

**Leandro Fregnani**

**Proposta de integração de sistemas de informação utilizando  
Arquitetura Orientada a Serviços.**

**São Paulo  
2011**

**Leandro Fregnani**

**Proposta de integração de sistemas de informação utilizando  
Arquitetura Orientada a Serviços.**

Monografia apresentada à Escola  
Politécnica da Universidade de São  
Paulo para obtenção do Título de  
MBA em Tecnologia da Informação

**Orientador**  
**Prof. Dr. Jorge Luis Risco Becerra**

**São Paulo**  
**2011**

MBA/TI /  
2010  
F858 p.

DEDALUS - Acervo - EPEL



31500020026

FICHA CATALOGRÁFICA

M 2010 S /

Fregnani, Leandro

Proposta de integração de sistemas de informação utilizando Arquitetura Orientada a Serviços/ L. Fregnani. -- São Paulo, 2011.

65 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. Arquitetura de informação  
3. Governança corporativa I. Universidade de São Paulo. Escola Politécnica. Programa de Educação Continuada em Engenharia II. t.

PECC

2195743

## **AGRADECIMENTOS**

Agradeço a Deus e Santo Expedito por toda força, coragem e determinação, pois através de proteção e iluminação, consegui alcançar os meus objetivos de vida.

Ao meu mentor e colega de trabalho Silvio De Paula, por estar sempre me ajudando desde o primeiro dia de USP até os dias de hoje, ao meu diretor Luis Moreira que me apoiou desde o início do MBA até o momento dessa monografia.

Aos meus pais e irmão que sempre me deram toda a infraestrutura necessária, apoio, incentivo, ajuda e são os meus pilares de sustentação da minha vida.

Ao meu orientador professor doutor Jorge Risco com toda sua sabedoria, que me ajudou desde o início do curso de MBA até me tendo como seu orientado me auxiliando no desenvolvimento desse trabalho.

Aos meus amigos por todo incentivo e paciência de me ouvir nas horas que mais precisava de ajuda.

À minha noiva e futura esposa que sempre teve muita paciência, amor, compreensão e nos momentos que mais precisava me cedia seu ombro e seu ouvido para ouvir as minhas lamentações, decepções, alegrias e conquistas.

## RESUMO

A evolução dos sistemas de informação tem cada vez mais destaque dentro de uma organização, com isso à integração de dados, de processos de negócio, entre outros são fatores fundamentais para que as decisões sejam precisas e rápidas. Desta maneira, este trabalho propõe a utilização de uma Arquitetura Orientada a Serviços para realizar tais integrações.

Como exemplo será apresentado o cenário atual do ambiente de TI da USP, mas especificamente do Departamento de Informática da Reitoria. Será apresentada uma proposta de integração de sistemas na universidade, utilizando os princípios da arquitetura SOA com o conceito de visões da especificação do RM-ODP de modo que as visões auxiliem o alinhamento entre o negócio e a TI.

Através desta metodologia alguns benefícios podem ser obtidos, entre eles, o encapsulamento da complexidade do negócio, reusabilidade, desacoplamento, facilitando assim o desenvolvimento de aplicações e a integração entre elas, melhorando o relacionamento com os clientes, possibilitando a melhoria de processos e gerando novas oportunidades de negócio.

Palavras-Chave: Sistemas de Informação, Integração de Sistemas, SOA, RM-ODP.

## **ABSTRACT**

The evolution of the information systems has each time inside more prominence of an organization, with this to the integration of data, processes of business, among others they are basic factors so that the decisions are necessary and fast. In this way, this work considers the use of a Guided Architecture the Services to carry through such integrations.

As example the current scene of the environment of TI of the USP, but specifically of the Department of Computer science of the rector will be presented. It will be presented a proposal of integration of systems in the university, using the principles of the architecture SOA with the concept of vision of the specification of the RM-ODP in way that the vision assist the alignment between the business and TI.

Through this methodology some benefits can be gotten, between them, the encapsulate of the complexity of the business, reusability, uncoupling, thus facilitating the development of applications and the integration between them, improving the relationship with the customers, making possible the improvement of processes and generating new chances of business.

**Keywords:** Information Systems, Systems Integration, SOA, RM-ODP.

## SUMÁRIO

|                                                                 |    |
|-----------------------------------------------------------------|----|
| <b>1. INTRODUÇÃO</b>                                            | 11 |
| 1.1. Objetivos                                                  | 11 |
| 1.2. Justificativa                                              | 12 |
| 1.3. Motivações                                                 | 13 |
| 1.4. Metodologia de elaboração                                  | 14 |
| <b>2. FUNDAMENTO CONCEITUAL</b>                                 | 15 |
| 2.1. Integração de aplicações corporativas                      | 15 |
| 2.2. Arquitetura orientada a serviços                           | 18 |
| 2.2.1. OASIS                                                    | 20 |
| 2.3. Serviços Web                                               | 22 |
| 2.3.1. Modelo de serviços Web                                   | 22 |
| 2.3.2. SOAP (Simple Object Access Protocol)                     | 23 |
| 2.3.3. WSDL (Web Services Description Language)                 | 24 |
| 2.3.4. UDDI (Universal Description, Discovery, and Integration) | 26 |
| 2.3.5. Barramento Corporativo de Serviços                       | 27 |
| 2.4. ISO Reference Model of Open Distributed Processing. RM-ODP | 28 |
| 2.4.1. Ponto de vista da Empresa                                | 30 |
| 2.4.2. Ponto de vista da Informação                             | 31 |
| 2.4.3. Ponto de vista da Computação                             | 31 |
| 2.4.4. Ponto de vista da Engenharia                             | 32 |
| 2.4.5. Ponto de vista da Tecnologia                             | 32 |
| 2.4.6. Considerações Finais                                     | 33 |
| <b>3. DEFINIÇÃO DO PROBLEMA</b>                                 | 34 |
| 3.1. Integração entre os sistemas                               | 34 |
| 3.2. Problemas encontrados                                      | 38 |
| 3.3. Definição de escopo do exemplo prático                     | 40 |

|                                                                   |           |
|-------------------------------------------------------------------|-----------|
| <b>4. PROPOSTA DE INTEGRAÇÃO DE SISTEMAS UTILIZANDO SOA .....</b> | <b>41</b> |
| 4.1. Metodologia adotada para integração .....                    | 41        |
| 4.1.1 Análise de Processos.....                                   | 45        |
| 4.1.2 Estrutura de informação para codificação dos serviços ..... | 46        |
| 4.1.3 Interação dos serviços .....                                | 47        |
| 4.1.4 Tecnologia Utilizada.....                                   | 48        |
| 4.1.5 Identificação da Comunicação .....                          | 48        |
| <b>5. CONCLUSÃO .....</b>                                         | <b>63</b> |
| 5.1. Trabalhos Futuros .....                                      | 64        |
| <b>6. REFERÊNCIAS BIBLIOGRÁFICAS.....</b>                         | <b>65</b> |



## **LISTA DE FIGURAS**

Figura 1 - Integração de sistemas utilizando EAI. Fonte[2].

Figura 2 - Agentes de uma arquitetura SOA.

Figura 3 - Relacionamento entre as especificações de serviços web.

Figura 4 - Pontos de Vista do RM-ODP.

Figura 5 - Cenário Atual DI USP.

Figura 6 - Etapas para a implementação de Serviços.

Figura 7 - Modelo de processo do Pagamento da Gratificação de Convênio.

Figura 8 - UC Pagamento Gratificação de Convênio.

Figura 9 - Diagrama de Classes Pagamento Gratificação do Convênio.

Figura 10 - Diagrama de componentes.

Figura 11 - Diagrama de Seqüência Pagamento da Gratificação do Convenio.

Figura 12 - Visão da Engenharia.

## **LISTA DE TABELAS**

Tabela 1 – Cenário atual no Departamento de Informática da USP

Tabela 2 - Tabela exemplo processo x problema a ser resolvido

Tabela 3 - Tabela comparativa visões RM-ODP X atividades para SOA.

## LISTA DE ABREVIATURAS E SIGLAS

|              |                                                                              |
|--------------|------------------------------------------------------------------------------|
| <b>B2B</b>   | Business to Business                                                         |
| <b>B2C</b>   | Business to Consumers                                                        |
| <b>BPMN</b>  | Business Process Management Notation                                         |
| <b>CRM</b>   | Customer Relationship Management                                             |
| <b>DI</b>    | Departamento de Informática                                                  |
| <b>EAI</b>   | Enterprise Application Integration                                           |
| <b>ERM</b>   | Enterprise Relationship Management                                           |
| <b>ESB</b>   | <i>Enterprise Service Bus</i> , ou Barramento Corporativo de Serviços        |
| <b>LDAP</b>  | Lightweight Directory Access Protocol                                        |
| <b>OASIS</b> | Organization for the Advancement of Structured Information Standards.        |
| <b>PRM</b>   | Partner Relationship Management                                              |
| <b>SCM</b>   | Supply Chain Management                                                      |
| <b>SOA</b>   | <i>Service-Oriented Architecture</i> , ou Arquitetura Orientada a Serviço    |
| <b>SOAP</b>  | Simple Object Access Protocol                                                |
| <b>TI</b>    | Tecnologia da Informação                                                     |
| <b>UDDI</b>  | Universal Description, Discovery, and Integration                            |
| <b>USP</b>   | Universidade de São Paulo                                                    |
| <b>XML</b>   | eXtensible Markup Language, ou Linguagem Extensível de Formatação            |
| <b>WSDL</b>  | Web Services Description Language, ou Linguagem de Descrição de Serviços Web |

## **1. INTRODUÇÃO**

A proposta apresentada surgiu com base na experiência obtida no Departamento de Informática da USP. Com a crescente demanda de distribuição de dados e integração dos sistemas corporativos, esse crescimento naturalmente acaba ficando um pouco desorganizado e cada vez mais difícil de integrarmos os sistemas de uma maneira mais centralizada.

Os desenvolvimentos tem se tornado cada vez mais dinâmico, exigindo uma maior agilidade por parte das organizações, consequentemente, de suas áreas de TI. É necessário que a arquitetura de sistemas da informação seja a de adaptar rapidamente as necessidades de negócio, em constante mudança.

Dentro deste cenário, vem se desenvolvendo e sendo aceito o conceito de Arquitetura Orientada a Serviços (SOA). O modelo de referência OASIS (OASIS SOA REFERENCE MODEL TC, 2006), define SOA como um paradigma, para organizar e utilizar competências distribuídas entre vários domínios. Trata-se de um modo de estruturar as aplicações de software em elementos chamados serviços. [10]

### **1.1. Objetivos**

O objetivo é propor a utilização de uma arquitetura orientada a serviços na integração dos processos de negócio, e como apoio utilizar o conceito de visões da especificação do RM-ODP [1], de maneira que se possam identificar alguns dos problemas enfrentados atualmente pelas equipes de desenvolvimento de sistemas e alguns benefícios alcançados através desse estudo.

Esse trabalho tem como propósito estabelecer uma metodologia para auxiliar a criação e utilização dos serviços alinhados aos requisitos de negócio e aderentes aos princípios de orientação a serviços.

Levando em conta o ciclo de vida dos processos de negócio dos sistemas de informação, pode-se dizer que a orientação a serviços serviria como ponte para chegar-se às especificações. Os serviços seriam os elementos responsáveis por realizar esta ligação, por possuírem a característica de serem elementos desenvolvidos pela TI para prover funções de negócio.

