

Gerenciamento de Rastreabilidade em Cadeias Produtivas Agropecuárias

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Andréia A. Kondo e aprovada pela Banca Examinadora.

Campinas, 10 de maio de 2007.

Profa. Dra. Claudia B. Medeiros
Instituto de Computação - UNICAMP
(Orientadora)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Substitua pela ficha catalográfica

(Esta página deve ser o verso da página anterior mesmo no caso em que não se imprime frente e verso, i.é., até 100 páginas.)

Substitua pela folha com as assinaturas da banca

Gerenciamento de Rastreabilidade em Cadeias Produtivas Agropecuárias

Andréia A. Kondo

Maio de 2007

Banca Examinadora:

- Profa. Dra. Claudia B. Medeiros
Instituto de Computação - UNICAMP (Orientadora)
- Prof. Dr. Edson Norberto Cáceres
Departamento de Computação e Estatística - UFMS
- Prof. Dr. Edmundo Roberto Mauro Madeira
Instituto de Computação - UNICAMP
- Prof. Dr. Ricardo Torres
Instituto de Computação - UNICAMP

© Andréia A. Kondo, 2007.
Todos os direitos reservados.

Resumo

Uma cadeia produtiva consiste de um conjunto de atividades desenvolvidas desde as matérias-primas até os consumidores finais. Sob o ponto de vista da Computação, cadeias produtivas apresentam vários desafios que vão desde a modelagem de seus processos até problemas relativos à comunicação entre seus componentes, logística de execução ou gerenciamento dos processos e produtos. Uma questão de importância crescente é permitir a rastreabilidade para garantir a procedência e a qualidade dos produtos. Há, no entanto, poucos trabalhos voltados a aspectos computacionais para resolver este problema. A maioria se concentra em aspectos específicos da cadeia, sem buscar soluções genéricas. A dissertação visa contribuir para preencher esta lacuna, abordando a questão da rastreabilidade de produtos, processos e serviços dentro de uma cadeia produtiva agropecuária. As principais contribuições são um modelo de armazenamento de dados de rastreabilidade e uma arquitetura baseada em Serviços Web associada. O trabalho foi validado por um protótipo da arquitetura, com testes que mostram ser uma solução genérica.

Abstract

A supply chain is a set of activities developed starting from raw materials to final consumers. Supply chains present many research challenges in Computing, such as the modeling of their processes, communication problems between their components, logistics or process and product management. An issues of increasing importance is to enable traceability to ensure the origin and quality control of the products. However, little has been published on implementation aspects to solve this problem. Most papers are related to specific aspects, and do not strive for generic solutions. This work contributes to fill this gap, considering product, process and service traceability within a supply chain. The main contributions are a model for traceability data storage, supported by Web Services-based architecture. This work presents was validated by a prototype, whose tests shows the genericity of the solution.

Agradecimentos

A Professora Claudia Medeiros por ter me guiado durante todo o mestrado.

Ao meu marido, Alan, pelo apoio, compreensão e amor que me ajudou a superar desafios e alcançar o meu objetivo.

A minha família, pela confiança, motivação e apoio que me deram durante a realização deste mestrado.

Ao Evandro Bacarin que contribuiu em todas as fases desta dissertação.

Ao Edmundo Madeira que ajudou a propor a arquitetura.

A todas as pessoas que passaram pelo LIS no período do meu mestrado, pelas contribuições e pelas amizades.

Este trabalho foi parcialmente financiado pela FAPESP processo número 05/53183-9 e CNPq, além dos projetos AgroFlow e WEBMAPS II do CNPq.

Sumário

Resumo	vi
Abstract	vii
Agradecimentos	viii
1 Introdução	1
2 Conceitos Básicos e Trabalhos Correlatos	4
2.1 Cadeias Produtivas	4
2.2 Rastreabilidade	5
2.2.1 Visão Geral	6
2.2.2 Rastreabilidade em Cadeias Produtivas Alimentares	7
2.2.3 Sistemas de Rastreabilidade em Cadeias Produtivas	9
2.2.4 Outros Aspectos de Rastreabilidade	10
2.3 Modelo de Bacarin	12
2.4 Serviços Web	15
2.5 Conclusões	17
3 Modelo para Rastreabilidade em Cadeias Produtivas	18
3.1 Repositórios	19
3.1.1 Repositório de Produtos	19
3.1.2 Repositório de Processos	20
3.1.3 Repositório de Serviços	21
3.1.4 Repositório de Participantes	21
3.1.5 Repositório de Regras	22
3.2 Sumários	23
3.2.1 Visão Geral	23
3.2.2 Sumário de Produtos	24
3.2.3 Sumário de Processos	26

3.2.4	Sumário de Serviços	28
3.3	Refinamento de Processos	30
3.4	Visão Integrada dos Registros de Repositórios e Sumários	32
3.5	Utilizando o Georreferenciamento	34
3.6	Expandindo o Cenário Básico da Cadeia Produtiva do Leite	34
3.7	Conclusões	38
4	Arquitetura	40
4.1	Visão Geral	40
4.2	Exemplos de Interação	42
4.3	Conclusões	44
5	Aspectos de Implementação	45
5.1	Ferramentas e Softwares adotados	45
5.2	Banco de Dados	46
5.3	Implementação da Arquitetura	49
5.3.1	Interface Básica das Consultas	49
5.3.2	Diagrama de Classes	50
5.3.3	Diagramas de Seqüência de algumas Operações Básicas	61
5.3.4	Diagrama de Atividades	63
5.4	Descrição de Consultas	66
5.4.1	Consulta Simples	69
5.4.2	Consultas Complexas	69
5.5	Conclusões	71
6	Conclusões e Trabalhos Futuros	72
6.1	Conclusões	72
6.2	Extensões	73
	Bibliografia	75

Lista de Tabelas

2.1	Tabela Comparativa de Artigos sobre Sistemas de Rastreabilidade.	10
2.2	Tabela Comparativa de Artigos.	13
3.1	Repositório de Produtos.	20
3.2	Repositório de Processos.	20
3.3	Repositório de Serviços.	21
3.4	Exemplo de um Repositório de Participantes.	22
3.5	Exemplo de um Repositório de Regras.	22
3.6	Sumário de Produtos para o Cenário Simples.	25
3.7	Tabela de Características dos Produtos	26
3.8	Sumário de Processos para o Cenário Simples.	27
3.9	Tabela de Características do Processo	28
3.10	Sumário de Serviço para o Cenário Simples.	30
3.11	Tabela de Deslocamento	30
3.12	Tabela de Armazenamento	30
3.13	Sumário de Processos Modificado para o Cenário Simples.	31
3.14	Sumário de Produtos Modificado para o Cenário Simples.	32
3.15	Sumário de Serviços Modificado para o Cenário Simples.	32
3.16	Sumário de Produtos para o Cenário Exemplo.	37
3.17	Sumário de Processos para o Cenário Exemplo.	38
3.18	Sumário de Serviços para o Cenário Exemplo.	39
5.1	Métodos de inserção, remoção e atualização da classe <i>WSSumManager</i> . . .	52
5.2	Métodos de consulta da classe <i>WSSumManager</i>	53
5.3	Tabela dos métodos da classe <i>WSProcSum</i>	54
5.4	Tabela dos métodos da classe <i>WSProdSum</i>	55
5.5	Tabela dos métodos da classe <i>WSServSum</i>	56
5.6	Tabela dos métodos da classe <i>WSProcRep</i>	57
5.7	Tabela dos métodos da classe <i>WSProdRep</i>	57
5.8	Tabela dos métodos da classe <i>WSServRep</i>	57

5.9	Tabela dos métodos da classe WSRegrasRep.	58
5.10	Tabela dos métodos da classe WSPartRep.	58
5.11	Tabela dos métodos da classe ClasseAuxiliar.	58
5.12	Métodos de inserção, remoção e atualização da classe InvocacaoWS.	59
5.13	Métodos de consultas da classe InvocacaoWS.	60

Lista de Figuras

2.1	Redes de Cadeias Produtivas [30].	5
2.2	Cadeia Produtiva do Leite [2].	14
2.3	Cadeia Encapsulada.	15
2.4	Arquitetura <i>Web Services</i> . Modificado de [16].	16
3.1	Exemplo Básico da Cadeia Produtiva do Leite.	18
3.2	Cenário simples da Cadeia Produtiva do Leite.	24
3.3	Registro de Sumário de Produtos: Visão Geral	25
3.4	Registro de Sumário de Processos: Visão Geral	26
3.5	Registro de Sumário de Serviços: Visão Geral	28
3.6	Encapsulamento no Processo de Fabricação do Queijo.	31
3.7	Sumários e Repositórios – Visão Global	33
3.8	Cenário Exemplo.	35
4.1	Arquitetura Proposta.	41
4.2	Exemplo de interação entre domínios.	42
4.3	Exemplo: seqüência de atividades.	43
4.4	Exemplo da Seqüência de Atividades em Domínios Distintos.	44
5.1	Modelagem do Banco de Dados.	46
5.2	Mapeamento ER para o Relacional.	47
5.3	Código SQL	48
5.4	Interface para consultas via Summary Manager.	50
5.5	Diagrama de Classes.	51
5.6	Diagrama de Seqüência para Inserir um registro no Sumário de Processos.	61
5.7	Diagrama de Seqüência para Inserir um registro no Sumário de Serviços.	62
5.8	Diagrama de Seqüência para obter os dados do produto genérico.	62
5.9	Diagrama de Seqüência para obter as características de um processo.	63
5.10	Diagrama de Seqüência para obter os dados dos serviços de armazenamento.	63
5.11	Diagrama de Atividades para o método de inserir eventos de produto.	65

5.12	Diagrama de Atividades para o método de atualizar campos e seus endereços no Sumário de Processo.	67
5.13	Diagrama de Atividades para o método de remover um serviço.	68
5.14	Exemplo de consulta realizada no Sumário de Produtos. (a) ilustra a álgebra relacional. (b) mostra a seqüência de invocações de serviços e suas respectivas respostas.	70
5.15	Exemplo de consulta obtendo dados do processo. (a) ilustra a álgebra relacional. (b) mostra a seqüência de invocações de serviços e suas respectivas respostas.	70
5.16	Exemplo de consulta obtendo dados de um serviço de transporte. (a) ilustra a álgebra relacional. (b) mostra a seqüência de invocações de serviços e suas respectivas respostas.	71

Capítulo 1

Introdução

Uma cadeia produtiva consiste de um conjunto de componentes interativos do qual fazem parte sistemas produtivos, fornecedores de insumos e serviços, indústrias de processamento e transformação, agentes de distribuição e comercialização, além dos consumidores finais [6]. Cadeias produtivas são estudadas sob vários aspectos, por exemplo, preocupação com os algoritmos adotados, logística, estratégias de armazenamento e escolha de parceiros. Gradativamente, os consumidores e as regras de qualidade têm se tornado mais exigentes. Com isso, para se determinar o critério de qualidade de um produto já são incluídos alguns itens como: se a técnica de produção de uma de suas matérias-primas é potencialmente prejudicial ao ambiente, se foi produzido por uma indústria ou num país que utiliza mão-de-obra infantil ou mal-remunerada, se o produto contém ingredientes geneticamente modificados, entre outros. Além disso, podem ocorrer, por exemplo, vazamentos químicos ou radioativos. Neste caso, seria necessário remover do mercado todos aqueles produtos que direta ou indiretamente tivessem sido contaminados.

Para que estes e outros itens possam ser verificados, é necessário um arcabouço que permite a rastreabilidade dos produtos. Rastreabilidade é um conceito que permeia vários tipos de atividades e refere-se à habilidade de descrever e seguir a vida de um elemento conceitual ou físico. A rastreabilidade pode ser realizada a partir da origem, desenvolvimento até a utilização dos produtos (chamada rastreabilidade para frente ou *forward*) e também pode ocorrer em ordem inversa (rastreabilidade para trás, *backward*) [27]. Em sistemas de bancos de dados, a noção de rastreabilidade está associada à execução de transações e é baseada em *log*. Em engenharia de *software*, está relacionada a todas as etapas da construção de um *software* desde a parte de requisitos até o produto final. Para um domínio alvo específico, rastreabilidade consiste na possibilidade de acompanhar as atividades e processos executados entre dois estágios no tempo.

As questões de rastreabilidade vêm tomando um papel importante em todos os setores. Alguns incidentes, como a epidemia da Encefalopatia Espongiforme Bovina (BSE)

ou “Vaca-louca” e problemas com produtos geneticamente modificados, têm preocupado as indústrias em relação aos assuntos de saúde e segurança. A rastreabilidade na indústria de alimentos está associada à preservação da identidade do alimento ao longo da cadeia produtiva [41]. Os alimentos podem estar contaminados devido às formas de armazenamento, manipulação, processamento e/ou preparação inadequada. O fator geográfico também é importante em questões de qualidade.

Outro incidente foi o ataque de 11 de setembro ao World Trade Center em Nova York, que causou a preocupação com o bioterrorismo. Assim, agências federais aumentaram a prevenção, regras, políticas e procedimentos relacionados a produtos falsificados. Dessa forma, surgiu um número de questões sobre segurança dos alimentos e também sobre a rastreabilidade dos produtos.

Com a doença da “vaca louca”, a União Européia regulamentou a rastreabilidade obrigatória para a carne bovina. Esta exigência afetou o comércio mundial de carnes, inclusive o Brasil. Isto motivou o surgimento de sistemas computacionais para a rastreabilidade, especialmente em agropecuária. Por exemplo, em Londrina-Paraná, o departamento de certificação animal do Instituto Gênesis [12] desenvolveu o Sistema Gênesis de Certificação SISBOV (SGCS), de acordo com as instruções normativas do SISBOV (Sistema Brasileiro de Identificação e Certificação de Origem Bovina e Bubalina). O SGCS tem por finalidade proporcionar maior transparência à metodologia de certificação e maior credibilidade ao programa SISBOV, cujo objetivo é identificar, registrar e monitorar todos os bovinos e bubalinos nascidos no Brasil ou importados. Através do SGCS, o Instituto [12] transfere, por meio de credenciamento, o serviço de identificação e rastreabilidade para empresas afins.

Na área de alimentos, tem havido grande evolução na cadeia de carne, com dispositivos que permitem marcar um animal e sistemas que armazenam eventos de sua vida. A ênfase, no entanto, vem sendo nos dispositivos rastreadores e não em uma infraestrutura computacional genérica. Alguns exemplos de dispositivos que auxiliam a rastreabilidade são: códigos de barras, cartões de microcircuitos e tags de frequência de rádio.

Aspectos que dificultam o desenvolvimento de sistemas e proporcionam a rastreabilidade dos componentes de uma cadeia são:

- Distribuição: os componentes da cadeia podem estar distribuídos por vários *sites* e os dados em múltiplos repositórios e banco de dados.
- Heterogeneidade: componentes da cadeia são heterogêneos, obedecendo a diferentes regras de negócios, serviços e funcionamento.

Ainda outros desafios incluem a coleta e armazenamento dos eventos, o gerenciamento de restrições associadas, participação e interação de agentes, protocolos de negociação e

recuperação de histórico, dentre outros. Embora alvo de vários estudos, há relativamente poucos trabalhos voltados a sistemas computacionais para a rastreabilidade de processos e produtos de uma cadeia produtiva. Uma das razões é a complexidade de cadeias reais, envolvendo dezenas de companhias potencialmente dispersas em todo o mundo. Mesmo em uma região geográfica mais restrita, o problema persiste.

O objetivo da dissertação é contribuir para resolver os problemas computacionais existentes para a rastreabilidade, desenvolvendo mecanismos que permitam monitorar as etapas e processos de uma cadeia produtiva, com ênfase em cadeias agropecuárias. Este trabalho está vinculado a uma tese de doutorado em andamento no Instituto de Computação da UNICAMP.

As principais contribuições são:

- Proposta de um modelo para gerenciar o armazenamento de dados de rastreabilidade.
- Proposta de uma arquitetura baseada em serviços Web.
- Implementação da arquitetura proposta, validando-a com a execução de consultas.

Parte dos resultados obtidos nesta dissertação foi publicada nos anais de um congresso internacional [21].

A dissertação está organizada da seguinte forma. O Capítulo 2 apresenta os conceitos e trabalhos relacionados. O Capítulo 3 discorre sobre o modelo de rastreabilidade proposto. O Capítulo 4 apresenta a arquitetura proposta para suportar o modelo de rastreabilidade. O Capítulo 5 descreve o protótipo implementado. Finalmente, o Capítulo 6 conclui a dissertação.

Capítulo 2

Conceitos Básicos e Trabalhos Correlatos

Este capítulo apresenta os vários conceitos utilizados no decorrer desta dissertação, além de mostrar alguns trabalhos relacionados. A Seção 2.1 descreve cadeias produtivas. A Seção 2.2 discute rastreabilidade. A Seção 2.3 apresenta o modelo de Bacarin [2], base para esta dissertação. A Seção 2.4 discorre sobre serviços *Web*. A Seção 2.5 conclui o capítulo.

Alguns trabalhos de rastreabilidade usam a noção de *workflow*. Um *workflow* denota a execução controlada de várias tarefas em um ambiente de processamento de elementos distribuídos. *Workflows* são freqüentemente associados à reengenharia de um processo de negócios, que é baseada na avaliação, análise, modelagem, definição e subseqüentes implementações operacionais do núcleo de processos de negócios de uma organização [18].

2.1 Cadeias Produtivas

Uma cadeia produtiva é um conjunto de componentes interativos do qual fazem parte sistemas produtivos, fornecedores de insumos e serviços, indústrias de processamento e transformação, agentes de distribuição e comercialização, além dos consumidores finais. Uma cadeia agrícola compreende todas as atividades que ocorrem desde a produção agrícola primária até a chegada do alimento processado na mesa do consumidor final [6, 30].

Uma cadeia produtiva pode ser vista de forma similar a uma rede, como ilustrada na Figura 2.1 (retirada de [30]). Os nós na rede representam instalações (*facilities*), que são conectadas por *links* de transportes. A figura apresenta em alto nível uma cadeia produtiva composta de quatro níveis: fornecedor, fábrica, centro de distribuição e mercados. O fluxo dos produtos na cadeia pode acontecer tanto da esquerda para a direita (fornecedor até mercado) quanto o inverso. O fluxo no sentido direita para esquerda (fluxo

de retorno) ocorre quando um produto intermediário retorna para uma etapa anterior, proporcionando o reuso ou a reciclagem de produtos [40].

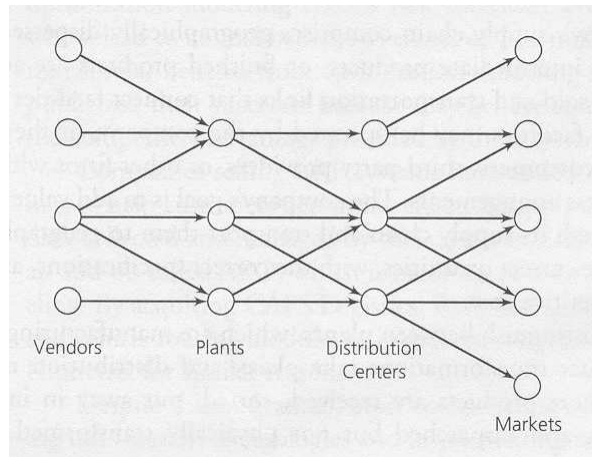


Figura 2.1: Redes de Cadeias Produtivas [30].

Com a crescente distribuição de serviços, o sucesso de uma cadeia produtiva depende da capacidade de integração das empresas. Esta integração é realizada com o gerenciamento de cadeias produtivas, que consiste na integração de processos de negócio (desde os fornecedores de matérias-primas até o usuário final) provendo produtos, serviços e informações para o cliente [30].

A dissertação aborda cadeias agropecuárias. O fluxo dentro deste tipo de cadeia está sujeito a vários controles específicos, por exemplo, se o processo de produção é prejudicial ao ambiente ou se há o uso de substâncias geneticamente modificadas. Isso requer o monitoramento em todas as etapas da produção. Por exemplo, o estado do Paraná está usando a rastreabilidade de forma a eliminar a colheita de produtos geneticamente modificados, devido à proibição de seu cultivo na região [22].

2.2 Rastreabilidade

Esta seção discorre sobre a rastreabilidade (Seção 2.2.1) e faz uma revisão de artigos que têm rastreabilidade como alvo. O principal ponto em comum de todos os trabalhos é a preocupação com a qualidade do sistema e dos produtos produzidos. Inicialmente, são apresentados alguns trabalhos que se preocupam com aspectos de cadeias alimentares (Seção 2.2.2). A Seção 2.2.3 aborda trabalhos que apresentam sistemas computacionais para a rastreabilidade. A Seção 2.2.4 discorre sobre artigos que abordam a rastreabilidade em outros domínios.

2.2.1 Visão Geral

Gryna [14] define rastreabilidade como a capacidade de preservar a identidade dos produtos e suas origens. A rastreabilidade é necessária principalmente para [14, 32]:

- garantir que somente materiais e componentes de qualidade estejam presentes no produto final.
- melhor alocar as responsabilidades.
- identificar produtos que são distintos, porém podem ser confundidos.
- possibilitar o retorno de produtos com defeitos ou suspeitos.
- localizar as causas de falhas e tomar medidas para repará-las a um custo mais baixo.

Segundo Moe [26], rastreabilidade na cadeia alimentar é a capacidade de localizar um produto e seu histórico através da cadeia produtiva, passando por etapas de colheita, transporte, armazenamento, processamento, distribuição e vendas. Além disso, a rastreabilidade pode ser realizada internamente aos passos da cadeia produtiva – por exemplo, rastreando as matérias-primas que foram utilizadas para a fabricação de um produto. Segundo o autor, algumas vantagens da rastreabilidade interna são: melhor controle de processo, correlação de dados dos produtos com características de matérias-primas e maior homogeneidade na qualidade das matérias-primas utilizadas.

Ainda neste contexto, Machado [23] afirma que rastreabilidade é um processo de práticas sistemáticas de segregação física e troca de informações entre diferentes agentes da cadeia produtiva, responsáveis pela execução e cumprimento de uma meta específica com a finalidade de preservar os atributos e a identidade de produtos transacionados de acordo com suas especificações.

Opara [31] enumera seis elementos importantes que devem ser considerados para a rastreabilidade:

- produtos: determina a localização de um produto em alguma fase da cadeia produtiva;
- processos: identifica o tipo e a seqüência de atividades que afetou um determinado produto;
- genética: determina a constituição genética do produto;
- entrada: determina o tipo e a origem da entrada como, por exemplo, fertilizantes e gado;

- doenças e pestes: rastreia a epidemiologia das pestes como, por exemplo, bactérias e vírus;
- medidas: determina os instrumentos de medição, além da especificação dos fatores ambientais, geo-espaciais e temporais que influenciam a qualidade dos dados.

A rastreabilidade pode ser realizada em dois sentidos para frente (*forward*) e para trás (*backward*) [19]. A rastreabilidade *forward* indica o que acontecerá com um produto – por exemplo, pode ser necessário saber o que foi produzido com determinadas matérias-primas. A rastreabilidade *backward* refere-se às informações sobre o passado na vida de um produto – por exemplo, quando um consumidor deseja saber as matérias-primas utilizadas para a fabricação de um produto.

2.2.2 Rastreabilidade em Cadeias Produtivas Alimentares

Cadeias produtivas alimentares têm grande preocupação com a qualidade do produto que chega ao consumidor. Exemplos de tais trabalhos são Hobbs [17], Wilson e Clarke [45] e Stock [41].

Segundo Hobbs [17], cada vez mais os clientes dos supermercados desejam obter mais informações sobre as práticas de produção e processamento dos produtos. O artigo identifica três aspectos da cadeia produtiva de carne que são importantes para os varejistas do Reino Unido: qualidade do produto, rastreabilidade e preocupação com a saúde animal. Além disso, realiza um estudo dos custos das transações comerciais necessárias para garantir esses três aspectos. Esses custos são aqueles correspondentes à troca de mercadorias e serviços.

Wilson e Clarke [45] tratam da qualidade, segurança e rastreabilidade de produtos em cadeias produtivas agrícolas no Reino Unido. O Sistema *Food Trak*, descrito no artigo, tem como objetivo projetar e trazer para o mercado um meio de comunicação comum (Internet) de segurança e informação da qualidade da comida.

Stock [41] apresenta a cadeia produtiva alimentar dos Estados Unidos, identifica os papéis dos seus vários componentes e discute as principais forças que estão por trás das mudanças ocorridas na indústria alimentar. O artigo trata a rastreabilidade como uma forma de preservação da identidade dos produtos, enfatizando sua importância com o crescimento de alimentos geneticamente modificados.

No Brasil, a rastreabilidade também está sendo bastante abordada em cadeias produtivas. Exemplos de tais trabalhos são Schuchmann e Dewes [39], Machado [24], Santo e Medeiros [38] e Instituto Gênese [12].

Schuchmann e Dewes [39] abordam a segurança alimentar e a rastreabilidade na cadeia produtiva da erva-mate da região do Alto Uruguai Rio-Grandense. O artigo faz o levantamento dos dados dessa cadeia produtiva realizando pesquisas com pessoas que trabalham

na área. A partir disso, sugere ações para que tal cadeia passe a dispor de mecanismos de rastreabilidade – como, por exemplo, integração dos elos da cadeia produtiva, incorporação de técnicas modernas de produção e priorizar a qualidade do produto.

Machado [24] discute a rastreabilidade para o controle da qualidade dos alimentos. Apresenta os custos, a importância e as dificuldades para a implantação da rastreabilidade em sistemas agroindustriais (com exemplos em cadeias de soja e de carne bovina). Segundo [24], a rastreabilidade exige adaptações ao longo da cadeia produtiva, o que acarreta aumento dos custos das transações.

Santo e Medeiros [38] abordam a coordenação e a qualidade na rastreabilidade da cadeia produtiva da carne bovina. O artigo faz uma análise comparativa dos sistemas francês e brasileiro de carne bovina. O sistema francês de rastreabilidade identifica os animais, ao nascer, com dois brincos. Estes brincos contêm um número de identificação e são aprovados pelo Ministério da Agricultura e Pesca. Os criadores informam às autoridades regionais da agropecuária (EDE - *Etablissement Departamental d'Eleavage*) o nascimento do bezerro e suas características. Com essas informações, o EDE envia um passaporte do animal que o acompanha por toda a vida. O sistema brasileiro é um sistema de rotulagem denominado “verde-amarelo” para garantir a rastreabilidade, possibilitando correlacionar o produto final com o lote de bovinos que lhe deu origem. Este sistema foi desenvolvido para cumprir a exigência de rotulagem com garantia de qualidade imposta pela União Européia, devido à crise provocada com a doença da “vacca louca”. Entretanto, a rotulagem é feita apenas nas indústrias frigoríficas, não estando associada a dados das propriedades rurais.

Dessa forma, novos procedimentos da Comunidade Européia tornariam o sistema brasileiro, abordado por Santo e Medeiros [38], inválido, pois requerem que a identificação animal seja registrada e os dados de movimentação e manejo controlados. Como consequência, em 2002, o Ministério da Agricultura, Pecuária e Abastecimento criou o Sistema Brasileiro de Identificação e Certificação de Origem Bovina e Bubalina (SISBOV). O SISBOV visa identificar, registrar e monitorar, individualmente, todos os bovinos e bubalinos nascidos no Brasil ou importados. Brisola et al [5] mostra uma pesquisa, com consumidores de Brasília, sobre o conhecimento da rastreabilidade e do SISBOV e os custos adicionais para a carne bovina rastreada.

