

BIANCA HABERLY PEREIRA

**PROPOSTA DE UTILIZAÇÃO DE UMA ARQUITETURA ORIENTADA A
SERVIÇOS PARA INTEGRAÇÃO DE SISTEMAS**

Monografia apresentada à Escola
Politécnica da Universidade de
São Paulo para obtenção da
certificação no curso de MBA em
Tecnologia da Informação

Orientador:
Prof. Dr. Reginaldo Arakaki

São Paulo

2007

MBA /TI
2007
P414 p

DEDALUS - Acervo - EPEL



31500017587

M2007 X

FICHA CATALOGRÁFICA

Pereira, Bianca Haberly

Proposta de utilização de uma arquitetura orientada a serviços para integração de sistemas / B.H. Pereira. -- São Paulo, 2007.

45 p.

Monografia (MBA em Tecnologia da Informação) - Escola Politécnica da Universidade de São Paulo. Programa de Educação Continuada em Engenharia.

**1. Sistemas de informação 2. Tecnologia da informação
3. Sistemas distribuídos I. Universidade de São Paulo. Escola Politécnica. Programa de Educação Continuada em Engenharia II.t.**

Aos meus pais, pela orientação e ensinamentos.

AGRADECIMENTOS

Ao Prof. Dr. Reginaldo Arakaki, pelo apoio, incentivo e confiança depositada, com que pude contar em todos os momentos durante a realização deste trabalho.

À minha família por sempre me apoiar em qualquer situação.

Aos colegas e amigos da USP, pela solidariedade e, principalmente, pela paciência ao longo da elaboração deste trabalho.

Aos meus amigos, por sempre torcerem pelo meu sucesso e compreenderem as minhas ausências para dedicar-me aos estudos.

Ao Departamento de Informática da Universidade de São Paulo, pela grande colaboração e incentivo.

RESUMO

O presente trabalho propõe a utilização de uma Arquitetura Orientada a Serviços na integração de processos de negócio e de sistemas distribuídos desenvolvidos e/ou geridos pelo Departamento de Informática da Reitoria da Universidade de São Paulo. Será utilizada como apoio a norma ISO/IEC 9126 para eleger alguns atributos de qualidade de software, de maneira que possamos identificar alguns dos problemas enfrentados atualmente pelas equipes de desenvolvimento de sistemas e os possíveis benefícios alcançados através deste estudo.

Inicialmente apresentaremos alguns conceitos relevantes à compreensão deste trabalho. A seguir, será apresentado o cenário atual do ambiente de TI corporativo da USP e, como resultado, apresentaremos uma proposta de integração de sistemas na universidade, utilizando SOA para apontar possíveis melhorias, tanto na qualidade como na integração e administração do ambiente de TI.

Através da análise dos resultados obtidos com este estudo, conclui-se que muitos dos benefícios obtidos com a utilização de serviços, como o encapsulamento da complexidade e detalhes do negócio, reuso, desacoplamento, entre outros, auxiliam na terceirização de sistemas ou módulos de sistemas, facilitam o desenvolvimento de aplicações e a integração entre elas, melhoram o relacionamento com os clientes, possibilitam a melhoria de processos e geram novas oportunidades de negócio.

Porém, vale ressaltar que a SOA não é um remédio para todos os males. Nem todo tipo de aplicação é apropriado para esta arquitetura. Modismos, compreensão superficial da SOA e falta de gerenciamento podem levar ao uso incorreto desta arquitetura.

Palavras-chave: Arquitetura de Software; Integração de Sistemas; Orientação a Serviços.

ABSTRACT

This work proposes the utilization of a Service-Oriented Architecture to integrate business process and distributed systems developed and/or managed by the Deanery's Computing Department of the University of São Paulo. ISO/IEC 9126 standard will be utilized as support to elect some software quality attributes, in order to allow us to identify a couple of the problems faced currently by system development teams and the feasible benefits achieved by this study.

At first we will present some concepts relevant to the comprehension of this work. Next, the present scenario of the corporate IT environment of USP will be presented and, as a result, we will present a system integration proposition in the university, using SOA to point feasible improvements, both quality and IT environment integration and administration.

By the analysis of the results gained with this study, we conclude that most of the benefits achieved with service utilization, like the encapsulation of business complexity and details, reuse, uncoupling, among others, aid the system or system modules outsourcing, ease the application development and the integration among them, improve the customer relationship, allow the processes improvement and generate new business opportunities.

However, it is worth to stand out that SOA is not a cure-all. Not every kind of application is appropriate to this architecture. Fashions, SOA superficial comprehension and lack of management can lead to the incorrect use of this architecture.

Keywords: Software Architecture; System Integration; Service Orientation.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

1	INTRODUÇÃO.....	1
1.1	OBJETIVO.....	1
1.2	METODOLOGIA	1
1.3	MOTIVAÇÃO	2
2	FUNDAMENTO CONCEITUAL	4
2.1	INTEGRAÇÃO DE APLICAÇÕES CORPORATIVAS	4
2.2	BUSINESS-TO-BUSINESS	6
2.3	ARQUITETURA ORIENTADA A SERVIÇOS.....	9
2.3.1	<i>Serviços.....</i>	9
2.3.2	<i>Paradigma SOA.....</i>	10
2.3.3	<i>Orquestração</i>	11
2.3.4	<i>Barramento Corporativo de Serviços.....</i>	12
2.3.5	<i>Benefícios da implementação da SOA.....</i>	12
2.4	EXEMPLOS DE IMPLEMENTAÇÃO DA SOA.....	13
2.4.1	<i>Siemens</i>	13
2.4.2	<i>Toyota</i>	14
2.4.3	<i>ABN AMRO Brasil.....</i>	15
2.5	QUALIDADE DE SOFTWARE	17
2.5.1	<i>Funcionalidade</i>	18
2.5.2	<i>Confiabilidade</i>	18
2.5.3	<i>Usabilidade</i>	19
2.5.4	<i>Eficiência</i>	19
2.5.5	<i>Manutenibilidade</i>	20
2.5.6	<i>Portabilidade</i>	20

3	DOMÍNIO DO PROBLEMA (EXEMPLO PRÁTICO).....	21
3.1	CENÁRIO ATUAL	21
3.1.1	<i>Integração entre os Sistemas Corporativos.....</i>	23
3.2	PROBLEMAS DE INTEGRAÇÃO	24
3.3	DEFINIÇÃO DE ESCOPO DO EXEMPLO PRÁTICO.....	26
3.4	REQUISITOS NÃO FUNCIONAIS X QUALIDADE DE SOFTWARE	27
4	ESPECIFICAÇÃO DA ARQUITETURA.....	30
4.1	VISÃO EMPRESA	30
4.2	VISÃO INFORMAÇÃO E COMPUTAÇÃO.....	32
4.3	VISÃO ENGENHARIA	34
4.4	VISÃO TECNOLOGIA	35
4.4.1	<i>BEA WebLogic.....</i>	35
4.4.2	<i>Microsoft BizTalk.....</i>	36
4.4.3	<i>IBM Websphere.....</i>	36
4.4.4	<i>Sun Java.....</i>	37
4.4.5	<i>Oracle</i>	37
4.5	MELHORIAS OBTIDAS	ERRO! INDICADOR NÃO DEFINIDO.
5	CONCLUSÕES.....	41
5.1	TRABALHOS FUTUROS.....	42
6	REFERÊNCIAS.....	43

LISTA DE FIGURAS

Figura 1 – Integração de sistemas utilizando EAI. Fonte: [4]	6
Figura 2 – Collaborations in a service-oriented architecture. Fonte: [9]	10
Figura 3 – Cenário atual da USP.	22
Figura 4 – Integração entre os sistemas corporativos	24
Figura 5 – Integração com os sistemas terceirizados.....	25
Figura 6 – Exemplo de diagrama de classes	32
Figura 7 – Exemplo de serviços.....	33
Figura 8 – Proposta SOA	34

LISTA DE TABELAS

Tabela 1 – Problemas de integração	26
Tabela 2 – Lista de requisitos não funcionais.....	27
Tabela 3 – Mapeamento RNFs x Características da ISO 9126.....	28
Tabela 4 – Mapeamento de problemas, requisitos não-funcionais e respectivas melhorias.....	39
Tabela 5 – SOA <i>antipatterns</i>	41

LISTA DE ABREVIATURAS

BD	Banco de Dados
B2B	Business-to-Business
DI	Departamento de Informática da Reitoria da USP
EDI	Electronic Data Interchange
IEC	International Engineering Consortium
ISO	International Standards Organization
SGBD	Sistema Gerenciador de Banco de Dados
SOA	Service-Oriented Architecture
XML	EXtensible Markup Language
ERP	Enterprise Resource Planning
EAI	Enterprise Application Integration
USP	Universidade de São Paulo

1 Introdução

A proposta deste trabalho surgiu com a experiência do Departamento de Informática da Reitoria da USP (DI) no desenvolvimento de sistemas, face a alguns problemas de integração como: relações B2B (*business-to-business*), desenvolvimento de sistemas idênticos por centros de informática distintos, entre outros.

1.1 Objetivo

O presente trabalho propõe a utilização de uma Arquitetura Orientada a Serviços (*Service-Oriented Architecture* ou SOA) na integração de processos de negócio e de sistemas distribuídos desenvolvidos e/ou geridos pelo DI, que são tratados neste trabalho como sistemas corporativos. Será utilizada como apoio a norma ISO/IEC 9126 [1] [2] para eleger alguns atributos de qualidade de software, de maneira que possamos identificar alguns dos problemas enfrentados atualmente pelas equipes de desenvolvimento de sistemas e os possíveis benefícios alcançados através deste estudo.

1.2 Metodologia

Inicialmente serão abordados alguns conceitos sobre integração de aplicações, relações B2B, SOA e a norma ISO/IEC 9126-1. A seguir, será apresentado o cenário atual do ambiente de TI corporativo, levantando alguns dos problemas de integração enfrentados atualmente e a identificação de alguns requisitos não-funcionais dos sistemas.

Como resultado, apresentaremos uma proposta de integração de alguns sistemas corporativos da Universidade de São Paulo, utilizando SOA para apontar possíveis melhorias, tanto na qualidade como na integração e administração do

ambiente de TI. Para uma melhor compreensão desta proposta, foi adotado o conceito de visões do RM-ODP (*Reference Model for Open Distributed Processing*). Não é escopo deste trabalho descrever o modelo, porém maiores detalhes podem ser encontrados em [3].