A utilização de um modelo de referencia se fez necessária nesse trabalho, pois assim desta maneira fornece uma linguagem comum e bem definida, sendo assim, o RM-ODP ajuda na divisão da especificação do sistema em pontos de vista, de forma a simplificar a descrição de sistemas complexos, um modelo de suporte para a infraestrutura, fornecimento de conceitos gerais como ferramentas de especificação e através dos pontos de vista, gerar artefatos de modo a auxiliar as fases de construção do sistema.

## **1.2. Justificativa**

Esse trabalho surgiu com a necessidade de organizar o desenvolvimento de sistemas diante do crescimento dos processos de negócio suportados no departamento, propondo assim o uso de uma nova arquitetura para auxiliar o Departamento de Informática da USP que nesse caso serve como objeto de estudo.

A arquitetura sugerida é SOA (Service Oriented Architecture – Arquitetura Orientada a Serviços), que surge como proposta para facilitar a integração, manutenção e comunicação de aplicações distribuídas que interagem entre si através de uma rede de computadores, fornecendo dessa forma, uma interoperabilidade entre os sistemas de informação.

A solução é uma metodologia para criação de serviços utilizando as visões do conceito da especificação do RM-ODP (*Reference Model of Open Distributed Processing*). A utilização de visões está na necessidade de tornar a especificação de sistemas distribuídos em etapas gerenciáveis que podem ser relevantes aos participantes do desenvolvimento do sistema.

### 1.3. Motivações

O crescimento da organização, a necessidade cada vez maior de novos sistemas, a integração de dados cada vez mais necessária, nos levam a analisar alguns pontos críticos para a integração, tais como:

1. Como obter dados de outros órgãos públicos (CNPQ, CAPES, FINNEP, FAPESP) para tomada de decisão;
2. Como integrar as diferentes aplicações corporativas que suportam os processos de negócio;
3. Como diminuir o volume de replicação de dados para outras unidades;
4. Preparar as mudanças na infra-estrutura de TI para suportar a implantação de uma arquitetura orientada a serviços;
5. Como diminuir o custo de infraestrutura que cada vez tem ficado mais alto por conta da replicação das tabelas;
6. Como aumentar a reusabilidade do código;
7. Como encapsular a complexidade do negócio de modo que o cliente utiliza e sua funcionalidade sem saber o que se encontra por trás dele;

A ideia desse trabalho é que essas questões sejam solucionadas ou mitigadas através da implementação SOA. A dúvida é em relação ao item 4 que fala sobre a mudança de TI para suportar uma arquitetura orientada a serviços. Trata de uma mudança de paradigmas no que diz respeito à cultura organizacional do desenvolvimento de sistemas na USP, além é claro dos impactos tecnológicos que possam surgir.

#### 1.4. Metodologia de elaboração

Inicialmente serão abordados os conceitos sobre:

- Integração de aplicações, relações *EAI (Enterprise Application Integration)*;
- SOA (*Service-Oriented Architecture*);
- Conceitos de visões da especificação do RM-ODP.

A seguir, será apresentado o cenário atual do ambiente de TI corporativo, levantando alguns dos problemas de integração enfrentados atualmente.

Como resultado, será apresentada uma proposta de integração de sistemas, utilizando Arquitetura Orientada a Serviços para apontar possíveis melhorias 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), utilizando UML e BPMN como uma forma de demonstrar os artefatos de cada etapa, mas não é escopo do trabalho detalhar o modelo de referência, assim como detalhar também o UML e o BPMN.

Por fim será apresentada a conclusão do trabalho e trabalhos futuros que possam ser realizados.

## 2. FUNDAMENTO CONCEITUAL

Nesse capítulo serão abordados alguns conceitos básicos acerca de integração de aplicações corporativas (EAI), arquitetura orientada a serviços (SOA), os conceitos de *web services* e a especificação das visões do RM-ODP, que vão auxiliar na compreensão do desenvolvimento desse 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 B2B (*Business to Business*) e B2C (*Business to Consumers*), 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 EAI (*Enterprise Application Integration*) - integração de aplicações corporativas.

O termo EAI ou *Enterprise Application Integration* é ainda, o termo formal que contempla a integração de aplicações corporativas e um conjunto de ferramentas e tecnologias. Como a dependência das corporações em relação à tecnologia tem crescido e se tornado mais complexa, a necessidade por um método de integração de aplicações em um único arsenal de processos de negócios tem sido a prioridade. Depois da criação de ilhas de negócio, usuários e dirigentes de grandes corporações têm demandado uma ponte que possa unir os diversos sistemas espalhados pelos departamentos das empresas. Eles estão demandando a unificação das aplicações.

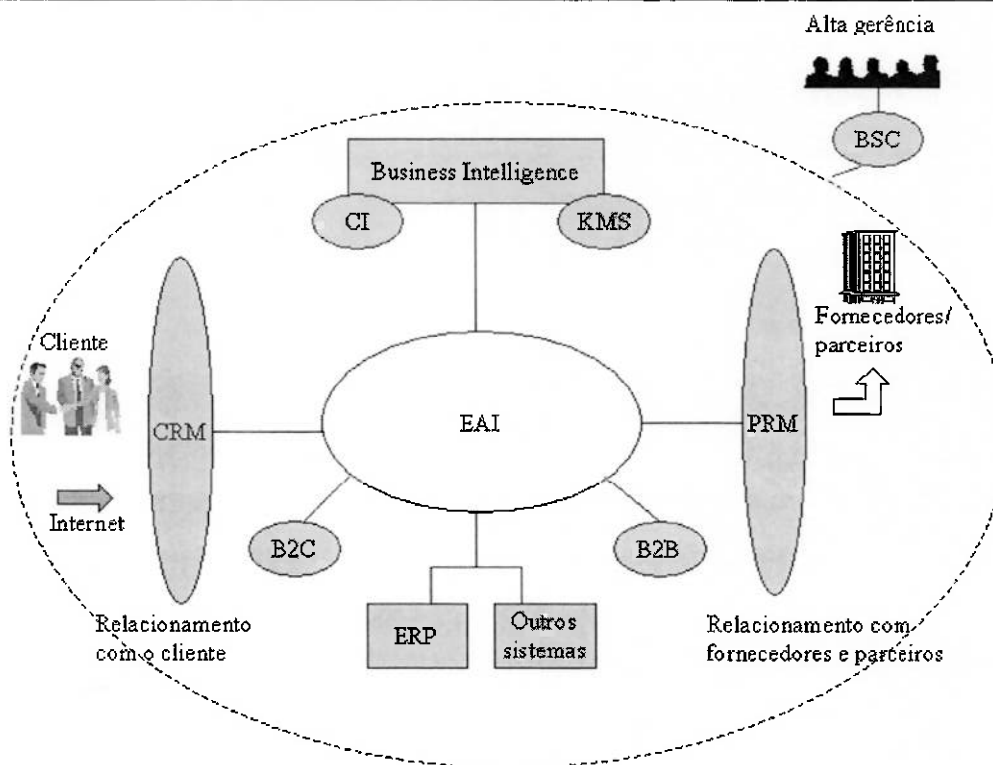


O crescimento do EAI permite o compartilhamento de processos e dados e vem atender a esta demanda [3].

Vários os motivos que estão levando a comunidade de informática a se interessar pela tecnologia de EAI, alguns exemplos são:

- Integração dos sistemas ERPs com o restante dos sistemas sobreviventes da fase empacotada, e até mesmo integração entre os diferentes módulos ERPs;
- 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 Warehouses (DW) e 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 e LDAP 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 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[2]**

Processos de negócio podem ser melhorados usando EAI, por exemplo, 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.Arquitetura orientada a serviços

É um paradigma de construção e integração de software que estrutura aplicações em elementos modulares chamados serviços.

Um estilo de arquitetura de software cujo intuito (a princípio) é disponibilizar as funcionalidades das aplicações em formas de serviços. Grande parte das vezes os serviços são organizados em um “barramento de serviços” (enterprise service bus) que disponibiliza contratos acessíveis através de Web Services, ou outra forma de comunicação entre sistemas da informação.

Uma arquitetura SOA básica é caracterizada pelas interações entre três tipos de agentes de software: Provedores de Serviços, Consumidores de serviços e o Registro do Serviço. As interações entre estes agentes podem ser vistas na figura abaixo:

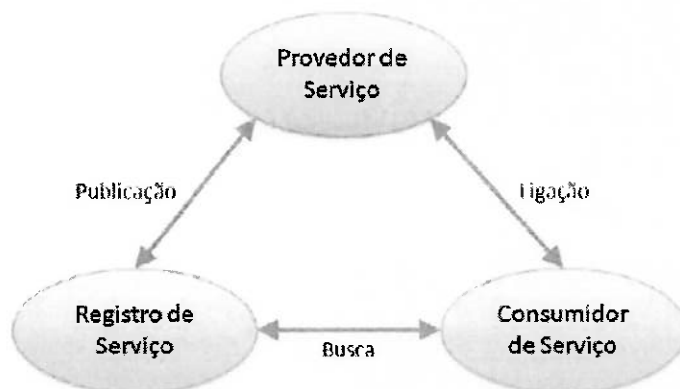


Figura 2 - Agentes de uma arquitetura SOA

A figura 2 representa a maneira que os serviços são publicados e utilizados segundo uma arquitetura orientada a serviços, o funcionamento ocorre às seguintes maneiras: 1) Um Provedor de Serviço publica a informação de implementação de um determinado serviço no UDDI que passa a documentar que esse provedor contém a implementação daquele serviço; 2) Um cliente acessa o serviço de diretório através de um protocolo conhecido, requisitando uma implementação de um determinado serviço; 3) O serviço de

diretório acessa sua documentação de provedores para o serviço indicado e responde dando uma identificação de um servidor/provedor que contém a implementação desse serviço juntamente com sua interface de comunicação; 4) Finalmente, o cliente acessa o serviço em questão fazendo uso correto da interface e recebe os resultados da execução [9].

Os arquitetos de software dizem que a criação de um bom aplicativo orientado a serviços envolve mais trabalho do que a tradicional integração de aplicativos. Para que este trabalho produza benefícios, portanto, SOA tem que eliminar trabalho em outro ponto qualquer, já que a própria metodologia não gera benefícios para o negócio. Assim, o primeiro passo é descobrir se existem aplicativos redundantes e mal integrados que poderiam ser consolidados ou eliminados como resultado da adoção. Se este for o caso, então há benefícios potenciais.

Para entender o panorama geral dos benefícios apregoados por SOA, é necessário examinar em dois níveis: primeiro, as vantagens táticas do desenvolvimento orientado a serviços e, segundo, as vantagens da SOA como estratégia de arquitetura global.

Vantagens do desenvolvimento orientado a serviços:

- Reutilização de Software: rotinas complexas utilizadas em mais de um sistema da sua empresa;
- Aumento de Produtividade: através da reutilização de códigos e processos, a equipe de desenvolvimento pode ser concentrar em outras tarefas do sistema;
- Interoperabilidade: troca de informações entre sistemas escritos em linguagens distintas;

- Escalabilidade: uma forma de melhorar o desempenho é usar uma estratégia que não é exclusiva dos Webservices, adicionar servidores e aumentar sua capacidade;

Vantagem de uma estratégia SOA:

- Melhor alinhamento com o negócio: Quando projetos de TI são apresentados em termos de atividades e processos de negócio e não na forma de aplicativos complexos, o pessoal de negócio pode apreciar e suportar melhor os projetos de TI. A visão SOA é quando TI se capacita para serviços, os processos importantes de um negócio, o pessoal de negócio poderá assumir controle sobre modificar e mesclar os diferentes serviços em novas combinações de processo próprias.
- Uma maneira melhor de defender o uso da arquitetura para o negócio e TI: Arquitetura corporativa sempre foi uma empreitada grande, complexa e cara. Padronizar, mapear e controlar ativos de TI não torna o negócio claramente mais flexível, capaz ou lucrativo. A arquitetura orientada a serviços proporciona o valor ao negócio que, na velha arquitetura corporativa, raramente passava de uma vaga promessa. Reutilização, maior produtividade e agilidade em TI e uma infraestrutura de software ajustada para processos de negócio específicos são as iscas para vender uma iniciativa de arquitetura corporativa para o negócio.