Em Londrina-Paraná, o departamento de certificação animal do Instituto Gênesis [12] desenvolveu o Sistema Gênesis de Certificação SISBOV (SGCS), de acordo com as instruções normativas do SISBOV. O SGCS tem por finalidade proporcionar maior transparência à metodologia de certificação e maior credibilidade ao programa SISBOV. Através do SGCS, o Instituto Gênesis transfere para empresas afins, por meio de credenciamento, o serviço de identificação e rastreabilidade.

Um fator complicador na rastreabilidade alimentar é a influência geográfica na quali-

dade dos produtos. A rastreabilidade geográfica é definida como o resultado da associação entre informações geográficas e dados tradicionais usados em esquemas de rastreabilidade. Informações geográficas são uma fonte importante para qualidade de alimentos, permitindo determinar padrões de produção relativos ao meio ambiente. Ainda assim, na maioria dos trabalhos sobre alimentos, a origem geográfica é determinada por análises químicas dos produtos. Poucas são as iniciativas que exploram plenamente o georreferenciamento. Uma exceção é o projeto europeu GeoTraceAgri [9], com resultados sobre milho e outros produtos agrícolas.

2.2.3 Sistemas de Rastreabilidade em Cadeias Produtivas

Há poucas descrições de sistemas computacionais que tratam da rastreabilidade de cadeias produtivas alimentares. A Tabela 2.1 mostra uma comparação de alguns desses sistemas, descritos a seguir.

Bello et al [3] propõe um sistema de informação colaborativo distribuído que utiliza XML para a representação de dados e serviços Web para a comunicação entre diferentes fornecedores através do protocolo HTTP. Neste sistema, todas as empresas trocam dados de rastreabilidade com outras empresas usando a rede.

Cimino et al [8] descreve um sistema de rastreabilidade que possibilita a troca de documentos entre alguns elementos da cadeia produtiva. Para a confiabilidade destes documentos foi utilizado ebXML (*Electronic Business using eXtensible Markup Language*). Trata-se de um conjunto de especificações que permite a empresas realizar negócios na Internet auxiliando a homogeneidade dos dados, escalabilidade e interoperabilidade do sistema.

Pinto et al [36] apresenta um pacote de software projetado para gerenciar a rastreabilidade. As principais características deste pacote são: registrar os parâmetros de controle de qualidade das matérias-primas dos produtos intermediários ou finais; gerenciar recepção, produção e expedição de pedidos; oferecer análise de custos, produtividade e consumo de produtos e matérias-primas.

Taniguchi e Sagawa [42] apresentam duas aplicações de sistemas de rastreabilidade baseados em IC-tags: sistema de inspeção de estoque automatizado e sistemas de previsão de informação orientada ao consumidor. IC-tags ou tags RFID (*Radio Frequency Identification*) são dispositivos que contém chips de silício e antenas que permitem a identificação automática de produtos através de sinais de rádio. O primeiro trata do uso de IC-tags em roupas para facilitar o trabalho da leitura de códigos de barra. O segundo apresenta como exemplo o sistema de rastreabilidade da indústria de carne. A partir desses dois exemplos, o autor descreve um sistema ideal de rastreabilidade. O sistema integra os sistemas mencionados anteriormente e utiliza banco de dados centralizados

armazenando o que o autor denomina de registros 5W (*Which - Object, What - Event, When - Time, Who - Worker, Where - Place*), que são entradas do log de eventos. Como se verá no Capítulo 3, o modelo proposto na dissertação é também baseado em log, com alguns campos similares.

Wakayama et al [44] descreve um sistema de rastreabilidade proposto pela Toshiba que emprega banco de dados distribuídos e valores *ID-hash*. Este sistema tem como característica o gerenciamento de banco de dados por domínio de gerenciamento, por exemplo, candidatos a domínio de gerenciamento seria fazendas, armazéns, mercados entre outros. Utiliza um mecanismo “*ID-DB resolution*” para encontrar o banco de dados correspondente a um identificador. Este mecanismo baseia-se em um método de gerenciamento de dados distribuídos chamado DHT (*Distributed Hash Table*) que usa um valor hash de uma chave (por exemplo, o id do produto) para distribuir os dados entre os nós da rede.

Artigo	Foco	Ambiente	Tecnologias utilizadas	Interoperabilidade	Armazenamento em BD
Bello et al [3]	Gerenciamento e transporte de dados baseados em XML. Avaliação do sistema de rastreabilidade	Distribuído	XML, Web Services	Sim	Não especificado. Informa apenas que haverá um repositório para armazenar dados de rastreabilidade XML.
Cimino et al [8]	Modelo de dados e comportamento dinâmico dos lotes	Centralizado e Distribuído	ebXML	Sim	Não é apresentado
Pinto et al [36]	Aplicação	Centralizado	SQL, Visual Basic	Desenvolvido para windows 95/98/2000/XP	Mostra uma figura com alguns relacionamentos, porém não detalha o que é armazenado
Taniguchi e Sagawa [42]	Sistemas de Rastreabilidade	Centralizado	IC tags	---	BD centralizado. Tabela 5W (which, when, who, where and what)
Wakayama et al [44]	Gerenciamento de Dados Distribuídos	Distribuído	RFID tags, PHP, PostgreSQL	Sim	BD distribuídos
Nosso trabalho	Modelo de armazenamento de dados	Distribuído	Web Services, PostGIS	Sim	Especificado em detalhes

Tabela 2.1: Tabela Comparativa de Artigos sobre Sistemas de Rastreabilidade.

2.2.4 Outros Aspectos de Rastreabilidade

Os artigos que abordam rastreabilidade podem ser classificados segundo diversos aspectos. Alguns, por exemplo, se preocupam com produtos ou processos da área alvo (agricultura, química, física) enquanto outros são voltados ao rastreamento da execução de programas em Engenharia de *Software*. Por exemplo, segundo Hamilton e Beeby [15], rastreabilidade é a habilidade de descobrir o histórico de todas as atividades envolvidas na construção de um sistema. A rastreabilidade aumenta a necessidade da documentação de projetos de *software*.

Outros usos para a rastreabilidade apontados são a análise de segurança, auditoria e controle de mudanças do projeto. Além disso, Hamilton e Beeby [15] apresentam ferramentas construídas para auxiliar os usuários a criar seus próprios ambientes integrados.

Estas ferramentas possuem mecanismos que provêm a oportunidade de automatizar a rastreabilidade além do controle das atividades.

Muitos trabalhos de rastreabilidade de *software* estão associados a sistemas críticos, como no Departamento de Defesa Americana. Por exemplo, segundo Gieszl [13], a noção de rastreabilidade foi compreendida, no relatório DOD STD 2167A (*Military Standard: Defense System Software Development*), como uma tarefa específica para desenvolvedores de *software* daquele Departamento. O conceito de rastreabilidade, nesse contexto, implica rastrear os passos do desenvolvimento do *software* desde os objetivos iniciais até o produto final. A rastreabilidade de *software*, nesse projeto, visava ajudar a integração e planos de teste de sistemas para a Marinha americana.

A noção de rastreabilidade em Engenharia de *Software* não se aplica apenas à execução. Conforme Ramesh [37], pode ser entendida no contexto de requisitos. A rastreabilidade de requisitos tem como objetivo verificar se o *software* desenvolvido satisfaz as expectativas do usuário e é uma garantia de qualidade do sistema.

Ainda outro exemplo de rastreabilidade associada a ferramentas em Engenharia de *Software* é o trabalho de Pinheiro e Gouguen [35] que descreve uma ferramenta denominada TOOR (*Traceability of Object-Oriented Requirement*), que utiliza várias formas de rastrear requisitos para permitir manutenção de atividades.

Os artigos de rastreabilidade em Engenharia de *Software* mencionados até agora têm como preocupação a qualidade do sistema. Já o trabalho de Maurer [25], apesar de ser relacionado ao mesmo domínio, preocupa-se com o gerenciamento de dependência entre decisões de projetos, que o autor considera a base para melhorar a rastreabilidade no processo de desenvolvimento de um sistema. O sistema CoMo-Kit, descrito nesse artigo, tem como foco métodos e técnicas para o planejamento e coordenação de projeto de *software*.

Outro conjunto de trabalhos na área aborda técnicas computacionais de rastreamento. Uma das formas de especificar e monitorar processos é por meio de *workflows*. Exemplos de tais trabalhos são Kaster et al [20], Pastorello et al [33] e, Xu e Ramesh[46].

Kaster et al [20] descreve o funcionamento do CBR (*Case-Based Reasoning*) como parte de mecanismos de recuperação e armazenamento para identificar modelos similares para reuso em novos processos de decisão. CBR, ou Raciocínio Baseado em Casos, consiste em uma técnica para resolução de problemas que ajustam as soluções usadas para resolver problemas anteriores. O trabalho foi realizado para o sistema WOODSS (*WOrkflOW-based spatial Decision Support System*). Neste sistema, *workflows* documentam seqüências de atividades, de forma semelhante ao armazenamento de eventos para rastreamento.

O trabalho de Pastorello et al [33] aborda, com mais detalhes, a parte de documentação baseada no WOODSS. O artigo preocupa-se com as formas de documentar processos de planejamento apresentando três aspectos diferentes: “o quê” foi feito, “como” e “por

quê” o plano foi desenvolvido. O trabalho mostra como tais documentos podem ser transformados no contexto de Web semântica.

Xu e Ramesh [46] propõem o uso de gerenciamento do conhecimento para garantir a capacidade da rastreabilidade na execução de *workflows* a partir de WFMS (Sistema de Gerenciamento de *Workflow*). Além disso, apresenta uma abordagem para gerenciar o processo de conhecimento no desenvolvimento e manutenção de sistemas de *workflow* e suas instâncias, usando a rastreabilidade de execução de *workflows*. Esta abordagem é baseada na premissa que uma variedade de ferramentas podem ser desenvolvidas para a especificação, manutenção e instanciação de *workflows*.

A tabela 2.2 apresenta os artigos analisados de forma comparativa. A tabela mostra que, por exemplo, poucos dos trabalhos revistos tratam de *workflows* no contexto de rastreabilidade ([46], [20] e [33]). Além disso, observou-se o uso da rastreabilidade na documentação e construção de *softwares* ([37], [13], [15], [25], [35]). Pode-se constatar a predominância da rastreabilidade em aspectos de alimentação e agropecuária ([17], [39], [45], [12], [24], [38], [41], [5]). Finalmente, vários dos trabalhos considerados envolvem algum tipo de implementação de sistemas, embora sem detalhes sobre estruturas de dados utilizadas.

2.3 Modelo de Bacarin

A dissertação se baseia no modelo de cadeias agrícolas de Bacarin [2]. Este modelo utiliza uma arquitetura em que os elementos da cadeia produtiva são mapeados como serviços Web e sua dinâmica é realizada com orquestração ou coreografia de serviços. O modelo, ilustrado na Figura 2.2, é composto pelos seguintes elementos básicos:

- Atores: são *softwares* ou humanos interagindo na cadeia;
- Produção: elemento que encapsula algum processo produtivo. A partir de entradas de insumos, produz um ou mais produtos para seguir pela cadeia produtiva;
- Armazenamento: elemento que armazena produtos ou matérias-primas;
- Transporte: elemento que faz o deslocamento de produtos e matérias-primas entre componentes de produção e/ou armazenamento.

Além disso, a dinâmica da cadeia é modelada a partir dos seguintes elementos:

- Regras: especificam as restrições impostas nas várias fases da cadeia produtiva;
- Contratos: são declarações de obrigações e autorizações mútuas que refletem o acordo entre parceiros de negócios, definindo qualidade, data de entrega e custos;

Nome do Artigo	Domínio	Workflow	Ênfase	Preocupação	Sistemas Relacionados
<i>Ramesh [37]</i>	Engenharia de Software	—	Fatores que influenciam o uso da rastreabilidade	Entender os fatores que impedem ou facilitam o uso da rastreabilidade	—
<i>Gieszl [13]</i>	Engenharia de Software	—	Integração	Documentar os problemas e a utilidade de tomar real a rastreabilidade em programas sendo desenvolvidos	Protótipo para o projeto da Marmha americana
<i>Hamilton e Beeby [15]</i>	Engenharia de Software	—	Integração de Ferramentas	Qualidade do Software	—
<i>Maurer [25]</i>	Engenharia de Software	—	Sistema	Dependência entre decisões	CoMo-Kit
<i>Pinheiro e Gouguen [35]</i>	Engenharia de Software	—	Ferramenta	Rastreamento de requisitos para melhorar desenvolvimento de software	TOOR (Traceability of Object-Oriented Requirement)
<i>Schuchmann e Devos [39]</i>	Cadeia produtiva da Erva-Mate	—	Segurança e Rastreabilidade dos Alimentos	Qualidade alimentar	—
<i>Instituto Gênesis [12]</i>	Cadeia Produtiva Agropecuária	—	Sistema	Identificação, Rastreabilidade e Certificação Animal	SCGS (Sistema Genesis de Certificação SISBOV)
<i>Hobbs [17]</i>	Cadeia Produtiva de Carne	—	Análise dos custos das transações	Qualidade, rastreamento e saúde animal	—
<i>Wilson e Clarke [45]</i>	Cadeia Produtiva Agrícola	—	Sistema	Qualidade e Segurança alimentar	Food Trak
<i>Machado [24]</i>	Cadeia produtiva de soja e carne bovina	—	Custos, importância e dificuldade para implantação da rastreabilidade	Qualidade alimentar	—
<i>Santo e Medeiros [38]</i>	Cadeia Produtiva de Carne Bovina	—	Coordenação da Rastreabilidade	Qualidade alimentar	—
<i>Stock [41]</i>	Cadeia produtiva alimentar	—	Tendências que afetam a indústria alimentícia	Qualidade e Segurança alimentar	—
<i>Brisola et al [5]</i>	Cadeia Produtiva de Carne	—	Pesquisa sobre rastreabilidade, SISBOV e custos	Interesse dos Consumidores na rastreabilidade da carne	—
<i>Xu e Ramesh [46]</i>	—	X	Baseado em gerenciamento do conhecimento	Reconfiguração de workflow e manter integridade	WFMS
<i>Kaster et al [20]</i>	Sistema de Informação Geográfica	X	Sistema	Uso de workflows e CBR (Raciocínio baseado em casos) no WOODSS	WOODSS
<i>Pastorello et al [33]</i>	Planejamento Ambiental	X	Cerenciamento de Documentação	Rastreamento e revisão das atividades	WOODSS

Tabela 2.2: Tabela Comparativa de Artigos.

- Plano de coordenação: é um conjunto de diretivas que descreve um plano para a execução da cadeia. Este elemento organiza as interações entre os componentes da cadeia produtiva.
- Sumários: são elementos propostos para permitir a rastreabilidade. São divididos em dois tipos: Sumários de Processos e de Produtos. Um Sumário de Processos contém informações sobre a execução de um processo de produção. Um Sumário de Produtos armazena informações de como, quando e onde um produto passou ao longo da cadeia.

A Figura 2.2, retirada de Bacarin [2], ilustra de forma simplificada a cadeia produtiva do leite, cujo objetivo é processar leite, produzir e comercializar produtos relacionados. O monitoramento dos produtos é realizado a cada etapa desta cadeia. A figura mostra que o produtor de leite disponibiliza o leite para o laticínio (“*Dairy*”), onde é processado de acordo com regras (*Regulation 1*) e transformado em derivados. Após isso, o laticínio fornece os produtos para o atacadista (“*Wholesale*”), que distribui (transportado por “*Transport 3*”) para o varejista (“*Retail*”) chegando ao consumidor. Pode-se notar que entre os elementos de produção (representadas por elipses) e armazenamento (representados por retângulos) são necessários elementos de transporte (representados pelos hexágonos).

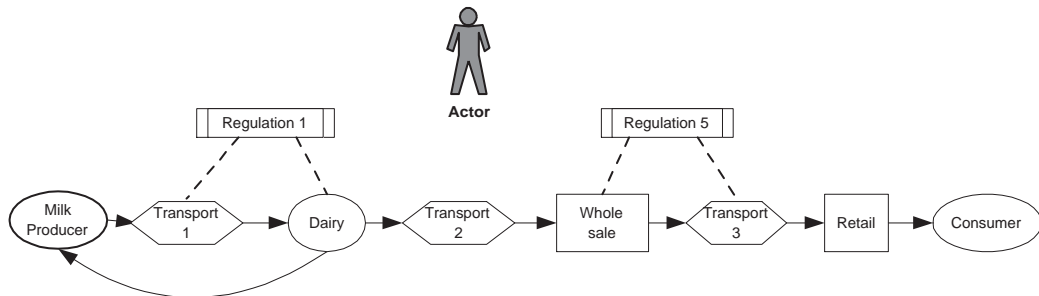


Figura 2.2: Cadeia Produtiva do Leite [2].

No modelo, cada componente da cadeia produtiva pode encapsular outras cadeias. Por exemplo, o componente “*Transport 2*” mostrado na Figura 2.2 pode ser encapsulado, conforme ilustra a Figura 2.3, por “*T2*” que faz o transporte dos produtos criados em “*Dairy*” para um armazém “*W1*”, o qual armazena o produto até que “*T3*” faça o transporte do produto para o atacadista (“*wholesale*”). O monitoramento também é realizado nas cadeias encapsuladas. Os elementos do modelo podem ser vistos como sendo encapsulados por serviços Web. Dessa forma, aspectos de integração são analisados a partir de estudos de gerenciamento e comunicação entre serviços.

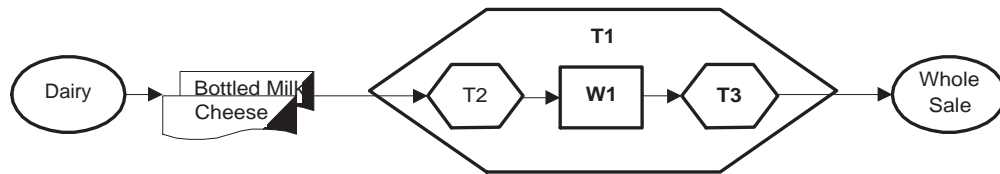


Figura 2.3: Cadeia Encapsulada.

Como se verá foi necessário estender a especificação do modelo de Bacarin para acomodar as várias exceções encontradas no armazenamento de registros para a rastreabilidade.

2.4 Serviços Web

Serviços Web ou *Web Services* são aplicações Web auto-descritas e modulares acessadas por meio de interfaces programáveis e de protocolos da Internet, com a finalidade de proporcionar formas de encontrar, assinar e invocar tais serviços [28]. Dessa forma, facilitam a comunicação entre diferentes aplicações e plataformas. Outras características incluem [10]:

- Promovem uma abordagem modular para a programação, de modo que várias organizações possam se comunicar com o mesmo serviço Web.
- São fáceis e baratos para implementar, devido ao emprego de uma infraestrutura existente, como a Web, para a troca de informações.

Serviços Web são implementados baseados em padrões e tecnologias específicos, tais como: XML (*Extensible Markup Language*), SOAP (*Simple Object Access Protocol*), WSDL (*Web Services Description Language*) e UDDI (*Universal Description, Discovery and Integration*) [7]. XML [11] é uma linguagem designada a descrever e estruturar dados. É baseado em padrões de tecnologias para a Web, definidos pelo W3C (*World Wide Web Consortium*).

SOAP [4] é um protocolo que permite a troca de mensagens em ambientes descentralizados e distribuídos. É uma das soluções para a invocação e a comunicação entre serviços, realizando o transporte de documentos XML por meio de protocolos de Internet como HTTP (*Hypertext Transfer Protocol*). Neste ambiente, as mensagens encapsulam informações que são transmitidas entre serviços Web e especificam a operação que será invocada. Uma mensagem é um arquivo XML com formato padronizado. A troca de mensagens SOAP baseia-se em uma estrutura para fazer chamadas remotas de procedimentos (RPC ou *Remote Procedure Calls*). Estas mensagens permitem que clientes invoquem

procedimentos, funções e métodos de um objeto remoto utilizando um protocolo baseado em XML [7].

WSDL [43] é uma linguagem baseada em XML para descrever um serviço Web [7]. Um documento WSDL especifica as funcionalidades do serviço, a localização do serviço na Web e as instruções para acessá-lo. Esse documento também define a estrutura das mensagens que um serviço envia e recebe [10].

UDDI [29] tem como objetivo possibilitar que desenvolvedores e negócios publiquem e localizem serviços Web na Internet. UDDI define um formato baseado em XML em que empresas podem descrever um determinado negócio. As empresas podem armazenar suas informações quer em registros UDDI privados, que são serviços acessíveis somente a sócios do negócio, quer em registros UDDI públicos, que qualquer pessoa pode acessar [16].

A Figura 2.4 apresenta a arquitetura geral (SOA - *Service Oriented Architecture*) em que um serviço Web opera. Nesta arquitetura, o provedor de serviços oferece serviços Web e publica a descrição de seus serviços em um registro de serviços por meio de uma especificação em WSDL. O registro de serviços é um repositório que contém informações sobre serviços Web. UDDI é um exemplo de tal repositório. Além de armazenar informações técnicas sobre como ter acesso ao serviço (URL - *Uniform Resource Locator*, dentre outras), o registro de serviços também armazena informações para a categorização de tais serviços e informações sobre a própria organização que oferece o serviço (por exemplo, endereço e telefone). O cliente de serviços é uma aplicação que procura por um serviço Web no registro de serviços e depois de encontrá-lo invoca-o utilizando o SOAP.

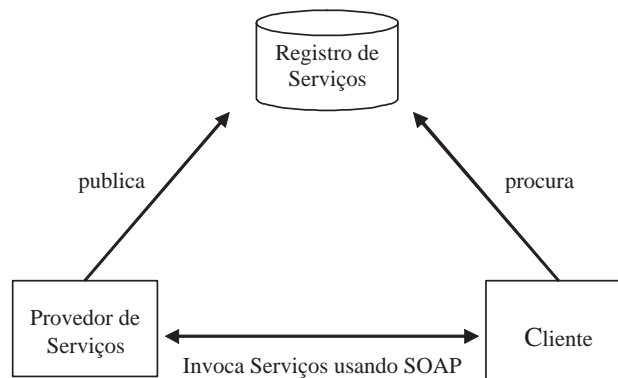


Figura 2.4: Arquitetura *Web Services*. Modificado de [16].

Os serviços podem ser implementados combinando funcionalidades providas por outros serviços Web, ou seja, a lógica de negócios envolve a invocação de operações de outros serviços Web. Isto é conhecido como serviço composto. O processo de desenvolvimento deste serviços Web compostos é chamado de composição de serviços [1].

2.5 Conclusões

Este capítulo apresentou os principais conceitos e trabalhos correlatos utilizados na dissertação. O próximo capítulo descreve o modelo de rastreabilidade proposto e o capítulo 4 apresenta a arquitetura proposta.

Capítulo 3

Modelo para Rastreabilidade em Cadeias Produtivas

Este capítulo apresenta o modelo proposto para cadeias produtivas, que combina sumários e repositórios.

A Figura 3.1 mostra um exemplo básico da cadeia produtiva do leite, utilizado de agora em diante para exemplificar a especificação do modelo. A cadeia produz apenas um produto (*Queijo*). Podem-se observar elementos de produção (*Fazenda de Leite1* e *Laticínio1*), transporte (*Transp1*, *Transp2* e *Transp3*) e armazenamento (*Armazém1* e *Mercado1*). Como mostra a figura, toda cadeia produtiva pode ser representada por um grafo orientado que possui ao menos uma fonte (*Fazenda de Leite1*) e um destino (*Mercado1*). Neste exemplo, o leite é produzido pela *Fazenda de Leite1* e transportado pelo *Transp1* para o *Laticínio1*. Em seguida, no laticínio, o leite passa por um processo de transformação para fabricar o produto *Queijo*. Este, por sua vez, é transportado (*Transp2*) para um armazém (*Armazém1*) onde o produto fica em estoque até ser transportado (*Transp3*) para o mercado (*Mercado1*). No mercado, o produto é comercializado e chega à mesa do consumidor. A seta de *Laticínio1* para a *Fazenda1* mostra um fluxo de retorno.

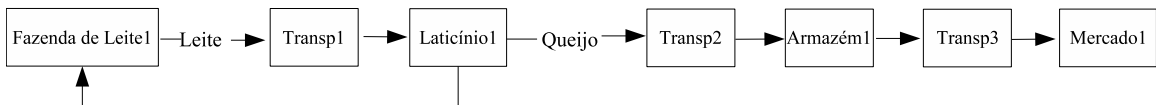


Figura 3.1: Exemplo Básico da Cadeia Produtiva do Leite.

A proposta para gerenciamento de rastreabilidade parte dos seguintes pontos principais:

- a execução de uma cadeia gera eventos que são registrados em repositórios (Seção 3.1).
- uma seqüência de registros de eventos corresponde a um sumário (Seção 3.2) que sintetiza a execução da cadeia.
- registros de eventos são interligados e podem referenciar outros sumários (Seção 3.3).

Uma visão integrada dos repositórios e dos sumários propostos pode ser vista na Seção 3.4. O uso do georreferenciamento é descrito na Seção 3.5. A Seção 3.6 apresenta outros cenários e suas implicações no modelo. A Seção 3.7 conclui o capítulo. Sumários estão armazenados em repositórios específicos. Sumários e todos os repositórios estão ligados entre si – a Figura 3.7 ilustra como todos os repositórios estão interligados.

3.1 Repositórios

Os repositórios partem da proposta de Bacarin [2] e armazenam informações sobre a execução e os elementos da cadeia produtiva. Bacarin [2] propôs cinco repositórios: Participantes, Produtos, Regras, Contratos e Sumários. Para garantir a rastreabilidade, a dissertação constatou a necessidade de mais dois repositórios: Processos e Serviços.

Este trabalho propõe o detalhamento do conteúdo dos Repositórios de Produtos (Seção 3.1.1), Processos (Seção 3.1.2), Serviços (Seção 3.1.3) e de Participantes (Seção 3.1.4). O Repositório de Regras é apresentado de forma simples na Seção 3.1.5. O Repositório de Sumários foi subdividido em vários repositórios (Seção 3.2).

3.1.1 Repositório de Produtos

O Repositório de Produtos, ilustrado na Tabela 3.1, contém a descrição genérica dos produtos utilizados ou gerados ao longo de uma cadeia produtiva. Serve para cadastrar e buscar os produtos de forma rápida. Além disso, possibilita obter informações sobre um produto que pode participar de várias cadeias produtivas distintas. A descrição genérica – por exemplo, produto Queijo Minas – deve ser obedecida por todos os produtos deste tipo, que podem pertencer a cadeias produtivas diferentes.

Um Repositório de Produtos é composto pelos seguintes campos:

- `id_reprod`: identifica um determinado produto genérico.
- `nome`: apresenta o nome do produto genérico.
- `descrição genérica`: descreve de forma genérica um determinado produto.

- unidade: indica a unidade de medida de um determinado produto. Por exemplo, na Tabela 3.1, a unidade de medida do produto *Queijo Minas Padrão* é *quilos (kg)*.

