

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**ELEVANDO A CAPACIDADE DE INTEGRAÇÃO
DE SISTEMAS DE MIDDLEWARE RFID
ATRAVÉS DO PROCESSAMENTO DE EVENTOS
COMPLEXOS DISTRIBUÍDOS ENTRE DIFERENTES
ORGANIZAÇÕES E NEGÓCIO**

LEONARDO ALBERNAZ AMARAL

TESE APRESENTADA COMO REQUISITO
PARCIAL À OBTENÇÃO DO GRAU DE
DOUTOR EM CIÊNCIA DA COMPUTAÇÃO
NA PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO GRANDE DO SUL

ORIENTADOR: PROF. DR. FABIANO PASSUELO HESSEL

PORTO ALEGRE
2011

Dados Internacionais de Catalogação na Publicação (CIP)

A485e Amaral, Leonardo Albernaz
 Elevando a capacidade de integração de sistemas de
 Middleware RFID através do processamento de eventos
 complexos distribuídos entre diferentes organizações e
 negócio / Leonardo Albernaz Amaral. – Porto Alegre, 2011.
 96 f.

 Tese (Doutorado) – Fac. de Informática, PUCRS.
 Orientador: Prof. Dr. Fabiano Passuelo Hessel.

 1. Informática. 2. RFID (Identificação por Rádio
 Frequência). 3. Sistemas Automáticos de Identificação.
 I. Hessel, Fabiano Passuelo. II. Título.

CDD 006.31

**Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS**



Pontifícia Universidade Católica do Rio Grande do Sul
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TERMO DE APRESENTAÇÃO DE TESE DE DOUTORADO

Tese intitulada "Elevando a Capacidade de Integração de Sistemas de Middleware RFID Através do Processamento de Eventos Complexos Distribuídos Entre Diferentes Organizações e Negócio", apresentada por Leonardo Albernaz Amaral, como parte dos requisitos para obtenção do grau de Doutor em Ciência da Computação, Sistemas Embarcados e Sistemas Digitais, aprovada em 15/07/2011 pela Comissão Examinadora:

Prof. Dr. Fabiano Passuelo Hessel -
Orientador

PPGCC/PUCRS

Prof. Dr. César Augusto Missio Marcon -

PPGCC/PUCRS

Prof. Dr. Cláudio Fernando Resin Geyer -

UFRGS

Prof. Dr. Eduardo Augusto Bezerra -

UFSC

Homologada em 30/08/11, conforme Ata No. 17, pela Comissão Coordenadora.

Prof. Dr. Fernando Luís Dotti
Coordenador.

PUCRS

Campus Central

Av. Ipiranga, 6681 - P. 32 - sala 507 - CEP: 90619-900

Fone: (51) 3320-3611 - Fax (51) 3320-3621

E-mail: ppgcc@pucrs.br

www.pucrs.br/facin/pos

AGRADECIMENTOS

Primeiramente gostaria de agradecer à minha esposa Raquel pela incondicional ajuda, motivação, compreensão, carinho e amor em todas as etapas da minha vida acadêmica. Foram anos difíceis e nunca deixastes de acreditar e me apoiar a seguir confiante nas minhas decisões. Do fundo do meu coração, muito obrigado! Espero poder sempre conseguir retribuir tanto amor e generosidade.

Gostaria de agradecer à minha família em geral e principalmente aos meus pais Paulo e Deolinda e a minha irmã Alessandra, pois tenho certeza que estão sempre ao meu lado me apoiando. Já faz alguns anos que me distanciei um pouco da convivência familiar e isso é um reflexo do esforço que venho fazendo para superar os meus desafios e atingir meus objetivos de vida. Mesmo assim, sempre encontramos uma forma de estarmos juntos. Muito obrigado por tudo!

Agradeço aos meus sogros Márcia e Romeu pelas palavras sábias e também pelos incondicionais conselhos e incentivos que me foram dados, sempre! Muito obrigado!

Agradeço ao meu orientador e amigo professor Fabiano Hessel por sua incansável dedicação, confiança, generosidade e carinho. Muitos anos se passaram até aqui. Aprendi e continuarei sempre aprendendo contigo, afinal, nossa relação pessoal e profissional permanece. Muito obrigado!

Agradeço à FINEP (Financiadora de Estudos e Projetos) pelo apoio financeiro prestado durante a realização do doutorado que originou esta Tese.

Agradeço ao professor e amigo Eduardo Bezerra por todas as oportunidades que me foram dadas tanto no mestrado quanto no doutorado. Sei que sempre poderei contar contigo. Obrigado!

Agradeço aos meus colegas de projeto e também aos demais integrantes do GSE/PUCRS (Grupo de Sistemas Embarcados) pela ótima convivência durante os anos de projeto. Foram inúmeras contribuições e discussões que engrandeceram meu trabalho de doutorado e as pesquisas acadêmicas feitas no laboratório.

Finalmente, agradeço a Deus por ter me dado saúde, paz de espírito, harmonia e principalmente força para perseverar e atingir meus objetivos de vida.

ELEVANDO A CAPACIDADE DE INTEGRAÇÃO DE SISTEMAS DE MIDDLEWARE RFID ATRAVÉS DO PROCESSAMENTO DE EVENTOS COMPLEXOS DISTRIBUÍDOS ENTRE DIFERENTES ORGANIZAÇÕES E NEGÓCIO

RESUMO

Com o passar dos anos a tecnologia RFID se tornou uma área de pesquisa e desenvolvimento extremamente estimulante e de rápida expansão. Embora existam sofisticados sistemas de software para RFID, muitos destes sistemas existentes estão baseados em tecnologias de middleware RFID com funcionalidades básicas de gerenciamento de dados de acordo com especificações do ALE (Eventos no Nível de Aplicação) segundo a *EPCglobal*. Estas funcionalidades do ALE agem como uma camada de integração entre dados RFID brutos (sem prévio processamento) e a lógica da aplicação. Nenhuma semântica de negócio é expressada nesta camada o que dificulta a contextualização de eventos. Este problema é considerado um *gap* tecnológico inerente às plataformas de middleware RFID que estão baseadas nas especificações do ALE.

Além das limitações dos mecanismos convencionais para gerenciamento de dados de eventos RFID, especificações existentes para tecnologias de software RFID (*EPC Network* segundo a *EPCglobal*) apresentam desafios nos serviços propostos para integração e compartilhamento de eventos entre diferentes organizações de negócio (ONS, DS e EPCIS). Tais serviços não permitem a utilização de esquemas de comunicação indireta de eventos o que dificulta a detecção de eventos complexos distribuídos.

Este trabalho apresenta um framework de software RFID cujo objetivo é elevar a capacidade de integração interorganizacional de sistemas de middleware RFID através de mecanismos de CEP (Processamento de Eventos Complexos) que cooperam entre si por meio de serviços de notificação de eventos. A tecnologia CEP é integrada a um sistema de middleware RFID que implementa especificações do ALE. O framework baseia-se no conceito de rede de CEPs para processamento e notificação de eventos complexos distribuídos. Cada nodo de processamento envolvido tem seu próprio mecanismo de CEP e a cooperação entre nodos provê um sistema de regras de negócio global que permite a decomposição e execução de regras de negócio em diferentes organizações.

Este trabalho é importante, pois, além de identificar limitações e problemas das tecnologias atuais de software RFID, resultados experimentais obtidos através de testes do framework proposto demonstram que as abordagens utilizadas na construção do framework melhoram atividades de projeto e desenvolvimento de aplicações RFID mantendo níveis aceitáveis de desempenho do mecanismo de CEP utilizado. Quando comparado com os principais sistemas RFID existentes, o framework apresenta melhorias e cumpre com os principais requisitos, tanto de gerenciamento quanto de integração de dados de eventos RFID. Além disso, o framework agrega novas funcionalidades para integração, gerenciamento e compartilhamento de dados de eventos RFID.

Palavras-chaves: *Framework de Software RFID, Middleware RFID, Processamento de Eventos Complexos, Serviço de Notificação de Eventos.*

IMPROVING THE INTEGRATION CAPABILITY OF RFID MIDDLEWARE SYSTEMS THROUGH THE PROCESSING OF DISTRIBUTED COMPLEX EVENTS AMONG DIFFERENT BUSINESS AND ORGANIZATIONS

ABSTRACT

Over the years the RFID technology has become an area of research and development extremely challenging and rapidly expanding. Although there are sophisticated RFID software systems, many of these systems are based on RFID middleware technologies with basic data management features according to specifications of the EPCglobal ALE (Application Level Events). These features of the ALE act as an integration layer between raw RFID data (data without prior processing) and the application logic. No business semantics is expressed in this layer, which hinders the contextualization of events. This problem is considered a technological gap of RFID middleware systems that are based on the specifications of the ALE.

Beyond the limitations of conventional mechanisms for data management of RFID events, existing specifications for RFID software technology (EPC Network according to EPCglobal) present challenges in the services proposed to integrate and share events among different business organizations (ONS, EPCIS and DS). Such services do not allow the use of indirect methods of event communication, which hinders the detection of distributed complex events.

This paper presents an RFID software framework whose goal is to raise the inter-organizational integration capability of RFID middleware systems. The systems integration is made through mechanisms of CEP (Complex Event Processing) that cooperate using event notification services. CEP technology is integrated with an RFID middleware system that implements the ALE specification. The framework is based on the concept of CEP network for processing and notification of distributed complex events. Each processing node has its own mechanism of CEP and the cooperation between nodes provides a system of global business rules, which allows the decomposition and execution of rules in different organizations.

This work is important because besides identify limitations of current RFID software technologies, experimental results of the proposed framework show that the approaches used in the framework improve project and development activities of RFID applications maintaining acceptable levels of performance of the CEP engine used. When compared with other RFID systems, the framework provides several enhancements and complies with key requirements for both management and integration of RFID event data. Moreover, the framework adds new features for integration, management and sharing of RFID event data.

Keywords: *RFID Software Framework, RFID Middleware, Complex Event Processing, Event Notification Service.*

LISTA DE FIGURAS

Figura 1.1 – Visão geral do trabalho	20
Figura 2.1 – Componentes principais do <i>EPC Network</i>	27
Figura 2.2 – Arquitetura básica de um middleware RFID	28
Figura 2.3 – Arquitetura básica do Serviço de Informação EPCIS	30
Figura 3.1 – Arquitetura básica de um sistema RFID	37
Figura 3.2 – Modelo para relacionamento dinâmico de ERs – DRER [WAN05]	40
Figura 3.3 – Arquitetura do Middleware RFID da SIEMENS [WAN05]	41
Figura 3.4 – Arquitetura do sistema SAP AII [BOR04]	46
Figura 3.5 – Camadas funcionais do middleware WINRFID [PRA06]	47
Figura 3.6 – Exemplo de uso da tecnologia CEP [PAL05].....	52
Figura 3.7 – Concentrador de dados RFID [PAL05].....	53
Figura 4.1 – Arquitetura do framework desenvolvido.....	60
Figura 4.2 – Exemplo de uma aplicação de captura contextualizando dados RFID	61
Figura 4.3 – Exemplo de regra de negócio para elevar o nível de abstração de dados RFID	62
Figura 4.4 – Modelos de integração de dados entre organizações [ZIE09].....	64
Figura 4.5 – Exemplo de uma arquitetura de CEP distribuída [DUT07]	66
Figura 5.1 – Módulos do Middleware RFID (<i>CloudRFID</i>)	67
Figura 5.2 – Módulo do Serviço de Informação EPCIS (<i>WindIS</i>)	69
Figura 5.3 – Visão geral do framework RFID	70
Figura 5.4 – Conceitos relacionados com CEP	73
Figura 6.1 – Visão geral do ambiente operacional do sistema.....	77
Figura 6.2 – Fluxo de dados do protótipo de teste.....	79
Figura 6.3 – Etapas do processo do armazém	80
Figura 6.4 – Etapas do processo das lojas.....	80
Figura 6.5 – Cenário 1: Tempos de processamento no armazém.....	82
Figura 6.6 – Cenário 1: Eventos detectados	82
Figura 6.7 – Cenário 2: Tempos de processamento no armazém.....	83
Figura 6.8 – Cenário 2: Tempos de processamento na loja 1.....	83
Figura 6.9 – Cenário 2: Tempos de processamento na loja 2.....	83
Figura 6.10 – Cenário 2: Eventos básicos detectados	84
Figura 6.11 – Cenário 2: Eventos complexos detectados	84

LISTA DE TABELAS

Tabela 2.1 – Tipos de eventos EPCIS e seus atributos suplementares.....	31
Tabela 3.1 – Abordagens sobre processamento de dados RFID.....	49
Tabela 3.2 – Plataformas para integração de dados RFID.	50
Tabela 5.1 – Construtores de Eventos do Drools.	73
Tabela 6.1 – Resumo das principais regras do cenário de teste.	81
Tabela 6.2 – Comparativo de funcionalidades arquiteturais.....	85

LISTA DE SIGLAS

ALE	<i>Application Level Events</i>
AOSE	<i>Agent Oriented Software Engineering</i>
API	<i>Application Programming Interface</i>
BI	<i>Business Intelligence</i>
CEP	<i>Complex Event Processing</i>
CORBA	<i>Common Object Request Broker Architecture</i>
CRM	<i>Customer Relationship Management</i>
DCOM	<i>Distributed Component Object Model</i>
DNS	<i>Domain Name System</i>
DRER	<i>Dynamic Relationship ER Model</i>
DS	<i>Discovery Service</i>
ECA	<i>Event Condition Action</i>
EPC	<i>Electronic Product Code</i>
EPCIS	<i>EPC Information Service</i>
ER	<i>Entity-Relationship model</i>
HTTP	<i>Hypertext Transfer Protocol</i>
JMS	<i>Java Message Service</i>
JPA	<i>Java Persistence API</i>
MARM	<i>Multi-Agent based middleware for Rfid asset Management system</i>
MDI	<i>Metaphysical Data Independence</i>
OLAP	<i>On-line Analytical Processing</i>
ONS	<i>Object Naming Service</i>
PASSI	<i>Process for Agent Societies Specification and Implementation</i>
PML	<i>Physical Markup Language</i>
POS	<i>Point of Sale</i>
RCEDA	<i>RFID Complex Event Detection Algorithm</i>
RDBMS	<i>Relational Database Management System</i>
RF	<i>Radio Frequency</i>
RFID	<i>Radio Frequency Identification</i>
RMI	<i>Remote Method Invocation</i>
SCM	<i>Supply Chain Management</i>
SMURF	<i>Statistical Smoothing of Unreliable RFID data</i>
SOAP	<i>Simple Object Access Protocol</i>
SQL	<i>Structured Query Language</i>
TCP	<i>Transmission Control Protocol</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
TI	<i>Information Technology</i>
WSDL	<i>Web Service Definition Language</i>
XML	<i>Extensible Markup Language</i>