Por fim, será exposta a conclusão do trabalho citando também alguns dos próximos passos a serem dados no projeto.

1.3 Motivação

O crescimento da organização, a necessidade de novos sistemas e a constante manutenção dos sistemas legados nos levam a analisar alguns pontos críticos para a integração, tais como:

1. Como terceirizar os sistemas ou módulos para atender a tempo as necessidades dos usuários, mantendo o padrão atual de desenvolvimento e garantindo a integridade do negócio?
2. Como disponibilizar acerto de dados para as unidades de negócio da organização?
3. Como integrar diferentes aplicações corporativas que suportam os processos de negócio?
4. É possível melhorar os serviços prestados a clientes, funcionários e parceiros de negócio?
5. A integração das informações pode ajudar a criar novos tipos de negócio e melhorar os processos já existentes?
6. Como facilitar fusões e aquisições?
7. O que muda na infra-estrutura de TI com a implantação de uma arquitetura orientada a serviços?

A procura por respostas a estas questões motivaram o desenvolvimento deste trabalho.

Esperamos que as cinco primeiras questões possam ser sanadas ou mitigadas através da implantação da SOA. A dúvida quanto às fusões e aquisições de organizações e seus respectivos sistemas de informação, deve ser melhor analisada, pois não é possível determinar, em um primeiro momento, se somente a SOA traria algum benefício a estes processos, visto que haveria dependência em relação aos sistemas adquiridos. Já a última questão trata de uma mudança de paradigmas no que diz respeito à cultura organizacional do desenvolvimento de sistemas na USP, além dos impactos tecnológicos.

2 Fundamentos

Neste capítulo serão abordados alguns conceitos básicos acerca de integração de aplicativos corporativos (EAI), *business-to-business* (B2B), arquitetura orientada a serviços (SOA) e a definição da norma ISO/IEC 9126 [1] que irão auxiliar na compreensão do desenvolvimento do trabalho.

2.1 Integração de aplicações corporativas

Com a evolução dos negócios, da globalização e da competitividade cada vez mais acirrada, as empresas passaram a sentir a necessidade de integrar todos os sistemas e buscar soluções baseadas na Internet. O ERP (*Enterprise Resource Planning*) é um componente importante para o comércio eletrônico (*e-business*), mas como integrá-los, se cada software ERP usa seu próprio padrão? É preciso comprar ou desenvolver vários componentes de produtos de diferentes fornecedores, juntá-los, e então implementar o comércio eletrônico. A solução para isso é a EAI (*Enterprise Application Integration*) - integração de aplicações corporativas [4].

Nesse cenário de EAI se posicionam tecnologias genéricas como *drivers* de integração, *middleware* síncronos e assíncronos, gerenciadores de transações, *brokers* orientados a objetos, camadas de servidores web, etc. E são vários os motivos que estão levando a comunidade de informática a se interessar pela tecnologia de EAI, Alguns exemplos são [4] :

- Necessidade de integração dos sistemas ERPs com o restante dos sistemas sobreviventes da fase empacotada, e até mesmo integração entre os diferentes módulos ERPs;
 - Necessidade de integração dos sistemas CRM (*Customer Relationship Management*) e sistemas PRM (*Partner Relationship Management*);
-

-
- Necessidade da definição de componentes integradores para ligar o mundo transacional dos pacotes e legados com o mundo informacional do *Data Warehouse* (DW) e dos *Data Marts* (DM), devido ao crescimento da área de BI (*Business Intelligence*);
 - Necessidade de cruzamento de fronteiras entre as aplicações de negócio na Internet, uma abordagem mais rápida e integrada entre as peças de sistemas já existentes e as novas que estão chegando. O SCM (*Supply Chain Management*) é um exemplo, pelo fato de representar os processos de produção e entrega do produto final, do fornecedor para o cliente;
 - O aporte de novas tecnologias indutoras de integração, como XML (*EXtensible Markup Language*) e LDAP (*Lightweight Directory Access Protocol*), pode servir como motivação para a melhoria do ambiente de negócios eletrônicos da empresa. XML é uma proposta para intercâmbio de documentos, baseado em uma definição de metadados (tags auto definidos), e já conta com a aposta dos grandes fabricantes de software do mercado, como Microsoft, Oracle e IBM. A outra abordagem, LDAP, é uma proposta para padronização de acessos aos diversos serviços de diretórios diferentes espalhados pelos ambientes de rede e de desenvolvimento.

Como mostra a Figura 1, o conceito de EAI pretende tornar possível a comunicação entre CRM, SCM, ERP, BI, DW, sistemas legados e outras propostas que aparecerão.

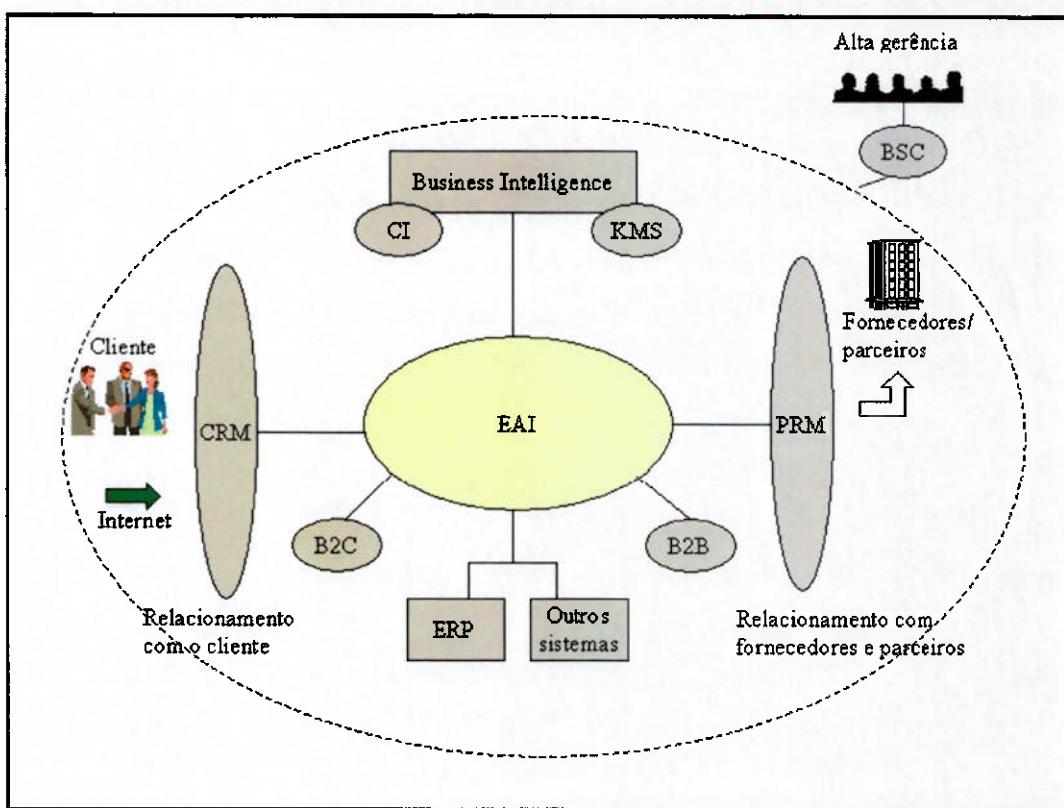


Figura 1 – Integração de sistemas utilizando EAI. Fonte: [4]

Processos de negócio podem ser melhorados usando a tecnologia EAI. EAI pode ser usada para ajudar a eliminar passos manuais no processo de negócio e evitar a entrada redundante de dados. As aplicações de EAI usam uma ferramenta de automação de fluxo de trabalho como ponte entre as aplicações que estão sendo integradas.

2.2 Business-to-Business

Organizações em todos os lugares estão implementando iniciativas B2B (*business-to-business*) buscando automatização e maior eficiência nas relações de negócio. Uma diversidade de iniciativas estão em vigor, abrangendo desde uma simples troca de informação transacional até sofisticadas redes de negócio [5].

Os benefícios de automatizar interações de B2B têm sido reconhecidos há muito tempo. Atualmente o padrão de comunicação B2B mais utilizado é o EDI (*electronic data interchange*). Organizações que tem implementado o EDI utilizam-

no para comunicar-se com menos de 20% de seus parceiros de negócio. Interações com os outros 80% ainda baseiam-se em métodos não automatizados como telefone ou fax. Além disso, não é raro duas organizações utilizarem o EDI para troca de ordens de compra e faturas, e outros métodos, como fax, para troca de documentos de remessa [6].

O campo do B2B é muito diversificado. Pode ser dividido nos seguintes segmentos: *marketplaces* do lado da venda, *marketplaces* do lado da compra, bolsas de comércio e comércio colaborativo[6] .

Sistemas tradicionais de EDI eram implementados em VANs (*Value-added networks*), o que impedia sua acessibilidade a pequenas empresas. Além de poder substituir a EDI tradicional ou suplementá-la, a EDI baseada na internet oferece diversas facilidades indisponíveis naquela. [6] .

A despeito de sua extensa adoção, o EDI tradicional é complicado e custoso. Seu custo e inflexibilidade têm limitado seu crescimento dentro da empresa e têm atuado como uma barreira para implementação por pequenas companhias. A falta de suporte à comunicação em tempo-real também é uma desvantagem adicional, levando à troca de documentos utilizando processos em lote (batch) [6].

Segundo Jones [5] , a Internet oferece um método muito mais barato e flexível para interações B2B. Todos reconhecem que para obter sucesso no futuro, produtores terão que abraçar a oportunidade oferecida pela internet de definir novos mercados, criar novas fontes de receita e atingir novos níveis de eficiência, fidelidade e satisfação do cliente. Além disso, os potenciais benefícios de compartilhar informações no meio empresarial são muito claros para empresas que atualmente estão colhendo os frutos de seu investimento em compartilhar informações entre seus sistemas de negócio internos.

Os maiores benefícios incluem a habilidade para responder mais rapidamente a variações em demanda para seus produtos, ao mesmo tempo em que minimizam o tamanho de seus estoques. E também a troca eletrônica de informações entre parceiros de negócio leva a transações e processos de negócio muito mais rápidos. Uma grande economia também pode ser conseguida, pois uma transação manual custa de seis a sete vezes mais do que a eletrônica.