### 2.2.1. OASIS

É uma organização sem fins lucrativos que trabalha no desenvolvimento de um modelo de padrões abertos de Tecnologia da Informação. Dentro destes padrões desenvolvidos encontram-se as diversas especificações e modelos relacionados à web services e SOA.

O OASIS elaborou um modelo de referência (OASIS SOA REFERENCE MODEL TC, 2006) com o objetivo de definir a essência da arquitetura orientada a serviço, e propor um vocabulário e um entendimento comum de SOA. Ele oferece uma referência normativa que permanece relevante, para SOA como um modelo abstrato e poderoso, independente das várias inevitáveis evoluções tecnológicas que venham a influenciar a realização de SOA.

Segundo definição do OASIS, SOA é um paradigma para organizar e utilizar competências distribuídas que podem estar sob o controle de diferentes domínios. Competência é definida como elemento criado para solucionar ou dar suporte para um determinado problema de negócio. E no cenário de computação distribuída, é comum os requisitos de uma entidade poderem ser satisfeitos por uma competência oferecida por outro domínio. [10]

De acordo com o modelo de referência OASIS, os principais conceitos do paradigma de uma arquitetura orientada a serviços são:

- Visibilidade: é o relacionamento entre os consumidores e provedores de serviços, que é satisfeito quando eles estão aptos a interagir entre si. As pré-condições para a visibilidade são consciência, concordância e acessibilidade. Devem ainda estar bem definidas as restrições, políticas e mecanismos para acesso e resposta.
- Interação: permite que uma competência seja utilizada, o que envolve a execução de ações na direção do serviço. Pode ser viabilizada por meio de troca de mensagens ou mudanças no estado compartilhado de alguma entidade.
- Efeitos no mundo real: são os resultados das interações. Eles estabelecem as expectativas para quem deseja utilizar as competências oferecidas e podem incluir a informação retornada em resposta a uma requisição, uma mudança no estado compartilhado de entidades definidas ou alguma combinação dos anteriores. Em SOA, as competências podem ser utilizadas sem que todos os detalhes sejam conhecidos.

## **2.3. Serviços Web**

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 [6] [7].

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 é um tipo de relacionamento (contrato) entre um provedor e um consumidor, sendo que esse provedor se compromete em realizar determinadas tarefas com resultados pré-estabelecidos para um consumidor, e que, por sua vez, se compromete a usar o serviço da forma contratada. Baseados em padrões XML, os serviços web podem ser desenvolvidos em forma de componente de aplicações fracamente acopladas usando qualquer linguagem de programação, protocolo ou plataforma.

É importante ressaltar que as estruturas e implementação de serviços web são particulares a cada empresa [10].

### **2.3.1. Modelo de serviços Web**

As tecnologias básicas dos serviços web são geralmente descritas em termos de SOAP, WSDL e UDDI, que serão definidas nas próximas seções. Porém, cada um desses padrões pode ser usado isoladamente. Por exemplo, existem muitas implementações que usam somente SOAP e outra que utilizam SOAP e WSDL.

O framework do W3C para serviços web consiste em uma base construída no topo de três especificações XML: WSDL, SOAP e UDDI. Esses padrões desenvolvidos com base nos princípios de SOA se formam baseado em serviços.

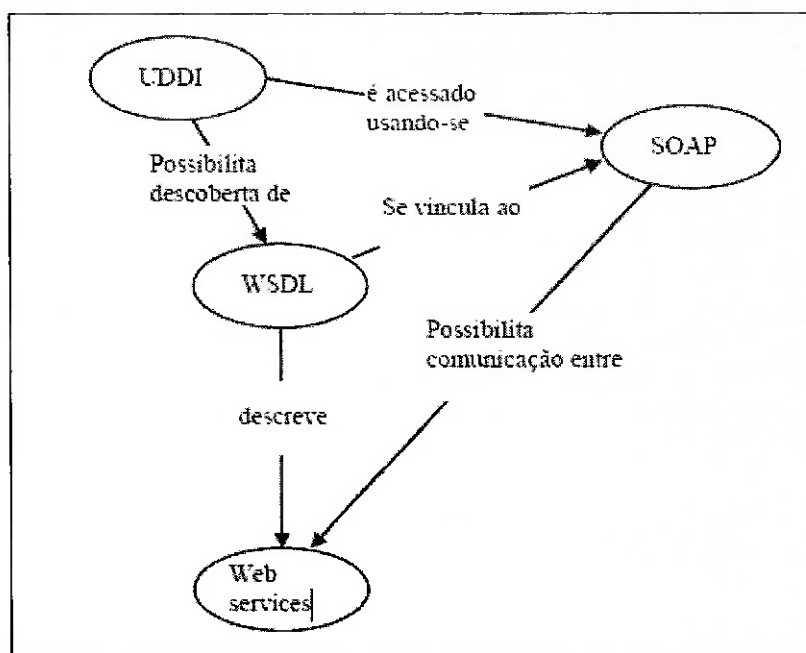


Figura 3 - Relacionamento entre as especificações de serviços web

### 2.3.2. SOAP (Simple Object Access Protocol)

O SOAP é um protocolo para troca de informações estruturadas em um ambiente descentralizado e distribuído. As mensagens SOAP podem ser transportadas por uma variedade de protocolos de rede como HTTP, SMTP, FTP, RMI/IIOP ou protocolos proprietários de troca de mensagens, e são independentes de quaisquer modelos de programação [10].

O framework de troca de mensagens SOAP consiste nos seguintes elementos XML:

- Envelope: elemento raiz do documento XML que representa a mensagem SOAP.
- Cabeçalho (header): elemento opcional, porém se estiver presente precisa ser o primeiro elemento. Contém informações de controle, como autenticação, informação dos intermediários, gerenciamento de transação, etc.



- Corpo (*body*): elemento obrigatório contém a carga da mensagem, ou seja, as informações a serem transportadas.
- Fault: elemento aninhado dentro do elemento *body*, traz informações de erro.

O SOAP estabelece dois estilos de mensagens: documento e RPC. O estilo documento que o corpo simplesmente contém um documento XML cujo formato está acordado tanto pelo remetente quanto pelo receptor. O estilo RPC indica contém uma representação XML de um método de chamada. Nesse caso a infraestrutura necessita das seguintes informações:

*URI (Uniform Resource Identifier)*, nome do método, nomes e valores de parâmetros, assinatura do método (opcional) e dados do header (opcional).

### 2.3.3. WSDL (Web Services Description Language)

É uma linguagem para especificação e implementação de descrições de serviços. Foi criada com uma gramática XML específica, desenvolvida para informar metadados sobre um serviço Web de maneira uniforme para todos os clientes em potencial.

Um serviço Web pode possuir várias portas, sendo capaz de aceitar vários protocolos. Tais protocolos podem ser definidos com a WSDL. Esta fornece a definição de implementação do serviço, que é uma forma padronizada para descrições abstratas das operações e mensagens. Essas descrições são vinculadas a um formato de mensagem e protocolo de rede específica para criar um endpoint que é a definição da implementação do serviço. Sendo assim, a definição abstrata de cada extremidade pode, então, ser vinculada a um protocolo de comunicação concreto. Uma definição de interface do serviço pode ser instanciada e referenciada por várias implementações de serviço.

Em um documento WSDL, os serviços são definidos usando seis elementos principais divididos entre a definição da interface de serviço e a definição da implementação do serviço.

A definição da interface do serviço contém os elementos que forma a porção reusável da descrição:

- Types: fornece as definições dos tipos de dados usados para descrever as mensagens trocadas entre as portas;
- Message: representa uma definição abstrata de dados que estão sendo transmitidos. Uma mensagem consiste em partes lógicas, e cada uma é associada a uma definição de tipo.
- PortType: conjunto de operações abstratas. Cada operação se refere a uma mensagem de entrada e mensagens de saída.
- Binding: especifica um protocolo concreto e especificações de formato de dados para as operações e mensagens definidas por um portType específico.

A definição de implementação do serviço contém os elementos que descrevem como uma interface particular é implementada por um dado provedor de serviço:

- Endpoint: um URL ou endereço de rede a um elemento WSDL: binding de uma definição de interface de serviço.
- Service: usado para agregar um conjunto de portas relacionadas. Um serviço web é modelado conforme esse elemento.

A definição da interface do serviço junto com a definição da implementação do serviço contém informação suficiente para descrever para o requisitante do serviço como invocar e interagir com o serviço Web. Sendo assim, desde que o remetente e o receptor tenham a mesma descrição do serviço (isto é, arquivo WSDL), as implementações por trás dos serviços Web podem ser construídos com diversas tecnologias.

#### 2.3.4. UDDI (Universal Description, Discovery, and Integration)

O UDDI pretende agir como um agente de informação entre consumidores e provedores de serviços. Ele especifica a maneira de armazenar e recuperar as informações sobre serviços, especialmente o nome do provedor e as interfaces técnicas.

Vimos que a definição de WSDL de um serviço informa ao requisitante como invocar e interagir com o serviço. Contudo, o requisitante do serviço poderia precisar de outras informações sobre o *endpoint* do provedor, tais como requisitos de qualidade de serviço, ou taxonomias associadas a um produto. Essas informações não constam das descrições na WSDL, mas podem ser recuperadas do UDDI, que também fornecem um mecanismo para armazenar as descrições dos serviços Web. Portanto o UDDI, não é somente um mecanismo de diretório, ele define também como padrão de estrutura de dados para representar informações de descrição de serviços em XML. Há quatro estruturas de dados fundamentais em uma entrada de UDDI[10]:

- *BusinessEntity*: modela informações da empresa, como nome, informação de contato, tipo de negócio, informações de identificação, e os elementos opcionais para categorização. Contém uma coleção de elementos *BusinessServices*, um para cada serviço Web que a empresa deseja publicar.

- *Business Service*: contém informações do serviço como nome do serviço, o identificador e os elementos opcionais para categorização. Cada *BusinessService* contém uma coleção de elementos *BindingTemplate*.
- *BindingTemplate*: descreve informações de acesso (por exemplo, URL) ou opções de roteamento. Esses elementos também contêm ponteiros para especificações técnicas. Uma especificação técnica é modelada com um *Tmodel* (modelo técnico).
- *Tmodel*: descreve diferentes conceitos de interface, como um tipo de serviço, uma plataforma tecnológica ou uma taxonomia. O *Tmodel* é uma parte importante para a estratégia de busca do UDDI fornecendo metadados sobre especificações adicionais, seu nome, publicador e o URL do seu esquema.

### 2.3.5. 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.

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 de 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 (Business Process Management).

No mundo da integração de sistemas de informação, 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.