A descrição genérica é especializada quando tipos ou lotes de um certo produto são gerados dentro de uma cadeia. A especialização é descrita em uma Tabela de Característica do Produto – vide Tabela 3.7, que detalha as propriedades de um lote de *Queijo Minas Padrão*.

Repositório de Produtos			
id_reprod	nome	descrição generica	unidade
LeiteA	Leite A	Produto Leite do tipo A	Litros
QMP	Queijo Minas Padrão	Produto Queijo	Quilos

Tabela 3.1: Repositório de Produtos.

3.1.2 Repositório de Processos

O Repositório de Processos contém registros que descrevem os processos pelos quais passam os produtos gerados ou transformados ao longo de uma cadeia produtiva. Esses processos são armazenados como workflows, os quais indicam as seqüências de atividades necessárias para a fabricação de um determinado produto. A Tabela 3.2 ilustra o Repositório de Processos para o cenário básico da Figura 3.1.

id_repproc	nome	workflow
ProcFabLeite	Processo de fabricação do Leite	<pre> graph LR Vacas --> Dar_banho[Dar banho] Dar_banho --> Ordenhar[Ordenhar] Ordenhar --> Armazenar_Leite[Armazenar Leite] Armazenar_Leite --> ProdLeite </pre>
ProcFabQueijo	Processo de fabricação do Queijo	<pre> graph LR ProdLeite --> Fabricar_Massa[Fabricar Massa] Fabricar_Massa --> Transportar[Transportar] Transportar --> Enformar[Enformar] Enformar --> ProdQueijo </pre>

Tabela 3.2: Repositório de Processos.

Um Repositório de Processos é composto pelos seguintes campos:

- id_repproc: identifica um determinado processo de fabricação ou transformação.
- nome: indica o nome do processo. Na Tabela 3.2, o processo identificado como “*ProcFabLeite*” possui o nome “*Processo de Fabricação do Leite*”.
- workflow: contém o identificador de um workflow, armazenado em banco de dados de workflows. Um workflow especifica as atividades necessárias para a fabricação de um determinado produto. A Tabela 3.2 apresenta *workflows* simplificados para

os processos de fabricação do leite e do queijo. Esta dissertação não contempla o armazenamento do workflow, por representar um problema em si e não influenciar o trabalho descrito. O trabalho de [34] detalha o armazenamento de workflows em um conjunto de relações de um banco de dados.

3.1.3 Repositório de Serviços

O Repositório de Serviços, ilustrado na Tabela 3.3, contém registros descrevendo os tipos genéricos de serviços realizados em uma cadeia produtiva. O modelo considera apenas dois tipos de serviços: transporte e armazenamento. Este repositório é composto pelos seguintes campos:

- *id_repserv*: identifica um serviço neste repositório, por exemplo, “*Transp*” na Tabela 3.3.
- *nomeserv*: contém o nome do serviço. Este campo identifica os tipos de serviços. Na Tabela 3.3, há dois tipos de serviços (“Armazenamento” e “Transporte”).
- *descrição*: descreve um determinado serviço, por exemplo, o serviço “*Armaz*” é descrito como “*Local para o estoque de produtos*”.

Repositório de Serviços		
<i>id_repserv</i>	<i>nomeserv</i>	<i>descrição</i>
Transp	Transporte	Mover produtos de um local para outro
Armaz	Armazenamento	Local para o estoque de produtos

Tabela 3.3: Repositório de Serviços.

3.1.4 Repositório de Participantes

O Repositório de Participantes, ilustrado na Tabela 3.4, armazena os dados cadastrais de todos os participantes da cadeia produtiva. Neste trabalho estamos consideramos apenas o conjunto mínimo de dados necessários para a rastreabilidade. Este repositório é composto pelos seguintes campos:

- *id_part*: identifica unicamente um determinado participante da cadeia produtiva. Por exemplo, o identificador *Part_Faz1* indica a *Fazenda de Leite1* e *Soft1* indica um *software* como participante (por exemplo, uma aplicação que executa um processo).
- *nome*: indica “quem” é o participante, por exemplo, o integrante identificado por “*Part_Lat1*” tem o nome “Laticínio1”.

- url: indica o endereço em que estão disponíveis os serviços de um participante. Na Tabela, o “*Armazém1*” possui o seguinte endereço *www.armazém1.com.br/servico.wsdl*.
- coordenadas: indica a posição geográfica em que está localizado o participante. Na Tabela 3.4, o participante identificado como *Part_Lat1* corresponde às coordenadas (x_2, y_2) .

Repositório de Participantes			
id_part	Nome	Url	Coordenadas
Part_Faz1	Fazenda de Leite1	www.fazendaleite1.com.br/servico.wsdl	(x_1, y_1)
Part_Transp1	Transp1	www.transp1.com.br/st1.wsdl	(x_{10}, y_{10})
Part_Lat1	Laticinio1	www.laticinio1.com.br/servico.wsdl	(x_2, y_2)
Part_Transp2	Transp2	www.transp2.com.br/st2.wsdl	(x_{11}, y_{11})
Part_Arm1	Armazém1	www.armazém1.com.br/servico.wsdl	(x_3, y_3)
Part_Transp3	Transp3	www.transp3.com.br/st3.wsdl	(x_{12}, y_{12})
Part_Merc1	Mercado1	www.mercado1.com.br/servico.wsdl	(x_4, y_4)
Soft1	Software1	www.software1.com.br/soft1.wsdl	(x_{13}, y_{13})

Tabela 3.4: Exemplo de um Repositório de Participantes.

3.1.5 Repositório de Regras

O Repositório de Regras, ilustrado na Tabela 3.5, armazena as instâncias das regras a serem obedecidas em uma determinada etapa da cadeia produtiva. Este repositório não faz parte do escopo deste trabalho. Dessa forma, será apresentado de forma simples contendo os seguintes campos:

- idrepregras: identifica uma regra.
- nomeregra: contém o nome da regra.

Repositório de Regras	
idrepregras	nomeregra
id_r1	Obedece as condições de higiene
id_r2	Obedece as normas de fabricação do Queijo

Tabela 3.5: Exemplo de um Repositório de Regras.

3.2 Sumários

Esta seção descreve e especifica os sumários (Seção 3.2.1), que são divididos em três tipos: Produtos (Seção 3.2.2), Processos (Seção 3.2.3) e Serviços (Seção 3.2.4).

3.2.1 Visão Geral

Um sumário é uma seqüência de registros que, semelhante a um log de banco de dados, descreve eventos em uma cadeia produtiva. Os sumários podem estar relacionados aos produtos, processos, transportes e armazenamento. Cada evento da cadeia ocasiona a criação de um ou mais registros de sumário.

Bacarin [2] propôs que os sumários fossem divididos em dois tipos: de produtos e de processos. Entretanto, no decorrer desta pesquisa, observou-se a necessidade de três tipos de sumários:

- Sumário de Produtos: acompanha as etapas de transformação, armazenamento e transporte de um produto, desde a sua criação a partir das matérias-primas até o seu consumo final. Os produtos estão sujeitos a várias restrições, por exemplo, legais ou fitosanitários.
- Sumário de Processos: armazena informações sobre o processo de fabricação de um determinado produto. Um processo consiste de uma atividade que transforma um conjunto de insumos/produtos (entradas) em um ou mais produtos distintos (saídas).
- Sumário de Serviços: armazena informações sobre um determinado serviço. Um serviço consiste de alguma atividade que influencia um produto, sem transformá-lo, estando atualmente restritos a transporte ou armazenamento de um produto.

A Figura 3.2 reproduz a Figura 3.1, acrescentando abaixo de cada elemento da cadeia a indicação dos sumários associados e suas ligações. A notação utilizada é a seguinte:

- Os retângulos com sombra representam os Sumários de Processos.
- Os retângulos normais representam os Sumários de Produtos – cada linha de texto simboliza um registro.
- Os retângulos com bordas arredondadas representam os Sumários de Serviços.

O retângulo mais à direita na parte (b) da Figura 3.2 simboliza o estado final do sumário do produto *Queijo*. Este sumário indica que o *Queijo* após ser produzido, foi transportado (*Transp2*), armazenado (*Armazém1*), de novo transportado (*Transp3*)

chegando ao *Mercado1*. Os registros desse sumário estão ligados aos Sumários de Serviços (quatro retângulos arredondados) e processo (*Sum. Proc. do Queijo* – retângulo com sombra). Estes registros são gerados à medida que eventos ocorrem na cadeia – acompanhar, na figura, a evolução do produto *Queijo*, da esquerda para a direita.

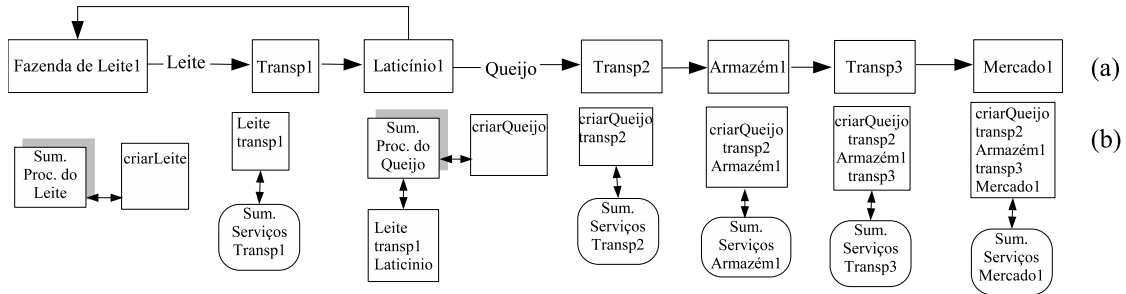


Figura 3.2: Cenário simples da Cadeia Produtiva do Leite.

3.2.2 Sumário de Produtos

A Figura 3.3 apresenta uma visão geral de um registro do Sumário de Produtos, associando cada campo com suas respectivas ligações com outras fontes de dados (registros de repositórios ou de sumários). Os campos de um registro de Sumário de Produtos relacionam-se com:

- um ou mais registros da Tabela de Características do Produto pelo campo “*id_produto*”;
- um registro de um Sumário de Processos ou de um Sumário de Serviços, como é o caso do campo “*processo ou serviço que deu origem ao evento*”.
- um registro do Repositório de Produtos, campo “*rep_prod*”, que indica o produto genérico correspondente.

Um Sumário de Produtos, ilustrado na Tabela 3.6, é composto pelos seguintes campos:

- *id_produto*: é o identificador do produto.
- *evento*: indica alguma etapa da cadeia produtiva. Exemplos de eventos são: criar, transportar, armazenar, embalar, replicar, misturar e consumir.
- *descrição do evento*: descreve o evento. Por exemplo, na Tabela 3.6, o evento “*criar*” para o produto *ProdLeite* corresponde à descrição “*obter leite vindo da Fazenda de Leite1*”.

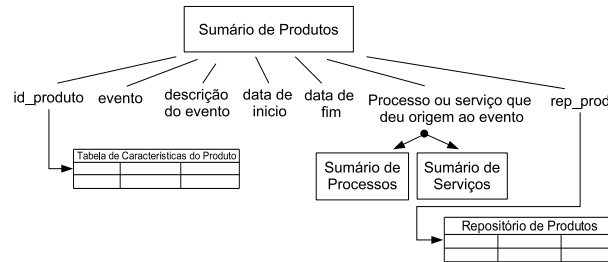


Figura 3.3: Registro de Sumário de Produtos: Visão Geral

SUMÁRIO DE PRODUTO						
id_produto	evento	descrição do evento	data de início	data de fim	processo ou serviço que deu origem ao evento	rep_produtos
ProdLeite	criar	obter leite vindo da fazenda de leite1	10/12/05 07:00	10/12/05 13:00	ProcLeite1	LeiteA
ProdLeite	transportar	transporte de leite da fazenda1 para o laticinio1	10/12/05 13:45	11/12/05 10:05	Stransp1	LeiteA
ProdQueijo	criar	obter queijo do laticinio1	12/12/05 14:00	12/12/05 17:00	ProcQueijo1	QMP
ProdQueijo	transportar	transporte de queijo do laticinio1 para o armazém1	12/12/05 16:40	13/12/05 09:00	Stransp2	QMP
ProdQueijo	armazenar	armazena queijo no armazém1	13/12/05 08:40	17/12/05 09:20	Sarm1	QMP
ProdQueijo	transportar	transporte de queijo do armazém1 para o mercado1	17/12/05 09:00	18/12/05 13:20	Stransp3	QMP
ProdQueijo	armazenar	queijo é armazenado no mercado1	18/12/05 13:00	18/12/05 13:20	Sarm2	QMP
ProdQueijo	Consumir	Queijo é consumido do mercado1	30/12/05 15:00	—	—	QMP

Tabela 3.6: Sumário de Produtos para o Cenário Simples.

- data de início: contém a data e horário de início de um evento de processo ou serviço.
- data de fim: contém a data e horário de fim de um evento de processo ou serviço.
- processo ou serviço que deu origem ao evento: é um identificador que referencia o registro do Sumário de Processos ou Serviços (exemplos, na Tabela 3.6, representados por *ProcLeite1* e *Stransp1*, respectivamente). Quando este campo estiver vazio, indica que a cadeia produtiva para um produto terminou. Isto pode ser visto na última linha da Tabela 3.6.
- rep_prod: é um identificador que aponta para informações genéricas do produto em questão. Essas informações estão armazenadas em um Repositório de Produtos, sendo este especificado na seção 3.1.1.

O Sumário de Produtos é complementado por uma tabela (ilustrada na Tabela 3.7) que define as características de instâncias do produto. Estas características são definidas, à semelhança de RDF (Resource Description Framework), por tuplas $\langle id, nome\ da\ propriedade, valor \rangle$. Na Tabela 3.7, o produto *ProdQueijo* apresenta, por exemplo, no campo “propriedade” a característica *formato* que é interessante para o Queijo e não para o Leite. Esta tabela permite que os registros do Sumário de Produtos, de tamanho fixo, tenham efetivamente tamanho variável. O mesmo tipo de artifício é usado no Sumário de Processos.

Tabela de Características dos Produtos		
id	nome da propriedade	valor
ProdLeite	quantidade	10000
ProdLeite	data de validade	10/06/2006
ProdQueijo	quantidade	2000
ProdQueijo	data de validade	12/02/2006
ProdQueijo	formato	redondo
ProdQueijo	teor de gordura	17,00%

Tabela 3.7: Tabela de Características dos Produtos

3.2.3 Sumário de Processos

A Figura 3.4 apresenta uma visão geral de um registro do Sumário de Processos associando cada campo com suas respectivas ligações com outros registros de repositórios ou de sumários. Os campos de um registro de Sumário de Processos relacionam-se com:

- um ou mais registros da Tabela de Características do Processo pelo campo “*id_processo*”;
- um registro do Repositório de Participantes, campo “*local*”;
- registros de Sumário de Produtos, campos “*input*” e “*output*”;
- registros do Repositório de Regras, campo “*regras*”;
- um registro do Repositório de Processos, em que processos são representados por *workflows*, indicado pelo campo “*detalhes processo*”.

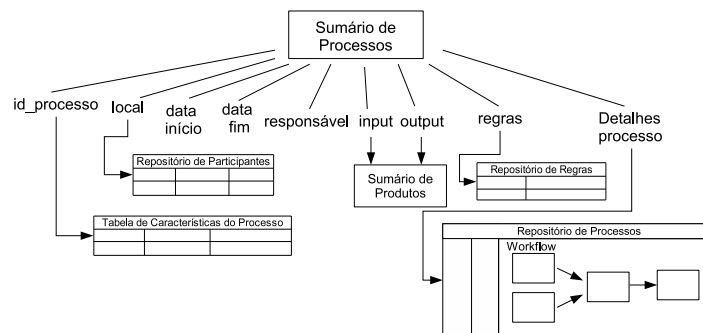


Figura 3.4: Registro de Sumário de Processos: Visão Geral

Um Sumário de Processos, ilustrado na Tabela 3.8, é composto pelos seguintes campos:

- *id_processo*: é um identificador único de uma instância de um processo. A cada execução de um processo gera-se um novo valor de identificação, como o processo de fabricação de *Queijo* (*ProcQueijo1*).

SUMÁRIO DE PROCESSO								
id_processo	local	data de início	data de fim	responsável	input	output	regras	detalhes processo
ProcLeite1	Part_Faz1	10/12/05 07:00	10/12/05 13:00	José	—	ProdLeite	R1	ProcFabLeite
ProcQueijo1	Part_Lat1	12/12/05 14:00	12/12/05 17:00	João	ProdLeite	ProdQueijo	R1, R2	ProcFabQueijo

Tabela 3.8: Sumário de Processos para o Cenário Simples.

- **local:** é um apontador para o Repositório de Participantes e permite determinar o local de fabricação de um determinado produto. Por exemplo, na Tabela 3.8, o *ProcLeite1* é realizado na *Fazenda de Leite1* identificado como *Part_Faz1* no Repositório de Participantes.
- **data de início:** contém a data e horário de início do processo.
- **data de fim:** contém a data e horário de fim do processo.
- **responsável:** identifica o responsável (humano, *software* ou máquina) por um determinado processo de fabricação. Na Tabela 3.8, *João* é o responsável pelo processo de fabricação do *Queijo*.
- **input:** indica as matérias-primas necessárias para a fabricação ou obtenção de um determinado produto. Essas matérias-primas podem referenciar os produtos presentes no Sumário de Produtos por meio do identificador do produto ou este campo pode estar vazio indicando o ponto de inicial da cadeia produtiva. Por exemplo, na Tabela 3.8 o campo “*input*” do processo de Queijo (*ProcQueijo1*) contém o identificador *ProdLeite* que está referenciado no Sumário de Produtos, ilustrado na Tabela 3.6. Se um insumo não foi obtido na cadeia produtiva, mas está sendo utilizado na cadeia, então este deve ser inserido no Sumário de Produtos indicando sua participação na cadeia como um elemento importado de outra cadeia. Por exemplo, frutas utilizadas na fabricação de iogurte podem ser associadas desta forma às suas respectivas cadeias produtivas.
- **output:** contém um ou mais identificadores dos produtos fabricados. Corresponde ao campo *id_produto* no Sumário de Produtos, para o evento “*criação*”. Por exemplo, na Tabela 3.8 o campo “*output*” do processo de Queijo (*ProcQueijo1*) contém o identificador *ProdQueijo* que está referenciado no Sumário de Produtos, como pode ser visto na Tabela 3.6.
- **regras:** referenciam instâncias de normas ou condições que estão sendo seguidas na fabricação de um determinado produto. Estas instâncias estão armazenadas no Repositório de Regras.

- `detalhes_processo`: é um apontador para um registro do Repositório de Processos, especificado na seção 3.1.2, identificando o *workflow* daquele processo.

Para complementar, os processos são detalhados por uma tabela de características, que, à semelhança da Tabela de Características de Produto, possui os campos $\langle id, nome\ da\ propriedade, valor \rangle$. Por exemplo, para o *ProcLeite1* as propriedades relevantes são *ph* e *densidade*.

Tabela de Características dos Processos		
id	nome da propriedade	valor
ProcLeite1	ph	6,65
ProcLeite1	densidade	1,03
ProcQueijo1	ph	5,5
ProcQueijo1	temperatura de aquecimento	65°C

Tabela 3.9: Tabela de Características do Processo

3.2.4 Sumário de Serviços

A Figura 3.5 apresenta uma visão geral de um registro do Sumário de Serviços, associando cada campo com suas respectivas ligações com outros registros de diferentes tabelas. Os campos do Sumário de Serviços relacionam-se com:

- um registro da Tabela de Deslocamento ou com um registro da Tabela de Armazenamento, por meio do campo “*id_serviço*”;
- um registro do Repositório de Participantes, campo “*local*”;
- um registro do Sumário de Produtos, campo “*id_prod*”;
- um registro do Repositório de Serviços, campo “*rep_serviço*”.

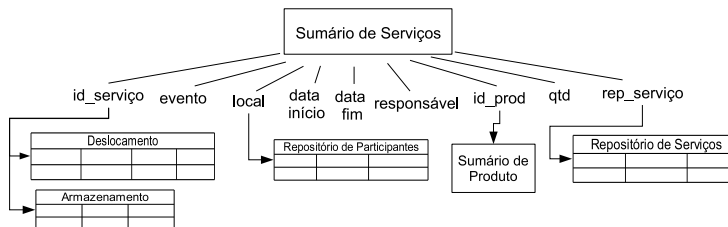


Figura 3.5: Registro de Sumário de Serviços: Visão Geral

Um Sumário de Serviços, ilustrado na Tabela 3.10, é composto pelos seguintes campos:

- *id_serviço*: é um identificador de um detalhamento do serviço, descrito em tabelas associadas (Deslocamento ou Armazenamento). Um serviço é identificado pela combinação de quatro campos: “*id_serviço*”, “*evento*”, “*id_prod*” e “*local*”. Por exemplo, suponha que um serviço de transporte deva fazer entregas de um mesmo produto em dois locais distintos. Desse modo, o Sumário de Serviços deverá ter dois registros com o mesmo identificador de serviço e de produto, porém com o ponto de entrega distinto.
- *evento*: indica a carga ou descarga de um determinado lote.
- *local*: é um apontador que permite determinar, pela ligação com o participante, o local onde o serviço é realizado (via Repositório de Participantes).
- *data de início*: indica a data e horário de início de um serviço.
- *data de fim*: indica a data e horário de fim de um serviço.
- *responsável*: identifica o responsável por este serviço. *Jorge* é o responsável pelo serviço apresentado como *Stransp2*.
- *id_prod*: identifica o produto para o qual o serviço é realizado. Por exemplo, na Tabela 3.10, o serviço apresentado como *Stransp2*, transporta o produto identificado como *ProdQueijo*.
- *qtd*: indica a quantidade manipulada pelo serviço. Por exemplo, no serviço *Sarm1* foram armazenados *2000 kg* do produto *Queijo* e foram retirados (evento *descarga*) *400 kg*.
- *rep_serv*: contém um identificador que aponta para um registro do Repositório de Serviços, especificado na seção 3.1.3.

A descrição de um serviço é complementada pelas Tabelas de Deslocamento (se transporte) ou Armazenamento. A Tabela de Deslocamento (Tabela 3.11) descreve um itinerário. Esta tabela possui os campos: *id_serv*, *descrição*, *itinerário*, *empresa* e *regras*. O campo “*id_serv*” identifica o serviço (campo do Sumário do Serviços). O campo “*descrição*” descreve o serviço. O campo “*itinerário*” contém o trajeto percorrido por um determinado transporte. O campo “*empresa*” identifica a empresa de transporte, referenciando um registro no Repositório de Participantes. O campo “*regras*” referencia instâncias de normas obedecidas pelo transporte durante o percurso inteiro.

A Tabela de Armazenamento (Tabela 3.12) detalha o elemento de armazenamento que efetua um serviço. Esta tabela possui os campos: *id_serv*, *empresa* e *regras*. O campo “*id_serv*” identifica o serviço de armazenamento. O campo “*empresa*” referencia

SUMÁRIO DE SERVIÇOS								
id_serviço	evento	local	data de início	data de fim	responsável	id_prod	qtd	rep_serviço
Stransp1	carga	Part_Faz1	10/12/2005 13:45:00	10/12/2005 14:05:00	José	ProdLeite	10000	Transp
Stransp1	descarga	Part_Lat1	11/12/2005 09:45:00	11/12/2005 10:05:00	José	ProdLeite	10000	Transp
Stransp2	carga	Part_Lat1	12/12/2005 16:40:00	12/12/2005 17:00:00	Jorge	ProdQueijo	2000	Transp
Stransp2	descarga	Part_Arm1	13/12/2005 08:40:00	13/12/2005 09:00:00	Jorge	ProdQueijo	2000	Transp
Sarm1	carga	Part_Arm1	13/12/2005 08:40:00	13/12/2005 09:00:00	Carlos	ProdQueijo	2000	Armaz
Sarm1	descarga	Part_Arm1	17/12/2005 09:00:00	17/12/2005 09:20:00	Carlos	ProdQueijo	400	Armaz
Stransp3	carga	Part_Arm1	17/12/2005 09:00:00	17/12/2005 09:20:00	João	ProdQueijo	400	Transp
Stransp3	descarga	Part_Merc1	18/12/2005 13:00:00	18/12/2005 13:20:00	João	ProdQueijo	400	Transp
Sarm2	carga	Part_Merc1	18/12/2005 13:00:00	18/12/2005 13:20:00	Marta	ProdQueijo	400	Armaz

Tabela 3.10: Sumário de Serviço para o Cenário Simples.

Deslocamento				
id_serv	Descrição	Itinerário	Empresa	Regras
Stransp1	transporte de leite da fazenda1 ao laticínio1	$\{(x_1, y_1, t_1), (x_2, y_2, t_2)\}$	Part_Transp1	id_r1
Stransp2	transporte de queijo do laticínio1 ao armazém1	$\{(x_2, y_2, t_2), (x_3, y_3, t_3)\}$	Part_Transp2	id_r1, id_r2
Stransp3	transporte de queijo do armazém1 ao mercado1	$\{(x_3, y_3, t_3), (x_4, y_4, t_4)\}$	Part_Transp3	id_r3

Tabela 3.11: Tabela de Deslocamento

um registro no Repositório de Participantes indicando a empresa que está realizando o armazenamento. O campo “*regras*” referencia instâncias da norma ou condição obedecida pela instância de um determinado armazenamento.

Armazenamento		
id_serv	empresa	regras
Sarm1	Part_Arm1	id_r1
Sarm2	Part_Merc1	id_r4

Tabela 3.12: Tabela de Armazenamento

3.3 Refinamento de Processos

A Seção 3.2.3 apresenta os registros do Sumário de Processos em uma certa granularidade. O campo “detalhes processo” aponta para um *workflow* que contém detalhes das etapas de um determinado processo. Este *workflow* pode, portanto, referenciar outros processos em um nível mais detalhado e, inclusive, também indicar etapas de transporte e armazenamento. Isto significa que é possível armazenar novos registros nos Sumários de Processos, Produtos e Serviços para refletir este nível mais refinado de granularidade.

Considere o exemplo do *Queijo* da Figura 3.2. Suponha, no entanto, que as atividades “Fabricar Massa” e “Enformar” encapsulam outros *workflows*, como mostrado na Figura 3.6. A figura apresenta o *workflow* para o processo de fabricação do queijo contendo o encapsulamento do processo de fabricação da massa do queijo (*Fabricar Massa*) e o encapsulamento do processo de colocar a massa em fôrmas (*Enformar*), obtendo o *Queijo*.

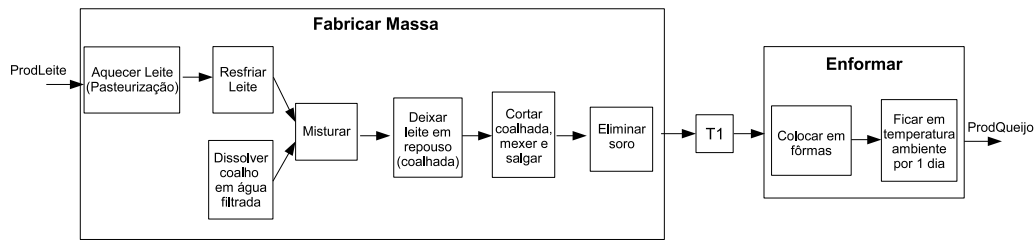


Figura 3.6: Encapsulamento no Processo de Fabricação do Queijo.