Ainda segundo Jones, em teoria, a internet torna muito fácil para as companhias começarem a se comunicar eletronicamente com seus parceiros de negócio. Na prática, porém, pode estar longe de implementar diretamente até mesmo relações simples de B2B com um número pequeno de parceiros, isto porque quaisquer duas companhias diferem em praticamente tudo: na sua infra-estrutura de comunicação, na forma como elas descrevem seus bens e serviços, na seqüência de passos que compõem uma típica transação, no nível de segurança com que elas se sentem confortáveis, etc. Além disso, para suportar efetivamente interações com parceiros de negócios externos, os sistemas de negócios internos precisam ser intimamente integrados com o mesmo processo.

Na maioria das instâncias, o foco imediato de implementar uma infra-estrutura B2B é atingir a automação dos processos de negócio. Interações com outra companhia são conseguidas eletronicamente de acordo com processos pré-definidos que asseguram que a transação seja significativa e confiável.

Esses processos públicos podem ser desenvolvidos como resultado de um acordo detalhado entre os parceiros de negócio ou podem ser codificados por um padrão ou consórcio. Num mundo onde diferentes companhias tendem a desenvolver suas próprias maneiras de descrever produtos, alocação de número de peças, formatação de faturas e numeração de ordem de compra, a padronização é essencial.

O problema com a internet é que não existiam formatos padrão até recentemente. Como padrão para comunicação B2B, o XML possui muitas vantagens significativas. Ele é altamente flexível, possui formato de dados genéricos, foi desenhado para múltiplas representações e idealmente estruturado para análise adicional de dados. Além disso, está preparado para interoperabilidade entre uma gama de dispositivos e suporta troca de informações em tempo real [5] [6] .

Enquanto o XML parece ter um grande potencial como padrão universal para a troca de documentos relativos a transação através da internet, existem também alguns obstáculos a serem superados, em sua maioria relacionados à falta de padronização (definição global). Embora muitas iniciativas de padronização tenham sido lançadas, elas ainda são muito independentes umas das outras, com diferentes pontos de partida, abordagens e vocabulários. Muitos padrões diferentes derivados do XML, como cXML (*Commerce XML*), xCBL (*XML Common Business Library*) e

BizTalk (arcabouço de integração de aplicações e comércio eletrônico), têm sido desenvolvidos. Contudo, em sua maioria, tem o escopo limitado, relacionado a interações, relações de negócio ou setores da indústria específicos.

Devido à falta de padronização é importante não subestimar o tamanho do investimento necessário em habilidades, organização e tecnologia, se a companhia pretende utilizar a internet para atingir a integração com seus parceiros de negócio. Claramente, conforme o número de parceiros aumenta, a necessidade de gerenciar a complexidade de múltiplas interações individuais irá aumentar também gradualmente [5] .

2.3 Arquitetura Orientada a Serviços

A Arquitetura Orientada a Serviços (*Service-Oriented Architecture* ou SOA) é um novo paradigma de desenvolvimento de aplicações, cujo objetivo é criar serviços com baixo acoplamento e permitindo a reutilização de código [7] .

A interoperabilidade proposta pela SOA, isto é, a possibilidade de *web services* distribuídos rodarem nas mais variadas plataformas de hardware e software, tem sido seu chamariz [8]

A SOA é um arcabouço (*framework*) de integração de ambientes heterogêneos de computação, onde as aplicações interagem através de serviços disponíveis numa rede, normalmente a internet.

2.3.1 Serviços

Um serviço é um componente que atende a uma função de negócio específica para seus clientes. Ele recebe requisições e as responde ocultando todo o detalhamento do seu processamento [7] [9] .

Disto se deduz que um serviço é estanque em relação a seu cliente, ou seja, quaisquer outros componentes acessórios devem ser utilizados somente dentro do código privado do serviço, não sendo expostos para os clientes.

Um serviço deve executar unidades completas de trabalho, não dependendo do estado de outros componentes externos. Assim, outra conclusão importante é que eles devem ser “*stateless*”, ou seja, sem armazenamento de estado de conversação. Isto aumenta sua reutilização [7].

Um serviço executa uma função atômica (ou transação). Todas as etapas intermediárias devem ser gerenciadas apenas pelo serviço e não pelo cliente.

2.3.2 Paradigma SOA

A Figura 2 ilustra o paradigma “encontrar-associar-executar” da SOA, onde um provedor publica (ou registra) seu serviço num repositório e este registro é utilizado por consumidores para encontrar serviços que satisfaçam determinada condição. Se o registro tiver o serviço requisitado, ele fornece ao consumidor seu contrato de interface e o endereço de destino [8].

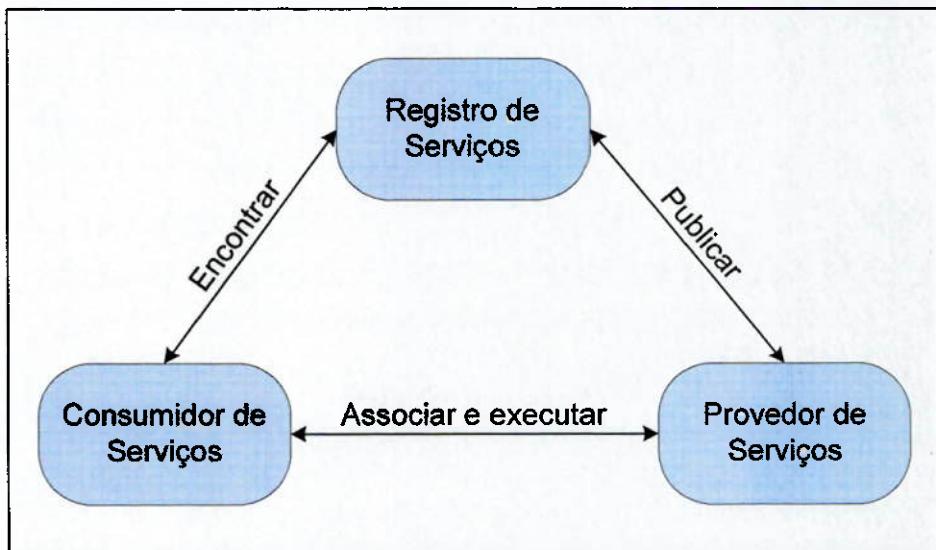


Figura 2 – Colaborações em uma arquitetura orientada a serviços. Fonte adaptada de: [9]

Os papéis definidos nesse paradigma são descritos a seguir:

-
- **Provedor de serviços** – é uma entidade endereçável na rede que aceita e executa requisições dos consumidores. Ela publica seus serviços e respectivos contratos de interface no registro de serviços para que o consumidor possa descobri-lo e acessá-lo.
 - **Registro de serviços** – é a entidade que possibilita a descoberta do serviço. Ela contém um repositório de serviços disponíveis e permite a busca das interfaces dos provedores para os consumidores interessados nos mesmos.
 - **Consumidor de serviços** – é uma aplicação, um módulo de software ou outro serviço que necessite de um serviço. Ele inicia a busca do serviço no registro, associa-se a ele e o executa, de acordo com o contrato da interface do serviço.

As operações realizadas por cada um desses papéis são:

- **Publicar** – para ser acessível, uma descrição do serviço deve ser publicada, de maneira que ele possa ser descoberto e executado por um consumidor.
- **Encontrar** – um consumidor localiza um serviço pesquisando no registro por um serviço que satisfaça seu critério.
- **Associar e executar** – após encontrar a descrição do serviço, o consumidor o executa de acordo com a informação na descrição.

2.3.3 Orquestração

A orquestração é composta por um fluxo de etapas, com verificações de pré e pós condições, e um coordenador, responsável por dar andamento ao fluxo, invocando e verificando todas as etapas necessárias. Cada etapa invoca um serviço que é oferecido por um provedor [7]

Desta maneira, podemos mudar a ordem das etapas, acrescentar outras, mudar critérios de verificação ou mesmo criar outros fluxos sem alterar o código dos serviços.

2.3.4 Barramento Corporativo de Serviços

O barramento corporativo representa uma camada de software ou abstração para arquitetura de integração do conjunto de esforços de integração de sistemas, sem estar necessariamente no escopo de construção de cada sistema. Isto é, o barramento corporativo é uma camada que admite regras especiais, lógica específica e uma grande quantidade de softwares e padrões pertinentes [10].

O ESB (*enterprise service bus*) ou Barramento de Serviços Corporativos é o barramento que incorpora a tecnologia SOA na forma de conexão e integração. Sua base tecnológica permite que a empresa conduza, na arquitetura de processos, a orquestração dos processos corporativos e seus fluxos de informação, e assim responda mais rápido para regras de negócio, conforme as prerrogativas das tecnologias que devem acompanhar seu uso: SOA e BPM.

No mundo da integração B2B e plataformas *e-business*, o ESB será um apoio mais consistente que as alternativas atuais pelo seu caráter aberto, padronizado e com alto poder de difusão entre empresas parceiras e fornecedoras.

2.3.5 Benefícios da implementação da SOA

Ela tem permitido que empresas realizem negócios de maneira mais eficiente, adaptando-as conforme as necessidades de mudanças surgem. A implementação de SOA através de *web services* traz alguns benefícios como [11] :

- **Reuso e baixo custo** – Códigos desenvolvidos para aplicações já existentes podem ser reutilizados para se adequar a novos requisitos de negócio, ou seja, SOA permite o reuso de recursos, onde um novo serviço pode ser criado a partir de uma infra-estrutura de TI ou sistema já existente. O reuso reduz o

tempo e custo de desenvolvimento das aplicações e, com a SOA, uma aplicação precisa conhecer apenas a interface de um serviço para utilizá-lo. Dessa maneira, as funções de uma aplicação ou sistema podem ser mais facilmente acessadas e a integração de sistemas pode se tornar mais simples;

- **Interoperabilidade** – Através dos padrões definidos para os web services, aplicações cliente e fornecedores de serviços podem se comunicar e trocar informações, mesmo em ambientes de hardware e software heterogêneos;
- **Escalabilidade e Flexibilidade** – Como os serviços numa SOA tem baixo acoplamento, eles são mais flexíveis e facilmente escaláveis, pois há menor dependência entre a aplicação cliente e o serviço utilizado. Isso permite compor mais rapidamente novos serviços para responder a novas necessidades de negócios, conforme elas surgem.

