carnegie Mellon University, Dezembro 2000.
<http://www.sei.cmu.edu/architecture/definitions.html>
- UML Proposal to the Object Management Group, 1997. <http://www.rational.com/uml>
- Sassoon, *Urbanisation des systèmes d'information*, 1998.
- SAP NetWeaver. <http://www.sap.com/solutions/netweaver/>, 2003
- Spewak, Steven, and Steven Hill, *Enterprise Architecture Planning: Developing a Blueprint for Data, Applications and Technology*, Wiley-QED, ISBN 0-471-599859, 1992.

- Tissot, Florence, and Wes Crump, *An Integrated Enterprise Modeling Environment*, P. Bernus, K. Mertins, G. Schmidt (Eds.), Handbook on Architectures of Information Systems, Springer, pp.59-79, ISBN 3-540-64453-9, 1998.
- Open Group, *The Open Group Architectural Framework (TOGAF) – Version 7*, Novembro 2001.
- Vasconcelos, A., A. Caetano, J. Neves, P. Sinogas, R. Mendes, and J. Tribolet, *A Framework for Modeling Strategy, Business Processes and Information Systems*, Proceedings of 5th International Enterprise Distributed Object Computing Conference EDOC, Seattle, USA, Setembro 2001.
- Vasconcelos, A., A. Caetano, P. Sinogas, R. Mendes, and J. Tribolet, *Arquitetura de Sistemas de Informação: A Ferramenta de Alinhamento Negócio / Sistemas de Informação?*, Proceedings da 3ª Conferência da Associação Portuguesa de Sistemas de Informação, Janeiro 2003a.
- Vasconcelos, A., P. Sousa, and J. Tribolet, *Information System Architectures: Representation, Planning and Evaluation*, Proceedings of International Conference on Computer, Communication and Control Technologies Orlando, U.S.A., Julho 2003b.
- Vernadat, François, *Enterprise Modeling and Integration*, London, Chapman & Hall, 1996.
- W3C, World Wide Web Consortium, *Web Services*, 2001.
<http://www.w3.org/2002/ws>.
- Zachman, John, *A Framework for Information System Architecture*, IBM System Journal Vol.26 Nº 3, 1987, p.276 – 292.
- Zachman, John, *Enterprise Architecture: The Issue of the Century, Database Programming and Design*, Março 1997.
- Zijden, Stefan, Hans Goedvolk, and Daan Rijsenbrij, *Architecture: Enabling Business and IT Alignment in Information System Development*, 2000.
<http://www.cs.vu.nl/~daan/>