O nível de detalhamento dos registros depende dos objetivos dos usuários dos sumários. As etapas mostradas na Figura 3.6 levam à criação dos seguintes registros adicionais:

- No Sumário de Processos, conforme pode ser visto na Tabela 3.13, há a criação de registros para o processo de fabricação da massa do queijo (*ProcMassa*) e para o processo de colocar a massa em fôrmas (*ProcEnf*) obtendo o produto *Queijo*. Observe que o processo de enformagem (*ProcEnf*) possui no campo “output” o identificador *ProdQueijo* indicando a produção do produto.

SUMÁRIO DE PROCESSO								
id_processo	local	data de início	data de fim	responsável	input	output	regras	detalhes processo
ProcLeite1	Part_Faz1	10/12/05 07:00	10/12/05 13:00	José	—	ProdLeite	R1, R5	ProcFabLeite
ProcQueijo1	Part_Lat1	12/12/05 14:00	12/12/05 17:00	João	ProdLeite	ProdQueijo	R1, R2	ProcFabQueijo
Procmassa	Part_Fábrica	11/12/05 13:00	11/12/05 14:30	Marta	ProdLeite	Pmassa	R1, R3	ProcFabMassa
ProcEnf	Empresa de Enformagem	11/12/05 14:50	11/12/05 17:30	Maria	Pmassa	ProdQueijo	R1, R4	ProcEnformagem

Tabela 3.13: Sumário de Processos Modificado para o Cenário Simples.

- No Sumário de Produtos, conforme pode ser visto na Tabela 3.14, houve a inserção de várias entradas com o identificador *PMassa* indicando que a massa foi obtida, transportada e enformada.
- No Sumário de Serviços, conforme pode ser visto na Tabela 3.15, foi adicionado o serviço de transporte identificado por *St1* indicando o transporte da massa para a enformagem.

SUMÁRIO DE PRODUTO						
id_produto	evento	descrição	data de início	data de fim	processo ou serviço que deu origem ao evento	rep_produtos
ProdLeite	criar	obter leite vindo da fazenda de leite	10/12/05 07:00	10/12/05 13:00	ProcLeite1	LeiteA
ProdLeite	transportar	transporte de leite da fazenda para o laticínio	10/12/05 13:45	11/12/05 10:05	Stransp1	LeiteA
PMassa	criar	obter massa do queijo	11/12/05 10:06	11/12/05 11:06	ProcMassa	QMP
PMassa	transportar	transporte da massa da fábrica para a empresa de enformagem	11/12/05 11:06	11/12/05 15:00	ST1	QMP
PMassa	enformar	colocar a massa em fôrmas	11/12/05 15:00	11/12/05 17:00	ProcEnf	QMP
ProdQueijo	criar	obter queijo do laticínio	12/12/05 14:00	12/12/05 17:00	ProcQueijo1	QMP
ProdQueijo	transportar	transporte de queijo do laticínio para o armazém1	12/12/05 16:40	13/12/05 09:00	Stransp2	QMP
ProdQueijo	armazenar	armazena queijo no armazém1	13/12/05 08:40	17/12/05 09:20	Sarm1	QMP
ProdQueijo	transportar	transporte de queijo do armazém1 para o mercado1	17/12/05 09:00	18/12/05 13:20	Stransp3	QMP
ProdQueijo	armazenar	queijo é armazenado no mercado1	18/12/05 13:00	18/12/05 13:20	Sarm2	QMP
ProdQueijo	Consumir	Queijo é consumido do mercado1	30/12/05 15:00	—	—	QMP

Tabela 3.14: Sumário de Produtos Modificado para o Cenário Simples.

SUMÁRIO DE SERVIÇOS								
id_serviço	evento	local	data de início	data de fim	responsável	id_prod	qtd	rep_serviço
Stransp1	carga	Part_Faz1	10/12/2005 13:45:00	10/12/2005 14:05:00	José	ProdLeite	10000	Transp
Stransp1	descarga	Part_Lat1	11/12/2005 09:45:00	11/12/2005 10:05:00	José	ProdLeite	10000	Transp
ST1	carga	Part_Lat1	11/12/2005 14:30:00	11/12/2005 14:40:00	Mário	Pmassa	2000	Transp
ST1	descarga	Part_Lat1	11/12/2005 14:40:00	11/12/2005 14:50:00	Mário	Pmassa	2000	Transp
Stransp2	carga	Part_Lat1	12/12/2005 16:40:00	12/12/2005 17:00:00	Jorge	ProdQueijo	2000	Transp
Stransp2	descarga	Part_Arm1	13/12/2005 08:40:00	13/12/2005 09:00:00	Jorge	ProdQueijo	2000	Transp
Sarm1	carga	Part_Arm1	13/12/2005 08:40:00	13/12/2005 09:00:00	Carlos	ProdQueijo	2000	Armaz
Sarm1	descarga	Part_Arm1	17/12/2005 09:00:00	17/12/2005 09:20:00	Carlos	ProdQueijo	400	Armaz
Stransp3	carga	Part_Arm1	17/12/2005 09:00:00	17/12/2005 09:20:00	João	ProdQueijo	400	Transp
Stransp3	descarga	Part_Merc1	18/12/2005 13:00:00	18/12/2005 13:20:00	João	ProdQueijo	400	Transp
Sarm2	carga	Part_Merc1	18/12/2005 13:00:00	18/12/2005 13:20:00	Marta	ProdQueijo	400	Armaz

Tabela 3.15: Sumário de Serviços Modificado para o Cenário Simples.

3.4 Visão Integrada dos Registros de Repositórios e Sumários

A figura 3.7 ilustra todos os sumários, repositórios e tabelas adicionais para o cenário simples, mostrando uma melhor visão do relacionamento entre eles.

Considere um produto genérico - Queijo Minas Padrão - registrado no Repositório de Produtos (topo da figura 3.7, à direita) com $id_repprod = QMP$.

O Sumário de Produtos mostra na terceira tupla que o produto *ProdQueijo* foi criado a partir do processo *ProcQueijo1*. Do lado direito deste sumário, na figura, estão propriedades adicionais do produto *ProdQueijo* que trata de uma instância do produto do tipo QMP (Queijo Minas Padrão), produzido na quantidade 2000, unidade quilos (devido ao tipo QMP) e com características de data e formato. O processo *ProcQueijo1* é detalhado na segunda tupla do Sumário de Processos, com entradas, saída, local do processo e identificador do *workflow* que detalha seus processos, PFQMP, especificado no registro

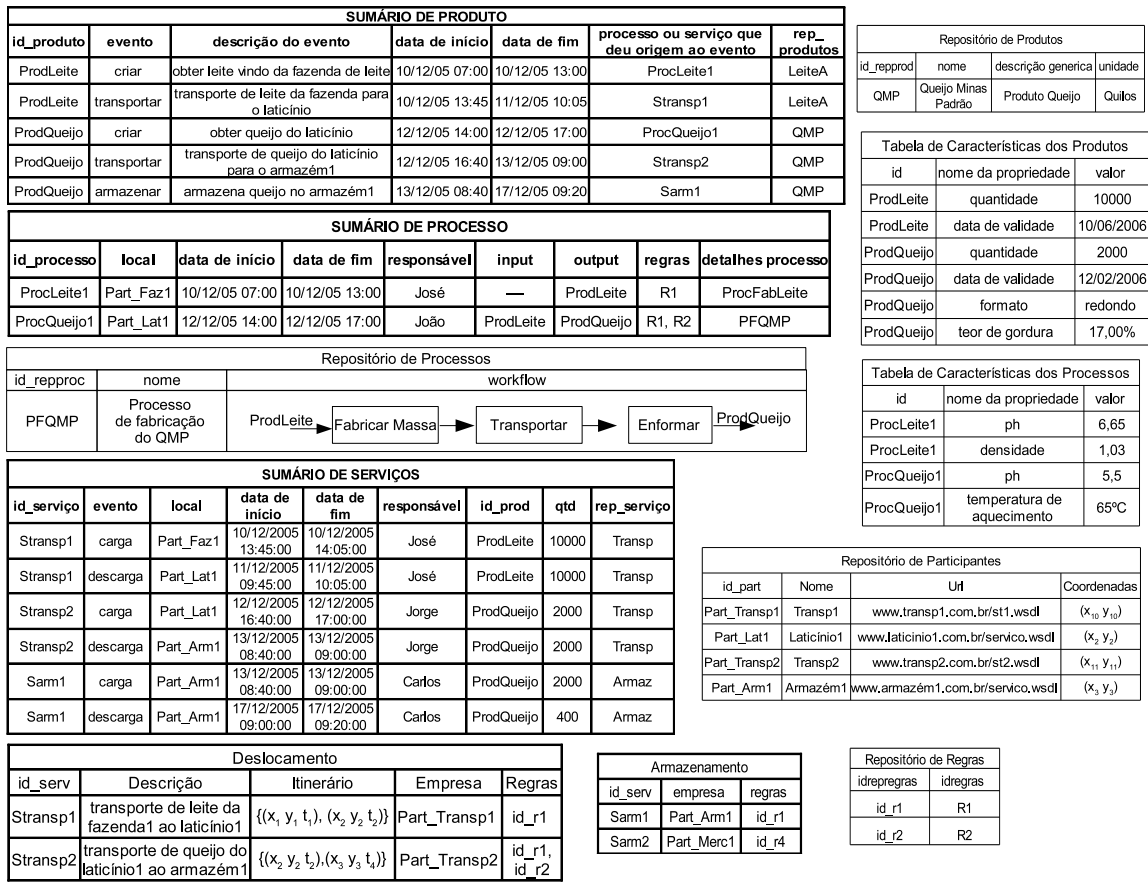


Figura 3.7: Sumários e Repositórios – Visão Global

do Repositório de Processos.

Esta instância de QMP passa por um evento de transporte, *Stransp2*, como mostra a quarta tupla do Sumário de Produtos. O Sumário de Serviços mostra que *Stransp2* teve eventos de carga e descarga do produto *ProdQueijo* e este serviço de transporte possui um itinerário na Tabela de Deslocamento. Nesta tabela, pode-se notar a empresa que realiza este serviço *Part_Transp2*. Além disso, este serviço possui instância de regras (*id_r1*, *id_r2*) que são obedecidas pela empresa no itinerário percorrido ((x_2 y_2 t_2), (x_3 y_3 t_3)). As informações da empresa e das regras estão no Repositório de Participantes e de Regras respectivamente.

A instância de QMP também passa por um evento de armazenamento, *Sarm1*, como pode ser visto na quinta tupla do Sumário de Produtos. No Sumário de Serviços, *Sarm1* tem eventos de carga e descarga do produto no armazém. Entretanto, devido ao tamanho da figura, só está sendo apresentada parte dos sumários. Este serviço de armazenamento possui suas características específicas na Tabela de Armazenamento e a empresa

responsável por este serviço é *Part_Arm1* que é descrita no Repositório de Participantes.

3.5 Utilizando o Georreferenciamento

O modelo requer o uso de coordenadas geográficas em diferentes repositórios e registros de sumários. Com isso, várias das questões em aberto em rastreabilidade podem ser respondidas usando junções espaciais. Por exemplo, o registro do Sumário de Processos contém o identificador da localização da execução daquele processo e conseqüentemente acesso aos dados ambientais. Combinado ao Repositório de Participantes, cujos registros são georreferenciados, permite obter informações cadastrais sobre a empresa responsável pelo processo.

O aspecto mais complicado envolve situações de transporte, considerado um tipo de serviço. Neste caso, certas consultas envolvem junções espaço-temporais. Cada serviço de transporte é considerado sob dois pontos de vista: descrição do serviço propriamente dito, na Tabela *Deslocamento*, que contém o itinerário completo de algum meio de transporte; e informações sobre cada evento ocorrido durante aquele serviço de Transporte. Cada evento gera registros nos Sumários de Serviços e de Produtos. Envolve ainda os seguintes elementos do modelo: Repositório de Participantes (cadastro da empresa de transportes), de Produtos (produto transportado), de Regras (condições para o transporte), de Serviços (descrição do tipo de serviço).

Considere o exemplo de um caminhão que descarrega parte de um lote em $(x_1 y_1 t_1)$ e o restante em $(x_2 y_2 t_2)$. Seu itinerário (Tabela de Deslocamento) contém uma seqüência $((x_i y_i t_i))$, não necessariamente especificando explicitamente os pontos $(x_1 y_1)$ e $(x_2 y_2)$, pois pode ter acontecido algum problema com o caminhão, não passando pelo segundo ponto. Já o Sumário de Serviços contém dois registros, indicando que aquele caminhão teve dois eventos de serviço nos instantes t_1 e t_2 nos locais com aquelas coordenadas. Dada uma unidade do produto, o consumidor pode rastrear a que lote pertencia e onde foi descarregada (registros interligados de Sumários de Produtos e Serviços). Pela ligação espaço-temporal com o itinerário de deslocamento, pode-se determinar se as regras do serviço foram obedecidas para aquela unidade, detectando desta forma problemas de qualidade.

3.6 Expandindo o Cenário Básico da Cadeia Produtiva do Leite

Há vários cenários adicionais ao cenário básico da Figura 3.1. A cada mudança são adicionados novos registros nos sumários. Exemplos são cadeias em que mais de um produto

é gerado, criando bifurcações na especificação; cadeias com múltiplos fornecedores; ou com vários pontos de carga/descarga de um produto. Cada um desses cenários gera problemas para a rastreabilidade como:

- a fabricação de mais de um produto acarreta o armazenamento de dois registros no Sumário de Produtos, um para cada produto.
- a utilização de vários fornecedores ou vários locais de carregamento de produtos em um determinado meio de transporte pode acarretar na mistura dos produtos. Em alguns, a mistura não pode ser desfeita dificultando a rastreabilidade. Esta mistura leva à geração de um processo fictício de mistura de produtos.
- os vários locais de descarregamento de produtos realizado por um determinado transporte levam à geração do processo fictício de replicação, em que os produtos são divididos em lotes que irão para locais diferentes.

A solução de criação de processos fictícios foi usada para facilitar o desdobramento de lotes ou sua fusão. Tais processos tem como entrada os produtos a serem fundidos e a saída é a fusão, ou, alternativamente, um lote a ser desdobrado, tendo como saída sublotes. A Figura 3.8 ilustra essas situações e indica alguns registros dos sumários.

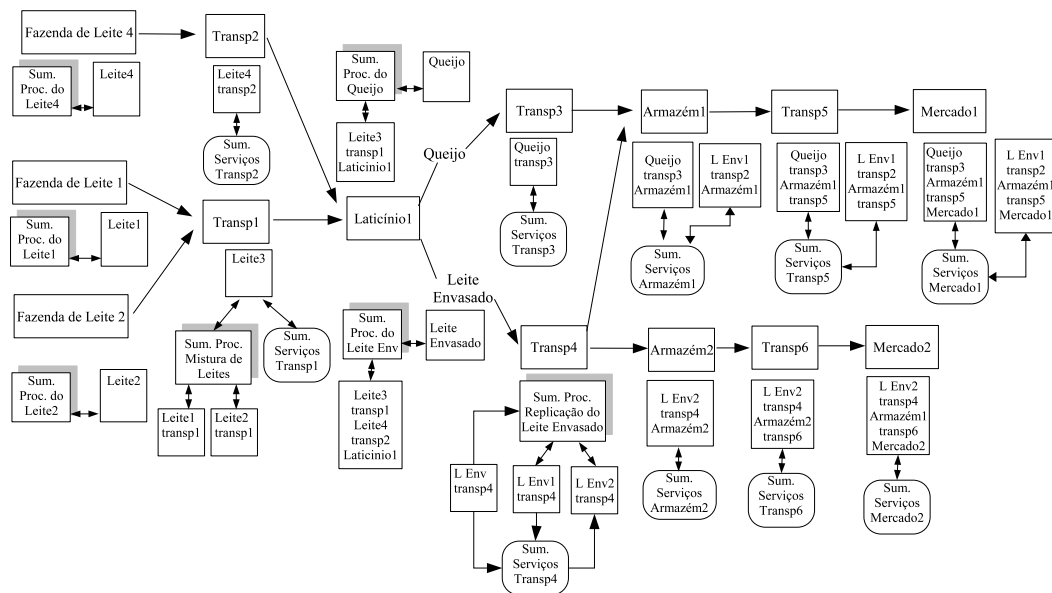


Figura 3.8: Cenário Exemplo.

O cenário da Figura 3.8 gera mais de um produto no *Laticínio1*: *Queijo* e *Leite Envasado*. Para a produção de *Queijo*, obtém-se leite vindo de dois fornecedores (*Fazenda*

de Leite1 e Fazenda de Leite2) e para a produção de Leite Envasado obtém-se leite de três fornecedores (os mesmos do Queijo mais a Fazenda de Leite4). Isto pode ser visto no Sumário de Processos (Tabela 3.17) em que o campo *input* para o *ProcQueijo1* contém *ProdLeite3* que é a união dos leites das Fazendas 1 e 2. Já o *ProcLeiteEnv1* contém como ingredientes *ProdLeite3* e *ProdLeite4*.

A situação de múltiplos fornecedores é ainda mais complicada quando há mistura de produtos, por exemplo, quando vários fornecedores de leite combinam seus produtos com a finalidade de contribuir para a fabricação de um único lote de queijo.

O problema de carga de um mesmo tipo de produto em lugares diferentes aparece em *Transp1* no carregamento de leite das fazendas (Fazendas de Leite 1 e 2). Esta carga pode misturar os produtos. Para a rastreabilidade desta mistura, no Sumário de Processos (Tabela 3.17), gera-se um processo artificial (*MisturaLeite*) que possui como entrada (campo “*input*”) os produtos a serem misturados (*ProdLeite1* e *ProdLeite2*) e como saída (campo “*output*”) um novo produto (*ProdLeite3*). Este artifício – eventos correspondentes a processos fictícios – resolve questões complexas relativas à rastreabilidade quando há misturas.

O problema de múltiplos pontos de descarga ocorre em *Transp4*, com entrega no *Armazém1* e *Armazém2*. Isto leva à geração de registros identificados como *Stransp4*, no Sumário de Serviços (Tabela 3.18). Pode-se observar que para este identificador há dois eventos de carga e descarga, significando que o *ProdLeiteEnv* foi carregado no *Laticínio1* ($x_3 y_3$) e descarregado no *Armazém1* ($x_5 y_5$) e *Armazém2* ($x_6 y_6$). Dessa forma, ocorreu um evento fictício de replicação, que a partir do *ProdLeiteEnv* gerou-se dois produtos *ProdLeiteEnv1* e *ProdLeiteEnv2*. Além disso, no Sumário de Produtos (Tabela 3.16) há dois registros de armazenamento (para o *ProdLeiteEnv1* e *ProdLeiteEnv2*), um para cada armazém. No Sumário de Serviços, também são adicionados registros, *Sarm1* e *Sarm2*, com eventos de carga e descarga do *ProdLeiteEnv1* e *ProdLeiteEnv2* em seus respectivos armazéns.

A multiplicidade de pontos de carga e descarga exige o gerenciamento de diferentes granularidades de um produto. Por exemplo, um caminhão pode transportar e descarregar lotes do mesmo produto em diversos locais. Cada um desses lotes deve possuir seu respectivo sumário, permitindo o rastreamento do produto ao longo da cadeia. Na ponta do consumo, a granularidade pode descer ao nível mais fino (pacote pertencente ao lote). Desse modo, é necessário que um pacote de leite possa ser rastreado de volta para um lote, até chegar a um tonel de uma cooperativa de gado leiteiro. No itinerário do transporte do lote, alguns incidentes podem ocorrer de forma a causar problemas para parte do lote. Por exemplo, um caminhão pode descarregar parte de um lote em $(x_1 y_1 t_1)$ e o restante em $(x_2 y_2 t_2)$. Neste percurso, entre $x_1 y_1$ (lote sem problemas) e $x_2 y_2$ (resto do lote) condições de funcionamento elétrico podem danificar o resto do carregamento. Se o

SUMÁRIO DE PRODUTOS						
id_produto	evento	descrição do evento	data de início	data de fim	processo ou serviço que deu origem ao evento	rep_produtos
ProdLeite1	criar	obter leite vindo da fazenda de leite1	10/12/05 07:00	10/12/05 13:00	ProcLeite1	LeiteA
ProdLeite1	transportar	transporte de leite da fazenda1 para a fazenda2	10/12/05 13:45	11/12/05 10:05	Stransp1	LeiteA
ProdLeite2	criar	obter leite vindo da fazenda de leite 2	11/12/05 07:00	11/12/05 10:00	ProcLeite2	LeiteA
ProdLeite3	misturar	mistura leites da fazenda1 e fazenda2	11/12/05 14:00	11/12/05 16:00	MisturaLeite	LeiteA
ProdLeite3	transportar	transporte de leite da fazenda2 para o laticínio1	11/12/05 10:45	12/12/05 08:05	Stransp1	LeiteA
ProdLeite4	criar	obter leite vindo da fazenda de leite4	11/12/05 09:40	11/12/05 16:00	ProcLeite4	LeiteA
ProdLeite4	transportar	transporte de leite vindo da fazenda de leite4	11/12/05 14:40	12/12/05 09:00	Stransp2	LeiteA
ProdQueijo	criar	obter queijo do laticínio1	12/12/05 10:00	12/12/05 14:30	ProcQueijo1	QMP
ProdQueijo	transportar	transporte de queijo do laticínio1 para o armazém1	13/12/05 09:40	14/12/05 07:20	Stransp3	QMP
ProdLeiteEnv	criar	obter leite envasado do laticínio1	12/12/05 11:00	12/12/05 12:45	ProcLeiteEnv1	LeiteA
ProdLeiteEnv	transportar	transporte de leite envasado do laticínio1 para o armazém1	13/12/05 07:00	13/12/05 17:30	Stransp4	LeiteA
ProdLeiteEnv1	replicação	replicar sumário do Leite Envasado	13/12/05 14:00	13/12/05 14:30	RepLeiteEnv	LeiteA
ProdLeiteEnv2	replicação	replicar sumário do Leite Envasado	13/12/05 14:00	13/12/05 14:30	RepLeiteEnv	LeiteA
ProdLeiteEnv2	transportar	transporte de leite envasado2 do armazém1 ao armazém2	13/12/05 17:30	14/12/05 07:40	Stransp4	LeiteA
ProdQueijo	armazenar	armazena queijo no armazém1	14/12/05 07:00	17/12/05 09:20	Sarm1	QMP
ProdLeiteEnv1	armazenar	armazena leite envasado1 no armazém1	13/12/05 17:20	17/12/05 09:40	Sarm1	LeiteA
ProdLeiteEnv2	armazenar	armazena leite envasado2 no armazém2	14/12/05 07:20	19/12/05 09:40	Sarm2	LeiteA
ProdQueijo	transportar	transporte de queijo do armazém1 para o mercado1	17/12/05 09:00	18/12/05 10:40	Stransp5	QMP
ProdLeiteEnv1	transportar	transporte de leite envasado1 do armazém1 para o mercado1	17/12/05 09:20	18/12/05 11:00	Stransp5	LeiteA
ProdLeiteEnv2	transportar	Transporte de leite envasado2 do armazém2 para o mercado2	19/12/05 09:20	20/12/05 09:40	Stransp6	LeiteA
ProdQueijo	armazenar	queijo é armazenado no mercado1	18/12/05 10:20	18/12/05 10:40	Sarm3	QMP
ProdLeiteEnv1	armazenar	leite envasado1 é armazenado no mercado1	18/12/05 10:40	18/12/05 11:00	Sarm3	LeiteA
ProdLeiteEnv2	armazenar	leite envasado2 é armazenado no mercado2	20/12/05 09:20	20/12/05 09:40	Sarm4	LeiteA
ProdQueijo	Consumir	Queijo é consumido do mercado1	15/02/06 10:40	—	—	QMP
ProdLeiteEnv1	Consumir	Leite Envasado1 consumido do mercado1	16/03/06 15:00	—	—	LeiteA
ProdLeiteEnv2	Consumir	Leite Envasado2 consumido do mercado2	17/03/06 18:00	—	—	LeiteA

Tabela 3.16: Sumário de Produtos para o Cenário Exemplo.

SUMÁRIO DE PROCESSO								
id_processo	local	data de início	data de fim	responsável	input	output	regras	detalhes processo
ProcLeite1	Part_Faz1	10/12/05 07:00	10/12/05 13:00	José	—	ProdLeite1	R1, R5	ProcFabLeite1
ProcLeite2	Part_Faz2	11/12/05 07:00	11/12/05 12:00	Marcia	—	ProdLeite2	R1, R4	ProcFabLeite2
MisturaLeite	Part_Faz2	11/12/05 14:00	11/12/05 16:00	Josué	ProdLeite1 e ProdLeite2	ProdLeite3	—	ProcMistura
ProcLeite4	Part_Faz4	11/12/05 09:40	11/12/05 14:10	Marcia	—	ProdLeite4	R1, R5	ProcFabLeite4
ProcQueijo1	Part_Lat1	12/12/05 10:00	12/12/05 14:30	João	ProdLeite3	ProdQueijo	R1, R2	ProcFabQueijo
ProcLeiteEnv1	Part_Lat1	12/12/05 11:00	12/12/05 12:45	Maria	ProdLeite3 e ProdLeite4	ProdLeiteEnv	R1, R3	ProcFabLeiteEnv
RepLeiteEnv	Part_Am1	13/12/05 14:00	13/12/05 14:30	Julio	ProdLeiteEnv	ProdLeiteEnv1 e ProdLeiteEnv2	R1, R2	ProcReplicação

Tabela 3.17: Sumário de Processos para o Cenário Exemplo.

consumidor constatar deterioração, isto não significa que todo o lote deve ser descartado, mas somente a parte que seguiu o trajeto de (x_1, y_1) a (x_2, y_2) .

3.7 Conclusões

Este capítulo apresentou o modelo de rastreabilidade proposto para cadeia produtivas, que envolve Sumários de Processos, Produtos e Serviços. O gerenciamento dos dados de rastreabilidade é realizado via armazenamento de registros em repositórios e sumários. Os registros de sumários caracterizam-se por serem interligados, estando associados a estruturas adicionais de dados que permitem um número variável de campos para cada registro. O capítulo a seguir apresenta a arquitetura que suporta este modelo.