SUMÁRIO

1. INTRODUÇÃO	15
1.1 Descrição do Problema	16
1.2 Motivações e Objetivos do Trabalho	18
1.3 Visão Geral do Trabalho Desenvolvido.....	20
1.4 Contribuições Originais da Tese	22
1.5 Organização do Documento.....	23
2. CONCEITOS E DEFINIÇÕES	24
2.1 Tecnologia RFID.....	24
2.1.1 Dispositivos e Comunicação RFID	24
2.1.2 Contexto Histórico	25
2.1.3 Arquitetura do EPC Network segundo a EPCglobal.....	26
2.1.4 Middleware RFID	28
2.1.5 Serviço de Informação EPC (EPCIS).....	29
2.1.6 Eventos EPCIS	30
2.2 Requisitos Funcionais de Middleware RFID.....	32
3. TRABALHOS RELACIONADOS	36
3.1 Visão Geral de Sistemas RFID.....	36
3.1.1 Arquitetura de um Sistema RFID	37
3.1.2 Requisitos de Sistemas RFID.....	38
3.1.2.1 Requisitos de Gerenciamento de Dados RFID	38
3.1.2.2 Requisitos de Integração de Dados RFID	39
3.2 Principais Tecnologias de Sistemas RFID	40
3.2.1 Sistemas para Gerenciamento de Dados.....	40
3.2.1.1 SIEMENS RFID Middleware.....	40
3.2.1.2 MDI-SMURF RFID Middleware	42
3.2.1.3 EPC-bitmap.....	42
3.2.1.4 UIUC'S RFID Middleware.....	43
3.2.1.5 REFILL	43
3.2.1.6 LIT Middleware	43
3.2.1.7 RF ² ID	44
3.2.1.8 MARM	44
3.2.2 Sistemas para Integração de Dados.....	45
3.2.2.1 SAP AII (Auto-ID Infrastructure).....	45
3.2.2.2 WINRFID	46
3.2.2.3 SUN JAVA System RFID Software.....	48
3.2.2.4 BEA WebLogic RFID	48
3.2.2.5 ACCADA	48
3.2.3 Análise Comparativa e Questões em Aberto	49
3.3 Processamento Semântico de Dados RFID	51
3.3.1 Princípios de um Processamento Semântico de Dados RFID.....	51
3.3.2 Abordagens para Processamento Semântico de Dados RFID	54
3.4 Resumo e Tendências Futuras.....	58
4. ABORDAGEM DE TRABALHO	59
4.1 Plano de Solução	59
4.2 Embasamento Teórico	60
5. DESCRIÇÃO DO TRABALHO	67
5.1 Plataforma de Referência	67
5.1.1 Middleware RFID (<i>CloudRFID</i>)	67
5.1.2 Serviço de Informação de Eventos EPCIS (<i>WindIS</i>)	68
5.2 Framework para Processamento e Notificação de Eventos Complexos.....	70
5.2.1 Visão Arquitetural e Funcionalidades do Framework	70
5.2.2 Drools como Motor de Regras	72
5.2.3 Eventos e Construtores de Eventos.....	72

5.2.4 Regras CEP	74
5.2.5 Modelando Eventos RFID	74
6. EXPERIMENTOS REALIZADOS	76
6.1 Aplicação da Plataforma de Referência em Casos Reais	76
6.1.1 Controle de Abastecimento de Frotas de ônibus.....	76
6.2 Testes do Mecanismo de CEP	78
6.2.1 Objetivos dos Testes.....	78
6.2.2 Metodologia dos Testes	78
6.2.3 Resultado dos Testes	82
6.3 Análise dos Resultados	84
6.3.1 Análise de Desempenho	84
6.3.2 Análise das Características Arquiteturais.....	85
6.3.3 Comparação com Trabalhos Existentes.....	86
7. CONCLUSÃO E TRABALHOS FUTUROS	88
7.1 Contribuições do trabalho.....	88
7.2 Publicações	88
7.3 Conclusões.....	89
7.4 Trabalhos futuros.....	91
REFERÊNCIAS BIBLIOGRÁFICAS	92

1. INTRODUÇÃO

Atualmente é possível perceber uma nova tendência em sistemas computacionais. Eles estão deixando de ser sistemas centralizados e orientados a computadores pessoais para se tornar uma computação portátil, baseada em dispositivos móveis e com diversidade de serviços e meios de comunicação. Essa evolução computacional é chamada “Computação Ubíqua” (do inglês, *Ubiquitous Computing*) e tem como objetivo melhorar a interação entre usuários e dispositivos eletrônicos através da integração do processamento de informações com atividades e objetos do cotidiano das pessoas.

A computação ubíqua se beneficia dos avanços da computação móvel e da computação pervasiva para prover uma computação onipresente (invisível e disponível para seus usuários), oferecendo acesso conveniente (a qualquer hora e em qualquer lugar) a informações relevantes. Segundo [WEI93], a computação móvel visa estimular a mobilidade física de serviços computacionais tornando o computador um dispositivo presente e disponível independente de sua localização. Já a computação pervasiva objetiva melhorar a integração entre sistemas computacionais e ambientes físicos, permitindo que dispositivos computacionais sejam embarcados no universo físico para obtenção de informações do meio em que estão inseridos.

A evolução da computação pervasiva ocorre de acordo com avanços tecnológicos de hardware e software e produz um ambiente computacional amigável e quase imperceptível ao usuário, o que de fato é o conceito de “pervasividade”.

Uma das definições mais atuais para pervasividade é “Internet das Coisas” (do inglês, *Internet of Things*). Internet das Coisas é uma área da computação pervasiva cujo objetivo é utilizar padrões da Internet para permitir a identificação única de objetos e dispositivos eletrônicos e com isso elevar a capacidade de integração destes objetos com sistemas de informação e processos de negócio¹. Tecnologias como código de barras, código 2D, RFID e redes de sensores sem fio, as quais podem ser diretamente associadas/acopladas a objetos físicos de valor agregado, têm capacidade funcional para transformar objetos comuns em “objetos inteligentes” a partir da integração destes objetos com sistemas computacionais. Desta forma, estes objetos fornecem dados que representam seus comportamentos em ambientes reais e que possam ser usados por sistemas na integração e automação de processos.

Uma tecnologia capaz de habilitar a pervasividade de serviços computacionais é RFID. Identificação por Radio Frequência ou RFID como é comumente conhecida, é uma tecnologia usada na identificação de itens, objetos e até mesmo seres humanos e que utiliza ondas de rádio frequência para comunicação entre objetos e dispositivos de leitura de dados.

RFID provê uma maneira automática de identificação sem necessidade de linha de visão nem contato físico direto com o objeto que está sendo identificado, elevando a velocidade do fluxo de informações e superando limitações de outros meios manuais de coleta de dados. Através do RFID, objetos do dia a dia se comportam como pontos de rede em sistemas pervasivos, visto que etiquetas RFID são capazes de enviar e receber dados de eventos facilitando o trabalho dos

¹ **Processos de negócio:** etapas ou atividades de negócio previamente estabelecidas e que podem ser automatizadas através de sistemas computacionais para aperfeiçoar a forma como o trabalho é realizado na organização.

sistemas computacionais nas atividades de localização, monitoramento e rastreabilidade de objetos.

Embora a tecnologia RFID tenha evoluído muito nos últimos anos, tanto em termos de dispositivos físicos e seus meios de comunicação, quanto na infraestrutura de software necessária para viabilizar todo sistema, sua real efetivação ainda está limitada a alguns desafios existentes. Boa parte desta limitação é causada por problemas nas atividades de **gerenciamento e integração de dados** provenientes de eventos de dispositivos de leitura RFID. Isso ocorre pois estes dados têm características particulares e necessitam de tratamento especial de acordo com o contexto onde estão inseridos. Além disso, muitas aplicações RFID existentes ainda são imaturas e foram projetadas de maneira *ad-hoc*, ou seja, foram criadas para prover determinadas funcionalidades sem atenção e planejamento necessários.

É importante perceber que implantar uma aplicação RFID conectada diretamente a dispositivos de leitura de dados é uma atividade trabalhosa e de risco. Isso ocorre, pois, dispositivos de leitura RFID fornecem uma enorme quantidade de dados brutos (sem prévio processamento) o que exige da aplicação funções específicas de processamento de dados para gerenciar/tratar o fluxo de dados gerados. Esta deficiência na capacidade de processamento muitas vezes acaba inviabilizando um projeto. Neste contexto, uma **arquitetura de software apropriada** é o termo chave para o sucesso de um sistema RFID.

Em uma arquitetura ideal, não basta ter capacidade para capturar e armazenar grandes volumes de dados de eventos, mas sim conseguir **capturar a informação contida nestes dados** e que **agregue valor operacional aos processos de negócio** envolvidos com a aplicação. Além disso, em uma rede de negócio, é indispensável que sistemas RFID provenham **funcionalidades de integração de dados entre diversas fontes de informação**.

1.1 DESCRIÇÃO DO PROBLEMA

Tentando oferecer sistemas com melhor suporte a atividades de **gerenciamento e integração de dados RFID**, pesquisadores e empresas envolvidas no assunto criaram o conceito de **middleware RFID** [FOS11] [LEO06] [EPC04]. Middleware RFID é um sistema de software que provê interfaces de comunicação e serviços para tratamento de dados entre dispositivos de leitura RFID e aplicações clientes. As funções básicas de um middleware RFID são: coleta de dados, roteamento de dados, gerenciamento de dados e gerenciamento de dispositivos de leitura [LEO06].

Muitas tecnologias de middleware RFID existentes oferecem funcionalidades básicas de gerenciamento de dados baseadas nas especificações do ALE [ALE11] (Eventos no Nível da Aplicação) segundo a *EPCglobal*². Estas funcionalidades não proveem suporte para atividades mais eficientes de processamento, como tratamento de dados de eventos complexos, e impõem uma elevada complexidade de projeto para a aplicação cliente pois esta fica diretamente responsabilizada pela transformação semântica de dados e também pela integração destes dados com processos de negócio. Por esta razão, desenvolvedores de aplicações RFID estão tendo que

² *EPCglobal*: organização sem fins lucrativos responsável pela definição, gerenciamento e disseminação mundial dos padrões da tecnologia EPC. Mais detalhes serão apresentados no Capítulo 2.

criar soluções de processamento em suas próprias aplicações para poderem derivar eventos significativos a partir de dados RFID entregues por sistemas de middleware RFID convencionais.

Este problema de integração ocorre porque ALE é uma especificação de software que provê funções para um conjunto padrão de operações de gerenciamento de dados de eventos, cujo resultado deste processamento é abstraído na forma de relatórios contendo eventos RFID ocorridos no ambiente de negócio que são encaminhados às aplicações requisitantes. Estas funções do ALE, além de filtrar e agregar dados de eventos gerados, agem como uma camada de roteamento entre dados de eventos RFID brutos (códigos EPC) e a lógica da aplicação (processos de negócio). Desta maneira, nenhuma semântica de negócio é expressada nessa camada o que é considerado um *gap* tecnológico inerente a maioria dos sistemas de middleware RFID baseados nas especificações do ALE.

Para elevar os benefícios do uso das tecnologias de middleware RFID com mínimo impacto no projeto da aplicação, é necessária a definição de uma camada de software adicional entre middleware e aplicações clientes. Esta camada deve gerenciar eventos RFID vindos do middleware e transformá-los em eventos de negócio que possam ser facilmente interpretados e integrados com a lógica da aplicação. Além disso, para reduzir o impacto de mudanças na lógica de negócio, processos de negócio devem ser modelados e gerenciados como entidades separadas da aplicação (módulos independentes e integrados) permitindo um gerenciamento mais eficaz da lógica de negócio envolvida sem alterar o código da aplicação.

Inicialmente e orientado pelas especificações da *EPCglobal*, a solução para este problema seria implementar na própria aplicação, um módulo de captura de eventos que oferecesse funções para capturar, transformar e armazenar dados de eventos RFID providos do middleware. No entanto, a aplicação de captura definida para o *EPC Network*³ não apresenta uma especificação efetiva para tais requisitos e carece de uma abordagem de inteligência de negócio capaz de abstrair do módulo de captura a heterogeneidade dos processos de negócio da aplicação. Sem esta abordagem de negócio é preciso criar novas aplicações de captura para cada cenário de negócio envolvido.

Como a tecnologia CEP (Processamento de Eventos Complexos) é muito utilizada no processamento semântico de grandes fluxos de dados de diversas fontes de informação, acredita-se que CEP possa ser uma opção adequada para elevar as funcionalidades da aplicação de captura nas atividades de processamento, integração e automação de processos de negócio em aplicações RFID. Além disso, CEP pode desempenhar o papel da aplicação de captura através da contextualização de eventos RFID em eventos de negócio (eventos *EPCIS*⁴) baseado em regras de negócio pré-estabelecidas.

Além dos problemas com ALE e aplicação de captura de eventos, especificações do *EPC Network* apresentam alguns desafios inerentes aos serviços propostos para compartilhamento de eventos RFID entre diferentes organizações (ONS, DS e EPCIS). Estes serviços não proveem esquemas de comunicação indireta de eventos o que dificulta a detecção de eventos complexos distribuídos. Isto ocorre porque EPCIS (Serviço de Informação EPC segundo o *EPC Network*)

³ **EPC Network:** especificações que lidam desde a captura de dados RFID até sua distribuição entre elementos de negócio envolvidos. Mais detalhes serão apresentados no Capítulo 2.

⁴ **Evento EPCIS:** é a forma contextualizada (através da semântica de negócio da aplicação RFID) de eventos RFID gerados pelo middleware. Mais detalhes serão apresentados no Capítulo 2.

suporta apenas endereçamento direto de recursos o que exige uma constante adaptação dos links de comunicação entre aplicações requisitantes (que consultam eventos de interesse) e fontes de informação. O endereçamento direto de recursos requer um fluxo de informação para cada relação entre produtor e consumidor de eventos degradando o sistema com a grande quantidade de comunicação necessária.