As principais características de um ESB são:

- Conectividade: propósito principal é promover a interconexão entre os participantes da arquitetura;
- Heterogeneidade de tecnologia: deve ser capaz de interconectar participantes baseados em diferentes linguagens de programação, diferentes sistemas operacionais diferentes ambientes de execução e também diferentes produtos de middleware e protocolo de comunicação;
- Heterogeneidade de conceitos de comunicação: deve suportar diferentes modos de comunicação. No mínimo para comunicação síncrona e assíncrona;
- Serviços técnicos: também deve fornecer serviços técnicos como logging, segurança e transações;

#### **2.4.ISO Reference Model of Open Distributed Processing. RM-ODP**

O RM-ODP estabelece um estilo e uma base para o projeto e implementação de sistemas, permitindo o uso de componentes, métodos de projeto e representação que já são comuns. O RM-ODP deve:

- Prover um modelo consistente, permitindo que a consistência possa ser mantida entre múltiplos padrões desenvolvidos separadamente;
- Definir as áreas onde o modelo de referência ODP é necessário, o relacionamento entre os padrões ODP e os outros padrões;

- Descrever um conjunto básico de ferramentas a serem usadas na definição do modelo de referência ODP;
- Prover um conjunto consistente de conceitos e terminologias comuns;

O objetivo do RM-ODP é o desenvolvimento de padrões que permitam que os benefícios dos serviços de processamento da informação distribuídos possam ser obtidos em um ambiente de recursos de tecnologia de informação heterogêneos e múltiplos domínios organizacionais.

Ao invés de lidar com toda a complexidade de um sistema distribuído, pode-se ver o sistema de pontos de vista diferentes, cada qual refletindo um conjunto de atributos diferentes [12].

Cada ponto de vista representa um nível de abstração diferente do sistema distribuído original, sem a necessidade de criar um grande modelo que descreva todos estes níveis em detalhes.

Especificando um sistema utilizando diferentes pontos de vista permite-se que um sistema grande e complexo possa ser separado em partes gerenciáveis, cada um focalizando assuntos relevantes para membros diferentes da equipe de desenvolvimento [12].

O RM-ODP possui cinco pontos de vista conforme figura abaixo, são eles: ponto de vista da empresa, ponto de vista da informação, ponto de vista computacional, ponto de vista da engenharia e ponto de vista da tecnologia.

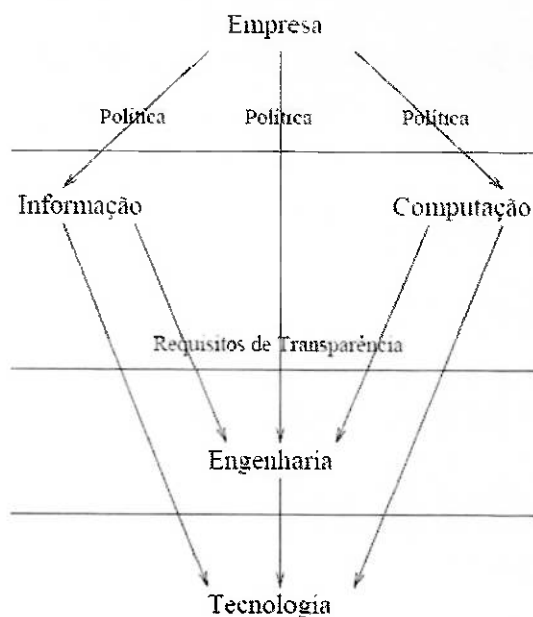


Figura 4 - Pontos de Vista do RM-ODP

#### 2.4.1. Ponto de vista da Empresa

Descreve os sistemas de informação em termos de que o mesmo deve fazer, capturando as necessidades e especificações administrativas e técnicas que guiam e justificam o projeto do sistema.

Este ponto de vista trata dos objetivos, escopos e políticas que governam as atividades de um sistema especificado, na organização da qual ele faz parte. Define as políticas sociais e organizacionais em termos de objetivos passivos e ativos, produzindo uma especificação que expressa às responsabilidades de todos os participantes do sistema distribuído.

É uma visão puramente funcional da arquitetura, definindo os objetivos gerais e o escopo do projeto, as entidades dentro do domínio do negócio e as regras e políticas que governam a interação entre essas várias entidades

### **2.4.2. Ponto de vista da Informação**

Descreve o sistema de informações em termos de estruturas de informações, fluxo de informações e restrições relacionadas com a manipulação das mesmas.

Trata com vários tipos de informação manipulada pelo sistema e restrições no uso e interpretação desta informação. Este ponto de vista captura a informação trocada entre as aplicações distribuídas em termos de metadados, estados e estrutura de um objeto.

Tem, portanto como foco, a semântica das informações, as classes de definição de objetos de informação, esquemas invariantes, estático e dinâmico e as regras associadas.

A informação, neste ponto de vista, pode ser estática, por exemplo, uma particular instância de um objeto: dinâmica, descrevendo possíveis alterações de estado e/ou estrutura; ou invariante, descrevendo propriedades, de um objeto, que não devem ser alteradas.

### **2.4.3. Ponto de vista da Computação**

O sistema de informações, neste ponto de vista, é descrito em termos de operações e características computacionais do processo de mudança das informações, preocupando-se com a distribuição dos objetos dentro do sistema, mas não com sua infraestrutura de comunicação.



Trata da decomposição funcional do sistema em um conjunto de objetos que interagem através de suas interfaces, habilitando a distribuição do sistema. Descreve a funcionalidade das aplicações ODP, de maneira transparente em relação à distribuição das mesmas, em termos de uma ou mais interfaces que os objetos oferecem para isolar seus dados privados e seus métodos (processo). A interface é, dentro da perspectiva de interoperabilidade, a chave para se entender os componentes.

#### **2.4.4. Ponto de vista da Engenharia**

Descreve o sistema de informações em termos de recursos de engenharia para suportar a natureza distribuída do processamento, preocupando-se com a maneira pela qual a interação entre os objetos ocorre e os recursos necessários para que isso se torne possível.

Trata da infraestrutura requerida para suportar a distribuição do sistema, considerando-se a heterogeneidade de hardware e software. Descreve os elementos das aplicações distribuídas que suportam a distribuição de software e a infraestrutura geral.

Define o conjunto de objetos que suportam os canais de comunicação, assim como a criação das estruturas de engenharia, tais como topologia de rede, padrões de comunicação e a infraestrutura de segurança.

#### **2.4.5. Ponto de vista da Tecnologia**

O sistema de informações é descrito, por este ponto de vista, em termos dos componentes que compõem o sistema em construção.

Trata da escolha da tecnologia para suportar a distribuição do sistema. Considera os elementos específicos de hardware e software da aplicação distribuída a ser implementada.

O RM-ODP não fornece muita orientação ou controle sobre esse ponto de vista, uma vez que tenta se manter, o mais possível, tecnologicamente independente.

#### **2.4.6. Considerações Finais**

Apesar do RM-ODP apresentar uma descrição essencial dos conteúdos de cada nível, que deve ser considerada e assim facilitar a comparação entre sistemas alternativos, aderir ao mesmo não resolverá todas as questões práticas envolvidas no projeto e implementação de sistemas de software. O RM-ODP não determina como o sistema de informações deve ser projetado e implementado, bem como não define quais as ferramentas ou técnicas que devem ser utilizadas para a representação de cada nível. Tal tarefa é deixada a critério do usuário. Por esta razão, além do RM-ODP, torna-se necessária a aplicação de métodos formais que possam guiar o projeto e implementação efetiva e consistente de sistemas de software através de cada um dos seus níveis.

### **3. DEFINIÇÃO DO PROBLEMA**

Neste capítulo, serão utilizadas algumas das atividades do Departamento de Informática da reitoria da USP que serviu como objeto de estudo. Será mostrado como o departamento encontra-se atualmente em relação ao desenvolvimento dos sistemas corporativos, como é tratada a integração de dados para outras unidades da USP e como seria a situação atual utilizando a metodologia proposta. A proposta do trabalho é a adoção da utilização da arquitetura orientada a serviços, para se ter uma maior reusabilidade de código, uma maior agilidade no desenvolvimento do projeto e a diminuição de replicação de dados.

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

#### **3.1. Integração entre os sistemas**

O Departamento de Informática da reitoria da USP desde sua criação é responsável pelo desenvolvimento dos sistemas corporativos de toda universidade. Atualmente está dividido em cinco áreas de projeto: institucional, acadêmica, administrativa, recursos humanos e infraestrutura. Cada área é responsável por um grupo de sistemas corporativos que por muitas vezes acabam tendo que fazer integração entre eles.

A Universidade é composta também por Unidades de negócio que são os centros de informática distribuídos nos campi e por sua vez acabam desenvolvendo sites ou aplicações locais para atender situações pontuais ou escopo localizado.

Os sistemas corporativos acessam diretamente os bancos de dados centrais, que seriam os vários bancos interconectados através de replicação on-line e dividido por áreas de negócio.

O cadastramento dessas informações é realizado na base central, porém como descrito acima existem sistemas locais desenvolvidos pelas áreas de tecnologia de cada unidade de negócio.

As unidades de negócio ainda podem possuir bases locais, com dados replicados, para alimentar os sistemas locais e específicos.

As informações enviadas para as aplicações desenvolvidas pelas unidades de negócio 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 de dados consolidado.

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 área de negócio.

O cenário atual da USP apresenta o seguinte panorama:

- Segurança: Os sistemas corporativos são compostos por componente de autenticação para sua identificação, e quais as permissões que possui de acordo com os níveis de segurança cadastrados.

*Login*: Um sistema de controle de acesso para evitar que os usuários não autorizados tenham acesso ao sistema.

*Identificação de perfil*: Mecanismo de controle de acesso por nível de segurança. Alguns sistemas corporativos já tem troca de perfil implementada onde um usuário pode ter mais de um nível de segurança cadastrado.

- Escalabilidade: Atualmente existe uma estabilidade em número de usuários, porém com o crescimento dos sistemas corporativos com novos projetos e novas funcionalidades, devido a este fato o número está aumentando significativamente.

*Quantidade de acesso ao banco de dados*: Os acessos são sazonais. Existem picos de acesso à base central em época de matrículas da graduação, inscrição de trabalho na iniciação científica, época de matrículas na pós-graduação e o acesso a holerites no início de cada mês.

- Componentização: Existe uma iniciativa de reutilização de código que se utiliza de algumas *stored procedures* e modelo de classes, mas ainda são muito poucos os casos que fazem uso dessa prática.
- Manutenção da Informação: Todas as transações são realizadas no banco de dados central independente da unidade de negócio, sendo replicadas quando necessárias para as outras unidades. Isso tem se tornado cada vez mais oneroso e custoso, devido ao grande aumento do número de tabelas que por sua vez acabam tendo de ser replicadas para suprir as necessidades de outras áreas de negócio.

Os sistemas corporativos são integrados através de um modelo de dados compartilhado e suportados através de replicação de tabelas para sua alimentação.

Eventualmente alguns cadastros locais, oriundo dos sistemas locais e de apoio podem ser carregados via batch (processo off-line) na base de dados central.

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 pelo conteúdo da sua informação e integridade do dado.

Porém, os dados que são comuns aos vários sistemas corporativos (Exemplo: Tabelas PESSOA, VINCULOPESSOAUSP, UNIDADE, SETOR, etc...) são replicados de forma on-line para as outras bases de dados, garantindo assim a integração das informações.

Existem também restrições de acesso 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, muito deles simultâneos. Esta entidade pode assumir o contexto de Aluno de Graduação, Aluno de Pós Graduação, Aluno de Pós Doutorado, Servidor (Docente ou Funcionário), entre outros. No sistema de recursos humanos essa entidade tem acesso de cadastro, mas nas outras áreas de negócio essa entidade somente pode ser consultada por isso em algumas entidades temos restrições de acesso em dados "comuns".