SUMÁRIO DE SERVIÇOS								
id_serviço	evento	local	data de início	data de fim	responsável	id_prod	qtd	rep_serviço
Stransp1	carga	Part_Faz1	10/12/2005 13:45:00	10/12/2005 14:05:00	José	ProdLeite1	3000	Transp
Stransp1	descarga	Part_Lat1	11/12/2005 09:45:00	11/12/2005 10:05:00	José	ProdLeite1	3000	Transp
Stransp1	carga	Part_Faz2	11/12/2005 10:45:00	11/12/2005 11:05:00	José	ProdLeite3	5000	Transp
Stransp1	descarga	Part_Lat1	12/12/2005 07:45:00	12/12/2005 08:05:00	José	ProdLeite3	5000	Transp
Stransp2	carga	Part_Faz4	11/12/2005 14:40:00	11/12/2005 15:00:00	Jorge	ProdLeite4	2000	Transp
Stransp2	descarga	Part_Lat1	12/12/2005 08:40:00	12/12/2005 09:00:00	Jorge	ProdLeite4	2000	Transp
Stransp3	carga	Part_Lat1	13/12/2005 09:40:00	13/12/2005 10:00:00	Marcos	ProdQueijo	2000	Transp
Stransp3	descarga	Part_Arm1	14/12/2005 07:00:00	14/12/2005 07:20:00	Marcos	ProdQueijo	2000	Transp
Stransp4	carga	Part_Lat1	13/12/2005 07:00:00	13/12/2005 07:20:00	Mateus	ProdLeiteEnv	4000	Transp
Stransp4	descarga	Part_Arm1	13/12/2005 17:20:00	13/12/2005 17:30:00	Mateus	ProdLeiteEnv	4000	Transp
Stransp4	carga	Part_Lat1	13/12/2005 17:30:00	13/12/2005 17:40:00	Mateus	ProdLeiteEnv2	2000	Transp
Stransp4	descarga	Part_Arm2	14/12/2005 07:20:00	14/12/2005 07:40:00	Mateus	ProdLeiteEnv2	2000	Transp
Sarm1	carga	Part_Arm1	13/12/2005 17:20:00	13/12/2005 17:40:00	Carlos	ProdLeiteEnv1	2000	Armaz
Sarm1	carga	Part_Arm1	14/12/2005 07:00:00	14/12/2005 07:20:00	Carlos	ProdQueijo	2000	Armaz
Sarm2	carga	Part_Arm2	14/12/2005 07:20:00	14/12/2005 07:40:00	Tadeu	ProdLeiteEnv2	2000	Armaz
Sarm1	descarga	Part_Arm1	17/12/2005 09:00:00	17/12/2005 09:20:00	Carlos	ProdQueijo	400	Armaz
Sarm1	descarga	Part_Arm1	17/12/2005 09:20:00	17/12/2005 09:40:00	Carlos	ProdLeiteEnv1	1000	Armaz
Stransp5	carga	Part_Arm1	17/12/2005 09:00:00	17/12/2005 09:20:00	João	ProdQueijo	400	Transp
Stransp5	carga	Part_Arm1	17/12/2005 09:20:00	17/12/2005 09:40:00	João	ProdLeiteEnv1	1000	Armaz
Sarm2	descarga	Part_Arm2	19/12/2005 09:20:00	19/12/2005 09:40:00	Tadeu	ProdLeiteEnv2	1500	Armaz
Stransp6	carga	Part_Arm2	19/12/2005 09:20:00	19/12/2005 09:40:00	Jeremias	ProdLeiteEnv2	1500	Transp
Stransp5	descarga	Part_Merc1	18/12/2005 10:20:00	18/12/2005 10:40:00	João	ProdQueijo	400	Transp
Stransp5	descarga	Part_Merc1	18/12/2005 10:40:00	18/12/2005 11:00:00	João	ProdLeiteEnv1	1000	Transp
Stransp6	descarga	Part_Merc2	20/12/2005 09:20:00	20/12/2005 09:40:00	Jeremias	ProdLeiteEnv2	1500	Transp
Sarm3	carga	Part_Merc1	18/12/2005 10:20:00	18/12/2005 10:40:00	Josué	ProdQueijo	400	Armaz
Sarm3	carga	Part_Merc1	18/12/2005 10:40:00	18/12/2005 11:00:00	Josué	ProdLeiteEnv1	1000	Armaz
Sarm4	carga	Part_Merc2	20/12/2005 09:20:00	20/12/2005 09:40:00	Marta	ProdLeiteEnv2	1500	Armaz

Tabela 3.18: Sumário de Serviços para o Cenário Exemplo.

Capítulo 4

Arquitetura

Este capítulo descreve a arquitetura proposta para o modelo de rastreabilidade, baseada em serviços Web. A Seção 4.1 apresenta uma visão geral da arquitetura proposta. A Seção 4.2 descreve alguns exemplos de interação entre os serviços *Web*. As conclusões deste capítulo são apresentadas na Seção 4.3.

4.1 Visão Geral

A Figura 4.1 mostra a arquitetura básica especificada, que é composta de três camadas: Gerente de Sumários, Serviços de Sumários e Serviços de Repositórios. Os componentes de cada camada estão encapsulados por serviços *Web*.

A figura mostra que os serviços estão organizados em níveis de complexidade. As setas mostram as possibilidades de invocação de serviços. No nível mais baixo, encontram-se os serviços para acesso e atualização dos repositórios-base. Supõe-se que cada repositório esteja sob responsabilidade de um serviço específico, para facilitar questões de gerenciamento de distribuição. No nível intermediário, encontram-se os serviços responsáveis por gerenciar sumários e tabelas associadas. Cada tipo de sumário está encapsulado por um serviço específico. Vale notar que os serviços de sumários podem fazer solicitações entre si. Finalmente, no nível mais alto está a interface do serviço do gerente de sumários, SMWS, um serviço que repassa pedidos para os serviços subjacentes, combina os resultados e os retorna.

A figura mostra que há dois tipos de acesso externo à arquitetura. O primeiro, no nível mais alto, é feito pela interface SM (Summary Manager) do Gerente de Sumários. O acesso a esta interface é realizado por aplicações-cliente, Gerente de Coordenação ou Gerente de Sumários. Os pedidos solicitados por aplicações-cliente se referem a consultas de rastreabilidade, sendo o alvo principal da dissertação. Cada evento ocorrido na cadeia gera um pedido para armazenamento dos registros correspondentes. O Gerente de Coordenação

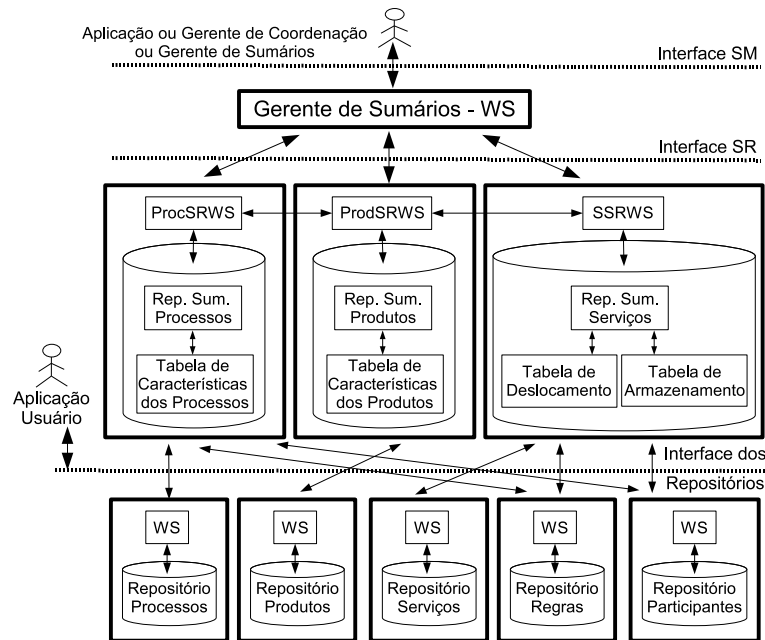


Figura 4.1: Arquitetura Proposta.

e o Gerente de Sumários fazem parte da arquitetura de Bacarin [2]. Entretanto, [2] não especifica estes elementos da arquitetura, apenas indica que o Gerente de Coordenação é responsável por coordenar a execução de uma cadeia produtiva e o Gerente de Sumários controla o acesso ao Repositório de Sumários.

O segundo nível de acesso ocorre no armazenamento de dados ou consultas aos repositórios-base (Processos, Produtos, Serviços, Regras e Participantes), sendo realizado por aplicações-cliente ou pelos serviços de sumários.

A interface de ProdSRWS (*Product Summary Repository Web Service*) aceita requisições vindas do serviço Web do Gerente de Sumários (SMWS). Além disso, interage com SSRWS (*Service Summary Repository Web Service*) e ProcSRWS (*Process Summary Repository Web Service*) enviando pedidos ou fornecendo respostas a estes serviços. Em outras palavras, o serviço Web do Repositório de Sumário de Produtos representa a principal peça no armazenamento e processamento dos dados de rastreabilidade. Os outros dois serviços de repositórios de sumários (ProcSRWS e SSRWS) não se comunicam diretamente, trocando informações e pedidos somente por intermédio do serviço do Gerente de Sumários, ProdSRWS e com alguns serviços da camada de Repositórios. Esta decisão de projeto foi tomada porque o Sumário de Produtos centraliza o armazenamento dos eventos e os Sumários de Processos e Serviços são usuários do Sumário de Produtos.

O dados de rastreabilidade são gerenciados pelo SMWS. Com a arquitetura distribuída, há vários SMWS interagindo entre si, controlando o acesso aos serviços disponibilizados

pelos Repositórios de Sumários de Produtos, de Processos e de Serviços.

A distribuição gera problemas relacionados à autorização de acesso e replicação, os quais estão fora do escopo deste trabalho. Para este trabalho, é suficiente supor que as interações realizadas via Gerente de Sumário estão de acordo com as chamadas restrições de domínio da rede. Para que dois Repositórios de Sumários estejam em um mesmo domínio, é necessário que eles possuam autorizações de acesso mútuo. A interação entre repositórios de sumários pertencentes a domínios diferentes é realizada através da troca de mensagens entre Gerentes de Sumários, como ilustrada na Figura 4.2. Neste caso, um serviço de Repositório de Sumário de Produto comunica-se com seu respectivo Gerente de Sumário (SMWS1) requisitando uma consulta em um repositório de outro domínio (*Domain 2*). Os gerentes (SMWS1 e SMWS2) dos dois domínios interagem e a consulta é executada sobre o Repositório de Sumários do domínio 2. A resposta faz o caminho inverso chegando ao emissor da consulta (Repositório de Sumário de Produto do domínio 1).

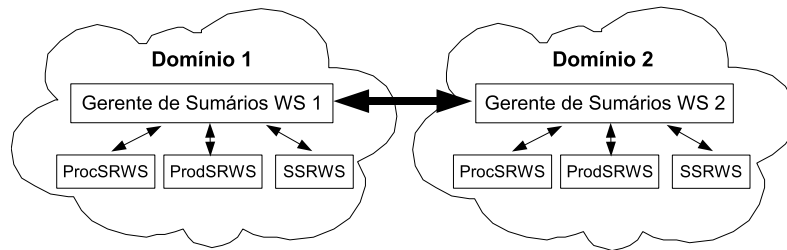


Figura 4.2: Exemplo de interação entre domínios.

Atualização, inserção e remoção também requerem interações entre repositórios de sumários e alguns outros repositórios da camada mais baixa. Por exemplo, a inserção de um registro no Repositório de Sumário de Produto requer comunicações com sumários de Processo e Serviços para garantir a existência de registros de processo ou serviço.

O uso de serviços Web para a comunicação entre gerentes de sumários e os vários repositórios facilita o gerenciamento de dados de rastreabilidade e provê a interoperabilidade para ambientes heterogêneos e distribuídos.

4.2 Exemplos de Interação

Para ilustrar as interações entre os serviços Web da arquitetura, mostraremos exemplos. Um primeiro exemplo apresenta as interações entre os Repositórios de Sumários de Processos e de Produtos. Considere que um pedido P1 foi feito para o SMWS1 com o intuito de encontrar quando ocorreu a produção do produto Queijo1. Esta consulta é realizada através dos elementos da arquitetura da seguinte forma. SMWS1 é responsável

por interagir com três serviços do repositório de sumários, ProcSRWS1, ProdSRWS1 e SSRWS1. Para a realização da consulta, SMWS1 envia um pedido ao ProdSRWS1 com os parâmetros de entrada “*id_produto = Queijo1*” e “*evento = criação*”, requisitando como saída uma data de fabricação. Inicialmente, ProdSRWS1 encontra a instância do processo que foi responsável pelo evento de criação do Queijo1 (suponha “*id_processo = proc_queijo1*”). A seguir, ProdSRWS1 envia um pedido ao ProcSRWS1 para encontrar a data de fabricação do “*proc_queijo1*”, ou seja, a data para responder R1.

Outro exemplo é baseado na seguinte consulta: “encontre o nome da empresa que realizou o transporte de um produto P do laticínio situado em (x_1y_1) ao armazém situado em (x_2y_2) e as datas de início e fim deste evento de transporte”. Suponha que uma aplicação usuário requisitou esta consulta ao SMWS. A Figura 4.3 apresenta o diagrama de seqüência para esta consulta. O SMWS interage com o SSRWS (serviço do Repositório de Sumário de Serviços) que encontra o identificador do serviço utilizando o identificador do produto P como informação. Com este identificador de serviço, é possível encontrar o identificador da empresa acessando a tabela de deslocamento. Provido do identificador da empresa, SSRWS acessa o serviço do Repositório de Participantes para encontrar o nome da empresa. Para encontrar as datas de início e fim deste serviço de transporte, SSRWS acessa ProdSRWS utilizando os identificadores do produto e do serviço. Após estes passos, o SMWS retorna o nome da empresa e as datas desejadas para a aplicação usuário.

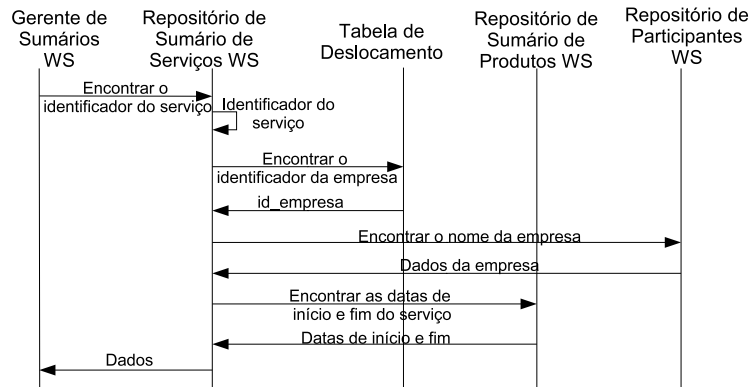


Figura 4.3: Exemplo: seqüência de atividades.

Os exemplos apresentados não tratam a comunicação entre os Serviços do Gerente de Sumários (SMWS). Um exemplo desta comunicação seria a inserção de um processo de fabricação no Repositório de Sumário de Processos. A Figura 4.4 ilustra a interação entre os Gerentes de Sumários antes do processamento da inserção, para quatro domínios diferentes. Nesta figura, pode-se observar que os serviços Web do Repositório de Processos e Regras estão nos domínios 3 e 4 respectivamente. A inserção do processo exige as

seguintes consultas:

- Consultar o serviço do Repositório de Processos: verifica se o identificador de um workflow existe no Repositório de Processos (atividade 1, na Figura 4.4);
- Consultar o serviço do Repositório de Regras: verifica se os identificadores das regras que estão sendo inseridas existem no Repositório de Regras (atividade 2, na Figura 4.4);
- Consultar o serviço do Repositório de Sumário de Produtos: verifica se os produtos correspondentes às matérias-primas (insumos) e ao produto final resultante do novo processo já estão inseridos no endereço do repositório especificado. No exemplo da Figura 4.4, supõe-se que uma das matérias-primas utilizadas (por exemplo, *Leite1*), no processo a ser inserido no Repositório de Processos do Domínio 1 está armazenada no repositório do Domínio 2. O serviço do Repositório de Sumário de Processos (ProcSRWS1) verifica se as matérias-primas estão em domínios diferentes. Como isso ocorre, ProcSRWS1 aciona o serviço de seu respectivo Summary Manager (SMWS1) requisitando a consulta do produto no outro domínio. Isto pode ser visto na atividade 3a da Figura 4.4. A seguir, SMWS1 entra em contato com o outro SMWS2 (atividade 3b), a consulta é executada e a resposta é gerada.

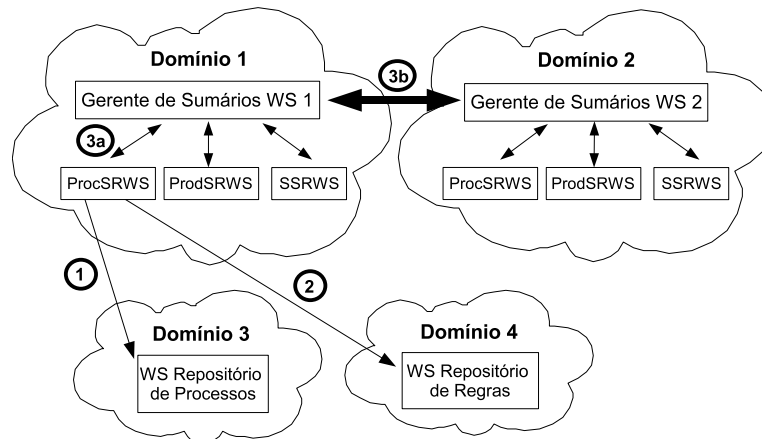


Figura 4.4: Exemplo da Sequência de Atividades em Domínios Distintos.

4.3 Conclusões

Este capítulo apresentou a infra-estrutura proposta para o modelo de rastreabilidade. Esta arquitetura é baseada em serviços Web. O próximo capítulo apresenta a implementação desta arquitetura.

Capítulo 5

Aspectos de Implementação

Este capítulo discute a implementação da arquitetura proposta. A Seção 5.1 discorre sobre as ferramentas utilizadas para a implementação. A Seção 5.2 mostra a organização do banco de dados. A Seção 5.3 descreve a arquitetura implementada. A Seção 5.4 exemplifica as consultas que podem ser processadas na arquitetura. A Seção 5.5 conclui o capítulo.

5.1 Ferramentas e Softwares adotados

Para a implementação da arquitetura foi utilizada a linguagem de programação Java (*Java™ 2 Platform Standard Edition Development Kit 5.0*) e o ambiente de programação Eclipse 3.2. A arquitetura é baseada em serviços Web, que na verdade são aplicações. Portanto, há a necessidade de um servidor Web para prover as condições de execução dessas aplicações. O servidor Web escolhido foi o *Apache Tomcat 5.5 Servlet/JSP*.

O tráfego de mensagens entre serviços Web ocorre por meio do protocolo SOAP (*Simple Object Access Protocol*). Para a construção de serviços Web foi utilizado o Axis 1.4, que é um *framework* para o desenvolvimento de processadores SOAP tais como: clientes, servidores, *gateways* entre outros.

O armazenamento dos dados de rastreabilidade utiliza o SGBD PostGIS. Este sistema de banco de dados foi escolhido porque alguns campos de dados necessitam de tipos geométricos. Para o acesso ao banco de dados através da linguagem Java utilizou-se o conector `postgresql-8.1-404.jdbc3.jar`.

Algumas diferenças entre o modelo proposto e o ER são:

- O atributo “sumario correspondente” na entidade “Sumário de Produtos” indica que um registro descreve um evento de Processo ou de Serviço.
- A entidade “Registro das Tabelas” é uma generalização de Sumário de Processos e Sumário de Serviços, contendo os campos comuns aos dois. Ou seja, um elemento desta entidade descreve um evento do tipo Processos ou Serviços.

A rastreabilidade é tratada para um ambiente em que os repositórios podem estar distribuídos na rede. O mapeamento ER dos repositórios foi feito em duas etapas. Inicialmente, realizou-se um mapeamento tradicional para banco de dados relacional, ilustrado na Figura 5.2, os atributos em itálico indicam as chaves estrangeiras. Na segunda etapa, as tabelas correspondentes foram encapsuladas por serviços distintos. A manutenção das chaves estrangeiras foi mapeada para endereços que fazem esta verificação de integridade.

```

RepProcessos(idrepproc, nomeproc, workflow)
RepProdutos(idrepprod, nomeprod, descrição, unidade)
RepServiços(idrepserv, nomeserv, descrição)
SumProcessos(responsavel, datainicio, datafim, id_reg, idreppart, idrepproc, idrepregras, input, output)
SumProdutos(idsprod, evento, descrição, sumariocorrespondente, datainicio, datafim, id_reg)
SumServiços(evento, qtd, responsavel, datainicio, datafim, idsprod, id_reg, idrepserv, idreppart)
TabCaracProc(propriedade, valor, id_reg)
TabCaracProd(propriedade, valor, idsprod, idrepprod)
TabDeslocamento(descrição, itinerário, idrepregras, idreppart, id_reg)
TabArmazenamento(idrepregras, idreppart, id_reg)
RepParticipantes(idreppart, nomepart, url, coordenadas)
RepRegras(idrepregras, nomeregra)

```

Figura 5.2: Mapeamento ER para o Relacional.

Para permitir processar ligações entre tabelas distribuídas (o equivalente a chaves estrangeiras), o valor da chave estrangeira passou a ser concatenado com uma URL de um determinado serviço *Web*, que encapsula a tabela correspondente. Cada serviço possui operações para verificar a existência de tal atributo no endereço especificado. Por exemplo, o atributo *idsprod* é chave no Sumário de Produtos e chave estrangeira no Sumário de Serviços (*SumServicos*). Seu conteúdo real no Sumário de Serviços passa a ser o par $\langle idspod, url_sumprod \rangle$ onde *idsprod* é o identificador original do produto no Sumário de Produtos e *url_sumprod* é a URL do Sumário de Produtos onde o registro correspondente está armazenado. Isto permite manter a integridade referencial em tabelas distribuídas.

```
create table RepParticipantes (  
    idreppart integer primary key,  
    nomepart text,  
    url text  
);  
  
SELECT AddGeometryColumn('repparticipantes', 'coordenadas', 128, 'POINT'  
    ', 2);  
  
create table TabDeslocamento (  
    idregserv integer primary key,  
    descricao text,  
    idrepregras integer[],  
    enderecorr text[],  
    idreppart integer,  
    enderecorp text  
);  
  
SELECT AddGeometryColumn('tabdeslocamento', 'itinerario', 4326, '  
    MULTIPPOINTM', 3);
```

Figura 5.3: Código SQL

Para mostrar a necessidade deste endereço, suponha que um processo necessite das matérias-primas leite e morango, para a fabricação de iogurte. Os eventos de criação dos produtos leite e morango não necessariamente foram registrados em um mesmo Repositório de Sumário de Produtos. Considere agora a inserção, no Sumário de Processos, de um registro de fabricação do iogurte, contendo como insumos leite e morango. Em um ambiente centralizado, tais insumos correspondem a chaves estrangeiras para os respectivos registros no Sumário de Produtos. Já em um ambiente distribuído, é necessário verificar a existência destas matérias-primas nos respectivos repositórios.

As tabelas *RepParticipantes* e *TabDeslocamento* possuem campos geográficos criados conforme pode ser visto na Figura 5.3. O PostGIS realiza isto da seguinte forma. Primeiro, a tabela é criada com seus campos normais. Depois disso, são inseridas as colunas geométricas. No caso de *RepParticipantes*, o tipo geométrico é *POINT*, representando as coordenadas x e y . Já em *TabDeslocamento* o tipo é *MULTIPOINT* representando os pontos do itinerário (x , y e t).

5.3 Implementação da Arquitetura

A Seção 5.3.1 mostra uma interface com as consultas disponibilizadas pelo SMWS. A arquitetura foi implementada conforme os diagramas de classes (Seção 5.3.2), de seqüência (Seção 5.3.3) e de atividades (Seção 5.3.4) apresentados a seguir. As seções 5.3.3 e 5.3.4 apresentam diagramas de apenas alguns dos métodos implementados.

Na implementação da arquitetura, foram encontradas várias dificuldades, algumas das quais descritas a seguir. Um primeiro problema foi ajustar o Tomcat e o Axis, para que um serviço Web pudesse invocar e combinar outros serviços Web para a realização de uma determinada tarefa.

Outra dificuldade foi a composição de serviços Web na parte da implementação. Normalmente, a composição seria especificada por mecanismos específicos de composição de serviços Web, como por exemplo, a linguagem BPEL4WS (*Business Process Execution Language for Web Services*). No protótipo, a composição foi realizada diretamente no código, usando a linguagem Java. Isto foi feito dessa forma devido ao tempo que levaria para pesquisar e aprender a utilizar uma máquina de composição (*composition engine*), o que escapava os objetivos da pesquisa.

Além disso, na parte do banco de dados, houve dificuldade na determinação do tipo do campo “itinerário”, que contém valores em três dimensões (coordenada x, coordenada y e o tempo), e na manipulação desses dados tridimensionais por meio das funções disponíveis no PostGIS. A solução encontrada foi utilizar uma extensão do PostGIS que suporta coordenadas 3D e 4D e contém funções específicas para a manipulação deste tipo de dados.

5.3.1 Interface Básica das Consultas

O acesso a SMWS é feito através de uma interface simples que disponibiliza várias consultas. Estas são traduzidas internamente em composição de serviços Web. A figura 5.4 mostra uma cópia de tela desta interface. Seu objetivo é validar a implementação.

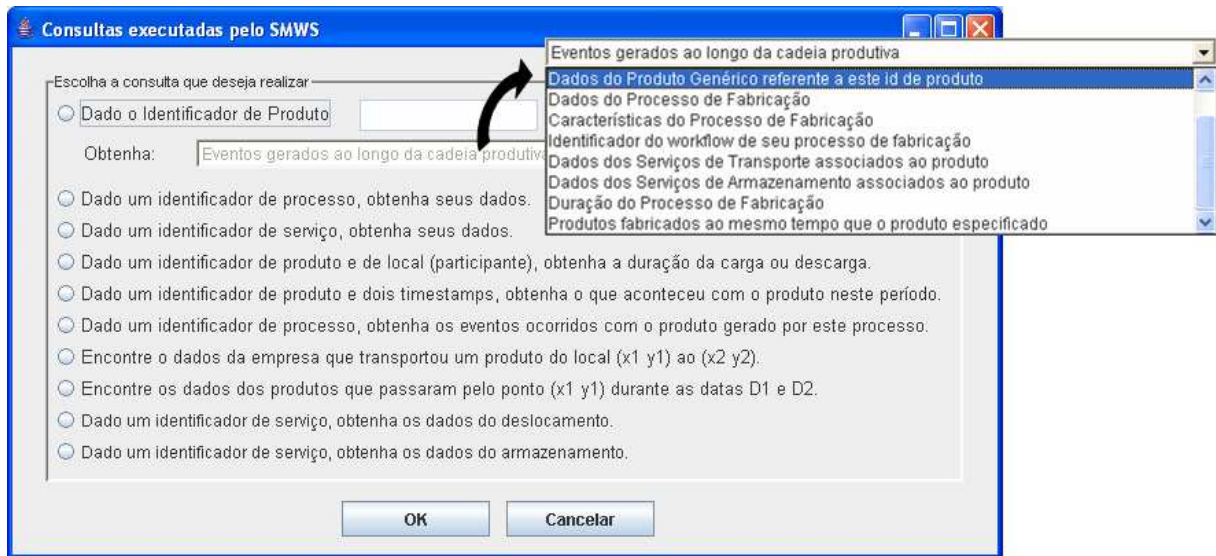


Figura 5.4: Interface para consultas via Summary Manager.