Dentro deste contexto de limitações das tecnologias de software RFID, o problema tratado neste trabalho concentra-se em **buscar melhorias para mecanismos de gerenciamento de dados RFID que elevem a capacidade de integração de sistemas de middleware RFID com aplicações de negócio de diferentes organizações**. Um framework de software RFID que minimize tais limitações deve estar baseado nos seguintes requisitos funcionais:

- A. *Mitigar os gaps existentes na especificação do ALE tanto para limitações na capacidade de processamento de eventos RFID complexos (gap no processamento semântico de dados) quanto na falta de contextualização de dados para integração com processos de negócio (gap de integração de dados entre middleware e repositório de dados EPCIS), provendo gerenciamento inteligente e flexível de dados RFID baseado em regras de negócio para contextualização de eventos;*
- B. *Suportar configurações dos mecanismos de gerenciamento e integração de dados de acordo com regras de negócio da aplicação, elevando a capacidade de manutenção e administração do sistema;*
- C. *Mitigar o gap existente na especificação do EPC Network nas limitações dos serviços de compartilhamento de dados RFID entre organizações (ONS, DS e EPCIS), os quais não permitem esquemas de interação indireta de eventos e dificultam a detecção de eventos complexos distribuídos;*
- D. *Prover uma arquitetura colaborativa que eleve a escalabilidade do sistema na integração de eventos.*

1.2 MOTIVAÇÕES E OBJETIVOS DO TRABALHO

As **motivações** deste trabalho estão relacionadas aos seguintes fatores:

- Embora a tecnologia RFID tenha evoluído muito nos últimos anos, ainda é necessário um grande esforço para tornar sua global efetivação uma realidade. Além disso, quanto mais se aprimora a tecnologia RFID, maior é a carência por arquiteturas e padrões unificados de projeto e desenvolvimento de sistemas RFID;
- A falta de soluções efetivas para problemas e limitações das tecnologias de software RFID existentes é outra motivação. Muitos dos problemas encontrados na literatura e também no uso da tecnologia RFID são pequenos *gaps* existentes nas especificações do *EPC Network*. Estes problemas, quando somados a novas tendências tecnológicas e requisitos de projeto, refletem em novos desafios, cujas soluções muitas vezes vão além das especificações existentes e precisam de propostas inovadoras que mitiguem estas deficiências tecnológicas e avaliem os padrões pré-estabelecidos;

- Por fim, esta tese é uma continuação dos trabalhos existentes do autor na área de middleware RFID e está sendo proposta como evolução tecnológica para os problemas encontrados no decorrer dos projetos relacionados.

Os **objetivos estratégicos** concentram-se nas seguintes atividades:

- Dominar a tecnologia de projeto de middleware RFID;
- Propor uma arquitetura de middleware RFID (plataforma de referência) que facilite o desenvolvimento de aplicações RFID;
- Aplicar a plataforma de referência em casos reais que possam validar o trabalho realizado e identificar problemas e deficiências das tecnologias usadas;
- Revisar os problemas e deficiências dos sistemas de software RFID existentes (estado da arte);
- Dominar e avaliar as especificações do *EPC Network*;
- Propor novos trabalhos e tendências futuras que mitiguem os problemas identificados no estado da arte e na plataforma de referência (proposta de tese – Framework de Software RFID);
- Desenvolver novos estudos e experimentos que embasem as tecnologias a serem usadas no framework proposto;
- Estender a plataforma de referência para suportar os requisitos do framework proposto;
- Aplicar o framework proposto em casos reais que possam validar o trabalho realizado e identificar problemas e deficiências das tecnologias usadas;
- Publicar artigos científicos que divulguem as ideias e os resultados conseguidos.

Assumem-se como **objetivos específicos**:

- Propor um **framework de software RFID** que seja inovador e que contribua tanto para a comunidade científica da área relacionada quanto para o andamento dos trabalhos realizados pelo autor com arquiteturas de middleware RFID;
- Definir e criar uma **arquitetura de software para processamento de eventos complexos** baseada na tecnologia CEP e que suporte os requisitos de processamento de dados do framework proposto;
- Definir e criar uma **arquitetura de software que provenha um ambiente para compartilhamento de eventos complexos entre organizações** e que suporte os requisitos de integração de dados do framework proposto;
- Integrar as arquiteturas criadas em um **framework de software RFID**;
- Validar o framework proposto através de aplicações e estudo de casos reais.

1.3 VISÃO GERAL DO TRABALHO DESENVOLVIDO

Este trabalho apresenta um framework de software RFID cujo objetivo é elevar a capacidade de integração interorganizacional de sistemas de middleware RFID através de mecanismos de CEP que cooperam entre si por meio de serviços de notificação de eventos. O framework baseia-se no conceito de CEPs colaborativos para processamento e notificação de eventos complexos distribuídos. Cada nodo de processamento envolvido tem seu próprio mecanismo de CEP e a cooperação entre nodos provê um sistema de regras de negócio global que permite a decomposição e execução de regras de negócio em diferentes organizações.

A Figura 1.1 apresenta uma visão geral do framework desenvolvido. O trabalho foi dividido em três etapas: (1) criação do middleware RFID e do serviço de informação EPCIS (plataforma de referência), (2) criação do módulo de CEP (processamento de eventos RFID complexos) e (3) extensão do módulo de CEP para criação do serviço de notificação de eventos RFID.

A etapa 1 concentrou atividades na definição e criação da plataforma de referência a qual é composta pelo middleware RFID (*CloudRFID*) e serviço de informação de eventos EPCIS (*WindIS*). Ambos os sistemas baseiam-se nos padrões de especificação da *EPCglobal* conforme apresentado em mais detalhes no capítulo 2. Através da plataforma de referência foi possível observar na prática alguns dos problemas e limitações dos sistemas de software RFID identificados por outros autores e que foram documentados neste trabalho por meio do estado da arte (Capítulos 3 e 4). Com os testes da plataforma de referência em aplicações de casos reais, identificou-se a carência de novas funcionalidades que originaram as etapas 2 e 3 deste trabalho.

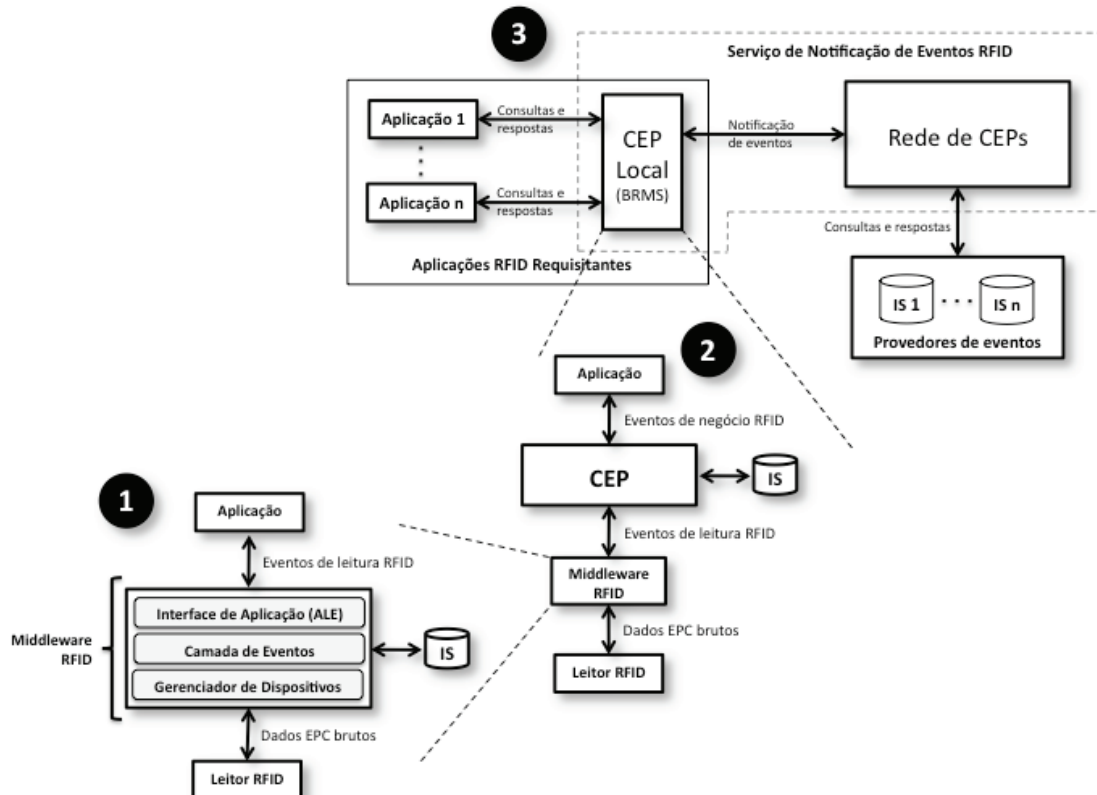


Figura 1.1 – Visão geral do trabalho

Na etapa 2 as atividades foram concentradas na criação de soluções e melhorias relacionadas ao **gerenciamento de dados de eventos RFID**. Nesta etapa foram agregadas novas funcionalidades à plataforma de referência e criou-se um mecanismo de CEP que está baseado em alguns serviços principais, como: **perfil organizacional**, **serviço único de captura de eventos e tratamento de eventos**.

O conceito de **perfil organizacional** permite definir tanto o contexto de negócio da aplicação quanto o contexto da infraestrutura lógica das fontes de informação usadas no processamento de eventos. No contexto de negócio é possível criar regras que representem a lógica do negócio envolvida no processo de contextualização de eventos RFID. O perfil concentra o armazenamento destas regras e facilita o processo de manutenção da lógica do negócio da aplicação. No contexto da infraestrutura lógica, o perfil permite configurar e definir a inter-relação entre fontes de informação (Middleware e nodos CEP) e repositórios de dados (*EPCIS*).

No serviço de captura de eventos, o mecanismo de CEP realiza o papel da aplicação de captura de eventos evitando que a aplicação tenha que ser reprojeta para prover essa funcionalidade para cada novo cenário de negócio a ser implantado. Para isso, criou-se o **serviço único para captura de eventos** o qual recebe eventos de diversas fontes de informação (N sistemas de middleware e N nodos CEP).

Já o módulo para **tratamento de eventos** está integrado ao **motor de regras** (sistema para gerenciamento de regras de negócio baseado no sistema *Drools* [DRO11]) e juntos realizam o processamento de eventos RFID complexos. Através do tratamento de eventos, eventos (RFID ou eventos *EPCIS*) são capturados pelo serviço de captura e processados pelo motor de regras. O resultado deste processamento pode ser a contextualização de um evento RFID em evento de negócio, a detecção de um evento complexo, o disparar de uma regra de negócio ou notificação de evento ou o armazenamento de eventos na memória de trabalho ou no repositório de dados para futura consulta.

Por fim, na etapa 3, as atividades foram concentradas na criação de soluções e melhorias relacionadas a **integração e compartilhamento de dados de eventos RFID**, o que resultou na extensão do mecanismo de CEP para suportar o conceito de **CEPs colaborativos para notificação de eventos**. Através deste conceito é possível decompor uma regra de negócio entre CEPs de diferentes organizações facilitando a detecção de eventos RFID complexos distribuídos. O serviço de notificação de eventos permite que aplicações compartilhem apenas eventos de interesse sem se preocupar com a busca por eventos ocorridos durante processos de detecção de eventos complexos.

Este trabalho é importante, pois, além de apresentar o estado da arte e identificar limitações e problemas das tecnologias atuais de software RFID e do próprio trabalho desenvolvido, resultados experimentais conseguidos com os testes do framework proposto demonstram que as abordagens utilizadas na construção do framework melhoram atividades de projeto e desenvolvimento de aplicações RFID mantendo níveis aceitáveis de desempenho do mecanismo de CEP utilizado. Além disso, o framework agrega novas funcionalidades para **integração, gerenciamento e compartilhamento de dados** de eventos RFID.

1.4 CONTRIBUIÇÕES ORIGINAIS DA TESE

As contribuições originais estão classificadas de acordo com os requisitos funcionais identificados para definição do framework proposto (Seção 1.1) e que norteiam os objetivos específicos da tese.

Para mitigar **os gaps existentes na especificação do ALE (gap no processamento semântico de dados e também de integração de dados entre middleware e repositório de dados EPCIS)**, nesta tese, usa-se a tecnologia CEP (Processamento de Eventos Complexos) para desempenhar o papel da aplicação de captura através da contextualização de eventos RFID em eventos *EPCIS* baseados em regras de negócio pré-estabelecidas. Embora usar mecanismos de CEP em aplicações RFID não seja considerada uma estratégia inovadora, nesta tese, a inovação se dá através da utilização do mesmo mecanismo de CEP tanto para processar eventos complexos de negócio, quanto para suprir as atividades da aplicação de captura e automatizar os processos de negócio da aplicação.

Em [FLO07], os autores propõem a implementação de uma aplicação de captura que forneça tais requisitos. No entanto, tal módulo não permite integração e configuração de diversas regras de negócio com elementos envolvidos no sistema e carece de uma inteligência de negócio para processamento de dados de eventos complexos e transformação semântica de eventos RFID que auxiliem na tomada de decisão.

Para suportar **configurações dos mecanismos de gerenciamento e integração de dados de acordo com regras de negócio da aplicação**, nesta tese, estendem-se as abordagens usadas em [KIM07] como uma forma inovadora de contextualizar tanto a automação de processos e configurações do sistema quanto o processamento de eventos distribuídos entre diversas organizações. Para isso, usa-se o conceito de perfil organizacional e serviço de notificação de eventos.

Em [KIM07] os autores apresentam um framework RFID ciente do contexto do negócio que é aplicado diretamente em processos de negócio e que se integra a instâncias do *EPC Network*. Embora a abordagem usada pelos autores seja eficiente na contextualização de dados e na automação dos processos de negócio, o mecanismo de CEP proposto não permite processamento de eventos complexos distribuídos entre organizações (diversas instâncias do *EPC Network*), o que é uma funcionalidade importante para os atuais requisitos da tecnologia RFID.

Para mitigar **o gap existente na especificação do EPC Network nas limitações dos serviços de compartilhamento de dados RFID entre organizações e elevar o nível de abstração da integração proposta nesta tese**, estende-se as abordagens usadas em [ZIE09] através de serviços de notificação de eventos para mitigar os problemas de integração de eventos RFID entre diferentes organizações. Com isso, é possível definir a decomposição de uma regra de negócio entre CEPs de diversas organizações e assim facilitar a identificação de eventos complexos distribuídos.

A abordagem proposta em [ZIE09] não especifica a integração do mecanismo de CEP com o processamento de eventos locais em uma organização, algo que também está sendo proposto nesta tese de doutorado. Sempre que uma regra de negócio é definida e decomposta entre nodos internos de uma organização, tal regra não é repassada para outros CEPs cooperantes e o processamento é local e hierárquico entre os nodos envolvidos [DUT07]. No entanto, quando uma

regra necessita de eventos interorganizacionais, a regra é declarada nos nodos CEP cooperantes. Dessa maneira, é possível decompor uma regra de negócio entre CEPs de diferentes organizações facilitando a detecção de eventos complexos distribuídos. O serviço de notificação de eventos, juntamente com a interface única para captura de eventos, reduz os problemas de sobrecarga na comunicação provocados pelas atividades de consulta de eventos e permite que aplicações compartilhem apenas eventos de interesse.

Por fim, para **prover uma arquitetura colaborativa que eleve a escalabilidade do sistema na integração de eventos**, a ideia de implementar um mecanismo de CEP colaborativo para processamento de eventos complexos através da execução de regras de negócio em tempo real está baseada nas propostas de [DUT07]. Cada nodo de processamento envolvido tem seu próprio mecanismo de CEP onde a cooperação entre eles provê um sistema de regras de negócio global. As ideias propostas em [DUT07] não especificam como o mecanismo de CEP monitora a execução de eventos complexos entre os nodos distribuídos. Além disso, os autores afirmam que as ideias propostas não foram implementadas.