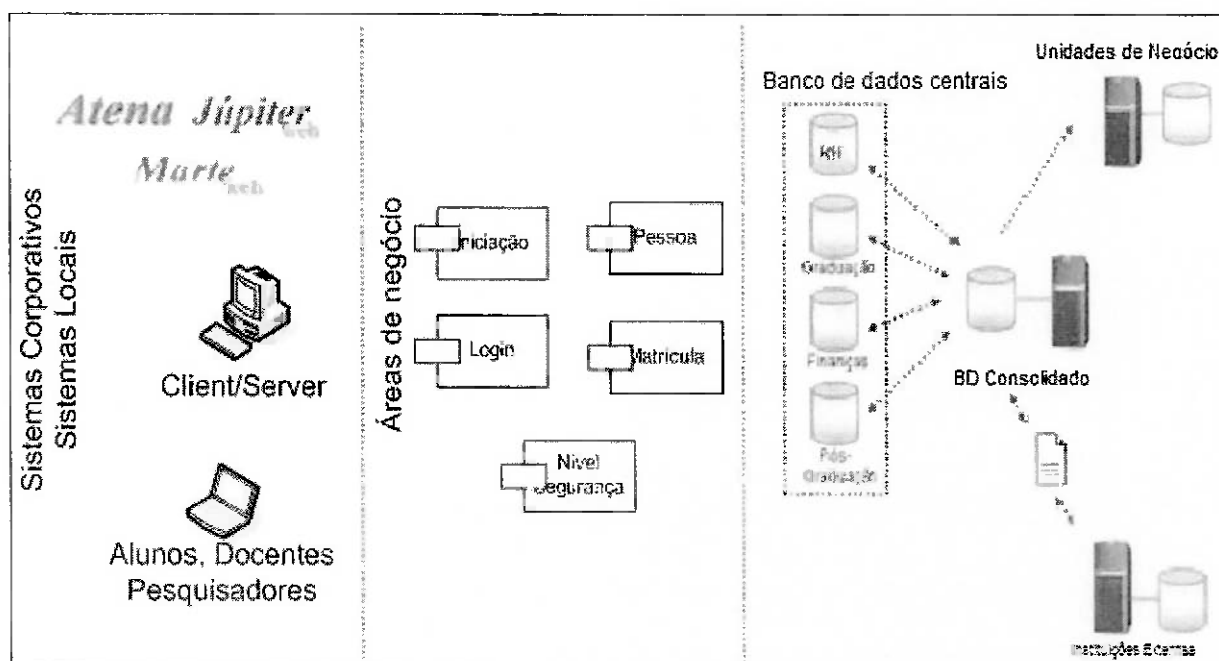


Figura 5 - Cenário Atual DI USP

A figura 5 ilustra como é o desenho tecnológico do departamento de informática mostrando os sistemas corporativos, às áreas de negócio correspondente e o modelo de banco de dados que está montado atualmente. Os sistemas corporativos e mesmo os sistemas locais muitas vezes não desenvolvidos pelo departamento, estão ligados a uma área de negócio instituída no departamento para que possa ser feita a devida correlação com o banco de dados corporativo que por sua vez está ligado ao negócio correspondente de cada área.

### **3.2. Problemas encontrados**

Com uma demanda cada vez maior, o departamento de informática deixou de ser responsável pelo desenvolvimento e manutenção de todos os sistemas corporativos que a Universidade de São Paulo possui atualmente.

Desta forma novas frentes de desenvolvimento de sistemas tiveram de ser abertas para que parceiros e unidades de negócio (centros de informática espalhados pelos campi da USP) começassem a desenvolver aplicações, para que se possa integrar ao corporativo. Com isso surgiu um cenário até então desconhecido: o modelo de dados da USP possui particularidades de sistemas, sendo necessário repassar parte da estrutura de modelagem e detalhes de negócio, de forma que as empresas ou unidades de negócio possam desenvolver seus aplicativos e tratar corretamente os dados. Isso causa dependência, como por exemplo, quando são necessárias mudanças no modelo, ou até mesmo no caso de mudança de servidores de banco de dados (troca de hardware e indisponibilidade de um servidor). O mesmo pode ocorrer com os sistemas locais de sites replicados.

Além disso, para receber os dados da replicação, as Unidades devem disponibilizar uma infraestrutura 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 essas unidades recebem a replicação.

No âmbito de informática central que atualmente é conduzida pelo departamento de informática, ocorrem alguns cenários que vem desde o início dos sistemas corporativos e que a proposta de mudança possa modificar esse modelo atual:

- Retrabalho – com as regras de negócio embutidas nos códigos dos sistemas ou espalhadas na forma de *stored procedure* nos bancos de dados, sempre que há uma manutenção no modelo de dados é necessário verificar todas as várias fontes diferentes;
- Reuso – devido às especificidades de cada área de negócio no tratamento do mesmo conceito dos dados, o reuso torna-se praticamente inexistente;
- 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 de dados;

Resumindo, os principais itens críticos agravados pelo tempo e pelo avanço da tecnologia são:

|           | <b>Cenário apresentado</b>                                                                              |
|-----------|---------------------------------------------------------------------------------------------------------|
| <b>P1</b> | Replicação de dados;                                                                                    |
| <b>P2</b> | Integração de sistemas corporativos desenvolvidos por cada área de negócio do departamento;             |
| <b>P3</b> | Custo da infra-estrutura;                                                                               |
| <b>P4</b> | Retrabalho;                                                                                             |
| <b>P5</b> | Complexidade da lógica de negócio;                                                                      |
| <b>P6</b> | Integração de dados de órgãos externos sendo eles, órgãos governamentais ou outras unidades de negócio; |

**Tabela 1 – Cenário atual no Departamento de Informática da USP**



### 3.3. Definição de escopo do exemplo prático

Neste exemplo prático utilizaremos dois cenários elencados na tabela 1. Para facilitar o entendimento e a proposta de uma nova arquitetura iremos estudar os seguintes cenários: *P1 – Replicação de dados* e *P2 – Integração de sistemas*.

Os cenários P1 e P2 serão exemplificados através do processo de Pagamento de Gratificação de Convênio que envolve dois sistemas: o sistema de Recursos Humanos responsável pelo pagamento do convênio e o sistema de execução orçamentária responsável pelo gerenciamento das informações referentes ao convênio (o cadastro da participação, cadastro de pagamento, autorização do convênio, e todas as atividades envolvidas diretamente com o cadastro e liberação dos convênios).

A explicação desse processo está localizada no item 4.2, pois é utilizado como um problema a ser resolvido com a metodologia proposta utilizando a arquitetura orientada a serviços.

Para um melhor entendimento a tabela 2 mostra o processo envolvido e o(s) problema(s) que esse processo esta relacionado, fazendo uma mescla com a tabela 1.

|    | <b>Processo envolvido</b>                         | <b>Problemas Envolvidos</b>                             |
|----|---------------------------------------------------|---------------------------------------------------------|
| C1 | Processo de Pagamento de Gratificação de Convênio | P1 – Replicação de dados<br>P2 – Integração de sistemas |

**Tabela 2 - Tabela exemplo processo x problema a ser resolvido**

## **4. PROPOSTA DE INTEGRAÇÃO DE SISTEMAS UTILIZANDO SOA**

O objetivo deste capítulo é apresentar a proposta de integração de sistemas utilizando a arquitetura orientada a serviços, baseada no conceito de visões existentes na especificação do RM-ODP.

### **4.1. Metodologia adotada para integração**

A idéia de implementar essa proposta da arquitetura orientada a serviços é utilizar como apoio a linguagem UML e também o BPMN, para representar algumas das visões do RM-ODP.

A utilização das visões está ligada à necessidade de fatiar a especificação de sistemas distribuídos em pedaços gerenciáveis que podem ser relevantes aos participantes do desenvolvimento do sistema.

Essa abordagem permite a implementação por etapas, o que facilita a comunicação aos usuários do sistema como será o planejamento estratégico da implementação do processo.

O valor do SOA é que ele oferece um paradigma escalável único para organizar grandes sistemas em rede que requerem interoperabilidade para realizar o valor inerente aos componentes individuais. Certamente, o SOA é escalável por que ele faz a menor suposição possível sobre a rede e também minimiza qualquer suposição de confiança que são frequentemente feitas em sistemas de escala menor. Um arquiteto usando os princípios do SOA é bem equipado para desenvolver sistemas que são, escaláveis, evolutivos e gerenciáveis. Pode ser fácil decidir como integrar as funcionalidades através dos limites proprietários.

Com base no livro de (Usländer, T. et al 2010) e algumas teses de dissertação de mestrado tais como, (Fugita et al 2009; Santana et al 2009) que serviram de referência para esse estudo. Sendo assim, através dessas referencias e depois de alguns estudos, as etapas importantes que se pode concluir para esse proposta de uma nova metodologia são:

- Análise de Processos;
- Estrutura de informação para codificação dos serviços;
- Interação dos serviços;
- Tecnologia Utilizada;
- Identificação da Comunicação;

A tabela 3 relaciona as visões do RM-ODP, as etapas para SOA, quais técnicas UML e o mapeamento de processo.

| <b>Visão<br/>RM-ODP</b> | <b>Atividade para SOA</b>                                                                                                                           | <b>Artefatos</b>                                                                   |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| Empresa                 | <u>Análise de Processos</u> – Reflete a análise e as necessidades do utilizador. Inclui normas que regem os atores, grupos de atores e seus papéis. | Casos de Uso (Use Case) e mapeamento de processos                                  |
| Informação              | <u>Estrutura de informação para codificação dos serviços</u> - Especifica a abordagem da modelagem de todas as categorias de informações.           | Diagrama de Classes e Diagrama de Componentes                                      |
| Computação              | <u>Interação dos serviços</u> – Especifica a interação e apoio para a “coreografia” dos serviços.                                                   | Diagrama de Seqüência                                                              |
| Tecnologia              | <u>Tecnologia Utilizada</u> - Especifica as opções tecnológicas, suas características e questões operacionais.                                      | Ferramentas ou tecnologias necessárias para criação do serviço                     |
| Engenharia              | <u>Identificação da comunicação</u> - Especifica o mapeamento dos serviços e como será o funcionamento em rede através do formato de três camadas.  | Como os serviços vão funcionar de acordo com a infraestrutura existente na empresa |

Tabela 3 - Tabela comparativa visões RM-ODP X atividades para SOA

O uso desta metodologia pressupõe uma análise “*top down*”, que parte do negócio em operação, detalhando sucessivamente as visões até como toda essa arquitetura será realizada.