2.4 Exemplos de implementação da SOA

Este capítulo apresentará os estudos de caso de algumas companhias que, com a implantação da SOA e a disponibilização de serviços, conseguiram integrar seus sistemas internamente e/ou externamente, trazendo ganhos de produtividade, entre outras vantagens.

Serão apresentados os cenários antes e depois da SOA e os benefícios alcançados com essa iniciativa.

2.4.1 Siemens

A Siemens é uma companhia multinacional de produtos eletroeletrônicos. Além das filiais, também faz parte dela um leque de empresas espalhadas ao redor do mundo [12].

O departamento de operações de TI da Siemens (*Siemens IT Operations*), localizado na Alemanha, é responsável por todas as operações de TI da companhia.

Um de seus maiores desafios é integrar uma variedade de plataformas e sistemas heterogêneos utilizados pelas filiais e demais empresas do grupo.

Eles adotaram a utilização do BizTalk Server da Microsoft, que utiliza *web services* baseados em XML para coordenar todas as interações entre os diversos sistemas que a Siemens utiliza. O BizTalk possibilitou a automatização de vários processos de negócio, como a contratação de novos funcionários, configuração de acesso a LAN remota, recuperação de dados perdidos e gerenciamento de e-mail. Clientes do Siemens IT Operations usam um portal web focado ao cliente para ver e solicitar serviços de TI.

O grupo esperava um crescimento exponencial no número de serviços de TI que eles poderiam prover em um prazo de 2 a 3 anos. Eles esperavam uma solução que suportasse de 3 a 4 vezes o número de aplicações e serviços oferecidos até então, sem degradar a performance do sistema. O departamento também queria assegurar que a sobrecarga no gerenciamento não crescesse exponencialmente junto com o crescimento esperado nas novas aplicações.

A ferramenta da Microsoft possibilitou o monitoramento de sistemas remotos, além daqueles baseados no BizTalk. Também foi possível integrar as informações do monitoramento diretamente dentro dos processos do BizTalk. Além disso, eles podiam monitorar os web services diretamente: se um serviço fica indisponível ou quando um serviço volta à ativa, os administradores são notificados automática e imediatamente.

Segundo o gerente de projeto da Siemens IT Operations, Tom Buse, do ponto de vista gerencial e técnico, é possível trabalhar muito mais rápido agora. Se o grupo pretende lançar um serviço novo, é definido o tipo de interface de comunicação necessário, pergunta-se qual o tipo de informação necessário para o consumidor final e então o serviço é integrado a todo o sistema. O que antes levava meses para ser disponibilizado, hoje leva apenas algumas semanas.

2.4.2 Toyota

A Toyota Austrália é uma subsidiária da Toyota, uma das maiores fabricantes de automóveis do mundo [13].

Recentemente a companhia adotou um programa de renovação de suas operações, de maneira a integrar os sistemas e processos que ligam a Toyota e seus parceiros de negócio, com o propósito de acessar informações em tempo-real, dar maior transparência às suas operações e aumentar a performance do negócio.

A empresa possui um ERP que conecta diversos parceiros na cadeia de valor através dos mais variados sistemas, alguns deles de tecnologia proprietária.

Um dos canais que precisam ser integrados é entre a companhia e suas diversas concessionárias. No passado, sistemas terceirizados que executavam as operações de uma concessionária, não eram integrados com o ERP da empresa. Informações vitais para os processos de pedido, produção e entregas, freqüentemente precisavam ser refeitos manualmente, o que consumia tempo e mão-de-obra, além de serem passíveis de erros.

Para baixar os custos e possibilitar uma transferência de dados estável por toda a cadeia de valor, a Toyota optou por uma abordagem baseada em padrões. De acordo com a companhia, desde 2003 foram finalizados com sucesso 6 projetos utilizando o software da BEA:

- Comprovante de entrega eletrônico;
- Status de pagamento de contas on-line para fornecedores;
- Integração de gerenciamento de frota;
- Sistema de processamento de pedidos de veículos Toyota;
- Aviso de remessa avançado para entrega de veículos;
- Sincronização de inventário de veículos.

Com o uso de padrões abertos, o valor do dado coletado aumentou, pois ele pode ser compartilhado entre os projetos.

2.4.3 ABN AMRO Brasil

O ABN AMRO Brasil é uma das maiores instituições financeiras do país. Para atender as preferências e necessidades do seu grande número de clientes, o

banco foi disponibilizando, ao longo do tempo, serviços financeiros através dos mais diversos canais de comunicação: pontos de auto-atendimento, *Internet banking*, correio eletrônico, telefone (*call center*), mensagens de texto por celular (*WAP*), etc. Na medida em que cada novo canal de comunicação era disponibilizado, repetiam-se os esforços de integração dos diversos sistemas de informação, que implementavam cada um dos principais serviços financeiros, com o software de interatividade específico do novo canal que seria disponibilizado [14].

Esta repetição de esforços para integrar múltiplos sistemas de informação a múltiplos canais, foi um dos motivadores para implementação da SOA no ABN AMRO Brasil. Em um primeiro momento, prevaleceu a necessidade de compartilhar recursos (no caso, sistemas de informação) com baixo acoplamento e de fácil reutilização entre os diversos canais. Assim, foi estabelecida uma SOA inicial onde os sistemas de informação foram compreendidos como serviços básicos e os softwares desenvolvidos para cada canal como sendo os aplicativos para o usuário final.

Apesar de haver a opção da compra de um barramento de serviços “comercial”, ou seja, na forma de pacotes de software chamados de *middleware* ou EAI, o ABN AMRO Brasil optou por construir seu próprio barramento. As soluções comerciais analisadas foram consideradas muito orientadas a plataforma baixa, especialmente para sistemas operacionais Windows e Unix, e muito focadas em aplicações no ambiente internet, o que não era o foco da empresa, cujo desafio era integrar um conjunto de sistemas de informação totalmente residente em mainframe.

A introdução de um novo canal de atendimento, chamado de *internet banking*, foi o fato que definiu a necessidade de construção do barramento de serviços que interligasse de forma eficaz todos os serviços bancários a todos os novos e futuros canais disponíveis aos clientes. O *internet banking* exigia comunicação bidirecional entre a plataforma baixa, onde os clientes acessavam o serviço através de seus computadores pessoais conectados à internet, e o ambiente mainframe, onde residiam os sistemas de informação, que implementavam os serviços bancários.

Após a consolidação da SOA básica, o ABN AMRO Brasil seguiu para o segundo estágio de maturidade da SOA, agregando serviços básicos para compor

serviços mais sofisticados. Por exemplo: o serviço básico “verificador de saldo de conta corrente” é utilizado por todos os serviços que gerem débito em conta-corrente, como: transferências, aplicações, pagamentos, etc. O serviço intermediário “autorizador de transação”, por sua vez, é o mesmo utilizado para transações de: transferência de valor entre contas, pagamentos de contas, aplicações, resgates, etc. E o “autorizador de transação” utiliza o “verificador de saldo de conta corrente” para efetuar as movimentações financeiras nas contas dos clientes.

2.5 Qualidade de software

O Padrão 9126 (*Software engineering – Product Quality*), definido pela ISO (*International Organization for Standardization*) e pela IEC (*International Electrotechnical Commission*) subdivide-se em quatro partes [1] :

- Parte 1: Modelo de Qualidade;
- Parte 2: Métricas Externas;
- Parte 3: Métricas Internas;
- Parte 4: Métricas de Qualidade em uso.

Este trabalho utilizará somente a parte 1, explicada neste capítulo, e que será utilizada no nosso estudo para definir atributos de qualidade software.

O modelo de qualidade nesta parte da série ISO/IEC 9126 categoriza os atributos de qualidade de software em seis características (funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade) as quais são, por sua vez, subdivididas em subcaracterísticas. Tais características são aplicáveis a qualquer tipo de software e não permitem sobreposição.

Essa norma é dirigida àqueles envolvidos com aquisição, desenvolvimento, uso, suporte, manutenção ou auditoria de software [1] [2] .

A subcaracterística “conformidade” está presente em todas as características, tendo em vista que o produto pode ser aderente a legislação, padrões internos e/ou normas diversas associadas a cada uma das características.

2.5.1 Funcionalidade

Um conjunto de atributos relativos a um conjunto de funções e a especificação de suas propriedades. Permitem medir a capacidade do software, isto é, o conjunto de funções oferecidas ao usuário, que satisfaça suas necessidades [1] [2] :

- Adequação – Atributos de software relativos à presença ou conveniência de um conjunto de funções para tarefas específicas;
- Acurácia – Atributos de software relativos à provisão de resultados ou efeitos corretos ou esperados;
- Interoperabilidade – Atributos de software relativos à sua habilidade de interagir com sistemas específicos;
- Conformidade – Atributos que fazem o software aderir a padrões ou convenções ou regulamentos legais relativos à aplicação ou prescrições similares;
- Segurança de Acesso – Atributos de software relativos à sua habilidade de prevenir acesso não autorizado, acidental ou deliberadamente a programas e dados.

2.5.2 Confiabilidade.

Um conjunto de atributos relativos à capacidade que o software tem em manter seu nível de desempenho sob condições estabelecidas por um período de tempo estabelecido [1] [2] :

- Maturidade – Atributos de software relativos à freqüência de fracassos por falhas do sistema;
 - Tolerância a Falhas – Atributos de software relativos à sua habilidade de manter um nível específico de desempenho em casos de falhas no software ou de violação nas interfaces especificadas;
-

-
- Recuperabilidade – Atributos de software relativos à sua capacidade de restabelecer seu nível de desempenho e recuperar os dados diretamente afetados, em caso de falha, e no tempo e esforço necessário para tal.

2.5.3 Usabilidade

Um conjunto de atributos relativos à facilidade de uso, e à avaliação individual deste uso, por um conjunto estabelecido ou implicado de usuários [1] [2] :

- Inteligibilidade – Atributos de software relativos ao esforço dos usuários para reconhecer os conceitos lógicos e sua capacidade de aplicação;
- Apreensibilidade – Atributos de software relativos ao esforço dos usuários para apreender a aplicação;
- Operacionalidade – Atributos de software relativos ao esforço dos usuários para operar e controlar a operação.

2.5.4 Eficiência

Um conjunto de atributos sobre o relacionamento entre o nível de desempenho do software e a quantidade de recursos utilizada, sob condições estabelecidas [1] [2] :

- Comportamento em relação ao tempo – Atributos de software relativos ao tempo de resposta de processamento e às taxas de rendimento quanto ao desempenho de suas funções;
- Comportamento em relação a recursos – Atributos de software relativos à quantidade de recursos utilizados e à duração deste uso quanto ao desempenho de suas funções.