Nesta tese, o perfil organizacional permite criar configurações individuais para cada nodo CEP onde a comunicação interorganizacional é feita de maneira ponto-a-ponto entre nodos envolvidos. Para isso, configura-se o serviço de notificação de eventos através de ações das regras criadas no perfil.

1.5 ORGANIZAÇÃO DO DOCUMENTO

Os capítulos seguintes estão organizados da seguinte maneira. O Capítulo 2 apresenta uma breve revisão da tecnologia RFID e identifica funcionalidades básicas e principais requisitos de sistemas de middleware RFID. Além disso, o Capítulo 2 esclarece os principais conceitos já tratados até o momento. O Capítulo 3 apresenta uma extensa revisão dos trabalhos relacionados com a presente tese de doutorado os quais foram divididos em dois grupos: “principais tecnologias RFID existentes (Seção 3.2)” e “abordagens para processamento semântico de dados RFID (Seção 3.3)”. O Capítulo 4 apresenta a abordagem usada no trabalho além do embasamento científico que justifique as hipóteses criadas. Já o Capítulo 5 apresenta a descrição do trabalho realizado relatando como a solução proposta foi implementada e suas funcionalidades. Já o Capítulo 6 apresenta os experimentos realizados e os resultados conseguidos. Por fim, o Capítulo 7 apresenta as considerações finais, as contribuições da tese, os artigos publicados e direções para trabalhos futuros.

2. CONCEITOS E DEFINIÇÕES

Este capítulo apresenta uma revisão da tecnologia RFID e aborda características e requisitos funcionais de sistemas de middleware RFID. Na seção 2.1 a tecnologia RFID é apresentada em mais detalhes abrangendo desde um contexto histórico até visões um pouco mais técnicas sobre dispositivos e meios de comunicação assim como serviços e elementos de infraestrutura de software RFID. Já a seção 2.2 apresenta detalhes sobre a tecnologia de middleware RFID tratando de funcionalidades e requisitos básicos da tecnologia.

2.1 TECNOLOGIA RFID

Identificação por Radio Frequência ou RFID é uma tecnologia usada na identificação de itens, objetos e até mesmo seres humanos e que utiliza ondas de rádio frequência para comunicação entre objetos e dispositivos de leitura de dados. Com isso, RFID provê uma maneira automática de identificação, elevando a velocidade do fluxo de informações e superando limitações de outros meios manuais de coleta de dados [QUA08].

Embora a tecnologia RFID exista há quase meio século [RIE06b], ela só começou a ganhar uma forte atenção há poucos anos atrás devido a convergência entre o baixo custo financeiro e o alto crescimento funcional e computacional das etiquetas RFID. Os responsáveis por sua popular difusão foram grandes mandatos impostos por empresas varejistas privadas e órgãos governamentais como *Wal-Mart*, *Target Metro AG* e *US DOD (Department of Defense)*, além das especificações da *EPCglobal*, as quais impulsionam o uso e o desenvolvimento da tecnologia RFID.

Atualmente RFID vem emergindo como uma tecnologia importante e revolucionária para uma gama de aplicações de negócio, tornando-se essencial em muitos aspectos e processos da vida moderna. Como exemplo desta evolução, temos o varejo, a logística, o transporte, a segurança e também áreas ligadas à saúde, como é o caso da área farmacêutica e hospitalar, as quais são nichos de mercado que têm apresentado grandes investimentos financeiros rumo à adoção do RFID como tecnologia padrão de identificação [NGA08] [ALE02] [UYS08] [KUU07].

Nestes diferentes domínios de mercado, RFID vem sendo aplicado como promessa tecnológica para elevar a eficiência dos negócios existentes e reduzir o *gap* econômico entre o mundo virtual dos sistemas de TI (Tecnologia de Informação) e o mundo real dos produtos e unidades de logística [ALE02].

Um sistema RFID é composto por três elementos principais: (1) uma etiqueta associada ao item/objeto que se deseja identificar, (2) um dispositivo de leitura usado para ler e extrair dados de etiquetas e (3) um sistema de aplicação que capture, filtre, armazene os dados lidos e os apresente a alguma entidade de negócio relacionada [LEO06] [THI06].

2.1.1 DISPOSITIVOS E COMUNICAÇÃO RFID

A comunicação na tecnologia RFID ocorre por meio de ondas de rádio frequência onde a troca de informação de uma etiqueta para um dispositivo de leitura ou vice-versa é realizada através de antenas RFID. Quando a etiqueta atinge o campo de rádio frequência gerado por

alguma antena RFID, esta etiqueta transmite seu código identificador (*ID*) para o leitor permitindo a identificação do objeto ou elemento etiquetado.

Etiquetas RFID contêm uma memória interna que armazena seu código de identificação. Um dispositivo de leitura RFID pode ler inúmeras etiquetas RFID simultaneamente e o alcance de leitura pode variar de poucos centímetros até alguns metros. A capacidade de leitura depende do tipo de etiqueta utilizada, da potência definida para o dispositivo de leitura, da interferência de outros dispositivos de rádio frequência, entre outros fatores.

Etiquetas RFID podem ser categorizadas em **ativas**, **semi-passivas** e **passivas**. Uma etiqueta ativa possui bateria interna a qual fornece energia para a comunicação necessária. Etiquetas ativas podem significativamente melhorar o alcance da comunicação e também iniciar a comunicação com algum dispositivo de leitura. Já etiquetas semi-passivas operam apenas quando recebem energia de algum leitor, ou seja, não tomam a iniciativa de iniciar uma atividade de comunicação. Etiquetas ativas e semi-passivas são considerados dispositivos caros quando comparados com etiquetas passivas. Este alto custo financeiro restringe o uso deste tipo de etiqueta a objetos de grande valor material agregado ou em grandes objetos que precisam ser rastreados por longas distâncias e que necessitam de outras tecnologias de comunicação embarcadas.

Ao contrario das etiquetas ativas e semi-passivas, etiquetas passivas trabalham sem necessidade de prover sua própria energia visto que se alimentam dos campos eletromagnéticos fornecidos pelos dispositivos de leitura RFID. Os sinais de rádio frequência dos leitores energizam as etiquetas RFID e habilitam suas funções de leitura, escrita e transmissão de dados. Etiquetas passivas têm seu alcance limitado pela potência que ela pode receber das ondas eletromagnéticas dos leitores RFID. Etiquetas passivas apresentam um custo financeiro mais acessível e podem ser utilizadas em larga escala.

Etiquetas RFID têm diferentes capacidades de armazenamento, são compostas por diferentes materiais e possuem diferentes tamanhos e formatos. Uma etiqueta pode ter antenas externas de metal, antenas embarcadas ou até mesmo antenas impressas. As frequências usadas na tecnologia RFID estão limitadas de acordo com padrões de regulamentação. A escolha de uma determinada frequência depende dos requisitos da aplicação, tais como: capacidade de absorção em líquidos, capacidade de reflexão a determinadas superfícies, densidade da etiqueta, necessidade de energia de alimentação, tamanho das etiquetas, variação da temperatura de exposição, velocidade e taxa de transmissão de dados, entre outros.

2.1.2 CONTEXTO HISTÓRICO

Recentemente o EPC ou Código Eletrônico de Produto [EPC04] – um código mundial para definição e identificação de objetos físicos – tornou-se tópico de grande interesse não só para entidades de pesquisa mas para indústrias e sociedade em geral. A rápida difusão do EPC se deu, principalmente, com ajuda do *Auto-ID Center*, um projeto que foi criado para desenvolver padrões dirigidos à tecnologia RFID.

Auto-ID Center foi fundado em 1999 no Instituto de Tecnologia de Massachusetts (*MIT*) em cooperação com diversas indústrias patrocinadoras. Pesquisadores do projeto *Auto-ID* criaram o padrão EPC para garantir interoperabilidade de dados entre sistemas de aplicação em cadeias de

distribuição de produtos. O código EPC foi concebido como novo esquema de numeração de objetos capaz de suportar a enumeração de todos os tipos e nomes de objetos existentes e também objetos futuros. A tecnologia EPC não foi projetada para substituir mas sim para integrar esquemas de numeração e identificação existentes.

A ideia por trás da tecnologia EPC é criar a “Internet das Coisas” a qual consiste em diversos componentes de hardware e software e uma família completa de padrões e especificações. Com isso é possível prover tecnologia de identificação para quase todos os tipos de objetos, melhorando a integração das informações de objetos com sistemas de informação empresariais.

Em outubro de 2003 o projeto *Auto-ID* foi transformado em uma rede de pesquisas internacionais formada pelo *Auto-ID Labs*, o qual se concentra na tecnologia e em pesquisas orientadas a aplicações, e também pela *EPCglobal*, uma organização sem fins lucrativos responsável pela definição e gerenciamento dos padrões EPC. Nos anos que se seguiram, a tecnologia EPC tornou-se uma base técnica para diversas iniciativas privadas (ex. *Wall-Mart* e *Metro Group*) assim como para indústria e organizações governamentais, tais como *Pfizer* e *US DoD*.

Dentro da comunidade *EPCglobal*, diversos grupos de trabalho discutem requisitos técnicos, desenvolvem casos de negócio e estabelecem padrões que promovem a disseminação mundial da tecnologia EPC. O framework comum para estas atividades é a arquitetura do *EPC Network* [LEO06] [EPC11], a qual inclui especificações que lidam desde a captura dos dados RFID até sua distribuição entre elementos de negócio envolvidos.

2.1.3 ARQUITETURA DO EPC NETWORK SEGUNDO A EPCGLOBAL

EPC Network é um framework conceitual criado pela *EPCglobal* para operar juntamente com padrões EPC definidos para etiquetas RFID. Especificações do *EPC Network* definem modelos de serviços e coleções de padrões inter-relacionados tanto para software e hardware quanto para interfaces de dados. Estas especificações não definem componentes individuais mas sim regras e interfaces que devem ser implementadas.

A Figura 2.1 representa a arquitetura do *EPC Network*. Esta arquitetura é composta pelos seguintes elementos (de baixo para cima): etiquetas RFID, dispositivos de leitura RFID, middleware RFID, camada de serviço de informação EPC (EPCIS), ONS e DS e aplicações empresariais.

Etiquetas e dispositivos de leitura são componentes indispensáveis em qualquer sistema RFID. Eles são considerados como fontes geradoras de dados do sistema e provêm informações que são consumidas pelos demais elementos envolvidos na arquitetura. A comunicação com dispositivos de leitura ocorre por meio de interfaces definidas pelos protocolos de leitura. Estas interfaces definem e controlam a entrega de dados dos leitores RFID para o Middleware RFID ou diretamente para a aplicação.

Middleware RFID ou *Savant*, como foi inicialmente chamado [FLO07] [LEO06], é uma camada de software que provê interfaces de comunicação entre dispositivos de leitura RFID e aplicações clientes que solicitam dados de eventos RFID. Este tipo de software é importante pois filtra e trata o grande volume de dados gerados pelos leitores RFID [FLO04] [LEO06].

A integração de dados entre middleware e aplicação cliente ocorre por meio da interface de eventos ALE (Eventos no Nível da Aplicação), a qual define funções de captura e entrega de dados de eventos tratados pelo middleware RFID (Figura 2.1). As interfaces do ALE devem ser implementadas pela aplicação de captura EPCIS a qual supervisiona as operações do middleware e provê contexto de negócio aos eventos RFID capturados, transformando eventos RFID em eventos EPCIS que são armazenados nos repositórios de eventos providos pelo EPCIS.

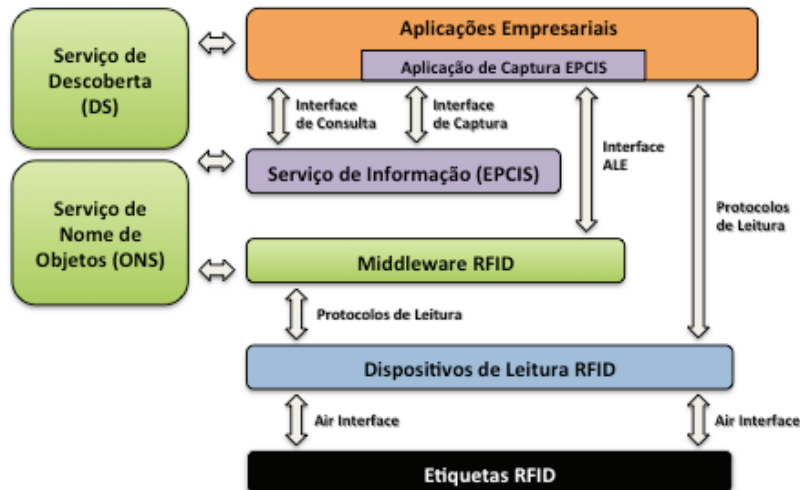


Figura 2.1 – Componentes principais do *EPC Network*.

EPCIS ou Serviço de Informação EPC é um repositório de informações a respeito de eventos e transações geradas pelo middleware RFID. O EPCIS permite que aplicações capturem (através da aplicação de captura EPCIS), armazenem (através da interface de captura) e consultem (através da interface de consulta) os dados lidos e tratados pelo middleware em seus ciclos de eventos⁵. Além disso, o EPCIS agrega informações de negócio aos eventos lidos, facilitando o entendimento e a integração com sistemas legados.

O ONS ou Serviço de Nome de Objetos apresenta um conceito semelhante ao DNS (Sistema de Nome de Domínios) usado na Internet indicando qual servidor possui informações relativas a determinado objeto EPC. Ou seja, permite que aplicações requisitantes tenham acesso à localização das fontes de informação EPC.

Já o DS ou Serviço de Descoberta, em contraste com o ONS, é um conjunto de serviços que quando integrado ao ONS, permite que usuários encontrem e acessem todas as fontes de dados (EPCIS) referentes a objetos específicos [LEO06] [EPC11].

As abstrações oferecidas pela hierarquia de elementos da arquitetura do *EPC Network* permitem um isolamento dos detalhes técnicos e funcionais entre as camadas envolvidas. Basicamente existem duas grandes abstrações, uma abstração de baixo nível (nível dos dispositivos físicos) e uma abstração de alto nível (nível dos processos de negócio). A primeira abstração é provida pela camada de middleware RFID a qual isola as camadas superiores da arquitetura do *EPC Network* do conhecimento necessário para gerenciar protocolos de *RF* e dispositivos usados na infraestrutura física RFID. Além disso, o middleware permite também uma

⁵ **Ciclos de eventos:** menor unidade de leitura entre middleware RFID e aplicações. São relatórios de eventos parametrizados pelas aplicações de captura EPCIS.

abstração das particularidades operacionais dos dispositivos de leitura RFID (como estes realizam atividades de leitura, tratamento e roteamento de dados de eventos observados).