5.3.2 Diagrama de Classes

A Figura 5.5 ilustra o diagrama de classes utilizado na implementação da arquitetura. A classe *InvocacaoWS* centraliza a invocação de serviços Web e a Classe Auxiliar é responsável pelo acesso ao SGBD. A classe *InvocacaoWS* foi implementada para fatorar invocações dos diferentes serviços, eliminando redundâncias de código e simplificando a implementação. A classe *WSSumManager* centraliza o recebimento de pedidos de consultas e os distribui aos demais serviços. Também é responsável por retornar o resultado da consulta. Este diagrama mostra que a classe *WSSumManager* invoca serviços Web utilizando métodos da classe *InvocacaoWS* e apresenta os resultados das operações utilizando métodos da classe *ImprimirResultado*. As classes *WSProcSum*, *WSProdSum* e *WSServSum* contêm os métodos correspondentes aos serviços dos Sumários de Processos, Produtos e Serviços. Elas também invocam serviços utilizando a classe *InvocacaoWS*. Além disso, estas classes e as classes *WSProcRep*, *WSProdRep*, *WSServRep*, *WSRegrasRep* e *WSPartRep* manipulam dados e fazem a conexão com o banco de dados através da criação de instâncias da classe *ClasseAuxiliar*. Todas as classes da Figura 5.5 possuem operações que são apresentadas sem seus respectivos parâmetros, para facilitar a visualização dos relacionamentos entre as classes. Todas as classes do tipo WS contêm métodos de consultas (a serviços ou a dados) e atualização (inserção, remoção e modificação de dados ou atributos).

A classe *WSSumManager* contém os métodos disponibilizados pelo serviço Web do Gerente de Sumário. Os métodos de inserção, remoção e atualização são ilustrados na Tabela 5.1 e os métodos de consultas aparecem na Tabela 5.2. Esta classe invoca serviços

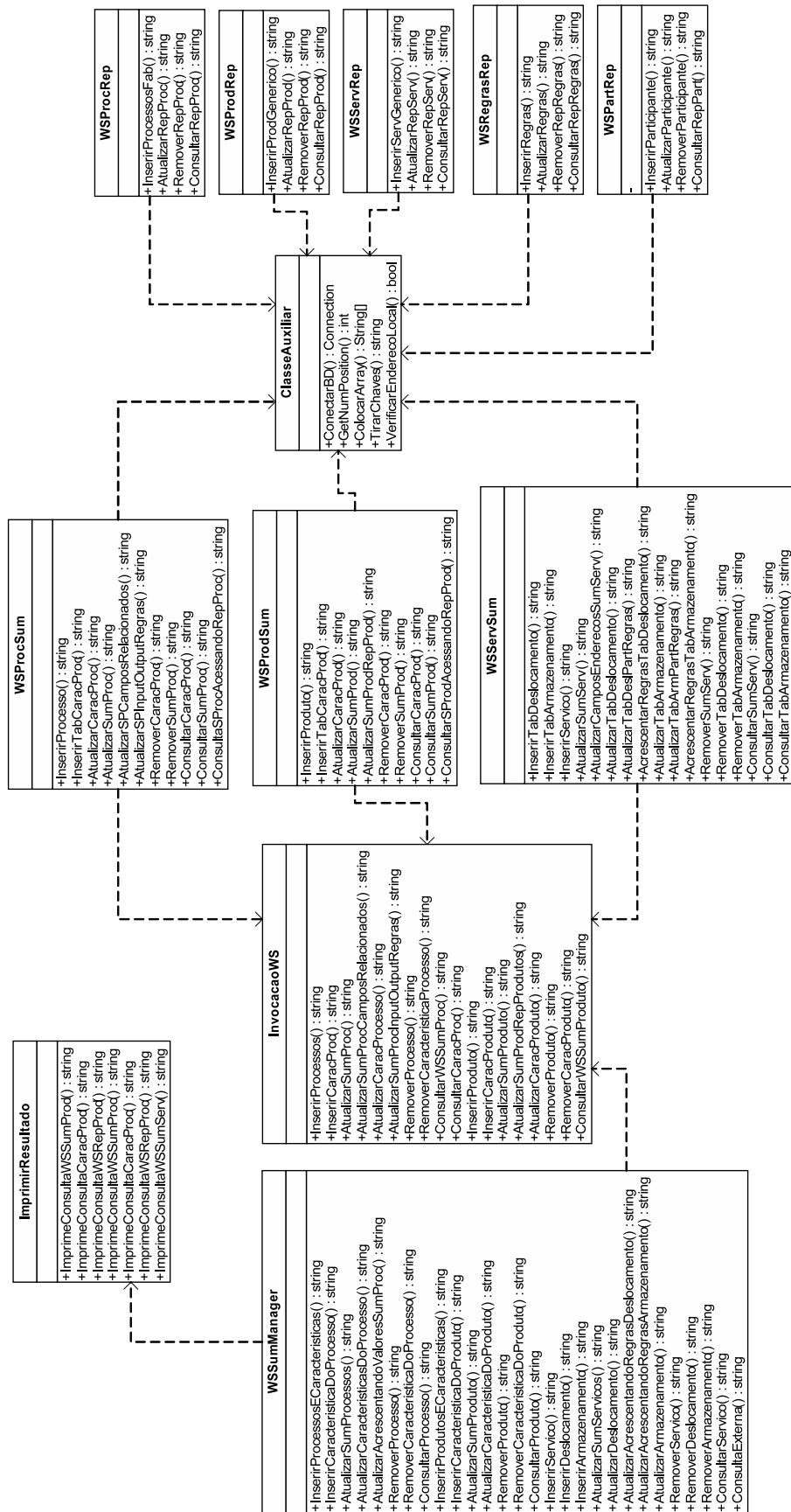


Figura 5.5: Diagrama de Classes.

disponíveis pelos repositórios de Sumários de Processos, Produtos e Serviços.

Nome	Parâmetros	Descrição
InserirProcessosECaracteristicas	Integer idregproc, Integer local, String enderecolocal, String dtfab, String responsavel, String dtini, String dtfim, Integer idrproc, String endrproc, String idrreg, String endrreg, String i, String ei, String o, String eo, String propriedade, String valor	Inserção de processos no Repositório de Sumário de Processo e de características do processo na Tabela de Características do Processo
InserirCaracteristicaDoProcesso	Integer idregproc, String propriedade, String valor	Inserção das características do processo na Tabela de Características do Processo
AtualizarSumProcessos	Integer idregproc, String campo, String valor, String campoend, String valorend	Atualiza um campo do Repositório de Sumário de Processos
AtualizarCaracteristicasDoProcesso	Integer idregproc, String propriedade, String campo, String valor	Atualiza um campo da Tabela de Características do Processo
AtualizarAcrescentandoValoresSumProc	Integer idregproc, String campo, String valor, String campoend, String valorend	Acrescenta valores aos campos que são vetores, input, output e regras, no Sumário de Processos.
RemoverProcesso	Integer idregproc	Remove um registro no Repositório de Sumário de Processos
RemoverCaracteristicaDoProcesso	Integer idregproc, String propriedade	Remove um registro na Tabela de Características do Processo
InserirProdutosECaracteristicas	Integer idsprod, Integer idpous, String enderecopous, String evento, String dataevento, String descricao, String sumcorresp, Integer idrepprod, String endrprod, String propriedade, String valor	Inserir um registro de produtos e uma de suas características no Repositório de Sumário de Produtos e na Tabela de Características respectivamente.
InserirCaracteristicaDoProduto	Integer idprod, String propriedade, String valor	Inserir um registro de características na Tabela de Características do Produto.
AtualizarSumProduto	Integer idprod, Integer idpous, String sumcorresp, String campo, String valor, String campoend, String valorend	Atualiza um campo de um registro no Repositório de Sumário de Produto.
AtualizarCaracteristicasDoProduto	Integer idprod, String propriedade, String campo, String valor	Atualiza um campo de um registro na Tabela de características do Produto.
RemoverProduto	Integer idprod	Remove um registro do Repositório de Sumário de Produtos.
RemoverCaracteristicaDoProduto	Integer idprod, String propriedade	Remove um registro na Tabela de Característica do Produto.
InserirServiço	Integer idregserv, String evento, Integer qtd, Integer localizacao, String enderecorp, String responsavel, String dtini, String dtfim, Integer idrs, String enderecors, Integer idprod, String enderecop	Inserir um registro no Repositório de Sumário de Serviços.
InserirDeslocamento	Integer idregserv, String descricao, String idreppreg, String enderecorr, Integer idreppart, String enderecorp, String itinerario	Inserir um registro na Tabela de Deslocamento.
InserirArmazenamento	Integer idregserv, String idrr, String enderecorr, Integer idreppart, String enderecorp	Inserir um registro na Tabela de Armazenamento.
AtualizarSumServicos	Integer idregserv, Integer idprod, String evento, Integer local, String campo, String valor, String campoend, String valorend	Atualiza um campo de um registro no Repositório de Sumário de Serviços.
AtualizarDeslocamento	Integer idregserv, String campo, String valor, String campoend, String valorend	Atualiza um campo de um registro na Tabela de Deslocamento.
AtualizarAcrescentandoRegrasDeslocamento	Integer idregserv, String campo, String valor, String campoend, String valorend	Acrescenta regras a um registro na Tabela de Deslocamento.
AtualizarAcrescentandoRegrasArmazenamento	Integer idregserv, String campo, String valor, String campoend, String valorend	Acrescenta regras a um registro na Tabela de Armazenamento.
AtualizarArmazenamento	Integer idregserv, String campo, String valor, String campoend, String valorend	Atualiza um campo de um registro na Tabela de Armazenamento.
RemoverServico	Integer idregserv, String evento, Integer idprod, Integer localizacao	Remove um registro do Repositório de Sumário de Serviços.
RemoverDeslocamento	Integer idregserv	Remove um registro da Tabela de Deslocamento.
RemoverArmazenamento	Integer idregserv	Remove um registro da Tabela de Armazenamento.

Tabela 5.1: Métodos de inserção, remoção e atualização da classe *WSSumManager*.

A classe *WSProcSum* possui os métodos disponibilizados pelo serviço Web do Repositório de Sumário de Processos. Os métodos, descritos na Tabela 5.3, são processados nos dados armazenados no Repositório de Sumário de Processos e na tabela de características do processo. Esta classe invoca serviços de consulta disponíveis pelos Repositórios de Participantes, Processos, Regras e Sumário de Produtos.

A classe *WSProdSum* contém os métodos, descritos na Tabela 5.4, do serviço Web do Repositório de Sumário de Produtos. Os métodos são processados no Repositório

Nome	Parâmetros	Descrição
ConsultarProduto	Integer idprod	Processa uma consulta no Repositório de Sumário de Produtos. A partir de um identificador, são obtidos todos os dados do registro.
ConsultarProcesso	Integer idregproc	Processa uma consulta no Repositório de Sumário de Processos. A partir de um identificador, são obtidos todos os dados do registro.
ConsultarServico	Integer idserv	Processa uma consulta no Repositório de Sumário de Serviços. A partir de um identificador, são obtidos todos os dados do registro.
ConsultaExterna	String endereco, String tipoconsulta, Integer id	Processa uma consulta em um repositório de outro domínio.
Consulta1	Integer idproduto	Dado um identificador de produto, obtenha os eventos gerados ao longo da cadeia produtiva e seus relacionamentos com outros sumários.
Consulta2	Integer idproduto	Dado um identificador do produto, obtenha suas características.
Consulta3	Integer idproduto	Dado um identificador do produto, obtenha o produto genérico referente a este identificador.
Consulta4	Integer idproduto	Dado um produto, obtenha os dados do processo de fabricação.
Consulta5	Integer idproduto	Dado um identificador de produto, obtenha as características do processo de fabricação.
Consulta6	Integer idproduto	Dado um identificador de produto, obtenha o identificador do workflow do processo de fabricação.
Consulta7	Integer idproduto	Dado um identificador de produto, obtenha os dados dos serviços de transporte associados.
Consulta8	Integer idproduto	Dado um identificador de produto, obtenha os dados dos serviços de armazenamento.
Consulta9	Integer idproduto	Dado um identificador de produto, obtenha a duração do processo de fabricação.
Consulta10	Integer idproduto, Integer local	Dado um identificador de produto, obtenha a duração da carga e/ou descarga no local (x y).
Consulta11	Integer idproduto, String data1, String data2	Obtenha o que aconteceu com o produto X desde a data D1 até D2.
Consulta12	Integer idprocesso	Dado um identificador de processo, obtenha os eventos ocorridos com o produto gerado por este processo.
Consulta13	Integer idproduto, Integer local1, Integer local2	Encontre o dados da empresa que transportou um produto do local (x1 y1) ao (x2 y2).
Consulta14	String ponto, String dt1, String dt2	Encontre os dados dos produtos que passaram pelo ponto (x1 y1) durante as datas D1 e D2.
Consulta24	Integer idproduto	Encontre os produtos fabricados ao mesmo tempo que o produto X.
ConsultarTabArmazenamento	Integer idserv	Processa uma consulta na Tabela de Armazenamento.
ConsultarTabDeslocamento	Integer idserv	Processa uma consulta na Tabela de Deslocamento.

Tabela 5.2: Métodos de consulta da classe *WSSumManager*.

Nome	Parâmetros	Descrição
InserirProcesso	Integer idregproc, Integer local, String enderecolocal, String dtfab, String responsavel, String dtini, String dtfim, Integer idproc, String endrproc, String idreg, String endreg, String i, String ei, String o, String eo	Inserir um processo de fabricação no Repositório de Sumário de Processos.
InserirTabCaracProc	Integer idregproc, String propriedade, String valor	Inserir uma característica do processo de fabricação na Tabela de Características do Processo.
AtualizarCaracProc	Integer idregproc, String prop, String campo, String valor	Atualiza um campo de um registro da Tabela de Características do Produto
AtualizarSumProc	Integer idregproc, String campo, String valor	Atualiza um campo de um registro do Repositório de Sumário de Processos
AtualizarSPCamposRelacionados	Integer idregproc, String campo, String valor, String campoend, String valorend	Atualiza um campo de um registro do Sumário de Processo. O campo deve estar relacionado a um campo de endereço. Desse modo, ambos devem ser apresentados.
AtualizarSPInputOutputRegras	Integer idregproc, String campo, String valor, String campoend, String valorend	Atualiza um campo de um registro do Sumário de Processo. O campo deve ser: input, output ou regras. Estes campos são vetores e possuem endereços relacionados.
RemoverCaracProc	Integer idregproc, String prop	Remove um registro da Tabela de Características do Processo.
RemoverSumProc	Integer idregproc	Remove um registro do Repositório de Sumário de Processos.
ConsultarCaracProc	Integer idregproc	Processa uma consulta na Tabela de Característica do Processo. A partir de um identificador, são obtidos todos os dados do registro.
ConsultarSumProc	Integer idregproc	Processa uma consulta no Repositório de Sumário de Processos. A partir de um identificador, são obtidos todos os dados do registro.
ConsultaSProcAcessandoRepProc	Integer idregproc	Processa uma consulta no Sumário de Processos e verifica se o identificador de um registro no Repositório de Processos realmente existe no endereço especificado..
ConsultarProdFabricadosMsmTempo	Integer idregproc	Processa uma consulta no Repositório de Sumário de Processos. Esta consulta obtém os registros de processos que fazem a interseção com o registro do processo de fabricação passado como parâmetro.

Tabela 5.3: Tabela dos métodos da classe *WSProcSum*.

de Sumário de Produtos e na tabela de características do produto. Esta classe invoca serviços de consulta disponibilizados pelos Repositórios de Produtos, Sumário de Processos e Sumário de Serviços.

Nome	Parâmetros	Descrição
InserirProduto	Integer idsprod, Integer idpous, String enderecopous, String evento, String dataevento, String descricao, String sumcorresp, Integer idrepprod, String endrp	Insere o registro de um produto no Repositório de Sumário de Produtos.
InserirTabCaracProd	Integer idsprod, String propriedade, String valor	Insere um registro na Tabela de Característica do Produto.
AtualizarCaracProd	Integer idprod, String prop, String campo, String valor	Atualiza um campo de um registro na Tabela de Características do Produto.
AtualizarSumProd	Integer idprod, Integer idpous, String sumcorresp, String campo, String valor	Atualiza um campo de um registro no Repositório de Sumário de Produtos.
AtualizarSumProdRepProd	Integer idprod, Integer idpous, String sumcorresp, String campo, String valor, String campoend, String valorend	Atualiza um campo de um registro no Repositório de Produtos. Este campo possui um endereço relacionado.
RemoverCaracProd	Integer idprod, String prop	Remove um registro da Tabela de Característica do Produto.
RemoverSumProd	Integer idprod	Remove um registro do Repositório de Sumário de Produtos.
ConsultarCaracProd	Integer idprod	Processa uma consulta na Tabela de Característica do Produto. A partir de um identificador, são obtidos todos os dados de um registro.
ConsultarSumProd	Integer idprod	Processa uma consulta no Repositório de Sumário de Produtos. A partir de um identificador, são obtidos todos os dados de um registro.
ConsultaAcessandoRepProd	Integer idprod	Verifica se o identificador de um registro no Repositório de Produtos realmente existe no endereço obtido no Sumário de Produtos.
ConsultarDuracao	Integer idprod, String evento, Integer local	Dado o identificador de um produto, obtenha a duração do respectivo processo de fabricação.
ConsultarSumProdD1D2	Integer idprod, String data1, String data2	Processa uma consulta no Repositório de Sumário de Produto. Dado o identificador do produto e as datas 1 e 2, encontra os eventos ocorridos com o produto.
ConsultarSumProdEvento	Integer idprod, String evento	Processa uma consulta no Repositório de Sumário de Produto. Dado o identificador do produto e o nome do evento, encontra os identificadores de processos ou serviços.

Tabela 5.4: Tabela dos métodos da classe *WSProdSum*.

A classe *WSServSum* possui os métodos disponibilizados pelo serviço Web do Repositório de Sumário de Serviços. Estes métodos, descritos na Tabela 5.5, são processados no Repositório de Sumário de Serviços, na tabela de deslocamento e na tabela de armazenamento. Esta classe invoca serviços de consulta disponibilizados pelos Repositórios de Participantes, Regras, Serviços e Sumário de Produtos.

ConsultarSumServ	Parâmetros	Descrição
InserirTabDeslocamento	Integer idregserv, String descricao, String idpreg, String enderecorr, Integer idrepart, String enderecorp, String itinerario	Inserir um registro na Tabela de Deslocamento
InserirTabArmazenamento	Integer idregserv, String idrr, String enderecorr, Integer idrepart, String enderecorp	Inserir um registro na Tabela de Armazenamento
InserirServiço	Integer idregserv, String evento, Integer qtd, Integer localizacao, String enderecorp, String responsavel, String dtini, String dtfim, Integer idrs, String enderecors, Integer idprod, String enderecop	Inserir um registro no Repositório de Sumário de Serviços
AtualizarSumServ	Integer idregserv, Integer idprod, String evento, Integer local, String campo, String valor	Atualiza um campo de um registro no Repositório de Sumário de Serviços
AtualizarCamposEnderecosSumServ	Integer idregserv, Integer idprod, String evento, Integer localizacao, String campo, String valor, String campoend, String valorend	Atualiza um campo de um registro no Repositório de Sumário de Serviços. Este campo possui um endereço relacionado.
AtualizarTabDeslocamento	Integer idregserv, String campo, String valor	Atualiza um campo de um registro na Tabela de Deslocamento
AtualizarTabDeslPartRegras	Integer idregserv, String campo, String valor, String campoend, String valorend	Atualiza um campo de um registro no Repositório de Sumário de Serviços. Este campo possui um endereço relacionado.
AcrescentarRegrasTabDeslocamento	Integer idregserv, String campo, String valor, String campoend, String valorend	Acrescenta uma regra a um registro na Tabela de Deslocamento
AtualizarTabArmazenamento	Integer idregserv, String campo, String valor	Atualiza um campo de um registro na Tabela de Armazenamento
AtualizarTabArmPartRegras	Integer idregserv, String campo, String valor, String campoend, String valorend	Atualiza um campo de um registro na Tabela de Armazenamento. Este campo possui endereços relacionados.
AcrescentarRegrasTabArmazenamento	Integer idregserv, String campo, String valor, String campoend, String valorend	Acrescenta uma regra a um registro na Tabela de Armazenamento
RemoverSumServ	Integer idregserv, String evento, Integer idprod, Integer localizacao	Remove um registro no Repositório de Sumário de Serviços
RemoverTabDeslocamento	Integer idregserv	Remove um registro na Tabela de Deslocamento
RemoverTabArmazenamento	Integer idregserv	Remove um registro na Tabela de Armazenamento
ConsultarSumServ	Integer idregserv	Processa uma consulta no Repositório de Sumário de Serviços. A partir de um identificador, são obtidos todos os dados do registro.
ConsultarTabDeslocamento	Integer idregserv	Processa uma consulta na Tabela de Deslocamento. A partir de um identificador, são obtidos todos os dados do registro.
ConsultarTabArmazenamento	Integer idregserv	Processa uma consulta na Tabela de Armazenamento. A partir de um identificador, são obtidos todos os dados do registro.
ConsultarCoordDatas	String ponto, String dt1, String dt2	Processa consultas na Tabela de Deslocamento e no Sumário de Serviços. Encontra o id dos produtos que passaram em um ponto durante um intervalo de tempo.
ConsultarProdutosLocais	Integer idprod, Integer local1, Integer local2	Encontra o registro da tabela de deslocamento que possui o id da empresa que realizou o transporte de um produto do local (x_1, y_1) ao (x_2, y_2) .
ConsultarSSIdEvento	Integer idprod, Integer local	Encontra o identificador do registro de serviço que carregou ou descarregou um produto em um determinado local.

Tabela 5.5: Tabela dos métodos da classe WSServSum.

A classe *WSProcRep* possui os métodos disponibilizados pelo serviço Web do Repositório de Processos. Estes métodos estão descritos na Tabela 5.6.

Nome	Parâmetros	Descrição
InserirProcessoFab	Integer idrepproc, String nome, Integer idworkflow	Inserir processos de fabricação no Repositório de Processos.
AtualizarRepProc	Integer idrepproc, String campo, String valor	Atualiza um campo de um determinado registro no Repositório de Processos.
RemoverRepProc	Integer idrepproc	Remove um registro do Repositório de Processos.
ConsultarRepProc	Integer idrepproc	Faz o processamento de consultas no Repositório de Processos. A partir de um identificador de processo, são obtidos todos os dados deste registro.

Tabela 5.6: Tabela dos métodos da classe *WSProcRep*.

A classe *WSProdRep* contém os métodos disponibilizados pelo serviço Web do Repositório de Produtos. Estes métodos estão descritos na Tabela 5.7.

Operações do WSRepProd		
Nome	Parâmetros	Descrição
InserirProdGenerico	Integer idrepprod, String nome, String descricao, String unidade	Inserir produtos genéricos no Repositório de Produtos
AtualizarRepProd	Integer idrepprod, String campo, String valor	Atualiza um campo de um registro no Repositório de Produtos.
RemoverRepProd	Integer idrepprod	Remove um registro do Repositório de Produtos.
ConsultarRepProd	Integer idrepprod	Faz o processamento de consultas no Repositório de Produtos. A partir do identificador de um produto genérico, são obtidos todos os dados deste registro.

Tabela 5.7: Tabela dos métodos da classe *WSProdRep*.

A classe *WSServRep* contém as operações disponibilizadas pelo serviço Web do Repositório de Serviços. Estas operações estão descritas na Tabela 5.8.

Nome	Parâmetros	Descrição
InserirServGenerico	Integer idrepserv, String nomeserv, String descricao	Inserir um serviço genérico no Repositório de Serviços.
AtualizarRepServ	Integer idrepserv, String campo, String valor	Atualiza um campo de um registro do Repositório de Serviços.
RemoverRepServ	Integer idrepserv	Remove um registro do Repositório de Serviços.
ConsultarRepServ	Integer idrepserv	Faz o processamento de consultas no Repositório de Serviços. A partir do identificador de um serviço genérico, são obtidos todos os dados deste registro.

Tabela 5.8: Tabela dos métodos da classe *WSServRep*.

A classe *WSRegrasRep* possui os métodos disponibilizados pelo serviço Web do Repositório de Regras. Estes métodos estão descritos na Tabela 5.9.

A classe *WSPartRep* contém os métodos disponibilizados pelo serviço Web do Repositório de Participantes. Estes métodos estão descritos na Tabela 5.10.

Os métodos da *ClasseAuxiliar*, descritos na Tabela 5.11, que são utilizados por várias classes.

InvocacaoWS invoca métodos de serviços Web. Esta classe serve como cliente de serviços Web, provendo os métodos de inserção, remoção e atualização descritos na Tabela 5.12, além dos métodos de consultas ilustrados na Tabela 5.13.

Nome	Parâmetros	Descrição
InserirRegras	Integer idrepregras, String nomeregra	Insere um registro de regras no Repositório de Regras.
AtualizarRegras	Integer idrepregras, String campo, String valor	Atualiza um campo de um registro no Repositório de Regras.
RemoverRepRegras	Integer idrepregras	Remove um registro do Repositório de Regras.
ConsultarRepRegras	Integer idrepregras	Faz o processamento de uma consulta no Repositório de Regras. A partir de um identificador, são obtidos todos os dados do registro.

Tabela 5.9: Tabela dos métodos da classe WSRegrasRep.

Nome	Parâmetros	Descrição
InserirParticipante	Integer idreppart, String nomepart, String url, String coordenadas	Insere um participante da cadeia produtiva no Repositório de Participantes
AtualizarParticipante	Integer idreppart, String campo, String valor	Atualiza algum dado do registro de um participante. Para isso, é necessário informar o campo que terá seu valor modificado.
RemoverRepParticipante	Integer idreppart	Remove um registro de um participante do Repositório de Participantes.
ConsultarRepPart	Integer idreppart	Processa uma consulta no Repositório de Participantes. A partir de um identificador, são obtidos todos os dados deste registro
ConsultarPartRegiaoInundada	String pontos	Encontra os participantes que se encontram dentro de um polígono. O polígono pode, por exemplo, representar uma região inundada.

Tabela 5.10: Tabela dos métodos da classe WSPartRep.

Nome	Parâmetros	Descrição
ConectarBD	---	Realiza a conexão com o Banco de Dados.
GetNumPosition	String str	Dada uma string que representa um vetor, em que as posições do vetor são separadas por vírgulas, obtém-se o número de itens na string.
ColocarArray	String str, Integer cont	Dada uma string representando um vetor e o número de itens dentro desta string, retorna um vetor.
TirarChaves	String entrada	A partir de uma string com abre e fecha chaves (por exemplo, {xxx,yyy}), é obtida uma string sem as chaves (ex. xxx,yyy).
VerificarEnderecoLocal	String endereco	Verifica se um endereço é local ou se é distribuído. Isso foi feito para simular endereços distribuídos.

Tabela 5.11: Tabela dos métodos da classe ClasseAuxiliar.