A figura abaixo ilustra as etapas da metodologia a serem seguidas para a implementação dos serviços:

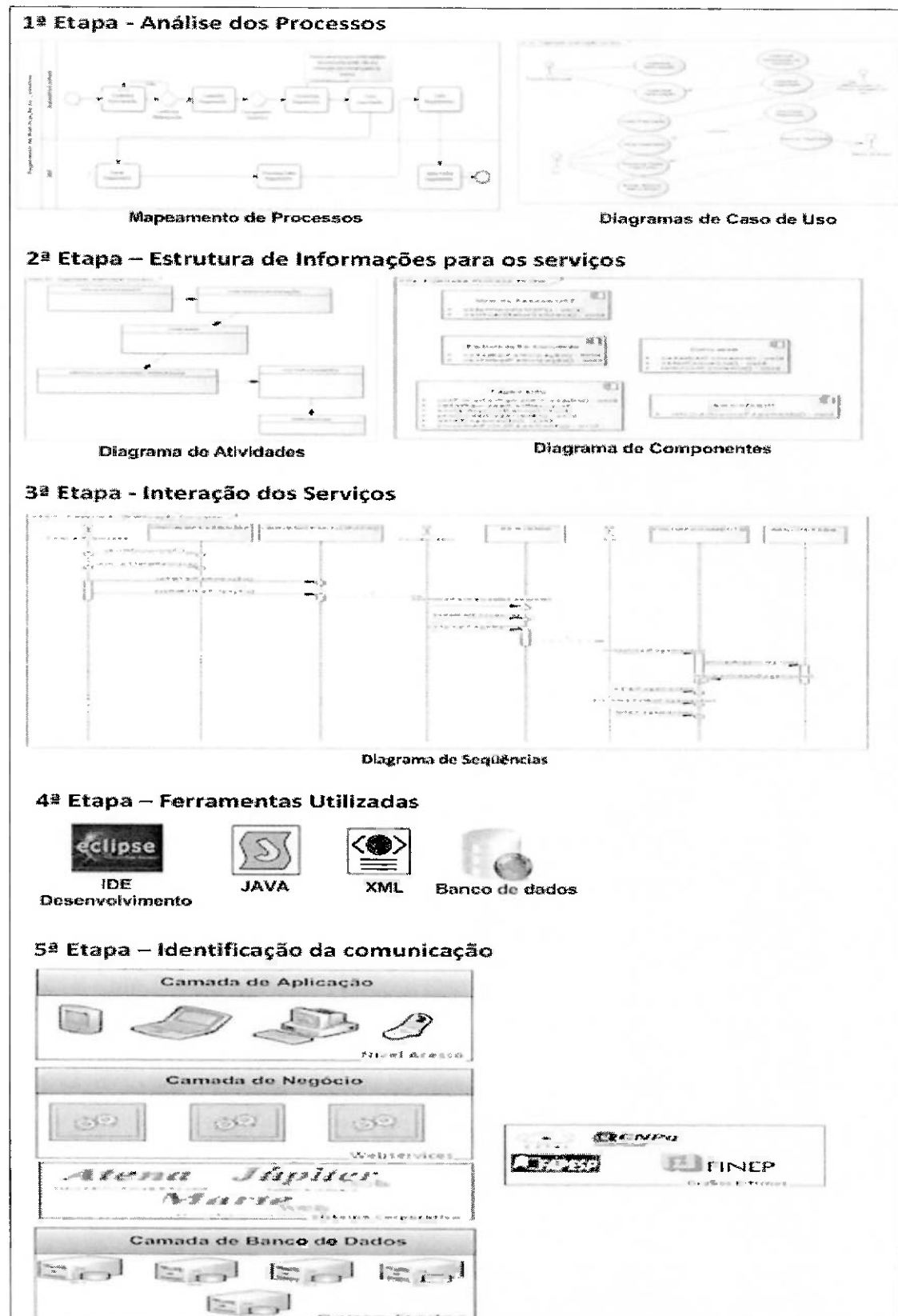


Figura 6 - Etapas para a implementação de Serviços

#### **4.1.1. Análise de Processos**

A primeira etapa consiste em apresentar através de uma visão do ponto de vista da empresa, as operações que são envolvidas no processo. O objetivo é mostrar quais são os sistemas corporativos que estão envolvidos, os usuários envolvidos, as políticas de acesso e as atividades do processo.

Para uma melhor compreensão dessas entidades, atores e interações, utilizaremos duas formas para representar essa etapa: mapeamento de processos e diagrama de caso de uso da UML.

A primeira representação é o mapeamento dos processos, pois permite documentar e identificar todos os passos e decisões de processos. A análise profunda deste conjunto, com base no mapeamento prévio, possibilita benefícios significativos, entre eles: eliminação de tarefas não necessárias, melhoria na comunicação, melhor visibilidade nos papéis responsabilidades e uma forma de visualizar quais processos dos sistemas corporativos estão envolvidos, as operações que serão explicitadas, o que cada sistema tem como responsabilidade e a sequência de execução do processo.

O mapeamento de processos também é importante, por que permite através de uma linguagem simbólica, o entendimento do cliente de uma maneira mais fácil e ilustrada, de como será a integração dos sistemas de informação.

A segunda representação são os diagramas de caso de uso, pois neles estão os atores responsáveis (usuários envolvidos) pela interação com o sistema e os casos de uso correspondentes ao processo.

Embora os casos de uso sejam parte importante na definição da especificação de um projeto, eles tentam capturar o que os usuários querem do sistema, mas não especificam como o sistema deve ser. Isso será definido nas próximas etapas.

A idéia de utilização dos casos de uso é mostrar as políticas de acesso que cada usuário possui em relação ao processo que está sendo construído, e com isso podemos decidir quais são as atividades que cada nível de acesso deverá executar.

#### **4.1.2. Estrutura de informação para codificação dos serviços**

Esta etapa esta relacionada diretamente com o modelo de informação a ser utilizado pelo processo. O nível da informação nesta etapa deve capturar uma representação o mais próximo possível da estrutura de informações.

O objetivo é apresentar uma estrutura mais detalhada do sistema, permitindo assim um maior refinamento na definição dos atributos e operações do processo.

As estruturas definidas para a representação dos modelos de informações estarão primariamente relacionadas com as informações a serem armazenadas na base de dados.

Os Diagramas de Classes permitem uma representação da estrutura interna de cada objeto, bem como os relacionamentos entre os objetos, fornecendo uma consistente representação da estrutura.

Através dos diagramas de classes, casos de uso e o mapeamento de processo, consegue-se obter subsídios importantes para nos auxiliar a identificar os serviços que serão necessários para a execução do processo bem como seus parâmetros de entrada e saída.

É importante dizer que não existe uma regra definida para identificação dos serviços, e nem todo caso de uso, ou atividade do processo necessariamente será um serviço. É válido explicarmos que isso vai além das etapas que estamos definindo como fonte de auxílio. O conhecimento do analista do seu ambiente de sistemas corporativos é que permite analisar quais atividades podem vir a se transformar em serviços e quais atividades apenas serão métodos do sistema.

Outro diagrama que utilizaremos nessa etapa é o Diagrama de componentes, pois é uma forma documentar os serviços agrupados de acordo com o negócio correspondente. Por exemplo, em “Pagamento” tudo que for referente a esse negócio deve ficar descrito nesse pacote, e podemos ter futuramente através do uso desse diagrama um grande facilitador para a criação de bibliotecas de serviços, nesse caso a UDDI.

Outro facilitador desse diagrama é identificar de forma rápida um serviço que possa ser reutilizado em outro processo e dessa maneira tornar o processo de desenvolvimento mais dinâmico e rápido. Além disso, se for preciso adequar esse serviço, a mudança é feita em um único ponto.

#### **4.1.3. Interação dos serviços**

Com os serviços já identificados, é preciso definir qual a seqüência de execução desses serviços para completar o processo.

Esta etapa está relacionada com os processos de mudança dos dados nos modelos de informações e captura os aspectos dinâmicos do sistema em desenvolvimento, ou seja, a representação das operações responsáveis por tais mudanças.

A proposta de utilizar diagramas de seqüência neste passo é fornecer uma representação de como os serviços (ou operações) podem interagir, para realizar cada uma das funcionalidades específicas do sistema.



Utilizando os diagramas de sequência, podemos mostrar de uma forma mais fácil como os serviços devem ser acessados e a sequência de execução dos serviços.

#### **4.1.4. Tecnologia Utilizada**

Esta etapa define o desenvolvimento dos serviços especificados nas etapas anteriores.

Aqui explicitamos as definições de quais ferramentas são utilizadas para o desenvolvimento dos serviços, quais os bancos de dados, os padrões de tecnologia que será adotado e qualquer outro item ligado à parte tecnológica.

#### **4.1.5. Identificação da Comunicação**

Esta etapa define como será a integração dos serviços criados com o parque tecnológico existente na empresa.

É importante nessa etapa definir com o pessoal responsável pela infraestrutura da empresa, como serão implementadas as políticas de acesso a esses serviços definidos nas etapas anteriores, a melhor maneira de publicar um serviço (HTTP, HTTPS ou FTP), como os órgãos externos vão enxergar os serviços sem que se precise expor detalhes do ambiente interno da empresa.

## 4.2. Aplicação da metodologia proposta

Para demonstrar o funcionamento dessa metodologia será utilizado um caso real do departamento de informática da reitoria da USP. Para tanto, primeiramente será detalhado o problema e depois aplicar as etapas para implementação da arquitetura orientada a serviços.

Inicialmente será descrito o processo usando o vocabulário empregado tanto na área de negócio quanto na área de TI em estudo.

O "Pagamento de Gratificação do Convênio" é um processo que se inicia no cadastramento de um servidor-docente ativo com contrato de trabalho RTC (Regime de Turno Completo) ou RTP (Regime de Turno Parcial). No caso de docentes com contratos de trabalho no formato de RDIDP (Regime de Dedicação Integral à Docência e Pesquisa), será checado em uma tabela de provimentos a validação de seu credenciamento. A data de início da inclusão dos participantes não poderá ser menor que a data início do convênio assim como a data fim não poderá ser maior que a data fim de vigência do convênio. O cadastro desse participante a gratificação de um convênio é realizado por uma "PESSOA AUTORIZADA" (Ator representado no diagrama de caso de uso na figura a seguir).

Com o cadastramento feito, uma análise manual fora sistema é realizada para autorizar ou não a participação do convênio. Uma vez esse participante autorizado, ele é incluído em um convênio e então passa para a fase seguinte que seria o pagamento da gratificação do convênio.

No sistema corporativo Marteweb, no item de "Pagamentos" é selecionado um convênio e dessa forma é listado todos os docentes participantes. Para a inclusão de valores aos participantes é feita algumas verificações pelo sistema:

- O docente deve possuir um registro válido de credenciamento no objeto Provimento;

- O valor informado não pode ultrapassar os vencimentos da categoria no caso de um docente na forma de trabalho RDIDP;
- Caso o docente possua um registro de folha para a mesma competência de outro convênio, o valor somado não pode ultrapassar ao novo valor no vencimento na forma de trabalho RDIDP;
- O convênio deve possuir saldo suficiente para cobrir o valor informado;
- Todo valor informado em uma folha para cada docente deve ser subtraído do montante destinado no convênio para o pagamento dessa gratificação;
- O sistema habilita automaticamente folha específica da competência relativa ao dia do cadastro, observando o dia do cadastro, cronograma de processamento e consolidação das folhas de pagamento, sem necessidade de nenhuma intervenção por parte do usuário;

Após todas essas verificações, o pagamento do valor referente ao convênio é autorizado pelo COORDENADOR, (ou por alguma pessoa designada que autoriza em nome dele).

A tabela contendo os dados desse convênio juntamente com os dados do pagamento, é replicada para o banco de dados que pertence ao sistema de recursos humanos.

Então o trâmite do pagamento começa a ser executado de forma automática, onde teremos os processos de geração de pagamento, listar o pagamento, processamento da folha e o envio do arquivo com a conta e o valor do convênio para o Banco do Brasil através de um arquivo XML.

O Banco do Brasil é considerado um agente externo, mas é quem concretiza o pagamento. É feita uma análise pelo banco (se possui conta no banco, verificação de conta ativa), e sendo todas as variáveis aceitas, é emitida uma ordem de pagamento. Esta ordem é recepcionada pelo sistema de recursos humanos que gera o processamento de folha de pagamento e paga efetivamente o servidor.

Após esse pagamento a tabela é replicada novamente para o sistema administrativo que executa os seguintes processos: lista os convênios que já sofreram algum tipo de pagamento, os docentes que já estão recebendo algum tipo de convênio e verifica se a folha de pagamento foi processada e paga, para só depois disso informar ao servidor contemplado o pagamento.

### 4.2.1. Análise de Processos:

Primeiro passo mapeamento dos processos, e os sistemas corporativos envolvidos nesse negócio.