Já a camada de abstração de alto nível é fornecida pelo serviço de informação EPCIS. Este serviço incorpora informações semânticas de negócio referentes aos processos de negócio envolvidos na coleta de dados EPC provendo um histórico de eventos observados. O EPCIS isola aplicações dos detalhes de execução dos processos envolvidos no negócio.

2.1.4 MIDDLEWARE RFID

Um sistema de middleware tradicional é considerado como sendo uma camada de software que conecta e gerencia componentes de aplicação em ambientes de execução distribuídos. Ele tem a capacidade e a responsabilidade de esconder e abstrair dos programadores muitos dos complexos detalhes envolvidos na programação distribuída.

O desenvolvimento de aplicações baseado em sistemas de middleware torna-se muitas vezes uma tarefa complexa. Isso ocorre devido à necessidade de um grande conhecimento, tanto dos requisitos impostos pelas funcionalidades das aplicações a serem desenvolvidas quanto dos requisitos impostos pelas próprias tecnologias de middleware disponíveis (*CORBA*, *RMI*, *DCOM*, *JGroups*, entre outros). No caso da tecnologia RFID, adotar a utilização de um middleware tem inúmeros benefícios inerentes às próprias funções básicas de um sistema de middleware tradicional. No entanto, usar um middleware tradicional em aplicações RFID apresenta uma série de limitações e desafios, principalmente quanto à capacidade de adaptar funções básicas em funcionalidades específicas para ambientes RFID.

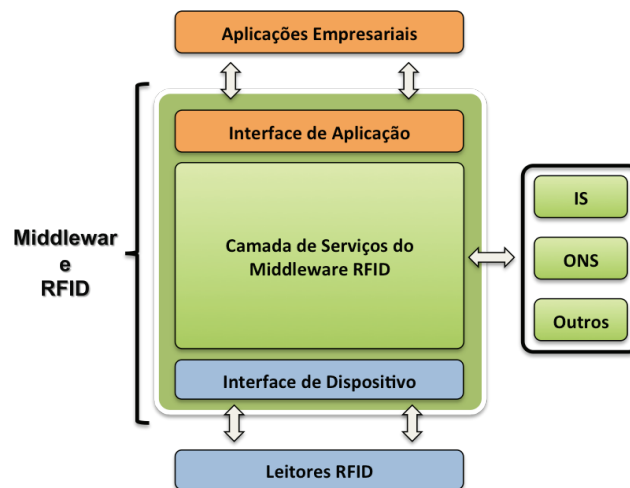


Figura 2.2 – Arquitetura básica de um middleware RFID.

Segundo [LEO06], um sistema de middleware RFID é uma camada de software que está localizada entre dispositivos de leitura e aplicações clientes (Figura 2.2), as quais devem operar no topo ou dentro do próprio ambiente operacional do middleware. O middleware deve realizar a troca de informações (requisições e repostas) entre aplicações e dispositivos de leitura, respectivamente. Para isso, ele deve oferecer um conjunto de funções específicas.

Conforme pode ser visto na figura 2.2, existem duas interfaces principais que um sistema de middleware RFID deve disponibilizar: uma interface de aplicação (alto nível) e uma interface de dispositivos (baixo nível). Ambas as interfaces visam a interoperabilidade com outros sistemas. A

primeira deve fornecer comunicação com aplicações de negócio em geral. Enquanto que a segunda é responsável pela comunicação com dispositivos físicos de leitura (RFID e outros tipos de dispositivos).

Além das duas interfaces principais, a tecnologia de middleware RFID deve fornecer também uma Camada de Serviços responsável por funções específicas, tais como: integração com EPCIS (para armazenamento de informações), interfaces de consulta para *ONS* (para que objetos possam ser localizados), entre outros. Sendo que cada função pode ser combinada para prover novas funcionalidades [LEO06].

Em sistemas simples de RFID como é o caso de um controle de acesso, muitas vezes não há necessidade de se utilizar a tecnologia de middleware RFID pois o escopo da aplicação é resumido e os dados RFID são consumidos apenas por uma única aplicação. No entanto, já em cenários mais complexos, que necessitam de modernas soluções de negócios e infraestruturas heterogêneas de dispositivos, tais como, gerenciamento e logística de cadeias de produção e distribuição, existe uma visão arquitetural distribuída que exige do sistema uma maior capacidade de integração (comunicação, interconexão e compartilhamento de dados) entre diferentes entidades de negócio. Neste caso, diversos dispositivos de leitura espalhados pelos elementos que compõem o sistema físico de distribuição (fornecedor, fabricante, distribuidor, atacadista, varejista e o cliente) capturam dados RFID lidos os quais devem ser disseminados para inúmeras e diferentes aplicações. Esse tipo de cenário de negócio introduz a necessidade de uma infraestrutura de informação RFID que abstraia as interfaces proprietárias dos dispositivos de leitura (leituras RFID), provendo gerenciamento e configuração, assim como filtragem, agregação e integração dos dados capturados.

Do ponto de vista do desenvolvimento da aplicação, é importante elevar o nível de abstração dos dados lidos e traduzi-los em um nível mais significativo, algo que pode ser conseguido com a utilização de um sistema de middleware RFID e serviços de informação EPC (*EPCIS*). Nesse tipo de abstração, a detecção de uma etiqueta RFID com um identificador "1112.5657298", feita por uma leitora "XYZ", deve resultar em um evento de negócio correspondente e que agregue valor à aplicação, como por exemplo, uma informação referente a uma carga de pneus de bicicleta que acabou de desembarcar do caminhão 50 e já está estocada no armazém 12, ou até mesmo disparar uma regra de negócio indicando que algum produto está sendo roubado.

2.1.5 SERVIÇO DE INFORMAÇÃO EPC (EPCIS)

Serviço de informação EPC é o módulo responsável pelo armazenamento de dados de eventos RFID provindos do middleware. Ele oferece uma estrutura de dados que permite a contextualização de eventos ocorridos, transformando eventos RFID em eventos EPCIS.

No EPCIS, dados de eventos RFID são armazenados através de uma *interface de captura* e disponibilizados para consulta por meio de uma *interface de consulta* (Figura 2.3). Desta maneira, aplicações podem requisitar dados RFID diretamente do middleware (dados brutos sem breve tratamento) ou via EPCIS (dados contextualizados). Informações RFID contidas no EPCIS são utilizadas como histórico de eventos e ajudam aplicações de negócio a entender detalhes físicos e operacionais dos eventos ocorridos.

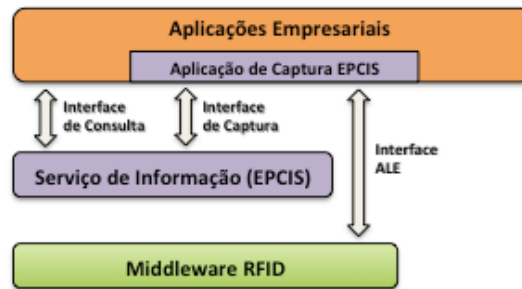


Figura 2.3 – Arquitetura básica do Serviço de Informação EPCIS.

O objetivo do serviço de informação EPC é definir métodos e estruturas de dados que agreguem semântica de negócio aos dados filtrados pelo middleware RFID. O EPCIS, juntamente com o middleware RFID, é capaz de compartilhar informações de interesse entre parceiros de negócio ou usuários da aplicação, reduzindo tempo e custo de acesso a dados e melhorando o processo de compartilhamento das informações RFID pertinentes ao negócio.

O EPCIS lida com dados semânticos, ou seja, com o histórico de dados de eventos RFID contextualizados através de atributos e dados de referência que caracterizam passos específicos de processos de negócio da aplicação (eventos EPCIS). Já um sistema de middleware RFID lida com dados RFID brutos com características de tempo real (dados de eventos RFID sem prévio tratamento). Um exemplo de evento gerado pelo middleware é: “No dispositivo de leitura L, entre os instantes T1 e T2, foram lidos os seguintes códigos EPC A, B e C.” Já um exemplo de evento EPCIS é: “Uma caixa com EPC X contendo 100 unidades do produto Y, foi transportada com sucesso do fornecedor A para o distribuidor B e expediu a nota fiscal W”.

2.1.6 EVENTOS EPCIS

Eventos RFID provindos de sistemas de middleware contêm o **número EPC da etiqueta lida (ID da etiqueta)**, o **timestamp da observação do evento** e o **identificador do dispositivo de leitura (ID do leitor RFID)** que gerou o evento. No entanto, estes eventos não fornecem informações a respeito do contexto de negócio do ambiente que gerou o evento. Estes atributos são agregados aos eventos RFID pelo próprio EPCIS através de informações descritivas do ambiente operacional de negócio e que geram eventos do tipo EPCIS.

Um evento EPCIS provê informações de negócio importantes, tais como, dia e hora da ocorrência do evento (*timestamp*), local da ocorrência do evento e alguns atributos transacionais relacionados ao processo de negócio envolvido.

Informações de eventos EPCIS estão relacionadas com características do tipo: **O QUE** (que produto/item está relacionado ao evento ocorrido), **ONDE** (o local de ocorrência do evento), **QUANDO** (o timestamp de ocorrência do evento) e **PORQUE** (etapa no processo de negócio envolvido com a aplicação). Eventos EPCIS são formados por cinco tipos de eventos (um evento genérico e quatro subclasses específicas de eventos RFID) e mais quatorze atributos que ampliam o significado dos cinco tipos principais de eventos os quais representam um conjunto de vocabulários específicos para dados de eventos EPCIS. Os tipos de eventos EPCIS especificados pela *EPCglobal* e seus atributos suplementares (requeridos e opcionais) estão apresentados na tabela 2.1.

Tabela 2.1 – Tipos de eventos EPCIS e seus atributos suplementares.

		Evento EPCIS	Evento de Objeto	Evento de Agregação	Evento de Quantidade	Evento de Transação	
Quando?	eventTime	R	R	R	R	R	Timestamp da observação do evento pela aplicação de captura
	recordTime	O	O	O	O	O	Timestamp da gravação do evento no EPCIS (repositório)
	eventTimezone-Offset	R	R	R	R	R	Timestamp para compensação de fuso horário pela aplicação de captura
O que?	epcList		R			R	Lista de EPCs observados
	parentID			O		O	ID do pai dos EPCs contidos no campo epcList
	childEPCs			R			Lista de EPCs dos objetos contidos no evento
	epcClass				R		ID do EPC que identifica o grupo de objetos identificados no evento
	quantity				R		Número de EPCs observados e identificados no campo epcClass
	action		R	R		R	Descreve o relacionamento do evento com o ciclo de vida dos EPCs nomeados no evento *
Onde?	readPoint		O	O	O	O	Nome e identificador do leitor RFID que gerou o evento
	bizLocation		O	O	O	O	Nome do local de ocorrência do evento
Porque?	bizStep		O	O	O	O	Etapa do processo de negócio no momento que ocorreu o evento
	disposition		O	O	O	O	Estado dos objetos associados aos EPCs na etapa do processo de negócio no momento que ocorreu o evento
	bizTransactionList		O	O	O	R	Lista de transações associadas que definem o contexto do evento
R - requerido, O - opcional							
* Uma ação pode ser um dos seguintes atributos: ADD, OBSERVE ou DELETE. ADD indica o primeiro evento em um ciclo de vida de produto onde EPCs são associados ao produto pela primeira vez. OBSERVE indica o que aconteceu entre o primeiro e último evento em um ciclo de vida de produto. E DELETE indica o último evento em um ciclo de vida de produto.							

Um evento EPCIS básico (*EPCISEvent*) define um tipo genérico de evento. Este tipo evento pode ser estendido por outros quatro eventos para definir um conjunto padrão de eventos que ocorre no compartilhamento de informações entre elementos de negócio em uma rede EPC (*EPC Network*). Conforme a Tabela 2.1, os tipos de eventos EPCIS especificados pela *EPCglobal* são:

- **Evento EPCIS (*EPCISEvent*):** define um tipo genérico de evento;
- **Evento de Objeto (*ObjectEvent*):** representa um evento simples que aconteceu para um ou mais EPCs (um ou mais objetos RFID). Este tipo de evento encapsula

informações sobre **o que** aconteceu, **quando** aconteceu e **onde** aconteceu determinado evento (coluna Evento de Objeto da tabela 2.1).

- **Evento de Agregação (*AggregationEvent*):** representa um evento que aconteceu para um ou mais EPCs (um ou mais objetos RFID) e que encapsula informações relacionadas com a agregação física de objetos identificados (ex. um conjunto de objetos agregados em um container). As informações relacionadas com este tipo de evento estão relacionadas com **o que** aconteceu, **quando** aconteceu e **onde** aconteceu determinado evento (coluna Evento de Agregação da tabela 2.1).
- **Evento de Quantidade (*QuantityEvent*):** representa um evento ocorrido que está relacionado a uma quantidade de objetos/itens pertencentes à mesma classe de objetos EPC, sem levar em consideração o tipo de objeto. Este tipo de evento encapsula informações sobre **o que** aconteceu, **quando** aconteceu e **onde** aconteceu determinado evento (coluna Evento de Quantidade da tabela 2.1).
- **Evento de Transação (*TransactionEvent*):** representa um evento que aconteceu para um ou mais EPCs (um ou mais objetos RFID) que estão associados ou desassociados a uma ou mais transações de negócio. Este tipo de evento encapsula informações sobre **o que** aconteceu, **quando** aconteceu, **onde** aconteceu e **porque** aconteceu determinado evento (coluna Evento de Transação da tabela 2.1).

De acordo com especificações da *EPCglobal*, eventos *EPCIS* devem lidar com dois tipos abstratos de dados: **event data** e **master data**. **Event data** são dados dinâmicos que são alterados conforme a ocorrência de eventos e estão relacionados ao ciclo de vida de objetos RFID. Estes dados são representados através de tipos específicos de eventos juntamente com seus atributos (Tabela 2.1). Estes dados devem ser capturados através da interface de captura do *EPCIS* e disponibilizados para consulta através da interface de consulta do *EPCIS* (Figura 2.3).

Já **master data** são informações adicionais que devem ser usadas para acrescentar contexto de interpretação de eventos, tais como, nome e endereço de locais de negócio ou nome, descrição e data de validade de determinado produto. Dados relacionados com **master data** devem ser disponibilizados para consulta através da interface de controle de consultas do *EPCIS*. No entanto, a maneira (o método) como eles devem ser adicionados no *EPCIS* não é especificada pela *EPCglobal*.

2.2 REQUISITOS FUNCIONAIS DE MIDDLEWARE RFID

Inicialmente, um middleware convencional é uma camada de software intermediária entre aplicação e o restante do sistema. Essa camada tem a função de oferecer abstrações para o desenvolvimento de aplicações distribuídas, bem como para o mascaramento da heterogeneidade das redes, do hardware envolvido, dos sistemas operacionais, das linguagens de programação, entre outros. Apesar dessa definição mais abrangente e genérica, normalmente uma plataforma de middleware é específica para um domínio de aplicação, como por exemplo, middleware para RFID, middleware para interoperabilidade de redes, entre outros. Para cada domínio de aplicação, uma plataforma de middleware, formada por *APIs*, frameworks e serviços provê abstrações para desenvolvimento de outras aplicações.