2.5.5 Manutenibilidade

É uma característica através da qual se mede o esforço necessário para fazer modificações específicas ou correções no software [1] [2] :

- Analisabilidade – Atributos de software relativos ao esforço necessário para o diagnóstico de deficiências ou causas de falha, ou para a identificação de partes a serem modificadas;
- Modificabilidade – Atributos de software relativos ao esforço necessário para modificação, remoção de falhas ou alteração de ambiente;
- Estabilidade – Atributos de software relativos ao risco de efeitos inesperados de modificações;
- Testabilidade – Atributos de software relativos ao esforço necessário para validação de modificações.

2.5.6 Portabilidade

Um conjunto de atributos relativos à habilidade do software de ser portado de um ambiente para outro [1] [2] :

- Adaptabilidade – Atributos de software relativos à oportunidade de sua adaptação a diferentes ambientes especificados, sem se aplicarem outras ações ou meios, que não aqueles providos para este propósito no software considerado;
- Capacidade para ser instalado – Atributos de software relativos ao esforço necessário para a instalação do software em um ambiente especificado;
- Conformidade – Atributos de software que o fazem aderir a padrões e convenções relativos à portabilidade;
- Capacidade para substituir – Atributos de software relativos à capacidade e esforço necessário para substituir outro software, no ambiente estabelecido para este outro software.

3 Domínio do Problema (Exemplo Prático)

Este capítulo aborda o cenário atual do ambiente corporativo de TI da USP, como pode ser visto na Figura 3, para exemplificar a utilização de SOA, suas vantagens e mudanças de paradigmas.

Em decorrência de alguns problemas enfrentados atualmente descritos na seção 3.2, será avaliada a implantação de uma nova arquitetura para que os sistemas corporativos e aplicações locais das diversas unidades de negócio (unidades administrativas ou de ensino) possam acessar os bancos de dados centrais, encapsulando a complexidade dos detalhes de acesso e regras de negócio.

3.1 Cenário atual

Todos os sistemas corporativos (client/server e web) acessam diretamente os bancos de dados centrais, que são vários bancos interconectados através de replicação on-line, e divididos por área de negócio.

Todo cadastramento é realizado na base central através desses sistemas, porém existem sistemas locais desenvolvidos pelas áreas de Tecnologia da Informação de cada unidade de negócio.

As unidades de negócio podem possuir bases locais, com dados replicados, para dar subsídio aos sistemas locais (e específicos).

As informações enviadas para esses sites são contextualizadas por *triggers* dos bancos de dados centrais e atualizadas em tempo real através de replicação para as bases locais, a partir de um banco consolidado. Ou seja, existem *triggers* especializadas em separar as informações por unidade de negócio.

Essas aplicações acessam somente os bancos locais para consultas. No caso de inserções ou atualizações de informações, estas devem ser feitas nas aplicações corporativas correspondentes a cada informação (RH, Graduação, Pós-Graduação etc.).

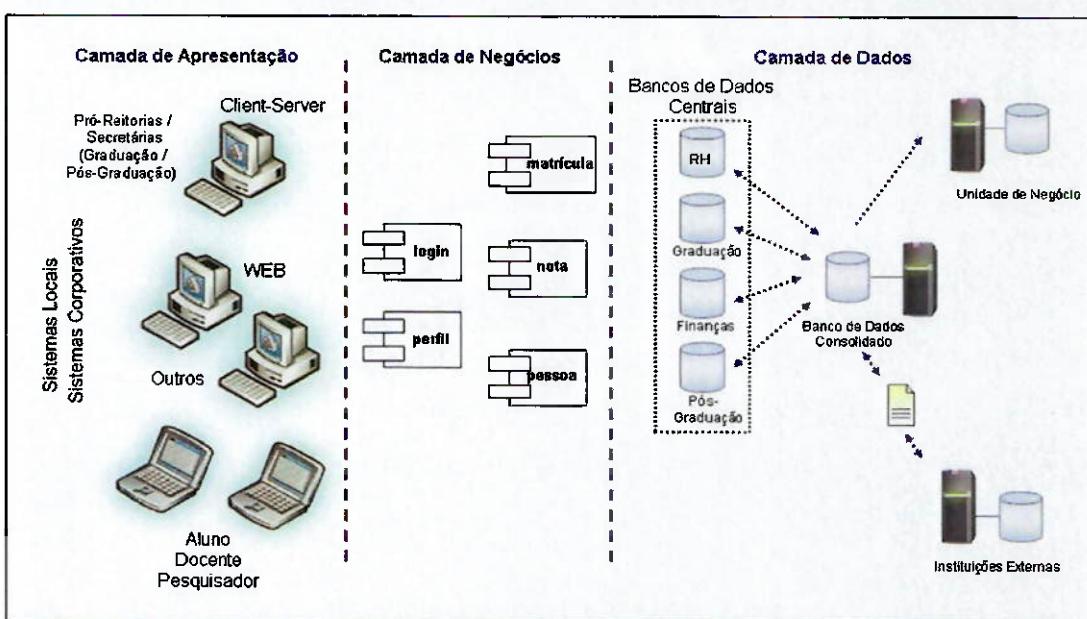


Figura 3 – Cenário atual da USP.

O cenário atual da USP apresenta o seguinte panorama:

Segurança:

Controle de login – O sistema possui um mecanismo de controle de segurança para evitar que pessoas não autorizadas tenham acesso ao sistema.

Identificação de perfil do usuário para acesso a determinadas funcionalidades dos sistemas – Atualmente os sistemas não possuem o mecanismo de identificação de perfil, apenas trabalha por níveis de acesso.

Escalabilidade:

Número de usuários / clientes – Atualmente existe uma estabilidade em número de usuários, porém o número de clientes tende a aumentar.

Quantidade de acessos ao banco de dados – Os acessos são sazonais. Existem picos de acesso à base central em época de matrículas.

Componentização:

Reuso de código – Existe uma iniciativa de reaproveitamento de código que se utiliza de algumas *stored procedures* e modelo de classe.

Manutenção da Informação:

Centralizada – Todas as transações são realizadas na base de dados central.

Integração entre os sistemas:

Os sistemas corporativos são integrados através de um modelo de dados único compartilhado entre os sistemas e suportados por um mecanismo de replicação de dados.

Eventualmente alguns cadastros locais, oriundo dos sistemas locais e de apoio (como matrícula) podem ser carregados via batch (processo de carga offline) na base central.

3.1.1 Integração entre os Sistemas Corporativos

O modelo de dados foi projetado para atender às diversas áreas de negócio da universidade e, ao mesmo tempo, integrá-las. As informações são manipuladas de forma isolada, ou seja, cada área de negócio é responsável apenas pela sua “visão” da informação. Porém, os dados comuns são replicados on-line para as outras bases de dados, garantindo a integração das informações.

Existe também uma classificação da informação: algumas áreas de negócio podem visualizar certas informações, mas não podem alterá-las.

Um bom exemplo dessa contextualização é a entidade “PESSOA” que, no modelo de negócio, possui vários papéis, muitos deles simultâneos. Esta entidade pode assumir o contexto de Aluno de Graduação, Aluno de Pós-Graduação, Docente, Funcionário, entre outros.

A Figura 4 ilustra integração entre os sistemas corporativos:

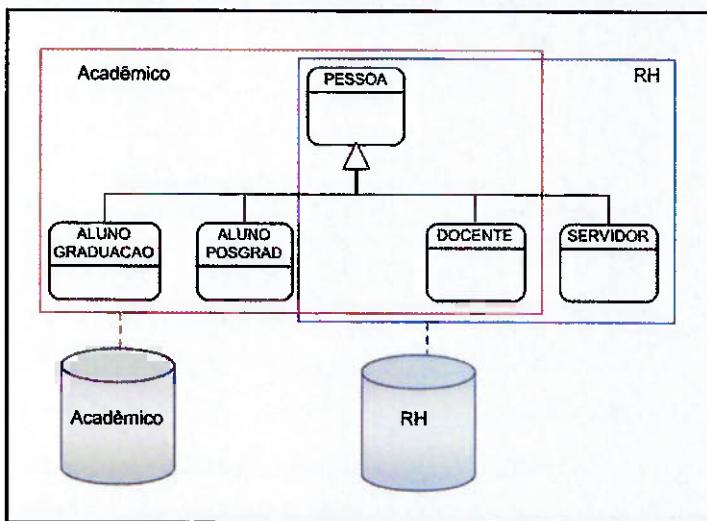


Figura 4 – Integração entre os sistemas corporativos

3.2 Problemas de Integração

Até meados de 2005, o DI era responsável pelo desenvolvimento e manutenção de todos os sistemas corporativos da universidade. A partir de então, a demanda de novos serviços exigiu a terceirização de determinadas atividades.

O desenvolvimento de sistemas por parte de empresas terceiras fez aflorar um problema: o modelo de dados da USP possui algumas particularidades de integração de sistemas, como explicado anteriormente, e isso faz com que seja necessário repassar parte da estrutura de modelagem e detalhes de negócio, conforme podemos ver na Figura 5, de forma que as empresas pudessem desenvolver seus aplicativos e tratar corretamente os dados. Isso pode causar problemas, por exemplo, quando são necessárias mudanças no modelo, ou até mesmo no caso de mudanças nos servidores de bancos de dados (troca de hardware e indisponibilidade de um servidor, por exemplo).