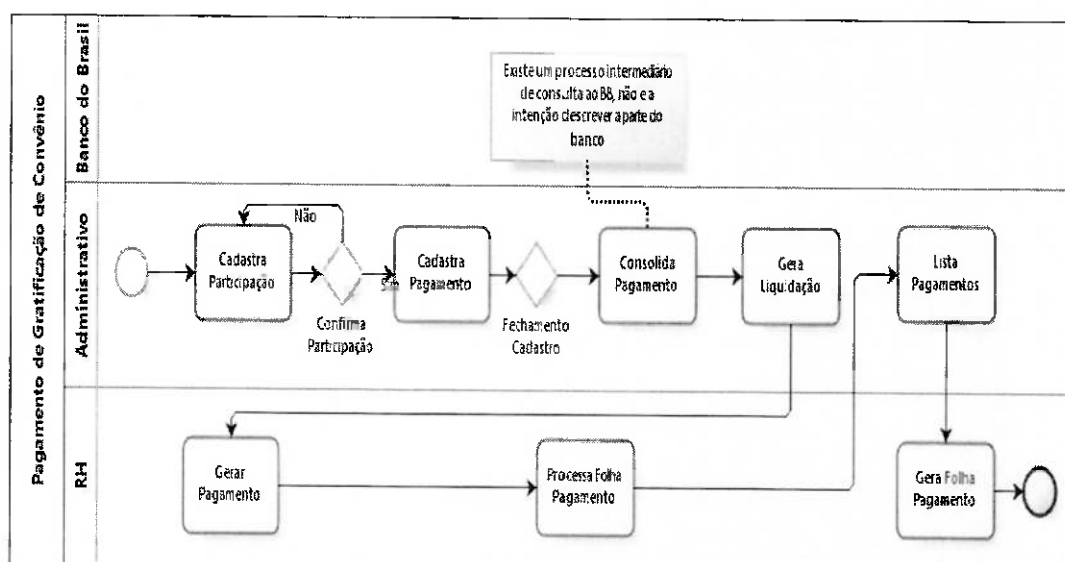


Figura 7 - Modelo de processo do Pagamento da Gratificação de Convênio

A seguir, é elaborado o diagrama de caso de uso, com os atores e suas responsabilidades.



- Banco do Brasil: Um agente externo que na verdade que auxilia no momento do pagamento efetivamente liberando o dinheiro na conta do participante ao convênio. O Banco também verifica se o mesmo possui conta e se a sua situação esta regularizada;

Esta etapa de análise de processos nos dá uma visão do ponto de vista da empresa, mostrando de uma forma bem detalhada, quais são os processos envolvidos e quais sistemas corporativos estão participando do processo de negócio, assim como também nos mostra as regras e políticas de acesso que cada ator poderá ter em relação ao negócio.

#### 4.2.2. Identificação dos serviços

Criação de um diagrama de classes. Nesse diagrama não colocaremos as operações, que estão no diagrama de componentes, e nem os atributos. Não é relevante nesse processo, mas nada impede que isso seja feito.

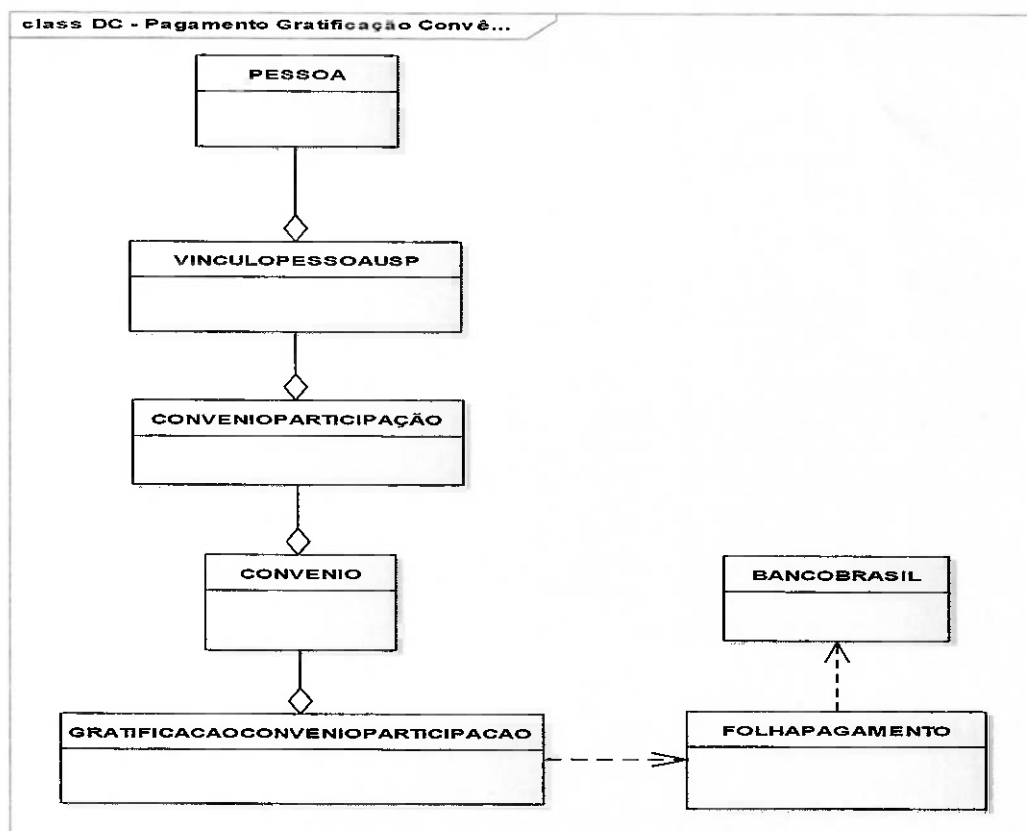


Figura 9 - Diagrama de Classes Pagamento Gratificação do Convênio

Na figura 9 temos todas as classes necessárias para a realização do processo de negócio. Nesse diagrama está incluída também a classe da entidade externa que nesse processo é representado pelo Banco do Brasil.



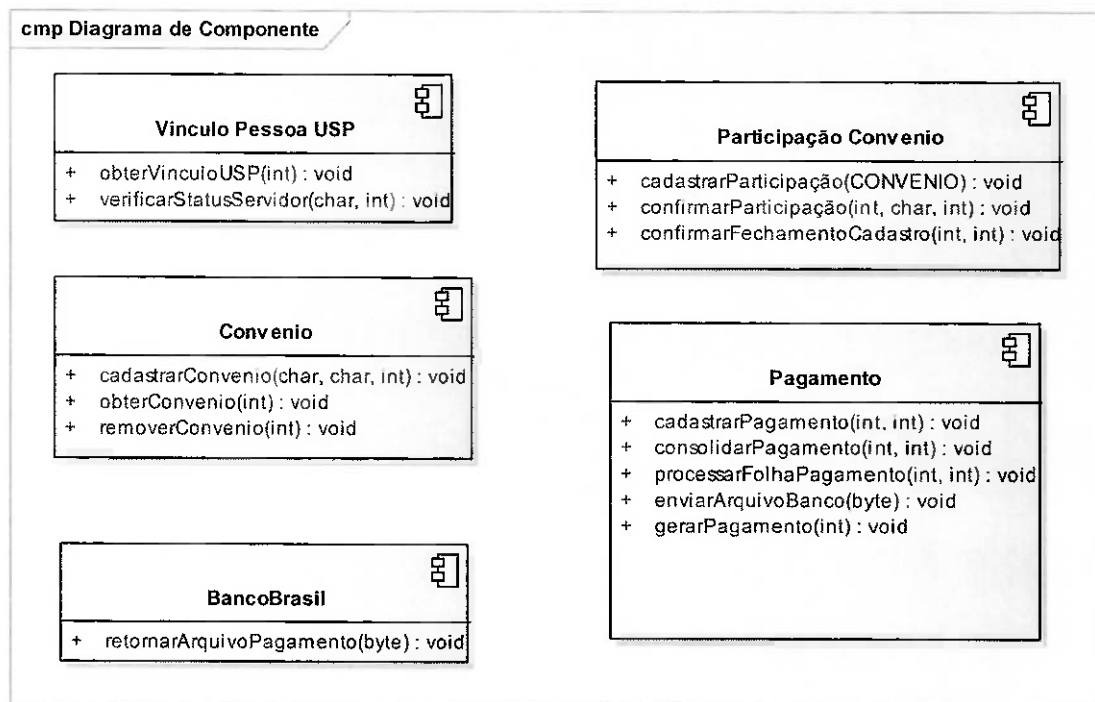


Figura 10 - Diagrama de componentes

No diagrama de componentes já temos os pacotes definidos e os serviços pertencentes a cada pacote, com os parâmetros de entrada e de saída.

Nesta etapa através de diagramas de classe e diagramas de componentes, temos a estrutura de informação necessária para auxiliar o projetista, elencando quais serviços são necessários para a execução do processo de negócio.

### 4.2.3. Interação dos serviços

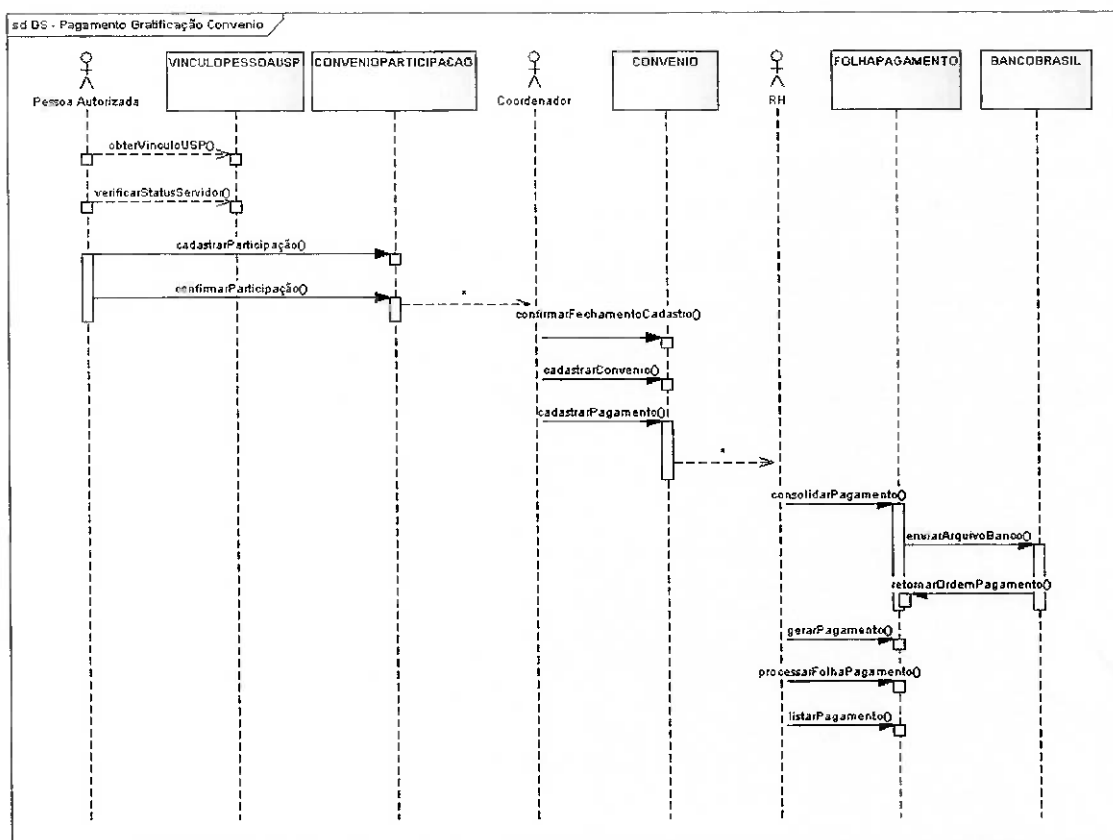


Figura 11 - Diagrama de Sequência Pagamento da Gratificação do Convênio

Descrevendo a figura acima temos a seguinte seqüência:

- obterVinculoUSP;
- verificarStatusServidor;
- cadastrarParticipação;
- confirmarParticipação;
- confirmarFechamentoCadastro;
- cadastrarConvênio;
- cadastrarPagamento;
- consolidarPagamento;
- enviarArquivoBanco;
- retornarOrdemPagamento;
- gerarPagamento;

- processarFolhaPagamento;
- listarPagamento;

Na etapa de interação de serviços a preocupação é ilustrar as realizações dos serviços definidos na etapa anterior, isto é, mostrar como os serviços interagem para executar o comportamento total ou parcial do processo de negócio.