Diferente de outras instâncias de middleware, um middleware RFID é basicamente projetado para operar nas extremidades das redes empresariais e não no centro dos sistemas de TI [PAL05]. Isso requer o uso de redes distribuídas além de uma infraestrutura descentralizada dos sistemas de informação envolvidos.

Um middleware RFID tem a responsabilidade de mover dados entre pontos de transação de negócio, ou seja, ele é responsável pelo fluxo de dados de eventos entre dispositivos de leitura e aplicações de negócio. Para isso, um middleware RFID deve ser capaz de realizar algumas funções básicas, tais como: coleta de dados, roteamento de dados, gerenciamento de dados e gerenciamento de dispositivos [LEO06].

Na coleta de dados, um sistema de middleware é responsável pela extração, agregação, limpeza/suavização e filtragem de dados oriundos de diversos dispositivos de leitura RFID. Ele trabalha como uma área de armazenamento temporário entre o grande volume de dados brutos que são coletados e os reais dados de interesse solicitados pelas aplicações. Sem esse *buffer* temporário de dados – o qual filtra os eventos que realmente importam para o negócio – torna-se muito custosa para aplicação as atividades de gerenciamento de dados.

Por sua vez, no roteamento de dados o middleware facilita a integração de dados entre leitores RFID e aplicações de negócio. Para isso o middleware direciona os dados solicitados para os destinos corretos, ou seja, ele determina “o que vai e para onde vai”.

Nas atividades de gerenciamento de dados o middleware é usado para identificar e disparar eventos baseados em regras de negócio pré-determinadas e com isso elevar o significado dos dados RFID e integrá-los com processos de negócio. Dessa maneira, é possível agregar valor operacional à tomada de decisão e assim facilitar a integração de dados.

Já no gerenciamento de dispositivos o middleware é usado para monitorar e coordenar dispositivos de leitura oferecendo uma camada de abstração de hardware capaz de desacoplar das aplicações toda a complexidade de interação com a infraestrutura física RFID. Com isso é possível elevar a produtividade do sistema e facilitar a manutenção e a integração de novos dispositivos à rede RFID.

Um middleware RFID, como artefato de software, pode ser implantado e executado em qualquer ambiente e dispositivo computacional que atenda aos requisitos mínimos para sua instalação e execução. Ou seja, um sistema de middleware RFID pode ser executado em um leitor RFID ou em um cluster de servidores dedicados.

Para atender necessidades comuns de diferentes aplicações RFID é desejável que middleware RFID satisfaça os seguintes requisitos funcionais [ALE11] [THI06] [FLO04] [BO06]:

- **Leitura e escrita em etiquetas RFID:** Podem existir informações implícitas agregadas em uma etiqueta RFID que além de representarem sua identificação, informam também algo a mais sobre determinado objeto. O Middleware RFID deve prover uma interface padrão para a aplicação realizar operações de leitura e escrita em etiquetas RFID, independente do formato de dados e tipo de etiqueta usada.
- **Gerenciamento de configuração e falhas:** Devido ao grande número de dispositivos de leitura RFID existentes no mercado e também da dependência que processos de negócio podem ter em relação ao bom funcionamento desses equipamentos de leitura,

torna-se indispensável um mecanismo para controle, manutenção e gerenciamento de possíveis defeitos que possam ocorrer. Além disso, é desejável que tal gerenciamento possa ser realizado remotamente e a partir dos próprios sistemas de informação existentes nas organizações.

- **Agregação de dados:** Do ponto de vista das aplicações empresariais, é importante elevar o nível de abstração dos dados RFID lidos e traduzi-los em um nível mais significativo e que acrescente valor operacional ao negócio da aplicação. Um exemplo prático é utilizar um sistema de middleware para agregar dados de eventos de leitura RFID e detectar eventos que acusam a entrada ou saída de um objeto por um portal de antenas RFID. Isso possibilita que determinada aplicação seja notificada com mais precisão sobre a ocorrência de um evento que pode desencadear uma série de outros eventos de negócio.
- **Filtro de dados:** Nem todas as informações geradas por uma fonte de dados RFID (dispositivos de leitura) são de interesse de seu consumidor (middleware ou aplicações clientes) e é interessante que tais informações possam ser resumidas antes de passarem para o nível de aplicação. É importante que o middleware ofereça serviços de filtragem e agregação de dados RFID de acordo com os interesses da aplicação.
- **Interpretação de dados:** Para uma aplicação RFID, é desejável que eventos identificados nos ciclos de leitura sejam traduzidos em eventos relacionados ao contexto de negócio da aplicação. Por exemplo, ao detectar que um lote de etiquetas passou por um determinado portal, pode-se deduzir que o carregamento de um determinado produto foi finalizado.
- **Disseminação de dados:** Informações capturadas por dispositivos de leitura RFID podem ter diversas aplicações clientes interessadas. Estas informações devem ser processadas pelo gerenciador de eventos e repassadas somente para aquelas aplicações que registraram interesse nos dados coletados. Nesse sentido, middleware RFID deve prover suporte para comunicação assíncrona e síncrona (requisição/resposta), visto que algumas aplicações precisam receber dados em tempo real ou com pouco atraso, e outras preferem receber dados em lote.
- **Integração com sistemas legados:** A tecnologia RFID é mais uma fonte de dados que pode ser integrada às diversas aplicações e fontes de dados existentes numa organização. Portanto, um sistema de middleware deve prover serviços de integração para diversos sistemas legados, oferecendo interfaces flexíveis para integração com outros sistemas e que permitam interindependência de linguagem de programação e de plataforma de execução.
- **Conformidade com padrões de middleware:** Para garantir a interoperabilidade entre diferentes implementações de middleware RFID utilizadas numa rede EPC (*EPC Network*), é importante que as soluções de middleware implementem os padrões de especificação existentes para minimizar eventuais problemas de integração e compartilhamento de informações.

Os requisitos citados acima são considerados requisitos funcionais e são contemplados por muitos projetos de middleware RFID existentes e fazem parte da especificação do ALE segundo a

EPCglobal. No entanto, além dos requisitos funcionais, um sistema de middleware RFID deve prover também características adaptativas para poder suportar diferentes aplicações e dispositivos de leitura, assim como elementos para gerenciamento e integração de dados. Escalabilidade, confiabilidade, capacidade de extensão, entre outros, fazem parte desse grupo de características adaptativas e que são apresentados em mais detalhes no capítulo 3.

3. TRABALHOS RELACIONADOS

Embora RFID seja uma tecnologia promissora e que provê inúmeros benefícios para diversas áreas de aplicação, ainda existem desafios importantes que precisam ser superados para sua real efetivação.

Um dos grandes desafios de aplicações RFID é o **gerenciamento de dados**. Este assunto tem gerado uma preocupação por parte de pesquisadores e desenvolvedores de aplicações pois dados de eventos RFID têm características particulares e necessitam de uma atenção e um tratamento especial [CHA04] [WAN05]. Grandes fluxos e volumes de dados, defeitos ou ruídos provocados por operações de leitura, dados com características temporais e espaciais e semântica implícita, são exemplos das particularidades de dados de eventos RFID.

Outro desafio importante é o **desenvolvimento e a integração de aplicações RFID de larga escala** o qual também possui requisitos desafiantes, tais como: escalabilidade, heterogeneidade, maleabilidade e integração de dados.

Privacidade e segurança também são questões que merecem atenção pois são requisitos importantes para integração da tecnologia RFID. No entanto, privacidade e segurança não são o foco deste trabalho e não serão abordadas. Mais detalhes podem ser encontrados em [RIE06a] [GAR05] [OHK05] [JUE06].

Este capítulo apresenta um estudo sobre as principais questões e soluções existentes para problemas relacionados com tecnologias de software RFID. O objetivo deste estudo é prover um melhor entendimento das pesquisas atuais e atividades vinculadas a área visando estabelecer um embasamento teórico e científico para definição da proposta arquitetural do framework de software RFID proposto nesta tese de doutorado (Capítulos 4 e 5).

O restante do capítulo está organizado da seguinte maneira. Na seção 3.1 é definida a arquitetura básica de um sistema RFID onde são identificados requisitos que serão usados na análise das soluções de software apresentadas. Na seção 3.2 é feita uma investigação detalhada das principais propostas tecnológicas para integração e gerenciamento de dados RFID. Estas propostas são avaliadas de acordo com os requisitos definidos na seção 3.1. Dando continuidade aos trabalhos relacionados, na seção 3.3 são apresentados os principais trabalhos para processamento semântico de dados RFID, onde são identificados novos desafios e tendências tecnológicas futuras. Por fim, na seção 3.4 é feito um resumo do que foi apresentado neste capítulo.

3.1 VISÃO GERAL DE SISTEMAS RFID

RFID vem emergindo como uma tecnologia promissora e compatível com uma grande variedade de aplicações, tais como: controle de acesso, gerenciamento de cadeias de produção e distribuição, varejo e logística, áreas ligadas à saúde, entre outras. Grande parte de aplicações RFID existentes ainda são imaturas e muitas foram projetadas de maneira *ad-hoc*, ou seja, foram criadas e implantadas para prover determinadas funcionalidades sem atenção e planejamento necessário que a tecnologia RFID necessita. Até o momento, não existe um padrão completamente estabelecido e unificado para desenvolvimento de aplicações RFID, e existe uma

carência por trabalhos que analisem a eficiência dos padrões já pré-estabelecidos e que proponham novas abordagens às especificações existentes.

3.1.1 ARQUITETURA DE UM SISTEMA RFID

A Figura 3.1 ilustra os componentes fundamentais de um sistema RFID. Interações neste sistema ocorrem em três camadas principais: dispositivos (camada de dispositivos e fontes de dados), processamento de dados (*camada de gerenciamento de dados*) e integração de dados (*camada de integração de dados*).

A camada de dispositivos é formada por etiquetas RFID, dispositivos de leitura de dados e protocolos de comunicação para leitura e escrita de dados RFID. A camada de processamento de dados é formada por diversos componentes de software para comunicação com dispositivos de leitura RFID, filtragem e limpeza de dados e adaptação de dados para integração com aplicações de negócio, incluindo filtragem semântica, transformação e agregação automática de dados.

Já a camada de integração de dados se preocupa em dar suporte às aplicações que exploram dados RFID gerados pelos dispositivos de leitura. O objetivo das interações nesta camada é facilitar, de maneira consistente e transparente, o processo de integração e compartilhamento de dados de eventos RFID entre processos de negócio envolvidos numa rede EPC.

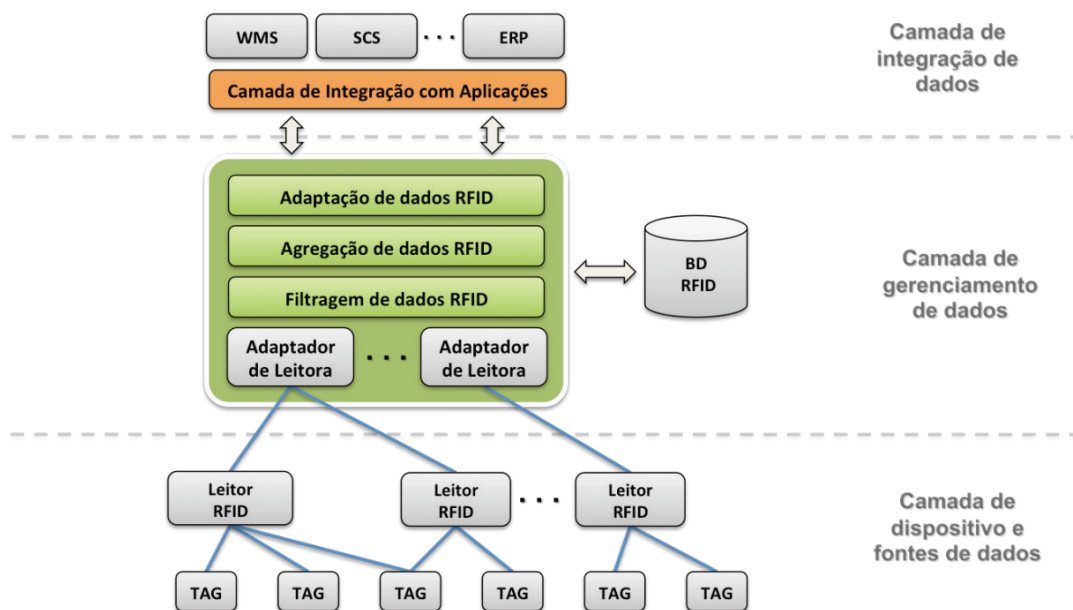


Figura 3.1 – Arquitetura básica de um sistema RFID.

Como neste trabalho o esforço é dedicado apenas a questões relacionadas a tecnologias de software RFID, detalhes sobre a camada de dispositivos físicos e seus aspectos relacionados não serão abordados e podem ser encontrados em [COL06] [ENG02] [FIN03] [FLO04] [SAR04] [WAN06].

3.1.2 REQUISITOS DE SISTEMAS RFID

3.1.2.1 REQUISITOS DE GERENCIAMENTO DE DADOS RFID

Apesar da diversidade funcional de aplicações RFID, dados de eventos RFID compartilham algumas características comuns e inerentes às três camadas da arquitetura genérica apresentada na figura 3.1. Através destas características é possível identificar os seguintes requisitos:

- **Dados imprecisos:** Enquanto houver problemas de confiabilidade nas operações de dispositivos de leitura RFID (leitores, protocolos de comunicação e meios de acesso), erros de leitura e escrita ainda serão considerados graves deficiências operacionais. Interferências e defeitos intermitentes ou até mesmo permanentes podem ser considerados como prováveis causas desta falta de confiabilidade de dados. Com isso, é indispensável um tratamento especial que tolere tais problemas e que permita um funcionamento correto dos sistemas.
- **Dados volumosos:** Dados de eventos RFID são gerados automaticamente e de maneira intensa onde o volume destes dados pode ser muito grande. Sistemas de rastreabilidade de itens é um exemplo disso, onde o fluxo diário de dados pode chegar à casa dos *terabytes*/dia. Neste sentido, é indispensável o uso de métodos que reduzam a quantidade de dados RFID produzidos.
- **Dados dinâmicos e temporais:** Dados de eventos RFID são gerados dinamicamente e a eles são associados *timestamps* durante o processo de leitura. Todas as transações relacionadas a dados RFID são, de fato, associadas ao tempo. Estas informações temporais são importantes para rastreabilidade e monitoramento de objetos RFID pois com elas é possível calcular a duração das atividades relacionadas, como por exemplo, quanto tempo leva o transporte de objetos de um determinado lugar para outro.
- **Inferências implícitas:** Dados de eventos RFID podem apresentar informações implícitas como mudanças de estado e etapas num processo de negócio. Estes dados podem também apresentar informações de agregação que descrevam relacionamentos entre objetos. Nesse sentido, dados RFID para serem integrados com aplicações de negócio precisam do contexto de outras informações, tais como, dados de referência, situações do ambiente, entre outros.
- **Limite de vida útil:** Dados de eventos RFID normalmente apresentam tempo de vida útil limitado, ou seja, determinado período de tempo em que tal dado é ativado, atualizado, rastreado, monitorado e destruído. Como exemplo, temos um sistema de cadeia de distribuição, onde o ciclo de vida de um objeto RFID inicia quando um produto é entregue pelo fabricante e termina quando este produto é vendido ao consumidor final. Neste caso, o gerenciamento do ciclo de vida de dados RFID é importante para o desempenho do sistema devido ao grande volume de dados RFID que podem ser lidos de maneira desnecessária.