Nome	Parâmetros	Descrição
InserirProcessos	Integer idregproc, Integer local, String enderecolocal, String dtfab, String responsavel, String dtini, String dtfim, Integer idproc, String endrproc, String idrreg, String endreg, String i, String ei, String o, String eo	Invoca o método de inserir um registro de processo de fabricação (InserirProcesso) do serviço WSProcSum.
InserirCaracProc	Integer idproc, String propriedade, String valor	Invoca o método de inserir um registro representando uma característica do processo (InserirTabCaracProc) do serviço WSProcSum.
AtualizarSumProc	Integer idproc, String propriedade, String valor	Invoca o método de atualizar o campo de um registro no Repositório de Sumário de Processos (AtualizarSumProc) do serviço WSProcSum.
AtualizarSumProcCamposRelacionados	Integer idregproc, String campo, String valor, String campoend, String valorend	Invoca o método de atualizar o campo de um registro no Repositório de Sumário de Processos (AtualizarSPCamposRelacionados) do serviço WSProcSum. Este método atualiza os campos: local, enderecolocal, idrepproc, enderecorp, idrepregras, enderecorr, input, enderecinput, output, enderecoutput.
AtualizarCaracProcesso	Integer idregproc, String propriedade, String campo, String valor	Invoca o método de atualizar um registro na Tabela de Características do Processo (AtualizarCaracProc) do serviço WSProcSum.
AtualizarSumProcInputOutputRegras	Integer idregproc, String campo, String valor, String campoend, String valorend	Invoca o método AtualizarSPInputOutputRegras do serviço WSProcSum. Este método atualiza acrescentando regras e matérias-primas.
RemoverProcesso	Integer idregproc	Invoca o método de remover processos (RemoverProcesso) do serviço WSProcSum.
RemoverCaracteristicaProcesso	Integer idregproc, String propriedade	Invoca o método de remover um registro da Tabela de Características do Processo (RemoverCaracProc) do serviço WSProcSum.
InserirProduto	Integer idsprod, Integer idpous, String enderecopous, String evento, String dataevento, String descricao, String sumcorresp, Integer idrepprod, String endrprod	Invoca o método de inserir um registro, que indica um evento do produto, no Repositório de Sumário de Produto (InserirProduto) através do serviço WSProdSum.
InserirCaracProd	Integer idprod, String propriedade, String valor	Invoca o método de inserir um registro na Tabela de Características do Produto (InserirTabCaracProd) via WSProdSum.
AtualizarSumProduto	Integer idprod, Integer idpous, String sumcorresp, String campo, String valor	Invoca o método de atualizar um registro no Repositório de Sumário de Produtos (AtualizarSumProd) via WSProdSum.
AtualizarSumProdRepProd	Integer idprod, Integer idpous, String sumcorresp, String campo, String valor, String campoend, String valorend	Invoca o método de atualizar o campo idrepprod e/ou enderecorp do Sumário de Produtos (AtualizarSumProdRepProd) via WSProdSum.
AtualizarCaracProduto	Integer idprod, String propriedade, String campo, String valor	Invoca o método de atualizar o campo de um registro na Tabela de Características do Produto (AtualizarCaracProd) via WSProdSum.
RemoverProduto	Integer idprod	Invoca o método de remover um registro do Sumário de Produtos (RemoverSumProd) via WSProdSum.
RemoverCaracProduto	Integer idprod, String propriedade	Invoca o método de remover um registro da Tabela de Características do Produto (RemoverCaracProd) via WSProdSum.
InserirServiço	Integer idregserv, String evento, Integer qtd, Integer localizacao, String enderecorp, String responsavel, String dtini, String dtfim, Integer idrs, String enderecors, Integer idprod, String enderecorp	Invoca o método de InserirServiço que insere um registro de serviço no Repositório de Sumário de Serviços via WSServSum.
InserirDeslocamento	Integer idregserv, String descricao, String idreppreg, String enderecorr, Integer idreppart, String enderecorp, String itinerario	Invoca o método de InserirTabDeslocamento que faz a inserção de um registro de deslocamento de um transporte na Tabela de Deslocamento via WSServSum.
InserirArmazenamento	Integer idregserv, String idrr, String enderecorr, Integer idreppart, String enderecorp	Invoca o método de InserirTabArmazenamento que faz a inserção de um registro na Tabela de Armazenamento via WSServSum.
AtualizarSumServico	Integer idregserv, Integer idprod, String evento, Integer local, String campo, String valor	Invoca o método AtualizarSumServ que faz a atualização de um registro de serviço no Repositório de Sumário de Serviços via WSServSum.
AtualizarSumServCamposRelacionados	Integer idregserv, Integer idprod, String evento, Integer local, String campo, String valor, String campoend, String valorend	Invoca o método de AtualizarCamposEnderecosSumServ que faz a atualização de registros no Repositório de Sumário de Serviços via WSServSum. Os registros alterados por este método são: idprod-enderecorp, local-enderecorp e idreppart-enderecors.
AtualizarDeslocamento	Integer idregserv, String campo, String valor	Invoca o método de AtualizarTabDeslocamento que atualiza um campo do registro na Tabela de Deslocamento via WSServSum.
AtualizarDeslPartRegrasRep	Integer idregserv, String campo, String valor, String campoend, String valorend	Invoca o método de AtualizarTabDeslPartRegras que atualiza o campo idreppart e idrepregras, além de seus campos relacionados, na Tabela de Deslocamento via WSServSum.
AtualizarAcrescentandoRegrasTabDeslocamento	Integer idregserv, String campo, String valor, String campoend, String valorend	Invoca o método de AcrescentarRegrasTabDeslocamento que atualiza o campo de regras da Tabela de Deslocamento via WSServSum. A atualização insere novas regras, sem apagar aquelas já inseridas.
AtualizarArmazenamento	Integer idregserv, String campo, String valor	Invoca o método AtualizarTabArmazenamento que atualiza o campo de um registro na Tabela de Armazenamento via WSServSum.
AtualizarArmPartRegrasRep	Integer idregserv, String campo, String valor, String campoend, String valorend	Invoca o método AtualizarTabArmPartRegras que atualiza o campo idreppart e idrepregras, além de seus campos relacionados, na Tabela de Armazenamento via WSServSum.
AtualizarAcrescentandoRegrasTabArmazenamento	Integer idregserv, String campo, String valor, String campoend, String valorend	Invoca o método AcrescentarRegrasTabArmazenamento que atualiza o campo regras da Tabela de Armazenamento via WSServSum. Esta atualização insere novas regras, sem apagar aquelas já inseridas.
RemoverServico	Integer idregserv, String evento, Integer idprod, Integer localizacao	Invoca o método RemoverSumServ que remove um registro no Repositório de Sumário de Serviços via WSServSum.
RemoverDeslocamento	Integer idregserv	Invoca o método RemoverTabDeslocamento que remove um registro da Tabela de Deslocamento via WSServSum.
RemoverArmazenamento	Integer idregserv	Invoca o método RemoverTabArmazenamento que remove um registro da Tabela de Armazenamento via WSServSum.

Tabela 5.12: Métodos de inserção, remoção e atualização da classe InvocacaoWS.

Nome	Parâmetros	Descrição
ConsultarWSSumProc	Integer idproc, String enderecosp	Invoca o método de consultar o Repositório de Sumário de Processos do serviço WSProcSum. A partir de um identificador, são obtidos todos os dados do registro.
ConsultarCaracProc	Integer idproc, String enderecosp	Invoca o método de consulta na Tabela de característica do Processo (ConsultarCaracProc) do Serviço Web do Repositório de Sumário de Processos. A partir de um identificador e um endereço, são obtidos todos as características do processo.
ConsultarWSSumProc	Integer idproc, String enderecosp	Invoca o método de consultar o Repositório de Sumário de Processos do serviço WSProcSum. A partir de um identificador e um endereço, são obtidos todos os dados do registro.
ConsultarWSSumProdutos	Integer idprod, String enderecop	Invoca o método ConsultarSumProd que faz o processamento de uma consulta no Repositório de Sumário de Produtos (WSProdSum). A partir de um identificador e um endereço, são obtidos todos os dados do registro
ConsultarCaracProduto	Integer idprod	Invoca o método ConsultarCaracProd que faz o processamento de uma consulta na Tabela de Características do Produto utilizando o Serviço Web do Repositório de Sumário de Produtos.
ConsultarSumProdArrays	Integer i, String ei	Invoca o método de ConsultarSumProd de vários endereços distintos.
ConsultarWSSumServ	Integer idserv, String enderecoss	Invoca o método ConsultarSumServ que processa uma consulta no Repositório de Sumário de Serviços. A invocação é realizada via Serviço Web do Repositório de Sumário de Serviços (WSServSum). A partir de um identificador e um endereço, são obtidos todos os dados do registro.
ConsultarWSRepProd	Integer idprod, String enderecorp	Invoca o método ConsultarRepProd que processa uma consulta no Repositório de Produtos. A invocação é realizada via Serviço Web do Repositório de Produtos (WSProdRep). A partir de um identificador e um endereço, são obtidos todos os dados do registro.
ConsultarWSRepPart	Integer local, String enderecolocal	Invoca o método ConsultarRePart que processa uma consulta no Repositório de Participantes. A invocação é realizada via Serviço Web do Repositório de Participantes. A partir de um identificador e um endereço, são obtidos todos os dados do registro.
ConsultarWSRepProc	Integer idregproc, String enderecorp	Invoca o método ConsultarRepProc que processa uma consulta no Repositório de Processos. A invocação é realizada via Serviço Web do Repositório de Processos. A partir de um identificador e um endereço, são obtidos todos os dados do registro.
ConsultarWSRepServ	Integer idregserv, String enderecors	Invoca o método ConsultarRepServ que processa uma consulta no Repositório de Serviços. A invocação é realizada via Serviço Web do Repositório de Serviços. A partir de um identificador e um endereço, são obtidos todos os dados do registro.
ConsultarWSRepRegras	Integer idrr, String enderecorr	Invoca o método ConsultarRepRegras que processa uma consulta no Repositório de Regras. A invocação é realizada via Serviço Web do Repositório de Regras. A partir de 2 strings uma contendo identificadores e a outra com endereços, são obtidos todos os dados de cada registro.
ConsultarDuracaoSumProd	Integer idproc, String evento, Integer local	Invoca o método ConsultarDuracao que processa uma consulta no Sumário de Produtos. A invocação é realizada via Serviço Web do Repositório de Sumário de Produtos. A partir dos parâmetros, é obtida a duração de um processo ou serviço.
ConsultarProdFabMsmTempo	Integer idproc	Invoca o método ConsultarProdFabricadosMsmTempo que processa uma consulta no Sumário de Processos. São obtidos os registros de processos que produziram um produto no mesmo período do processo especificado como parâmetro.
ConsultarSSCoordDatas	String ponto, String dt1, String dt2	Invoca o método ConsultarCoorddatas do serviço Web do Sumário de Serviços com o objetivo de encontrar o identificador do produto.
ConsultarSSProdLocais	Integer idprod, Integer local1, Integer local2	Invoca o método ConsultarProdutosLocais do serviço Web do Sumário de Serviços com o objetivo de encontrar o registro da tabela de deslocamento que possui o id da empresa que transportou um produto de (x_1, y_1) ao (x_2, y_2) .
ConsultarSumProcAcessandoRepProc	Integer idproc	Invoca o método ConsultaSProcAcessandoRepProc que faz o processamento de uma consulta no Repositório de Sumário de Processos, verificando se o identificador de um registro no Repositório de Processos realmente existe no endereço especificado. Esta invocação é realizada utilizando o Serviço Web do Repositório de Sumário de Processos.
ConsultarSumServIdEvento	Integer idprod, Integer local	Invoca o método ConsultarSumServ que processa uma consulta no Repositório de Sumário de Serviços. A invocação é realizada via Serviço Web do Repositório de Sumário de Serviços (WSServSum). A partir de um identificador e um endereço, são obtidos todos os dados do registro.
ConsultarSumProdAcessandoRepProd	Integer idprod	Invoca o método ConsultarTabCaracProdAcessandoRepProd que faz o processamento de uma consulta na Tabela de Características do Produtos, verificando se o identificador de um registro e um endereço existem no Repositório de Produtos.
ConsultarWSSumProdD1D2	Integer idprod, String data1, String data2	Invoca o método ConsultarSumProdD1D2 que faz o processamento de uma consulta no Sumário de Produtos. Esta invocação é feita via Serviço Web do Repositório de Sumário de Produtos. Dado os parâmetros, encontra os eventos ocorridos com o produto.
ConsultarWSSumProdEvento	Integer idprod, String evento	Invoca o método ConsultarSumProdEvento que faz o processamento de uma consulta no Repositório de Sumário de Produtos (WSProdSum). A partir de um identificador, um endereço e o nome de um evento, são obtidos todos os dados dos registros.
ConsultarTabArmazenamento	Integer idserv, String enderecoss	Invoca o método ConsultarTabArmazenamento. A invocação é realizada via Serviço Web do Repositório de Sumário de Serviços (WSServSum).
ConsultarTabDeslocamento	Integer idserv, String enderecoss	Invoca o método ConsultarTabDeslocamento. A invocação é realizada via Serviço Web do Repositório de Sumário de Serviços (WSServSum).

Tabela 5.13: Métodos de consultas da classe InvocacaoWS.

A classe *ImprimirResultado* fornece métodos para mostrar os resultados das consultas.

5.3.3 Diagramas de Seqüência de algumas Operações Básicas

Esta seção ilustra alguns diagramas de seqüência de inserções e consultas com o objetivo de visualizar as trocas de mensagens efetuadas entre os serviços.

Inserção de um registro de Sumário de Processos

A Figura 5.6 ilustra a seqüência de invocações de serviços para a inserção de um processo no Repositório de Sumário de Processos. Antes de ser executada a inserção no banco de dados, estas invocações de consultas são necessárias para verificar a existência dos identificadores do participante, do workflow e das regras nos endereços especificados. Em outras palavras, todo procedimento de inserção de um registro em um determinado sumário só é efetuado se garantida a integridade referencial. Remoções e modificações também verificam integridade referencial e são processados de forma análoga às inserções.

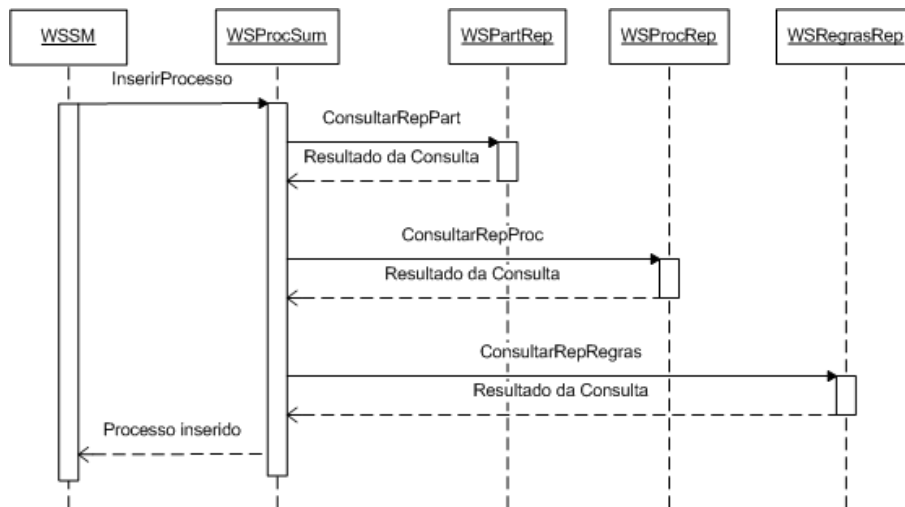


Figura 5.6: Diagrama de Seqüência para Inserir um registro no Sumário de Processos.

Inserção de um registro no Sumário de Serviços

A inserção de um determinado serviço, no Repositório de Sumário de Serviços, envolve uma seqüência de invocações de serviços (Figura 5.7). A inserção somente é realizada, depois de verificar a existência dos identificadores do serviço genérico, do participante e do produto nos endereços especificados. Estas verificações são realizadas via invocações de serviços de consulta, garantindo a integridade referencial. A inserção cuida apenas de campos do próprio Sumário de Serviços, não tratando de inserções em tabelas auxiliares.

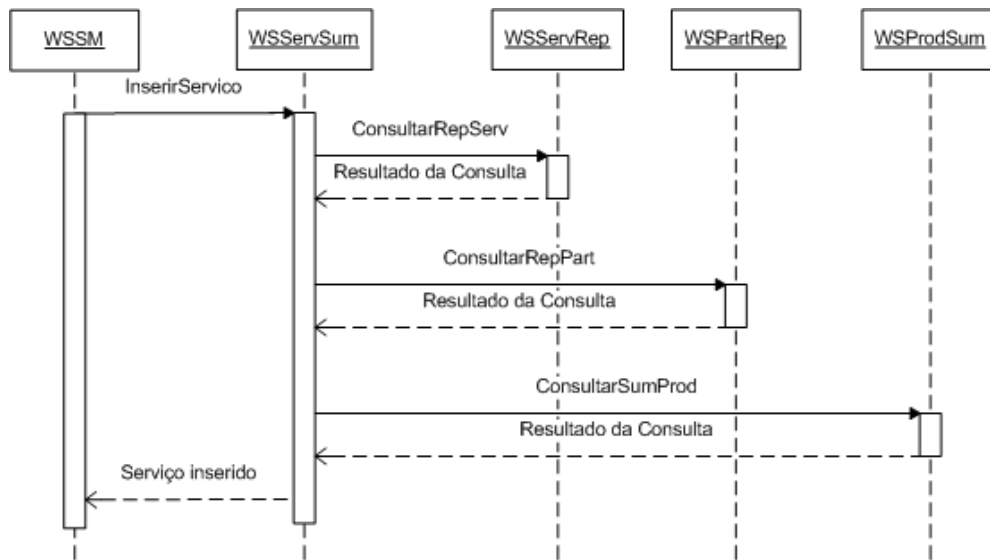


Figura 5.7: Diagrama de Seqüência para Inserir um registro no Sumário de Serviços.

Obter os dados de um produto genérico referente a um identificador de produto

Para obter o produto genérico que está associado a um determinado identificador de produto, são realizadas as invocações de serviços ilustradas na Figura 5.8. Primeiro, uma invocação de serviço é realizada no Sumário de Produtos, para conseguir obter o identificador de um registro no Repositório de Produtos e o endereço (URL) deste repositório. Com este identificador e endereço, os dados do produto genérico podem ser obtidos do Repositório de Produtos via invocação do serviço de consulta.

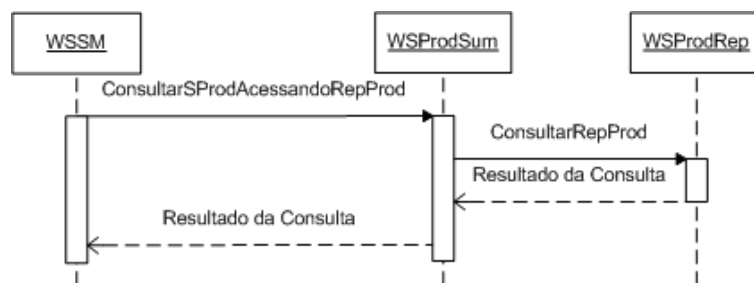


Figura 5.8: Diagrama de Seqüência para obter os dados do produto genérico.

Obter as características do processo de fabricação de um produto

A Figura 5.9 ilustra a seqüência de invocações de processos para, dado um identificador de produto, obter as características de seu processo de fabricação. O WSSM solicita ao

WS do Sumário de Produtos (*WSProdSum*) o identificador do processo de fabricação correspondente. De posse deste identificador, acessa o Sumário de Processos para consultar a Tabela de Características do Processo e obter dados daquele processo.

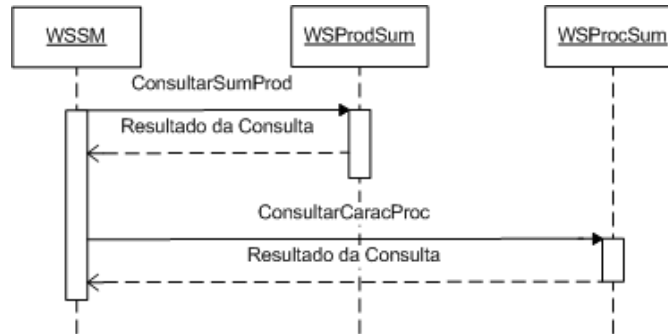


Figura 5.9: Diagrama de Seqüência para obter as características de um processo.

Obter os dados dos serviços de armazenamento de um determinado produto

Para conseguir os dados dos serviços de armazenamento, a partir de um identificador de produto, são necessárias invocações de serviços ilustradas na Figura 5.10. O WSSM consulta o serviço do Sumário de Produtos para a obtenção dos identificadores dos serviços de armazenamento registrados para aquele produto. Com estes identificadores, são obtidos os dados dos serviços no Sumário de Serviços.

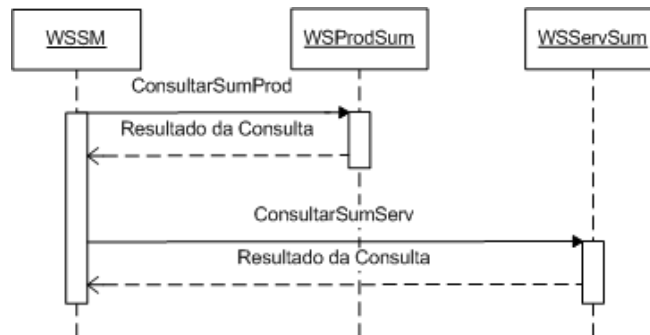


Figura 5.10: Diagrama de Seqüência para obter os dados dos serviços de armazenamento.

5.3.4 Diagrama de Atividades

Um diagrama de atividades mostra o fluxo de atividades de uma determinada operação. Alguns exemplos de operações são apresentados a seguir.

Inserir o evento de um produto

A inserção de um registro que descreve um evento de produto envolve a seguinte seqüência de atividades (Figura 5.11). Esta operação contém como entrada um conjunto de dados para a criação de um registro no Sumário de Produtos.

1. Verifica se o identificador de um registro do Repositório de Produtos existe no endereço especificado. Isto é necessário pois as chaves estrangeiras foram concatenados com endereços. Dessa forma, a verificação da integridade das chaves estrangeiras é tratada no código via invocações de serviços. Se esta verificação for verdadeira, o fluxo segue para o passo 2, senão a inserção é cancelada.
2. Verifica se o evento gerado por um produto é realizado por um Processo ou Serviço. Isto pode ser visto através do campo “*sumariocorrespondente*”.
3. Se o evento corresponde a um Processo, verifica se o identificador do campo “*id-pous*” existe no Sumário de Processos, antes da inserção do registro no Sumário de Produtos. Caso contrário, verifica a existência do identificador no Sumário de Serviços.
4. Se a verificação no Sumário de Processos ou Serviços for verdadeira, o registro correspondente ao produto é inserido no banco de dados, na tabela contendo dados de Sumário de Produtos. Se falsa, a inserção é cancelada.

Atualizar campos e seus endereços relacionados no Sumário de Processos

No Sumário de Processos, a atualização do par $\langle \textit{atributo}, \textit{uri} \rangle$ envolve as seguintes atividades (Figura 5.12). Para esta operação, a entrada é composta dos dados: identificador do processo, $\langle \textit{atributo que será atualizado}, \textit{uri relacionada} \rangle$, $\langle \textit{novo valor do atributo}, \textit{novo valor da uri} \rangle$.

1. Verifica a existência do identificador do registro que será atualizado no Sumário de Processos.
2. Verifica se o par $\langle \textit{atributo}, \textit{uri} \rangle$ que sofrerá modificação corresponde a uma das seguintes opções: $\langle \textit{local}, \textit{enderecolocal} \rangle$, $\langle \textit{idrepproc}, \textit{enderecorp} \rangle$, $\langle \textit{idrepregras}, \textit{enderecorr} \rangle$, $\langle \textit{input}, \textit{enderecoinput} \rangle$ ou $\langle \textit{output}, \textit{enderecooutput} \rangle$. Identificado o tipo de atributo que será atualizado, verifica a existência daquele identificador no endereço especificado. Este endereço pode corresponder ao Repositório de Participantes, Repositório de Processos, Repositório de Regras ou Repositório de Sumário de Produtos respectivamente.

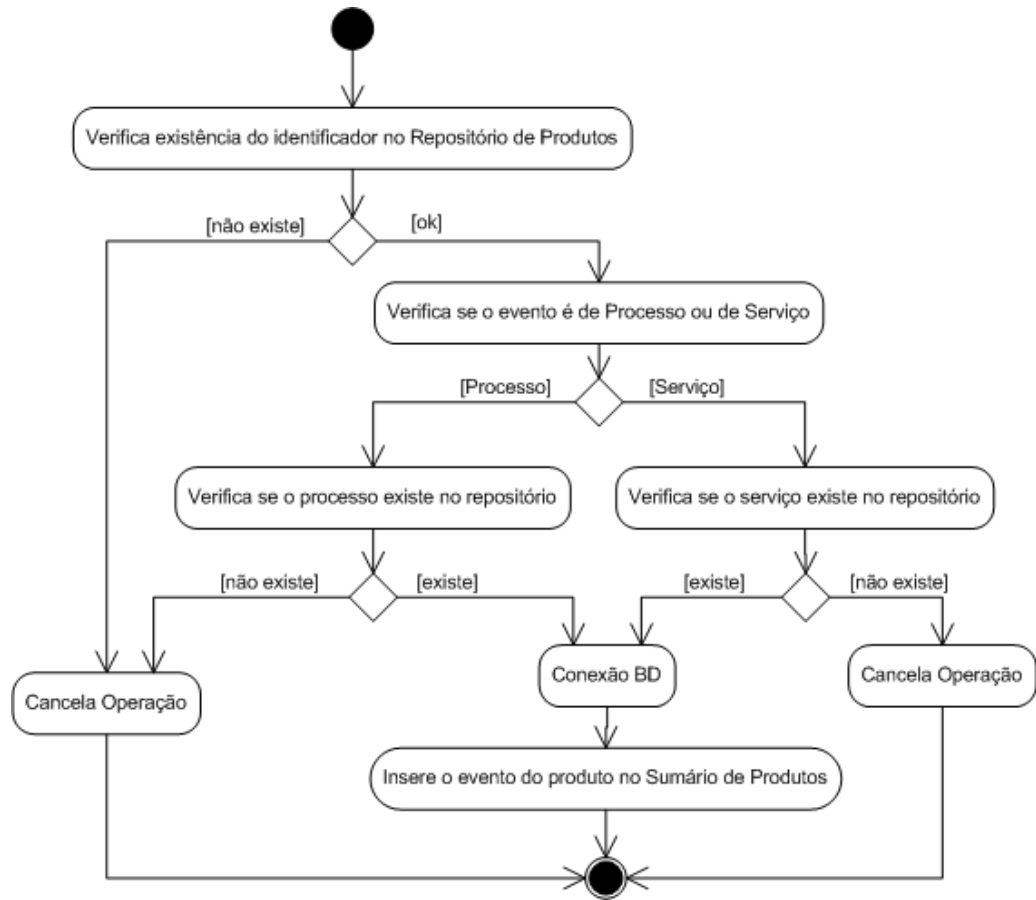


Figura 5.11: Diagrama de Atividades para o método de inserir eventos de produto.

3. Se a verificação do passo anterior for verdadeira, os campos são atualizados no banco de dados. Se falsa, a atualização é cancelada.

Remover um serviço

A remoção de um registro no Sumário de Serviços que contém como entrada o identificador de um registro de serviço envolve as seguintes atividades (Figura 5.13):

1. Verifica se o identificador do registro que será removido não existe no Sumário de Serviços.
2. Verifica se o identificador do registro que será removido existe na Tabela de Deslocamento.
3. Verifica se o identificador de registro que será removido existe na Tabela de Armazenamento.
4. Se alguma das atividades acima for verdadeira, a remoção do serviço é cancelada. Se todas forem falsas, a remoção é processada no Repositório de Sumário de Serviços.