Esse tipo diagrama é importante para o designer do projeto, pois esclarecem os papéis em um fluxo mostrando a sequência explícita de serviços, e também sua linha temporal.

#### 4.2.4.Tecnologia Utilizada:

Como estamos nos referindo a implementação de uma nova arquitetura orientada a serviços, as ferramentas necessárias para o desenvolvimento desses serviços são:

- Banco de dados SYBASE: Atualmente é o banco de dados, no departamento de informática da reitoria da USP para os sistemas corporativos;
- Bizagi BPMN: Software que atualmente possui licença livre para fazer os mapeamentos de processos dos sistemas corporativos. Pode-se utilizar qualquer aplicativo que faça um mapeamento de processo adequado para o entendimento da integração;
- JAVA: Linguagem de programação atualmente empregada pelo departamento de informática da Reitoria da USP no desenvolvimento de sistemas corporativos web;
- JSP e CSS: Padrão para construção de interfaces, para acesso visual aos serviços criados (camada de visualização);
- Servlet 2.4 ou superior: Servlets como camada de negócio, utilizando atualmente o modelo três camadas, onde os servlets executam e controlam o acesso aos serviços;
- ECLIPSE: *IDE* utilizado para o desenvolvimento de sistemas e também para a geração dos serviços. Atualmente o departamento utiliza a versão Galileo 3.5
- AXIS2: Framework que auxilia na criação de serviços.O eclipse utiliza esse framework e facilita bastante a criação dos serviços sem precisarmos criar classes de forma manual.

- Plugin WTP: Plugin utilizado no Eclipse onde podemos ter um ambiente de desenvolvimento Web e JEE já configurado. O WTP é um projeto que faz parte do Eclipse.Org e tem por finalidade prover um ambiente de desenvolvimento padrão para web sob a plataforma Eclipse, sendo formado por um conjunto de APIs para JEE e aplicações web, além de ferramentas para suporte de deploy e testes de aplicações.

Nesta etapa são definidas quais as ferramentas serão utilizadas para o uso da arquitetura orientada a serviços. Como essa metodologia está sendo aplicada em cima de um estudo de caso do departamento de informática da reitoria da USP, foram utilizadas algumas ferramentas já em uso pelo departamento, e algumas outras que são necessárias para executar esse tipo de arquitetura. Isso não significa que são essas ferramentas necessárias e obrigatórias para qualquer que seja a organização, isso vai variar muito de acordo com a particularidade e o conhecimento das pessoas envolvidas no projeto.

#### 4.2.5. Identificação da Comunicação

Neste ponto é descrito como será a infraestrutura para suportar a interação entre os serviços definidos anteriormente, como será a sincronização entre os serviços e o acesso pelo usuário, conforme mostra a figura 12.



Figura 12 - Visão da Engenharia

➤ **Camada de Aplicação:**

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

➤ **Camada de Negócio:**

- Webservices: Camada que fornece os processos de negócio, softwares de integração, padrões de integração, orquestração e disposição dos serviços.
- Sistemas Corporativos: Esta camada representa os sistemas desenvolvidos pelo Departamento de Informática ou parceiros de desenvolvimento como as unidades de ensino ou centros de informática que façam parte também do parque tecnológico corporativo.
- Órgãos Externos: Representa os agentes externos que possuam algum tipo de integração via WebService dos quais precisamos recuperar alguma informação, neste caso, por exemplo, o agente externo seria o Banco do Brasil.

➤ **Camada de Dados:**

- Banco de dados: Fornece acesso às informações (banco de dados, arquivos texto, etc.)

O modelo três camadas oferece vantagens significativas no desenvolvimento de aplicações, através da separação das camadas, possibilitando implementar com maior facilidade e clareza questões programáticas importantes como a persistência de dados, controle de negócio e fluxo de visualização. Esta separação permite que os aplicativos possam ser mais facilmente modificados e expandidos para atender a novas exigências, e possibilita também ter várias interfaces gráficas referente a um só modelo sem interferir com a lógica.

## 5. CONCLUSÃO

No trabalho apresentado foram mostradas quais as deficiências que enfrentamos atualmente no departamento de informática da reitoria da USP, não por uma má administração, mas sim pelo avanço tecnológico cada vez mais rápido. E para superar essa deficiência apresentamos uma proposta de implementação de arquitetura orientada a serviços.

A proposta de implementação de SOA é começar com pequenos processos que envolvam os sistemas corporativos, para aos poucos ao adquirir maturidade e conhecimento, expandir gradativamente a adoção da arquitetura por completo.

A conclusão que se pode chegar, é que com essa nova proposta de arquitetura, algumas vantagens podem ser obtidas tais como o encapsulamento da complexidade do negócio, reusabilidade, desacoplamento, diminuição do número de tabelas replicadas, inicialmente.

A proposta do desenvolvimento de um sistema baseado em serviços é interessante e bastante satisfatória. Levando em consideração as possibilidades que se pode atingir com o uso de serviços os quais podem ser consumidos por aplicações diferentes tais como: web, desktop, palmtops, celulares, etc.

Com essa metodologia também visa-se diminuir o custo da infraestrutura, facilitar o desenvolvimento de aplicações e o acesso dos dados por outras unidades sem que se precise ter algum especialista técnico, por exemplo, para cuidar da replicação de dados.



### **5.1. Trabalhos Futuros**

O método proposto é uma versão inicial de uma abordagem a implementação de serviços. Possíveis melhorias nessa proposta podem ser sugeridas para complementar essa metodologia.

Novos estudos de caso com cenário diferente do apresentado pode refinar e aprimorar essa metodologia, inclusive podendo gerar novos artefatos para contemplar de uma forma mais madura essa nova proposta.

Um trabalho futuro interessante também, seria a extensão desse trabalho pensando em uma forma de construir uma orquestração de serviços criando ou implementando através de uma ferramenta existente no mercado, que seria um ESB (Enterprise Server Bus).

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Farooqui K.; Logrippo L.; Meer J. The ISO Reference Model for Open Distributed Processing - An Introduction. Disponível em: [http://citeseerx.ist.pst.edu/viewdoc/download?doi=10.1.1.99.5064\[1\].pdf](http://citeseerx.ist.pst.edu/viewdoc/download?doi=10.1.1.99.5064[1].pdf). Acesso em 10 de outubro de 2010.
- [2] Garcia S. S.; Shinotsuka T. H.; EAI – Enterprise Application Integration. Disponível em <http://www.bax.com.br/Bax/orientacao/alunos/PSI/EAI1.pdf>. Acesso em 18 de outubro de 2010.
- [3] Chaves L. G.; O EAI como abordagem de integração de sistemas corporativos para a obtenção de vantagem competitiva; Disponível em <http://www.linhadecodigo.com.br/artigo/429/O-EAI-como-abordagem-de-integracao-de-sistemas-corporativos-para-a-obtencao-de-vantagem-competitiva.aspx>. Acesso em 19 de outubro de 2010.
- [4] Mahmoud Q. H.; Service-Oriented Architecture (SOA) and Web Services: The Road to Enterprise Application Integration (EAI); Disponível em: <http://www.oracle.com/technetwork/articles/javase/soa-142870.html>. Acesso em 01 de novembro de 2010.
- [5] Malinverno P.; Hill J. B.; SOA and BPM Are Better Together. Disponível em [http://www.gartner.com/it/content/1128400/1128412/soa\\_and\\_bpm\\_are\\_better\\_together.pdf](http://www.gartner.com/it/content/1128400/1128412/soa_and_bpm_are_better_together.pdf). Acesso em 01 de novembro de 2010.
- [6] Koch C.; ABC da SOA. Disponível em <http://cio.uol.com.br/tecnologia/2006/07/17/idgnoticia.2006-07-17.3732358054/>. Acesso em 01 de novembro de 2010.

- [7] Marzullo F. P.; SOA na Prática Inovando seu negócio por meio de soluções orientadas a serviços. Editoria: Novatec; Primeira Edição Setembro de 2009. 390p.
- [8] Sampaio, C.; SOA e Web Services em Java. Rio de Janeiro: Brasport Livros e Multimídia Ltda, Primeira Edição 2006.
- [9] Ganci, J. et al; Patterns: SOA Foundation Service Creation Scenario. RedBooks, IBM. Edição Ilustrada Abril de 2004
- [10] Fugita S. H.; MAPOS: Método de análise e Projeto Orientado a Serviços. 2009. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3141/tde-15092009-171050/pt-br.php>>. Acesso em 14 de novembro de 2010.
- [11] Freitas D. N.; Metodologia de desenvolvimento de Software Livre com Arquiteturas Orientadas a Serviços: Um estudo de caso em um ambiente de tradução automática. 2009. Disponível em: <<http://dspace.c3sl.ufpr.br/dspace/bitstream/1884/23534/1/DissertacaoDanielFreitas.pdf>>. Acesso em 17 de novembro de 2010.
- [12] Degan J. O. C.; Integração de dados corporativos: uma proposta de arquitetura baseada em serviços de dados. 2005. Disponível em: <<http://cutter.unicamp.br/document/?code=vtls000378466>>. Acesso em 01 de novembro de 2010.
- [13] Costa C. A.; A aplicação da linguagem de modelagem unificada (UML) para o suporte ao projeto de sistemas computacionais dentro de um modelo de referência. 2001. Disponível em: <<http://www.scielo.br/pdf/gp/v8n1/v8n1a02.pdf>>. Acesso em 03 de novembro de 2010.

[14] Usländer T.; Service-Oriented Design of Environmental Information Systems. KIT Scientific Publishing. Kit Scientific Publishing 20 de janeiro de 2010.

[15] 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.

[16] Bizagi BPM 2010. Disponível em: <<http://www.microsoft.com/biztalk/default.mspx>>. Acesso em 20 de novembro de 2010.

[17] Santana F. S.; Uma infraestrutura orientada a serviços para a modelagem de nicho ecológico. 2009. Disponível em <<http://www.teses.usp.br/teses/disponiveis/3/3141/tde-13072009-165044/pt-br.php>>. Acesso em 19 de novembro de 2010.

[18] Miranda P. A. P; SOA – Arquitetura Orientada a Serviços. 2008. Disponível em <<http://www.cci.unama.br/margalho/portaltcc/tcc20082s/pdf/sampaio01.pdf>>. Acesso em 05 de novembro de 2010.

[19] Medeiros T. F.; Integração de sistemas através de WebServices. 2007. Disponível em: <<http://bibdig.poliseducacional.com.br/document/?view=72>>. Acesso em 01 de novembro de 2010.

[20] OASIS SOA Reference Model TC. Disponível em: <[http://www.oasis-open.org/committees/tc\\_cat.php?cat=soa](http://www.oasis-open.org/committees/tc_cat.php?cat=soa)>. Acesso em 19 de novembro de 2010.