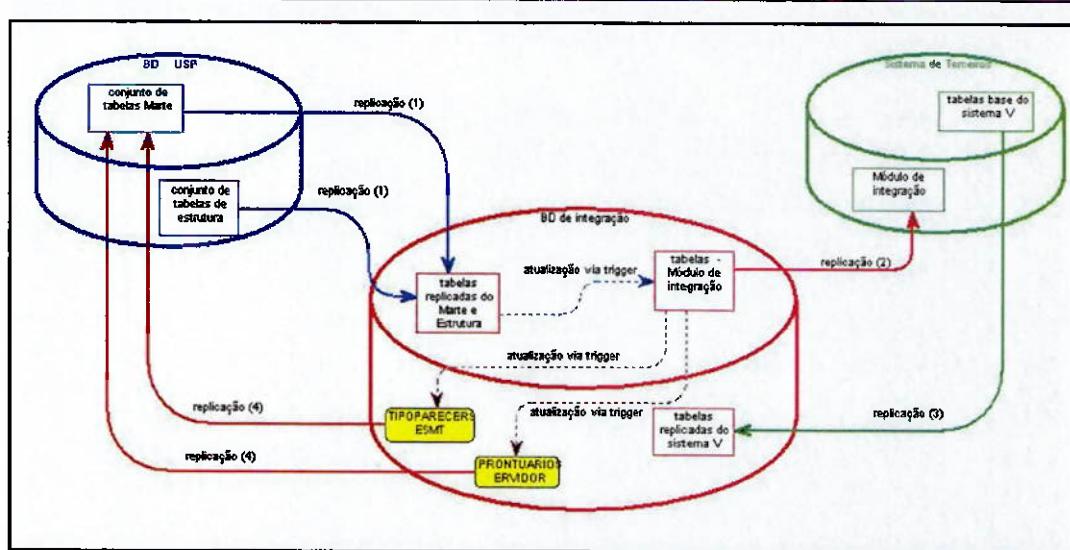


Figura 5 – Integração com os sistemas terceirizados

O mesmo ocorre com as aplicações locais dos sites replicados.

Além disso, para receber os dados da replicação, as Unidades devem disponibilizar toda uma infra-estrutura de TI e de pessoal capacitado para manutenção do novo ambiente, o que na maioria das vezes pode ser um empecilho, tanto é que nem todas recebem a replicação.

Mesmo na área de informática central da corporação ocorrem problemas decorrentes do modelo atual:

- Retrabalho – com as regras de negócio embutidas nos códigos dos aplicativos ou espalhadas na forma de *stored procedures* nos bancos de dados, sempre que há uma manutenção no modelo é necessário verificar várias fontes diferentes.
- Reuso – devido às especificidades de cada área de negócio no tratamento do mesmo conceito de dados, o reuso torna-se praticamente nulo.
- Complexidade da lógica de negócio – praticamente todos são obrigados a assimilar a complexidade inerente a cada área de negócio, visto que por compartilharem dados comuns, um sistema pode acabar

interferindo no outro, caso ele não considere determinadas regras de tratamento dos dados.

Resumindo, os problemas identificados são:

Tabela 1 – Problemas de integração

Problemas de integração	
P1	Repasso do modelo de dados
P2	Indisponibilidade da informação
P3	Indisponibilidade do sistema
P4	Custo de infra-estrutura
P5	Retrabalho
P6	Reuso
P7	Complexidade da lógica de negócio

3.3 Definição de Escopo do Exemplo Prático

Para facilitar o entendimento e a proposta de uma nova arquitetura iremos estudar a integração de informação entre o sistema A, que realiza a gestão de cursos de especialização, aperfeiçoamento e difusão, com outros sistemas similares que não utilizam o ambiente corporativo da USP, que neste caso denominaremos de sistema X.

Esta integração é necessária principalmente para que se possa efetuar a emissão de certificados, cadastro e controle de matrícula de forma independente por sistemas locais e garantir a homogeneidade e controle das informações e regras de negócio da corporação.

3.4 Requisitos Não Funcionais x Qualidade de Software

Com base no cenário descrito anteriormente, nos atributos e problemas de integração elegemos alguns requisitos não funcionais descritos na Tabela 2 para nos ajudar no estudo de uma nova arquitetura. Esses requisitos foram selecionados observando-se o levantamento pré-existente realizado nos sistemas corporativos.

Tabela 2 – Lista de requisitos não funcionais

	Requisitos Não Funcionais
RNF1	O sistema deve garantir a integridade de dados
RNF2	O sistema deve estar preparado para atender vários usuários simultâneos e concorrentes no ambiente da Internet/Intranet.
RNF3	O sistema deverá ser suficientemente robusto para permitir acesso 24h por dia, todos os dias da semana.
RNF4	Controle de login com perfil de usuários
RNF5	Dificuldade de Reuso devido ao alto grau de acoplamento

De posse desses requisitos e dos problemas relacionados na seção 3.2, utilizamos a norma ISO/IEC 9126-1 para definirmos as metas de melhoria desejadas conforme pode ser visto na Tabela 3, porém vale notar que nem todas as subcaracterísticas serão utilizadas nesta proposta.

Tabela 3 – Mapeamento RNFs x Características da ISO 9126

Característica	Subcaracterística	Descrição dos problemas encontrados atualmente	Requisitos Não Funcionais	Metas
Funcionalidade	Adequação			
	Acurácia			
	Interoperabilidade	Para integrar com sistemas de terceiros muitas vezes é necessário redundância de informação, redundância de código e criação de modelo de integração	RNF5	Diminuir o grau de acoplamento entre os sistemas Legados e facilitar a criação de novas aplicações
	Conformidade			
	Segurança de acesso	Hierarquia de acesso (herança de permissões)	RNF4	Permitir o controle de login por perfil de usuários
Confiabilidade	Maturidade	Dificuldade de acesso no período de matrículas	RNF3	Criar mecanismos que possibilitem o balanceamento de acesso
	Tolerância a falhas			
	Recuperabilidade			
Usabilidade	Inteligibilidade	Conhecimento do modelo ER para as unidades	RNF5	Criar mecanismos que possibilitem a disponibilização de serviços
	Apreensibilidade			
	Operacionalidade			

Característica	Subcaracterística	Descrição dos problemas encontrados atualmente	Requisitos Não Funcionais	Metas
Eficiência	Tempo	Dificuldade de acesso no período de matrículas (demora no tempo de resposta)	RNF2	—
	Recursos	Falta de balanceamento de acesso por definição do negócio	RNF2	Criar mecanismos que possibilitem o balanceamento de recursos
		Redundância de HW e SW para as unidades	RNF1	Criar serviços que permitam a manutenção das informações

Manutenibilidade	Analisabilidade	São identificadas facilmente, porém a identificação do local a ser corrigido não é simples.	RNF1 RNF5	Criar mecanismos que possibilitem a disponibilização de serviços
	Modificabilidade	Dificuldade quando a mudança envolve várias aplicações ou Bancos de Dados	RNF1 RNF5	Criar mecanismos que possibilitem a disponibilização de serviços
	Estabilidade			
	Testabilidade			

Portabilidade	Adaptabilidade	Mudança de plataforma requer muito esforço de desenvolvimento	RNF5	Criar mecanismos que possibilitem a disponibilização de serviços
	Capacidade para ser instalado			
	Conformidade			
	Capacidade para substituir			

4 Especificação da Arquitetura

Este capítulo apresenta a proposta de SOA, baseada no conceito de visões existentes na especificação do RM-ODP [3] , divididas da seguinte forma: visão empresa, visão informação, visão computacional e visão engenharia.

A utilização de visões está na necessidade de tornar a especificação de sistemas distribuídos em pedaços gerenciáveis que podem ser relevantes aos participantes do desenvolvimento do sistema.

4.1 Visão Empresa

Esta visão descreve os principais procedimentos necessários para integração dos sistemas relatados no capítulo anterior.

1. O sistema A é responsável pelo cadastro e manutenção dos catálogos de Cursos, Programas, Edições dos Cursos, Disciplinas, Disciplinas de cada Programa e seus Oferecimentos.
 2. O sistema X poderá consultar os catálogos de Cursos, Programas, Edições dos Cursos, Disciplinas, Disciplinas de cada Programa e seus Oferecimentos.
 3. Para inscrever um aluno no Curso o sistema X utiliza o serviço “consultar pessoa” para descobrir se o aluno já existe na base de dados, caso este não exista será necessário utilizar o serviço “gerenciar pessoas”, para incluir ou alterar os dados do aluno, seguindo as regras de negócio dos sistemas corporativos.
-

-
4. Com o código do aluno, o Sistema X fica responsável pela matrícula e demais controles dos alunos através dos serviços disponíveis pelo sistema A e ao mesmo tempo possui autonomia.
 5. Para gerenciar docentes do Curso, o sistema X utiliza o serviço “consultar pessoa” para descobrir se o docente já existe na base de dados, caso este não exista será necessário utilizar o serviço “gerenciar pessoas”, para incluir ou alterar os dados do docente, seguindo as regras de negócio dos sistemas corporativos.

Serviços necessários para a integração entre os sistemas A e X

- Consultar catálogo de Cursos
- Consultar catálogo de Edições do Curso
- Consultar catálogo de Disciplinas
- Consultar catálogo de Programa de Curso
- Consultar catálogo de Disciplinas do Programa
- Consultar catálogo dos Oferecimentos da Disciplina
- Gerenciar Pessoas
- Matricular o aluno no Curso
- Matricular o aluno em Disciplinas
- Gerenciar o andamento da Matrícula (Histórico Escolar / Notas/ Freqüência)
- Solicitar Emissão do Certificado
- Consultar Certificados

4.2 Visão Informação e Computação

Nesta seção apresentamos um esboço do diagrama de classes e dos principais serviços de acordo com as necessidades descritas na seção anterior.