3.1.2.2 REQUISITOS DE INTEGRAÇÃO DE DADOS RFID

Além dos requisitos de gerenciamento de dados, a integração de dados RFID com aplicações e processos de negócio também apresenta alguns desafios que precisam ser tratados [BOR04]. A seguir são listados os principais requisitos de integração de dados RFID.

- **Escalabilidade:** A escalabilidade refere-se à habilidade do sistema em crescer em uma ou mais dimensões, seja em volume de dados ou número de transações e sem afetar o desempenho do sistema. Com a adoção da tecnologia RFID, empresas precisam gerenciar dados de diversos dispositivos de leitura RFID distribuídos entre elementos de negócio. Como tais ambientes são propícios a mudanças (alianças e fusões empresariais), a escalabilidade é um critério importante a ser considerado quando se avalia soluções de integração em sistemas RFID.
- **Heterogeneidade:** Muitas vezes o desenvolvimento e a implantação de um sistema RFID ocorre de maneira distribuída, envolvendo diversas companhias ou até mesmo países, onde diferentes dispositivos RFID, estruturas de dados e padrões podem ser usados no processo de integração. Este aspecto requer que sistemas RFID suportem técnicas de abstração que escondam a complexidade dos sistemas e dispositivos envolvidos e que facilitem as atividades de integração de dados.
- **Capacidade de administração:** Esse requisito refere-se a capacidade de gerenciamento e administração dos componentes de um sistema RFID. Por componentes de sistema, entende-se infraestrutura de rede e dispositivos RFID, módulos de gerenciamento e integração de dados, estruturas de dados e regras e processos de negócio, entre outros. Sistemas RFID devem facilitar a supervisão, administração, manutenção e o controle dos componentes envolvidos.
- **Flexibilidade:** Esse requisito refere-se ao grau de adaptabilidade de um sistema RFID. Para isso, é desejável que um sistema RFID seja adaptável a diferentes cenários de negócio e também a suas constantes alterações. Mudanças são necessárias para adaptar aplicações RFID (ex. econômicas, políticas ou mudanças organizacionais). Além disso, sistemas RFID devem ser flexíveis para prover respostas rápidas a situações anormais.
- **Openness:** Esse requisito refere-se a capacidade do sistema em prover interoperabilidade de sistemas RFID. Por exemplo, um bom projeto de interfaces para adaptação de um dispositivo de leitura RFID facilita a capacidade de integração do sistema. Além disso, sistemas RFID devem ser baseados em protocolos de comunicação existentes (TCP/IP, HTTP), assim como padrões de sintática e semântica de dados (XML, PML [FLO04], EPC [EPC04]). Uma arquitetura RFID baseada em padrões permite heterogeneidade e facilita a escalabilidade e integração do sistema.

3.2 PRINCIPAIS TECNOLOGIAS DE SISTEMAS RFID

3.2.1 SISTEMAS PARA GERENCIAMENTO DE DADOS

Dados RFID devem ser apropriadamente processados antes de serem utilizados por qualquer aplicação. A comunidade científica vem desenvolvendo um forte interesse para questões relacionadas com gerenciamento de dados em sistemas RFID [BOR04] [CHA04] [GON06a] [GON06b] [HU05] [JEF06a] [JEF06b] [RAO06] [WAN05] [WAN06], e as principais tecnologias existentes serão apresentadas a seguir.

3.2.1.1 SIEMENS RFID MIDDLEWARE

Siemens RFID Middleware [WAN05] [WAN06] é um sistema integrado para gerenciamento de dados RFID que (1) permite filtragem semântica e transformação automática de dados RFID, (2) provê suporte eficiente para consultas sobre monitoramento e rastreabilidade de objetos e (3) permite fácil integração com outras aplicações.

A proposta dos autores é a criação de um sistema de middleware RFID baseado em relacionamentos dinâmicos de ERs (*Dynamic Relationship ER*) e em modelos de dados temporais (*Temporal-based Data Model*) para representação de dados RFID (*DRER - Dynamic Relationship ER Model*), conforme apresentado na figura 3.2. Através desta abordagem, o sistema abstrai uma série de entidades (objetos, leitores, localização e transações) e modelos de interação entre entidades, tais como, relacionamentos baseados em eventos ou estados (*State-based* ou *Event-based Relationships*).

Para relacionamentos baseados em estados, uma combinação de atributos (*tstart* e *tend*) representa a vida útil de um produto identificado, como por exemplo, o período de tempo que um produto permanece em fabricação ou em estoque na fábrica. Já para relacionamentos baseados em eventos, um atributo de tempo (*timestamp*) é usado para indicar a ocorrência de determinado evento. Estas funcionalidades são características temporais e espaciais dos objetos identificados e facilitam a rastreabilidade e o monitoramento de itens. Todas as entidades e relacionamentos do modelo *DRER* podem ser implementadas como tabelas em um sistema de gerenciamento de banco de dados relacional o que permite uma fácil manutenção e consulta do histórico dos dados dos objetos persistidos no sistema.

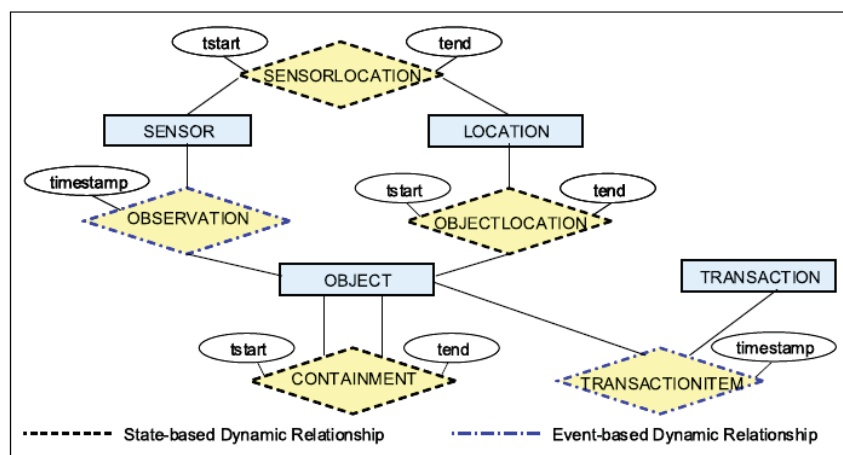


Figura 3.2 – Modelo para relacionamento dinâmico de ERs – DRER [WAN05].

Para prover suporte a transformação automática de dados RFID, uma abordagem declarativa baseada em regras e orientada a eventos também foi proposta pelos autores, onde a lógica da aplicação pode ser representada em eventos complexos. Os autores propõem também um mecanismo baseado em grafos para detecção de eventos RFID chamado *RCEDA (RFID Complex Event Detection Algorithm)*. Com este mecanismo, os autores afirmam conseguir representar e tratar eventos RFID que apresentem requisitos temporais.

A arquitetura do middleware RFID da Siemens (Figura 3.3) é composta por três módulos principais: (1) leitor RFID, (2) gerenciador de eventos e (3) servidor de dados RFID. O servidor de dados RFID é dividido em (3.1) gerenciador de dados RFID, (3.2) armazenamento de dados RFID, (3.3) arquivamento de dados, (3.4) armazenamento de dados de produtos e (3.5) interface de integração.

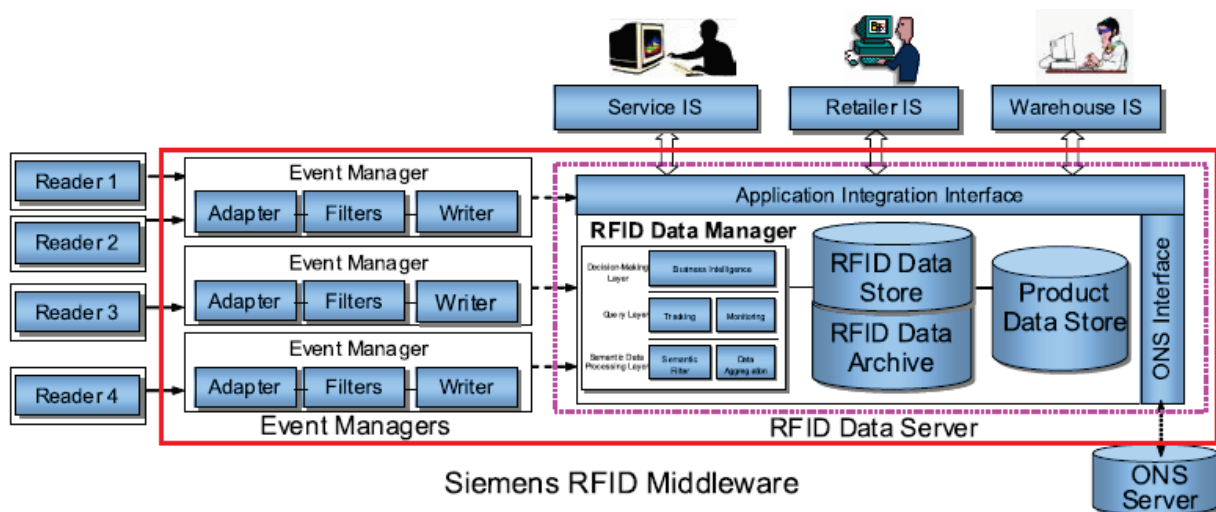


Figura 3.3 – Arquitetura do Middleware RFID da SIEMENS [WAN05].

O módulo de gerenciamento de eventos é o ponto de entrada do sistema. Ele é responsável por, dinamicamente, receber dados vindos dos leitores RFID, realizar uma filtragem preliminar destes dados recebidos e repassar tais dados para o servidor de dados RFID. Além destas funcionalidades, o módulo de gerenciamento de eventos tem outras três funções básicas (camadas funcionais) que são: adaptadores de dispositivos de leitura (componente de software responsável pela comunicação e gerenciamento de leitores RFID), filtro de dados (componente de software que realiza uma filtragem preliminar dos dados RFID lidos) e escrita de dados (componente de software responsável pela formatação dos dados lidos em um formato padrão de comunicação).

O módulo de gerenciamento de dados RFID é o módulo principal do sistema e está inserido no servidor de dados RFID. Esse módulo está dividido em três camadas funcionais: (a) processamento de dados semânticos (filtragem semântica, transformação baseada em regras e agregação de dados), (b) camada de consulta (define métodos para rastreabilidade e monitoramento de objetos) e (c) camada de tomada de decisão (provê suporte ao gerenciamento de processos através de *BI*).

O módulo de armazenamento de dados provê esquemas de armazenamento através do modelo de dados *DRER*, e com isso permite o armazenamento de objetos RFID. O módulo de arquivamento de dados usa dados RFID não ativos (que já encerraram os ciclos de vida) como

histórico de dados. Já o módulo de armazenamento de dados de produtos tem como funcionalidade, armazenar informações estáticas relacionadas a dados RFID dinâmicos, tais como, descrição e modelo de produto.

Por fim, o módulo de interface de integração provê meios de comunicação padronizados para que aplicações consigam ter fácil acesso a dados RFID relacionados. Informações tanto no nível do objeto quanto no nível do produto podem ser compartilhadas entre empresas através de um servidor ONS (*Object Naming Service*) usando diferentes padrões de comunicação de dados (*XML*, por exemplo).

3.2.1.2 MDI-SMURF RFID MIDDLEWARE

SMURF (*Statistical Smoothing for Unreliable RFID data*) [JEF06a] é um mecanismo adaptativo para filtragem e limpeza de dados RFID brutos. O sistema se baseia na modelagem de dados RFID em amostras randômicas de etiquetas do mundo físico, explorando uma abordagem baseada em porções estatísticas (*Statistical-sampling-based approach*) para compensar a falta de confiabilidade dos dispositivos de leitura de dados e etiquetas dinâmicas (etiquetas que entram e saem do campo de detecção das leitoras) que podem ocorrer durante o processo de filtragem e limpeza de dados. A habilidade para determinar automaticamente o índice de suavização e adaptação necessária baseado nas observações realizadas pelos dispositivos de leitura, facilita o processo de integração do *SMURF* com plataformas de middleware RFID. Tal integração visa facilitar o processo de limpeza de dados e também prover dados RFID mais confiáveis.

Em [JEF06b] os autores estenderam as funcionalidades do *SMURF* para atender características de middleware RFID e criaram o *MDI-SMURF* (*Metaphysical Data Independence for Statistical Smoothing of Unreliable RFID data*). Este novo sistema consiste em uma plataforma organizada através de estágios de processamento (*pipeline*) de dados RFID e que usa *MDI* como camada de independência física que blinda aplicações dos fenômenos e características do mundo físico. Segundo os autores, a filosofia do *MDI* é que aplicações não devem tratar nenhum aspecto físico de dispositivos de leitura e sim interagir com reconstruções do mundo físico criado a partir de atividades de dispositivos. Baseado nos conceitos de *SMURF*, *MDI-SMURF* tem a visão de dados RFID como fluxos randômicos de etiquetas do mundo físico. Desta forma, o mecanismo incorpora técnicas de amostragens estatísticas para guiar operações de processamento de dados RFID.

3.2.1.3 EPC-BITMAP

Oracle EPC-bitmap [HU05] é um novo tipo de dado para tratamento e manipulação eficiente de grandes volumes de dados RFID gerados por aplicações de rastreabilidade no nível do item. Este tipo de aplicação rastreia itens em grupos baseados em suas localizações, data de expiração, fabricante ou alguma outra propriedade comum entre todos os objetos do grupo. Nesse contexto, *EPC-bitmap* representa uma coleção de EPCs com segmentos comuns, tais como cabeçalho, número de identificação e classe do objeto. Com isso, a aplicação acessa e manipula estes dados usando um conjunto de operações *EPC-bitmap*, como por exemplo, funções *epc2bmp* e *bmp2epc* para conversão de dados.

Esta abordagem permite métodos significantes de armazenamento de dados mantendo o mesmo desempenho ou até melhor em termos de consulta de dados quando comparado com

sistemas tradicionais de banco de dados que armazenam coleções de EPCs em bancos de dados relacionais.