5.4 Descrição de Consultas

A literatura correlata faz distinção entre as abordagens de rastreabilidade para frente (*forward*) e para trás (*backward*). A idéia está associada à noção de processamento do histórico de eventos de algum ponto para frente na cadeia, ou do fim para trás. Muitas soluções só permitem um tipo de abordagem. A modelagem e a implementação realizadas nesta dissertação possibilitam os dois tipos de percursos, indistintamente. Assim, não é preciso realizar processamentos distintos para as consultas:

- Qual o histórico do produto X? (*backward*)
- A partir do processo Y, obtenha os eventos ocorridos com o produto gerado por este processo (*forward*).

O Summary Manager Web Service (SMWS) permite realizar uma gama de consultas, aqui divididas em simples e complexas. As consultas simples estão descritas na Seção 5.4.1 e as consultas complexas, na seção 5.4.2. Como os repositórios estão distribuídos, supõe-se que as consultas estão sendo realizadas nos repositórios corretos. Caso contrário, seria

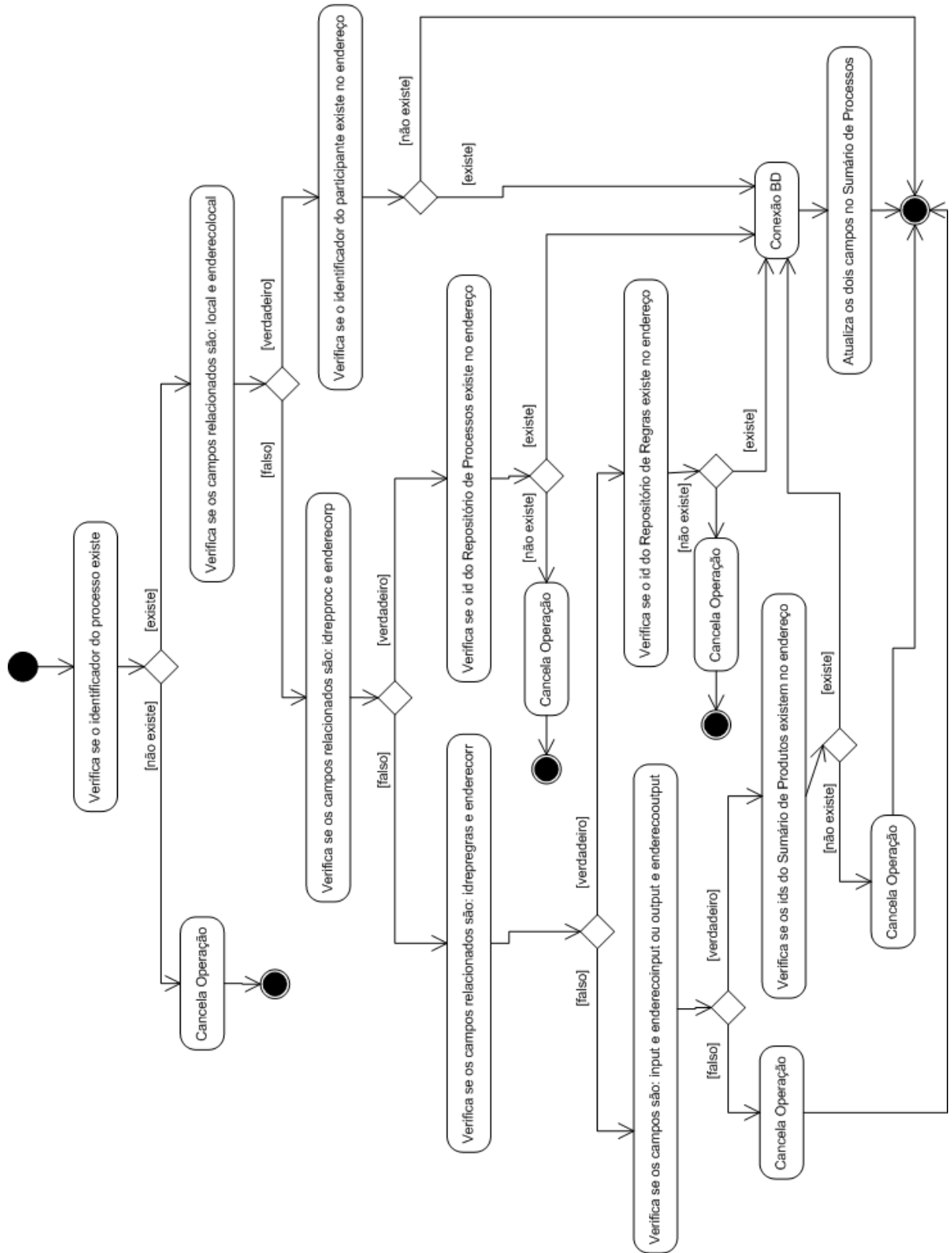


Figura 5.12: Diagrama de Atividades para o método de atualizar campos e seus endereços no Sumário de Processo.

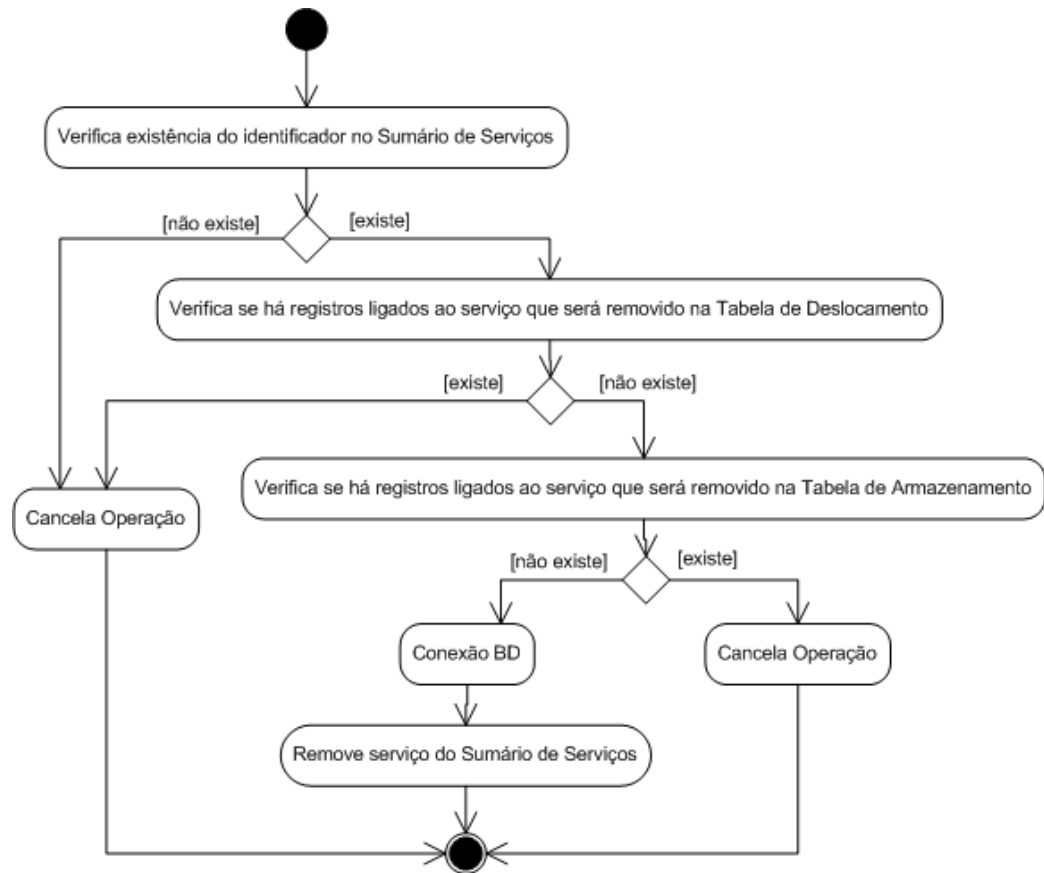


Figura 5.13: Diagrama de Atividades para o método de remover um serviço.

necessária a conversação entre os SMWSs para conseguir os dados que podem não necessariamente estar no repositório em que está sendo feita a consulta (vide discussão sobre comunicação entre seus domínios na Seção 4.1). Dessa forma, todo SMWS deveria conhecer todos os SMWSs. Isto pode ser resolvido de vários modos, que fogem ao escopo deste trabalho – por exemplo, via uso de UDDI, ou por meio de broadcast para vários outros serviços.

5.4.1 Consulta Simples

A dissertação considera consultas simples aquelas que envolvem acesso a um único serviço e cujos predicados sejam igualmente simples. Todas as demais consultas são ditas complexas. Predicados simples, neste contexto, são os que não exigem desenvolvimento de funções específicas. São aqueles disponibilizados por qualquer SGBD padrão. Um exemplo de consulta deste tipo é “mostrar o histórico do produto X”. Esta consulta apenas invoca o serviço Web do Repositório de Sumário de Produtos (ProdSRWS) e retorna todos os registros correspondentes ao produto X.

Outro exemplo é “determinar algumas características do processo Y”. Esta consulta envolve o acesso ao ProcSRWS (Figura 4.1), onde são encontradas as características do processo.

Outras consultas possíveis podem, por exemplo, solicitar eventos ocorridos com um produto, processo ou serviço dentro de um período, ou atributos de algum registro, entre outros.

5.4.2 Consultas Complexas

As consultas complexas envolvem a invocação de mais de um serviço e/ou predicados que exigem um processamento complexo.

Por analogia à álgebra relacional, pode-se dizer que consultas simples não envolvem junções e não há processamento complexo para o cálculo de predicados envolvidos. Consultas complexas, nesta analogia, requerem junções e processamento de predicados que exigem implementação de funções específicas. A existência de serviços Web distintos facilita a independência e a distribuição. Entretanto, requer maior sofisticação de código no tratamento de junções que, em um SGBD relacional, se limitaria ao processamento de chaves estrangeiras. A seguir são apresentados exemplos dos casos de junções via invocação de vários serviços Web e de predicados complexos.

Junções via invocação de vários serviços Web

Um exemplo de consulta que envolve invocações de serviços Web é “dado um identificador de um produto X, obtenha o produto genérico referente a este identificador”. Suponha que esta consulta seja enviada a um WSSM, o qual invoca um método de ProdSRWS. Este método possui uma lógica que faz a invocação de serviços de WSRepProd, para obter o resultado esperado. Esta seqüência de invocações de serviços está ilustrada na Figura 5.14(b). Já a Figura 5.14(a) apresenta esta consulta em uma notação semelhante à álgebra relacional.

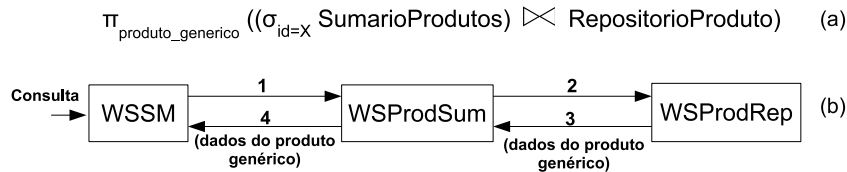


Figura 5.14: Exemplo de consulta realizada no Sumário de Produtos. (a) ilustra a álgebra relacional. (b) mostra a seqüência de invocações de serviços e suas respectivas respostas.

Outra consulta que realiza junções via invocação de serviços é “dado um identificador de produto X, obtenha os dados do seu processo de fabricação”. Esta consulta é processada por um WSSM que faz a composição de serviços invocando métodos de WSProdSum e WSProcSum. Esta composição pode ser vista na Figura 5.15(b). Na parte (a) da figura, está a “álgebra relacional” correspondente a esta consulta.

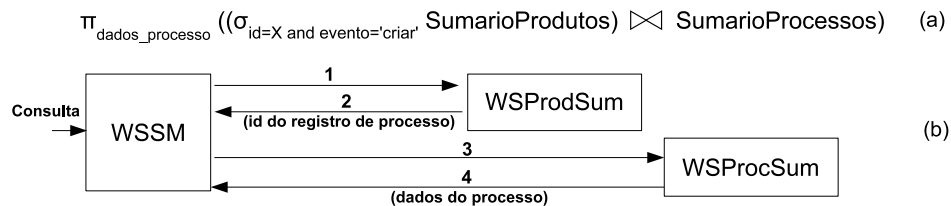


Figura 5.15: Exemplo de consulta obtendo dados do processo. (a) ilustra a álgebra relacional. (b) mostra a seqüência de invocações de serviços e suas respectivas respostas.

Ainda outro exemplo de consulta é “dado um identificador de produto X, obtenha os dados dos serviços de transporte associados”. O WSSM processa a consulta fazendo a composição dos serviços WSProdSum e WSServSum. Isto pode ser visto na Figura 5.16(b). A parte (a) mostra a “álgebra relacional” correspondente.

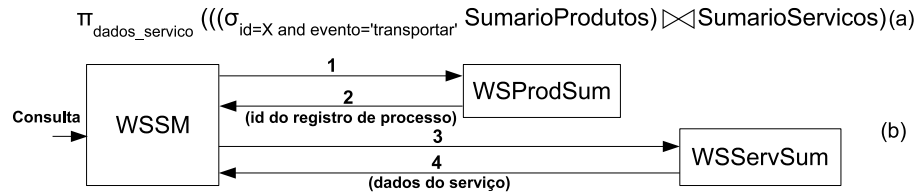


Figura 5.16: Exemplo de consulta obtendo dados de um serviço de transporte. (a) ilustra a álgebra relacional. (b) mostra a seqüência de invocações de serviços e suas respectivas respostas.

Predicados Complexos

Exemplos típicos de predicados complexos são aqueles aspectos que envolvem relacionamentos temporais ou espaciais:

1. Produtos fabricados ao mesmo tempo que o produto X.
2. Armazéns situados em uma região inundada durante um certo período.
3. Transportes que atravessam uma região inundada durante um certo período (implicação - possivelmente os produtos transportados podem ter sido contaminados) [47].

A implementação do terceiro item não foi considerada na tese, sendo deixada como trabalho futuro.

5.5 Conclusões

Este capítulo apresentou os principais aspectos da implementação realizada. Mostrou, através de exemplos, como diferentes tipos de consultas sobre rastreabilidade podem ser solucionadas, pela combinação de chamadas aos serviços Web desenvolvidos.

Capítulo 6

Conclusões e Trabalhos Futuros

6.1 Conclusões

Cadeias produtivas apresentam vários desafios para pesquisas em Computação. Este trabalho teve como ponto de partida o artigo de Bacarin [2], o qual propôs o processamento de dados de rastreabilidade realizado via Sumários. A partir daquela proposta, a dissertação especificou um modelo detalhado de rastreabilidade, centrado em diferentes tipos de registros interligados. Cada registro documenta um evento da cadeia. O modelo proposto permite seguir a vida de um produto em vários níveis de granularidade distintos através do uso de repositórios e sumários distribuídos. Este modelo foi avaliado para cenários específicos utilizando dados fictícios.

Registros são armazenados em repositórios que podem estar distribuídos. Desta forma, a arquitetura proposta é baseada em serviços Web para permitir interações entre os repositórios. O uso de serviços Web para a comunicação entre os repositórios facilita o gerenciamento dos dados de rastreabilidade e provê a interoperabilidade para ambientes distribuídos e heterogêneos. A arquitetura foi totalmente implementada e testada para alguns cenários. Os testes se basearam em conceitos de consultas simples e complexas, o que permitiu validar a execução de vários outros tipos de cenários.

Esta dissertação difere de outros trabalhos relacionados pelo nível de detalhamento de aspectos de armazenamento, pela implementação de serviços e pelo uso de dados espaciais associados aos participantes da cadeia. Estes aspectos aumentam o número de entidades e fatos rastreáveis.

Este trabalho apresentou várias dificuldades. A principal dificuldade encontrada foi conseguir propor um modelo de rastreabilidade que pudesse ser utilizado em qualquer cadeia produtiva na agropecuária. Cada produto fabricado em uma cadeia produtiva possui características ou informações específicas diferentes, o que complicou a especificação do modelo.

Outra dificuldade encontrada foi a modelagem da mistura de produtos durante o transporte ou o armazenamento. A mistura ocorre quando um transporte leva vários tipos de mercadorias, carregando-as e descarregando-as em locais distintos ou quando vários elementos de transporte descarregam em um mesmo local de armazenamento. Misturas também ocorrem durante um processo. Este problema torna-se mais difícil quando o produto efetivamente perde a identidade, como o leite ou os grãos. Por exemplo, determinados tipos de café são produzidos a partir de misturas de lotes de diferentes produtores. A solução encontrada foi criar um processo fictício, que “*gera*” a mistura, obtendo-se, assim, um novo produto. O modelo trata esta mistura como um processo que gera um novo produto. Desse modo, é possível identificar o momento em que houve a mistura e de “onde” vieram os produtos que foram misturados.

As principais contribuições desta dissertação são:

- Proposta de um modelo para gerenciar o armazenamento de dados de rastreabilidade.
- Proposta de uma arquitetura baseada em serviços Web, para permitir a implementação do modelo em um ambiente distribuído.
- Implementação da arquitetura proposta, validando-a com execução de algumas consultas de rastreabilidade.

6.2 Extensões

A dissertação tem várias extensões possíveis, algumas das quais listadas a seguir.

Uso de Ontologias

O uso de ontologias pode ser adotado para explicitar a semântica dos campos dos registros de sumários. No Sumário de Produtos e Serviços, o campo *evento* pode conter vários tipos de eventos. Para este campo pode-se criar uma ontologia especificando os eventos que poderão ser gerados ao longo da cadeia produtiva. Ontologias poderiam fornecer a identificação correta dos produtos e sua respectiva descrição, ou definir os tipos de serviços realizados ao longo da cadeia produtiva.

Explorar o Uso de Regras

Regras aparecem em vários pontos de uma cadeia produtiva. Este trabalho não se ocupou com seu uso. Por exemplo, a tabela de deslocamento contém os itinerários e as regras seguidas no caminho completo. Como trabalho futuro sugerimos que ocorra a associação das regras obedecidas a cada trecho do itinerário.

Arquitetura

Para processar consultas distribuídas, supomos que a consulta está sendo executada no repositório correto. Caso isso não fosse considerado, seria necessário algum tipo de centralizador para controlar os SMWSs existentes e fazer broadcast para encontrar o repositório que contenha dados sobre o elemento procurado.

Outras extensões do trabalho neste item seriam o controle de acesso aos serviços Web, controle de concorrência aos repositórios e a replicação dos dados de rastreabilidade nos repositórios.

Outros

Sugerimos a integração do Repositório de Processos, que contém os workflows dos processos de fabricação, com o banco de dados de workflows proposto por [34].

Também pode ser feito futuramente o gerenciamento da rastreabilidade quando ocorre o encapsulamento de serviços dentro de serviços. O problema encontrado é a dificuldade em diferenciar os serviços devido à coordenada geográfica ser a mesma, pois os serviços são realizados em um mesmo local.

Ainda outra possibilidade seria verificar a aplicabilidade do modelo proposto a outros domínios de cadeia produtiva – por exemplo, na indústria automobilística.

Referências Bibliográficas

- [1] G. Alonso et al. *Web Services: Concepts Architectures and Applications*. Springer, Germany, 2004.
- [2] E. Bacarin, C. B. Medeiros, and E. R. M. Madeira. A Collaborative Model for Agricultural Supply Chains. In R. Meersman and Z. Tari, editors, *CoopIS/DOA/ODBASE 2004, LNCS 3290*, pages 319–336. Springer-Verlag, 2004.
- [3] L. L. Bello, O. Mirabella, and N. Torrasi. Modelling and evaluating traceability systems in food manufacturing chains. In *WETICE '04: Proceedings of the 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 173–179, USA, 2004. IEEE Computer Society.
- [4] D. Box et al. Simple object access protocol (SOAP) 1.1. <http://www.w3.org/TR/soap>, 2000.
- [5] M. V. Brisola, E. E. Santo, and M. T. A. Paludo. Interest of the consumer from brasiliana in beef traceability. In *IV Congresso Internacional de Economia e Gestão de Redes Agroalimentares*, pages 1–14, Ribeirão Preto, São Paulo, 2003.
- [6] A. M. G. Castro et al. *Cadeias Produtivas e Sistemas Naturais - Prospecção Tecnológica*. EMBRAPA/Serviços de Produção de Informática, Brasília, 1998.
- [7] D. A. Chappell and T. Jewell. *Java Web Services*. O'Reilly & Associates, Inc., 981 Chestnut Street, Newton, MA 02164, USA, 2002. Using Java in service-oriented architectures.
- [8] M. G. C. A. Cimino et al. Cerere: an information system supporting traceability in the food supply chain. In *CECW '05: Proceedings of the Seventh IEEE International Conference on E-Commerce Technology Workshops*, pages 90–98, USA, 2005. IEEE Computer Society.

- [9] M. Debord et al. Geotraceability: an innovative concept for the qualification of crop production, 2001. Disponível em: http://www.geotraceagri.net/doc/GeoTraceAgri_Finalreport_EN.pdf/.
- [10] H. M. Deitel et al. *Web Services. A Technical Introduction*. Prentice Hall, Estados Unidos, 2003.
- [11] H. M. Deitel et al. *XML Como Programar*. Bookman, Porto Alegre, 2003. Tradução: Luiz Augusto Salgado and Edson Furmankiewicz.
- [12] Instituto Genesis. Rastrear com credibilidade é para profissional do ramo agropecuário. Disponível em: http://www.institutogenesis.org.br/internas/certificacao/p_rastrear_api.asp. Acesso em: 16 abr. 2005.
- [13] L. R. Gieszl. Traceability for integration. In *2nd International Conference on Systems Integration*, pages 220–228, 1992.
- [14] F. M. Gryna. Manufacturing planning. In *Juran, J. M. and Gryna, F. M. - Quality Control Handbook*, pages 38–40. McGraw-Hill, 1988.
- [15] V. L. Hamilton and M. L. Beeby. Issues of traceability in integrating tools. In *IEE Colloquium on Tools and Techniques for Maintaining Traceability During Design*, pages 41–43, 1991.
- [16] M. Hendricks et al. *Professional Java Web Services*. Alta Books, Rio de Janeiro, 2002.
- [17] J. E. Hobbs. A transaction cost analysis of quality, traceability and animal welfare issues in UK beef retailing. *British Food Journal*, 98(6):16–26, 1996.
- [18] D. Hollingsworth. Workflow management coalition - the workflow reference model. In *The Workflow Management Coalition Specification*, 1995.
- [19] M. H. Jansen-Vullers, C. A. van Dorp, and A. J. M. Beulens. Managing traceability information in manufacture. *International Journal of Information Management*, 23:395–413, 2003.
- [20] D. S. Kaster, C. M. B. Medeiros, and H. V. Rocha. Supporting modeling and problem solving from precedent experiences: the role of workflows and case-based reasoning. *Environmental Modelling & Software*, 20(2005):689–704, 2004.
- [21] A. A. Kondo et al. Web service-based traceability in food supply chains. In *Proceedings of the 3rd International Conference on Web Information Systems and Technologies (WEBIST 2007)*, pages 171–177. INSTICC, 2007.

- [22] S. Lewis. Soybean traceability used as policy weapon in brazil. *Food Traceability Report*, 4(5):4, 2004.
- [23] R. T. M. Machado. Rastreabilidade, tecnologia de informação e coordenação de sistemas agroindustriais. Doutorado em administração, Universidade de São Paulo, Faculdade de Economia, Administração e Contabilidade, 2000.
- [24] R. T. M. Machado. Signals of food quality and traceability. In *IV Congresso Internacional de Economia e Gestão de Redes Agroalimentares*, pages 1–15, Ribeirão Preto, São Paulo, 2003.
- [25] F. Maurer. Coordinating system development processes. In *KAW'96: Tenth Workshop on Knowledge Acquisition*, volume 2, pages 1–20, Banff, Canada, 1996. Modeling and Management.
- [26] T. Moe. Perspectives on traceability in food manufacture. *Trends in Food Science & Technology*, 9:211–214, 1998.
- [27] K. Mohan and B. Ramesh. Managing variability with traceability in product and service families. In *HICSS '02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 3*, pages 76–85, Washington, DC, USA, 2002. IEEE Computer Society.
- [28] R. Nagappan, R. Skoczylas, and R. P. Sriganesh. *Developing Web Services. Architecting and Developing Secure Web Services Using Java*. Wiley, Estados Unidos, 2003.
- [29] OASIS. Uddi version 3.0.2, 2004. Disponível em: <http://www.oasis-open.org/committees/uddi-spec/doc/spec/v3/uddi-v3.0.2-20041019.htm>.
- [30] Committee on Supply Chain Integration. *Surviving Supply Chain Integration*. National Academy Press, 2000.
- [31] L. U. Opara. Traceability in agriculture and food supply chain: A review of basic concepts, technologic implications, qne future prospects. *Food, Agriculture and Environment*, 1(1):101–106, 2003.
- [32] L. U. Opara and F. Mazaud. Food traceability from field to plate. *Outlook on Agriculture*, 30(4):239–247, 2001.
- [33] G. Z. Pastorello et al. Interoperability for gis document management in environmental planning. *Journal on Data Semantic III*, (LNCS 3534):100–124, 2005.

- [34] G. Z. J. Pastorello. Publicação e integração de workflows científicos na web. Mestrado em ciência da computação, Universidade de Campinas, Instituto de Computação, 2005.
- [35] F. A. C. Pinheiro and J. A. Goguen. An object-oriented tool for tracing requirements. *IEEE Softw.*, 13(2):52–64, 1996.
- [36] D. B. Pinto, I. Castro, and A. A. Vicente. The use of tic's as a managing tool for traceability in the food industry. *Food Research International*, 39:772–781, 2006.
- [37] B. Ramesh. Factors influencing requirements traceability practice. *Commun. ACM*, 41(12):37–44, 1998.
- [38] E. E. Santo and J. X. Medeiros. Coordenação e qualidade na cadeia da carne bovina: O caso da exigência da rastreabilidade. In *III Congresso Internacional de Economia e Gestão de Redes Agroalimentares*, pages 1–14, Ribeirão Preto, São Paulo, 2001.
- [39] C. E. Z. Schuchmann and H. Dewes. Actions for formulation of a protocol of traceability for erva-mate. In *IV Congresso Internacional de Economia e Gestão de Redes Agroalimentares*, pages 1–12, 2003.
- [40] J. F. Shapiro. *Modeling the Supply Chain*. Duxbury, 2001.
- [41] J. R. Stock. *Food Supply Chain Management*, capítulo 14 - The US Food Supply Chain, pages 211–220. Blackwell Publishing, Reino Unido, 2004.
- [42] Y. Taniguchi and N. Sagawa. Ic tag based traceability: System and solutions. In *ICDE '05: Proceedings of the 21st International Conference on Data Engineering (ICDE'05)*, pages 13–17, Washington, DC, USA, 2005. IEEE Computer Society.
- [43] W3C. Web services description language (wsdl) 1.1, 2004. Disponível em: <http://www.w3.org/TR/wsdl>.
- [44] S. Wakayama et al. Extendable product traceability system from small start. In *International Symposium on Applications and the Internet Workshops (SAINTW'06)*. IEEE Computer Society, 2006.
- [45] T. P. Wilson and W. R. Clarke. Food safety and traceability in the agricultural supply chain: using the internet to deliver traceability. *Supply Chain Management: An International Journal*, 3(3):127–133, 1998.

- [46] P. Xu and B. Ramesh. Supporting workflow management systems with traceability. In *HICSS '02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 3*, pages 91–101, Washington, DC, USA, 2002. IEEE Computer Society.
- [47] Bei Yi. A data model for moving objects. Mestrado em ciência da computação, Universidade de Campinas, July 2004.