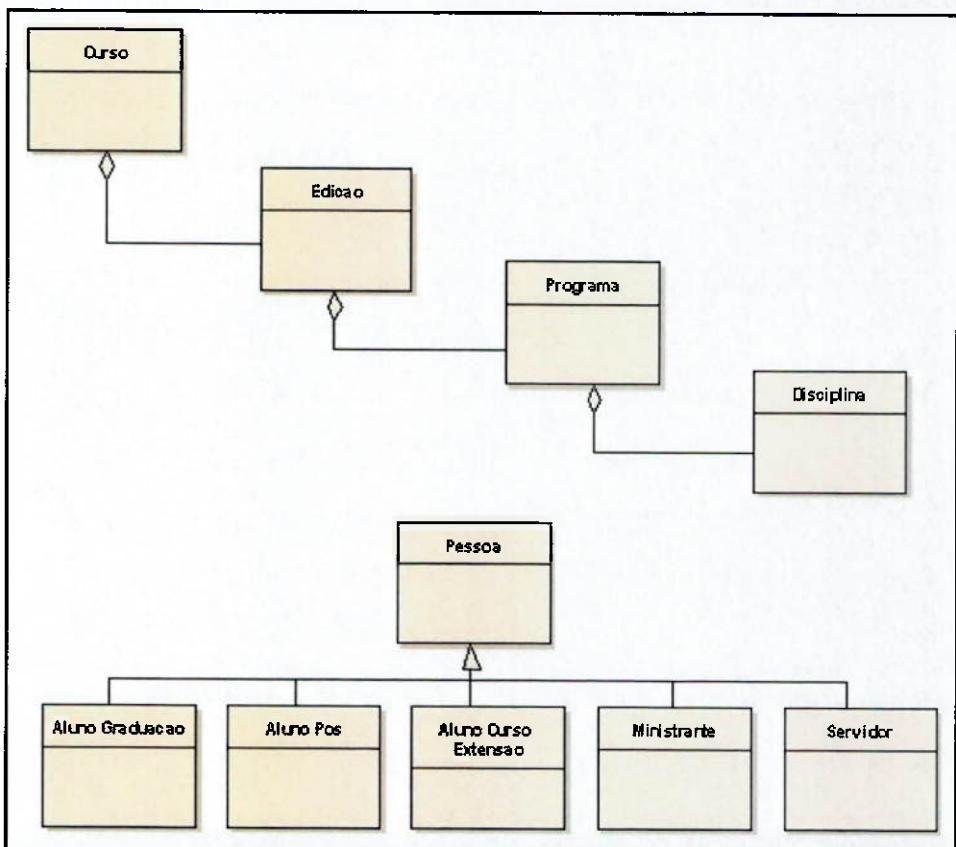


Figura 6 – Exemplo de diagrama de classes

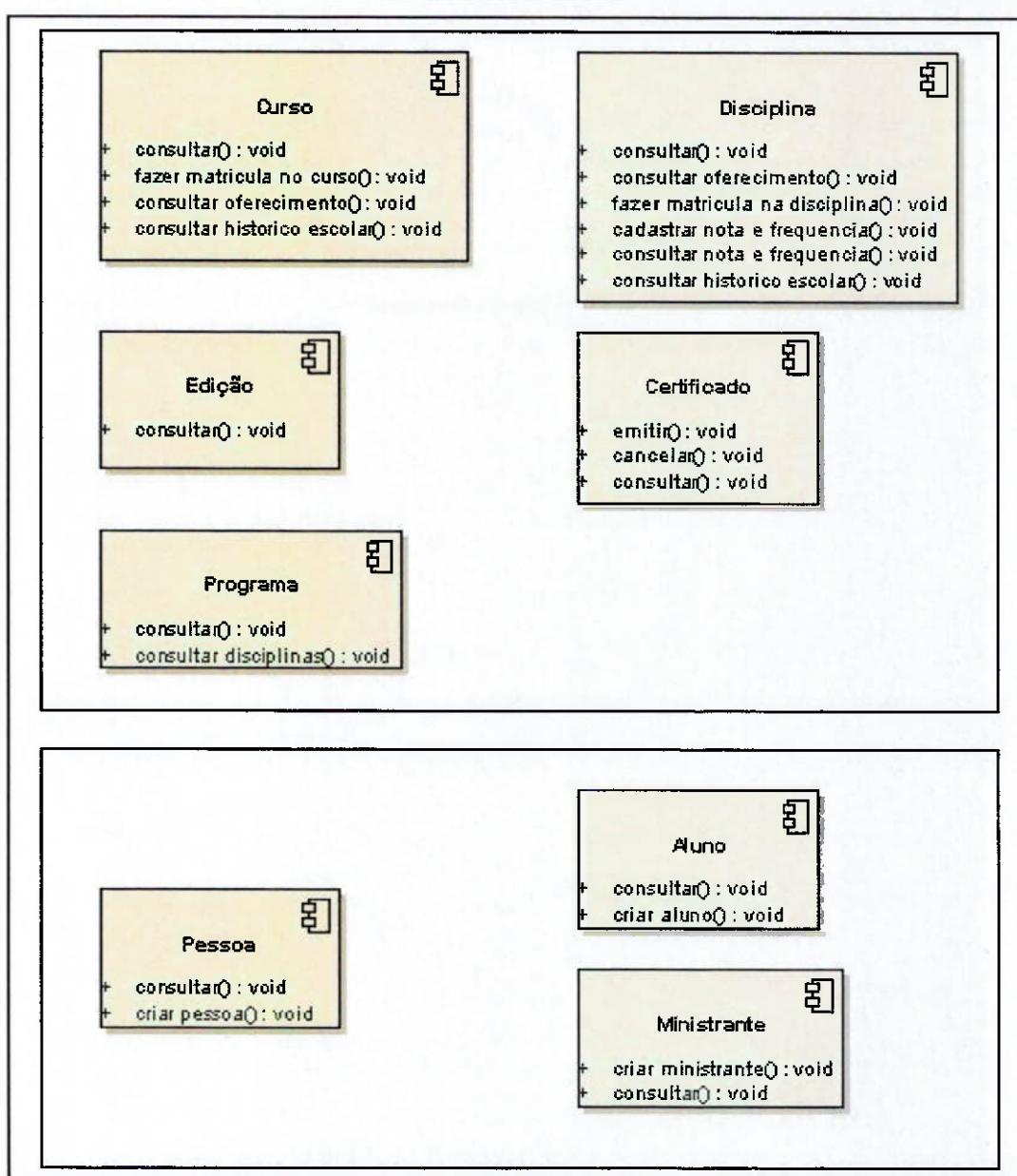


Figura 7 – Exemplo de serviços

4.3 Visão Engenharia

A fim de sanar os problemas apresentados anteriormente, propõe-se utilizar uma arquitetura orientada a serviços.

Assim, tanto as aplicações *client/server* e *web* corporativas quanto as locais, desenvolvidas pelas Unidades, bem como aplicações construídas por terceiros acessariam o banco de dados central através desses serviços.

Isso possibilitaria, além de encapsular a complexidade do modelo e da lógica de negócio, controlar melhor o acesso à base de dados. Visto que as consultas são responsáveis pela quase totalidade das conexões, seria possível, por exemplo, criar uma base replicada onde fossem efetuadas somente as consultas e as demais transações continuassem na base primária.

A Figura 8 ilustra o novo cenário:

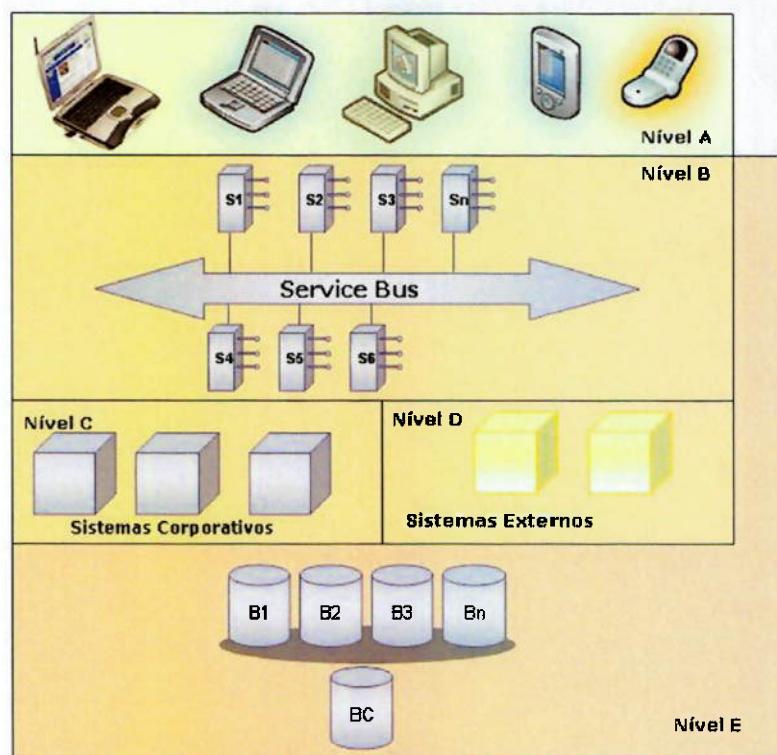


Figura 8 – Proposta SOA

Nível A: Ponto único de acesso para clientes, empregados, parceiros e fornecedores interagirem com conteúdo, aplicações, processos e pessoas da corporação.

Nível B: Camada que fornece os processos de negócio, softwares de integração, padrões de integração, orquestração e disposição dos serviços.

Nível C: Esta camada representa os sistemas desenvolvidos pelo Departamento de Informática.

Nível D: Representa os sistemas externos, desenvolvidos por parceiros ou unidades de negócio da universidade.

Nível E: Fornece acesso às informações (banco de dados, arquivos texto, etc.)

4.4 Visão Tecnologia

Diversas empresas têm desenvolvido softwares para auxiliar no desenvolvimento, implantação e manutenção de um sistema baseado em SOA. A seguir serão apresentadas algumas dessas ferramentas.

4.4.1 BEA WebLogic

Dentre a suíte de ferramentas de SOA da BEA, destacam-se [15] :

- BEA WebLogic Server – Servidor de aplicações para aplicações de missão crítica e Arquitetura Orientada a Serviços (SOA). Permite que as aplicações estejam ativas, mesmo trocando de versão, mudando a configuração do servidor ou redirecionando entre centros de dados (*data centers*);

-
- BEA WebLogic Portal – Permite a criação de um portal da empresa que simplifica a produção e gerenciamento de portais customizados orientados a serviço. O suporte a “portais confederados” (*federated portals*) baseados em padrões aumentam a produtividade, agilidade e velocidade, permitindo a adaptação a mudanças de negócios com complexidade e esforços mínimos;
 - BEA WebLogic Integration – Constrói e amplia serviços corporativos, incluindo web e legado, ao integrar aplicações, processos e usuários. Oferece uma solução para integração de aplicações e também atua como um elemento estratégico fundamental nas soluções de SOA para integração corporativa.

4.4.2 Microsoft BizTalk

O Microsoft BizTalk Server é um servidor de integração, que permite às companhias automatizarem e otimizarem seus processos de negócio. Ele possui ferramentas para desenhar, desenvolver, implementar e gerenciar esses processos [16].

Com o Microsoft BizTalk Server, um desenvolvedor pode criar soluções mais flexíveis para processos de negócio integrados e os administradores e usuários de negócio podem monitorar mais efetivamente processos de negócio em andamento.

4.4.3 IBM Websphere

A suíte de ferramentas Websphere da IBM apresenta softwares para [17] :

- Modelagem – Websphere Business Modeler. Permite que analistas de TI e de negócios possam modelar e desenhar fluxos de processos antes deles serem implementados.

-
- Desenvolvimento – Websphere Integration Developer. Trata-se de uma ferramenta de desenvolvimento utilizada para compor aplicações baseadas em processos de negócio específicos.
 - Tratamento de dados – Websphere Data Interchange. Permite tratamento e conversão de dados e gerenciamento de transações.

4.4.4 Sun Java

A Sun possui diversos softwares para implementação da SOA. Entre as principais ferramentas da companhia, podemos citar [18] :

- Java CAPS – Pacote de ferramentas para integração, infra-estrutura e desenvolvimento, que incluem softwares para modelagem, testes, implementação, monitoramento, gerenciamento e otimização de processos de negócio; extração, transformação e carga de dados; projeto e construção de web services, entre outros.
- Java System Application Platform Suite – Direciona o uso de web services para integrar diversos sistemas empresariais.

4.4.5 Oracle

A suíte de ferramentas SOA da Oracle possui softwares para os mais diversos fins, dentre os quais destacam-se [19] :

- BPEL Process Manager – Mecanismo de BPEL (business process execution language) para orquestração de web services que permite o desenho, definição e execução de processos de negócio.

-
- Web Services Manager – Console único para proteger e gerenciar web services.
 - Enterprise Service Bus – Produto baseado em padrões para conectar sistemas de informação existentes e parceiros de negócios através de uma série de serviços.
 - JDeveloper – Ferramenta de desenvolvimento para criação e composição de aplicações.

4.5 Mapa de Melhorias (Metas)

Tabela 4 – Mapeamento de problemas, requisitos não-funcionais e respectivas melhorias

Problemas elencados	Requisitos	Visões			Metas
		Empresa	Informação e Computação	Engenharia	
P1	RNF5	✓	✓	✓	<ul style="list-style-type: none"> • Encapsulamento da complexidade e detalhes do processo de negócio; • Permite que as Unidades que não possuem infra-estrutura tenham acesso às informações corporativas; • Implementação dos conceitos de Negócio de forma transparente; • Diminuição de custos de Desenvolvimento e Manutenção.
P2	RNF2 RNF3	✓	-	✓	<ul style="list-style-type: none"> • Desacoplamento, permitindo manutenção pontual; • Flexibilidade dos sistemas (Plataforma / Saída de Dados) - facilidade da migração de ambiente (troca de plataforma / banco de dados, eventual roteamento devido à indisponibilização do banco de dados central, etc.).
P3	RNF2 RNF3	✓	-	✓	<ul style="list-style-type: none"> • Desacoplamento, permitindo manutenção pontual; • Flexibilidade dos sistemas (Plataforma / Saída de Dados) - facilidade da migração de ambiente (troca de plataforma / banco de dados, eventual roteamento devido à indisponibilização do banco de dados central, etc.).

Problemas elencados	Requisitos	Visões			Metas
		Empresa	Informação e Computação	Engenharia	
P4	RNF1				<ul style="list-style-type: none"> •Flexibilidade dos sistemas (Plataforma / Saída de Dados) - facilidade da migração de ambiente (troca de plataforma / banco de dados, eventual roteamento devido à indisponibilização do banco de dados central, etc.);
	RNF2	✓	-		<ul style="list-style-type: none"> •Expansão do acesso às informações corporativas para potenciais usuários / parceiros;
	RNF3		-		<ul style="list-style-type: none"> •Permite que as Unidades que não possuem infra-estrutura tenham acesso às informações corporativas;
	RNF5				<ul style="list-style-type: none"> •Redução de custos de manutenção do ambiente replicado nas Unidades que participam da replicação hoje; •Diminuição de custos de Desenvolvimento e Manutenção.
	P5	RNF5	-	✓	<ul style="list-style-type: none"> •Reaproveitamento de código - facilitar e estimular o reuso; •Desacoplamento, permitindo manutenção pontual; •Diminuição de custos de Desenvolvimento e Manutenção.
P6	RNF5	-	✓	-	<ul style="list-style-type: none"> •Reaproveitamento de código - facilitar e estimular o reuso; •Criação de novas oportunidades de Negócio; •Implementação dos conceitos de Negócio de forma transparente; •Diminuição de custos de Desenvolvimento e Manutenção.
	RNF1 RNF4 RNF5				<ul style="list-style-type: none"> •Encapsulamento da complexidade e detalhes do processo de negócio; •Expansão do acesso às informações corporativas para potenciais usuários / parceiros; •Criação de novas oportunidades de Negócio; •Implementação dos conceitos de Negócio de forma transparente.
P7					

5 Conclusões

Através da análise dos resultados obtidos com este estudo, detalhados através do exemplo prático e da proposta de SOA apresentados nos capítulos 3 e 4, respectivamente, conclui-se que muitos dos benefícios obtidos com a utilização de serviços, como o encapsulamento da complexidade e detalhes do negócio, reuso, desacoplamento, entre outros, auxiliam na terceirização de sistemas ou módulos de sistemas, facilitam o desenvolvimento de aplicações e a integração entre elas, melhoram o relacionamento com os clientes, possibilitam a melhoria de processos e geram novas oportunidades de negócio. Pode-se dizer, então, que a SOA atende a muitas das questões que motivaram a execução deste trabalho.

Porém, vale ressaltar que a SOA não é um remédio para todos os males. Nem todo tipo de aplicação é apropriado para esta arquitetura. A Tabela 5 descreve alguns usos incorretos dessa arquitetura (os *antipatterns*) [7] [9] .

Tabela 5 – SOA antipatterns

Trem da tecnologia	Quando as empresas “pulam” para dentro do trem da tecnologia sem avaliar cuidadosamente se é apropriada para suas necessidades.
Qual é a novidade?	A falta de entendimento da diferença de aplicações SOA para outros paradigmas pode levar a conclusão de que é apenas um novo nome para velhas técnicas, implicando em aplicações inadequadas e resultados contraditórios.
Registros mal comportados	Registros duplicados de serviços, sobreposição de funções e falta de quem controla qual parte (cross application), podem gerar custos maiores e performance ruim.

5.1 Trabalhos futuros

Entre alguns dos próximos passos a serem dados no projeto, podemos citar:

- Garantir que todos os sistemas corporativos estejam utilizando os serviços disponíveis;
- Implementar melhores técnicas de segurança no acesso aos dados;
- Abranger / estender os serviços para outras áreas de negócio ou assuntos, como Matrículas de Graduação, Solicitação de Bolsas de Pesquisa, etc;
- Eliminar a necessidade de manutenção de um parque tecnológico e profissional capacitados nos sites replicados;
- Permitir o acesso aos dados corporativos a todas as Unidades da USP e parceiros de negócios;
- Exportar, publicar e gerenciar esses serviços.

6 Referências

- [1] ISO/IEC 9126-1:2001, International Standard. **Information Technology – Software engineering –Product quality – Part 1:Quality model.** First edition 2001-06-15.
- [2] NBR 13596, **Tecnologia de Informação – Avaliação de Produto de Software – Características de Qualidade e Diretrizes para o seu uso.** ABNT – Abril 1996 (versão brasileira da Norma ISO/IEC 9126,1991).
- [3] ISO IEC; **Open Distributed Processing – Reference Model: Overview:** 10746-1. Switzerland. 1998.
- [4] Barbieri, C.; **BI-Business Intelligence – Modelagem & Tecnologia**, Axcel Books do Brasil Editora, 2001.
- [5] Jones, R.; **B2B integration. Manufacturing Engineer.** Volume 80, Issue 4, Aug. 2001, Page(s):165 - 168.
- [6] Turban, E.; King, D.; **Comércio Eletrônico Estratégia e Gestão.** São Paulo: Prentice Hall, 2004. Cap.5
- [7] Sampaio, C.; **SOA e Web Services em Java.** Rio de Janeiro: Brasport, 2006
- [8] Endrel, M. et al; **Patterns: Service-Oriented Architecture and Web Services.** RedBooks, IBM, Apr. 2004.
Disponível em: <http://www.redbooks.ibm.com/redbooks/pdfs/sg246303.pdf>
Acesso em: jul. de 2006.
- [9] Ganci, J. et al; **Patterns: SOA Foundation Service Creation Scenario.** RedBooks, IBM, Apr. 2004
-

Disponível em: <http://www.redbooks.ibm.com/redbooks/pdfs/sg247240.pdf>
Acesso em: dez de 2007.

[10] Botto, R.; **Arquitetura Corporativa de Tecnologia da Informação**. Rio de Janeiro: Brasport,2004.

[11] Ort, E.; **Service-Oriented Architecture and Web Services: Concepts, Technologies, and Tools**. Article, Sun Microsystems, April 2005.

Disponível em:

<http://java.sun.com/developer/technicalArticles/WebServices/soa2/soa2.pdf>

Acesso em: ago. de 2006.

[12] Microsoft; **SIEMENS AG GERMANY**. Case Study, October 2005.

Disponível em:

http://download.microsoft.com/documents/customerevidence/21494_Siemens_BizTalk2006_Final.doc

Acesso em: Dez. de 2006

[13] BEA System Inc; **Case Study – Toyota Australia Drives Out Cost and Increases Visibility with BEA Solution for Dealers**, June 2004.

Disponível em:

http://www.bea.com/content/news_events/white_papers/IDC_ToyotaAustralia_cs.pdf

Acesso em: Dez. de 2006

[14] Sordi, J.; Marinho, B.; Nagy, M.; **Case Benefícios da Arquitetura de software orientada a serviços para as empresas: Análise da Experiência do ABN AMRO Brasil**, TECSI FEA USP, 2006.

Disponível em:

<http://www.tecsifea.usp.br/revistatecsi/edicoesanteriores/v03n01-2006/v3n1a2.pdf>

Acesso em: Jan. de 2007

[15] BEA WebLogic Product Family

Disponível em:

<http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products/weblogic/>

Acesso em: Jan. de 2007

[16] Microsoft BizTalk Server 2006

Disponível em: <http://www.microsoft.com/biztalk/default.mspx>

Acesso em: Jan. de 2007

[17] IBM

Disponível em: <http://www-306.ibm.com/software/br/websphere/>

Acesso em: Jan. de 2007

[18] Sun

Disponível em: <http://www.sun.com/products/soa/index.jsp>

Acesso em: Jan. de 2007

[19] Oracle

Disponível em: <http://www.oracle.com/technologies/soa/soa-suite.html>

Acesso em: Jan. de 2007