3.2.1.4 UIUC'S RFID MIDDLEWARE

Pesquisadores da Universidade de *Illinois (University of Illinois at Urbana-Champaign - UIUC)* [GON06a] [GON06b] propuseram um novo modelo de dados RFID para ambientes de *warehousing*. O modelo proposto permite uma análise eficiente e flexível de espaços multidimensionais no estilo *OLAP (Online Analytical Processing)* que explora a tendência de objetos individuais mesmo levando em consideração sua movimentação. Com isso, é possível identificar múltiplos movimentos de objetos em apenas um único registro e sem perda de informação.

Este modelo consiste em uma hierarquia de resumos de eventos altamente compactados representados na forma de cuboides de dados agregados em diferentes níveis de abstração e onde as análises dos dados RFID são realizadas (*RFID-cuboid*). Cada modelo contém a movimentação de objetos e permite (1) o armazenamento de informações de produtos para cada objeto RFID, (2) permite que informações de objetos se mantenham juntas em uma determinada localização e (3) mantém informações necessárias para unir múltiplos armazenamentos. O modelo proposto resulta em um banco de dados pequeno e provê suporte eficiente para grandes variedades de consultas RFID.

3.2.1.5 REFILL

REFiLL (Rfid Event FiLtering tooLchain) [ANA09] é um middleware RFID baseado na arquitetura do *EPC Network*. Ele foi projetado como um ambiente programável que provê flexibilidade para desenvolvedores de aplicações RFID mediante tanto a heterogeneidade de hardware quanto de requisitos de aplicação. *REFiLL* está logicamente posicionado entre a camada de virtualização de leitores e o ALE. O mecanismo de processamento de dados proposto pelos autores consiste em filtros e gerentes de saída (*output managers*). Os filtros são usados para receber dados de entrada vindos dos leitores ou outros filtros. Os filtros são definidos e interconectados com o editor de arquivos XML, o qual é compilado para código JAVA visando prover a funcionalidade de filtragem e coleta de eventos RFID. Já os gerentes de saída proveem a funcionalidade de receber dados processados dos filtros e abstrações dos sistemas de saída, tais como *RBDMS*, arquivos, *Sockets* e *Web Services*.

3.2.1.6 LIT MIDDLEWARE

LIT (Logistics Information Technology) [KAB07] é um middleware RFID que se propõe a satisfazer tanto necessidades das aplicações quanto restrições da tecnologia RFID. Ele foi projetado usando conceitos de ALE e EPCIS segundo a *EPCglobal*. A arquitetura de software proposta para o ALE é composta por quatro camadas: abstração da aplicação, execução baseada em estados, consulta contínua e abstração de leitores.

Para alcançar um alto desempenho e com isso diminuir tanto o tráfego da rede quanto o tempo de processamento de dados, *LIT* usa em sua arquitetura um modelo de execução baseado em estados [CAR03] e processos de consultas contínuas. Na execução baseada em estados,

ocorre um escalonamento dos processos que monitoram o estado do sistema, mantendo assim uma fila de execução controlada e não susceptível aos problemas tradicionais relacionados ao uso de modelos baseados em threads. Já no processo de consultas contínuas, os autores usam um mecanismo para gerenciamento de consultas. Este mecanismo recebe as especificações (*ECSpecs*) das aplicações requisitantes e cria índices de consultas, os quais servem de identificação para a atividade de roteamento e envio dos dados consultados na forma de relatórios (*ECReports*).

A arquitetura de software proposta para o EPCIS consiste em três camadas: camada de consulta, camada de repositório de dados e camada de captura de dados. O componente de acesso a fonte de dados foi projetado para prover persistência e conexões com diferentes tipos de dados, tais como *Oracle*, *MS SQL Server*, *MySQL*, entre outros.

3.2.1.7 RF²ID

RF²ID (*Reliable Framework for Radio Frequency Identification*) [AHM07] foi projetado e implementado com a intenção de prover um middleware com as seguintes características: confiabilidade, balanceamento de carga, alta vazão no tráfego de dados, escalabilidade e organização de dados. Sua arquitetura é baseada no conceito de leitores virtuais (*VR – Virtual Reader*) e caminhos virtuais (*VP – Virtual Path*). Cada *VR* é associado a um conjunto de leitores RFID físicos (*PR – Physical Readers*) em seus locais determinados. As principais responsabilidades de cada *VR* são: filtragem dos dados vindos dos leitores físicos (*PRs*) conectados e de outros leitores virtuais (*VRs*), adicionar um *timestamp* ao dado lido, gerenciamento de caminhos virtuais (*VPs*) e também de consultas. Os caminhos virtuais (*VPs*) são canais lógicos criados dinamicamente por leitores virtuais (*VRs*) visando prover funcionalidades como: alta escala de confiabilidade dos dados, respostas eficientes a consultas e balanceamento de carga entre leitores.

3.2.1.8 MARM

MARM (Multi-Agent based middleware for Rfid asset Management system) [MAS09] é um projeto de middleware RFID que provê suporte ao processamento e gerenciamento de dados RFID e que se baseia no conceito de agentes de software (*AOSE - Agent Oriented Software Engineering*) como tecnologia base para sua implementação.

Os autores usam técnicas do projeto *SMURF* [JEF06a] para tratamento de dados RFID não confiáveis e um modelo de dados temporal proposto em [WAN05] para facilitar o processamento de eventos. Além disso, a arquitetura do *MARM* está baseada na metodologia *PASSI (Process for Agent Societies Specification and Implementation)* [CHE04], a qual é uma das metodologias *AOSE* para projeto e desenvolvimento de sistemas multiagentes.

A arquitetura do *MARM* é dividida em três camadas principais: gerenciamento de dispositivos, gerenciamento de dados e camada de interfaces de aplicação. Nesse contexto, as aplicações clientes, através da camada de interfaces de aplicação, especificam seus dados de interesse baseados na comunicação com agentes de gerenciamento de dados que são *ECReport-Request* e *ECReportSPEC*. Por sua vez, agentes de gerenciamento de dados se comunicam com

agentes dos dispositivos de leitura na camada de dispositivos, os quais realizam o processamento de dados e geram dados requisitados pela aplicação.

Dois diagramas de classe de ontologia foram modelados para suportar a coordenação entre agentes, o *DOD* (*Domain Ontology Description*) e o *COD* (*Communication Ontology Description*). *DOD* descreve a ontologia do projeto de middleware através da representação de predicados de domínio e algumas ações do agente. Já o *COD* descreve uma ontologia sobre o conhecimento dos agentes e sobre regras de associação entre eles. *MARM* foi projetado para prover não apenas funcionalidades de captura e tratamento de dados e processamento de eventos, mas também etapas de processamento de eventos de alto nível da aplicação.

3.2.2 SISTEMAS PARA INTEGRAÇÃO DE DADOS

Grandes empresas desenvolvedoras de software como *Sun Microsystems*, *SAP*, *BEA Systems*, *IBM* e também algumas grandes organizações de pesquisa, tais como *WINMEC*, *Auto-ID Labs* e *Siemens Research* têm trabalhado na criação de plataformas de integração de sistemas RFID. Embora cada sistema proposto tenha suas próprias soluções, ambas as abordagens primam por um objetivo comum, melhorar a capacidade de integração de sistemas RFID com aplicações de negócio.

Nesta subseção serão apresentados alguns trabalhos referentes às plataformas mais significativas do mercado. Vale a pena lembrar que algumas informações foram pesquisadas tendo como base manuais de usuários e artigos demonstrativos de sistemas, os quais não apresentam muitos detalhes técnicos que facilitem a classificação dos trabalhos selecionados. Além disso, como estes trabalhos são plataformas de software RFID, eles nativamente apresentam funcionalidades básicas de gerenciamento de dados RFID, mesmo que em sua maioria não suportem o processamento de eventos RFID complexos.

3.2.2.1 SAP AII (AUTO-ID INFRASTRUCTURE)

SAP AII (Auto-ID Infrastructure) [BOR04] [SAP11] é um middleware RFID que oferece uma camada de inteligência que visa facilitar a integração entre dispositivos automatizados de comunicação (leitores RFID, dispositivos *Bluetooth* e leitores de códigos de barras) e aplicações empresarias. Ele converte dados lidos pelos dispositivos em informações de processos de negócio através da associação de dados com regras de negócio pré-definidas. *SAP AII* foi construído baseado no servidor de aplicações web da *SAP* (*SAP Web Application Server*) o qual faz parte da plataforma de integração de aplicações *NetWeaver*.

A arquitetura do sistema é composta por quatro camadas (Figura 3.4). A camada de dispositivos (*Device Layer*) suporta diferentes tipos de sensores através de uma interface de baixo nível e independente de hardware. Tal interface consiste em um conjunto de operações básicas tanto para escrita e leitura de dados RFID quanto para operações de assinatura de eventos (*publish/subscribe*).

A camada de operação de dispositivos (*Device Operation Layer*) consiste em um ou mais controladores de dispositivos (cada um similar a um middleware RFID). Ela provê também, funções para tratamento de dados (filtragem, agregação, transformação/ajuste/enriquecimento) e é responsável por repassar dados recebidos e tratados para nodo(s) *Auto-ID* envolvido(s). A camada

de operação suporta dois modos de operação: síncrono e assíncrono. No modo síncrono, *DC* recebe mensagens de configuração de operação de dispositivos de algum nodo *Auto-ID* (requisição da aplicação), tal como operações de escrita e leitura. Já no modo de operação assíncrona, o nodo *Auto-ID* espera por eventos ocorridos nas fontes de dados, o que desencadeia uma série de etapas para tratamento de dados que é feito por um conjunto de processadores de dados (filtragem, enriquecimento dos dados, agregação, escrita, armazenamento temporário de mensagens, e envio de mensagens).

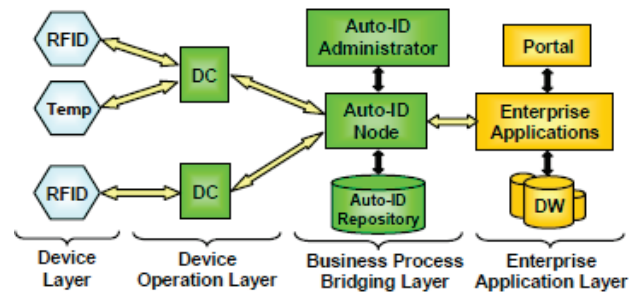


Figura 3.4 – Arquitetura do sistema SAP All [BOR04].

A camada de ligação com processos de negócio (*Business Process Bridging Layer*) associa mensagens de observação com processos de negócios existentes. Esta camada, através do módulo *Auto-ID Node*, armazena informações de status de objetos e históricos de eventos, tais como: localização de objetos, informações de agregação e informações referentes ao ambiente de operação dos objetos identificados.

Por fim, a camada de aplicação empresarial (*Enterprise Application Layer*) dá suporte a processos de negócio, tais como gerenciamento de cadeias de suprimentos (SCM), gerenciamento do relacionamento com clientes (CRM), entre outras aplicações compatíveis ou não com sistemas *SAP*.

Para permitir o teste de aplicações sem a necessidade de instalar dispositivos físicos, o sistema provê um ambiente de simulação. Além disso, *SAP* oferece também, suporte a uma infraestrutura flexível de sistema, através de uma abordagem baseada em regras customizáveis para especificação e execução da lógica de negócio nos nodos *Auto-ID*.

Visto que o sistema permite uma arquitetura distribuída e que cada camada de operação possui seus próprios processadores de dados, os quais implementam a mesma interface para assinatura de eventos (*publish/subscribe*), o conjunto de processadores de dados instanciados no sistema podem operar na forma de cadeia de processamento.

3.2.2.2 WINRFID

WinRFID [PRA06] é um middleware RFID que possui uma arquitetura baseada em cinco camadas (Figura 3.5): hardware RFID, protocolos, processamento de dados, framework *XML* e camada de apresentação de dados. A camada de hardware trata questões relacionadas com implantação, configuração e comunicação dos dispositivos (leitores, etiquetas e seus módulos de I/O), tendo um componente para cada um desses três dispositivos.

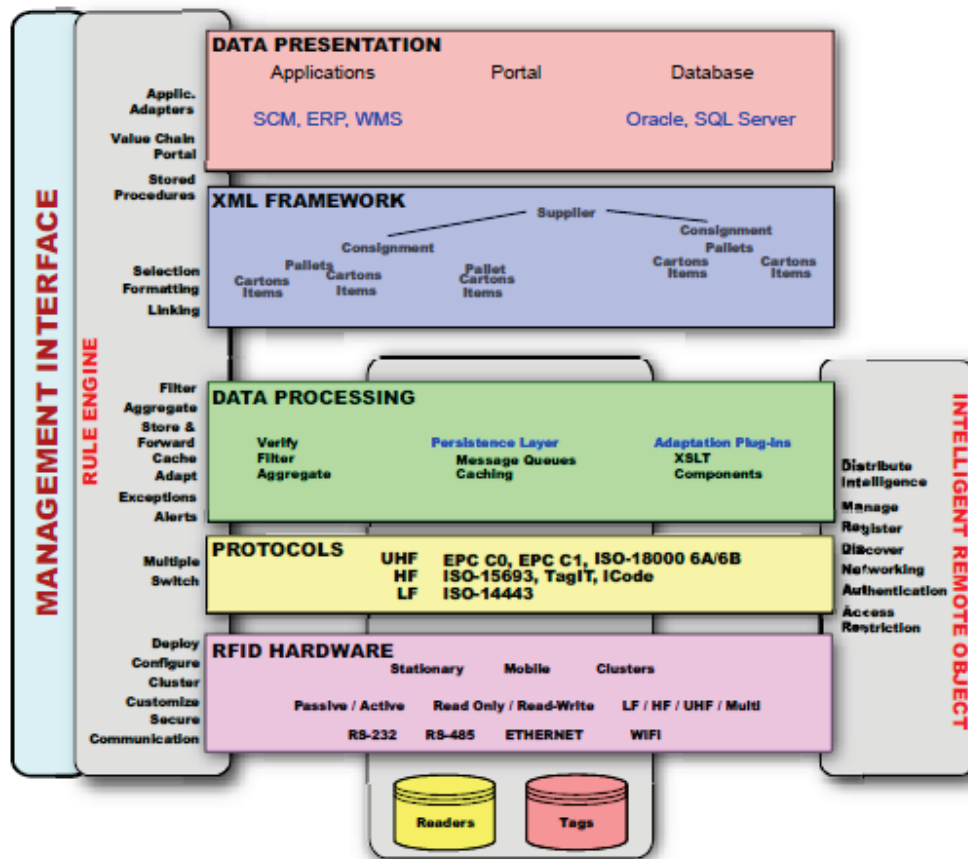


Figura 3.5 – Camadas funcionais do middleware WINRFID [PRA06].

A camada de protocolos é responsável pela abstração da maioria dos protocolos comuns existentes, tais como *ISO 15693*, *ICode*, *EPC Class 0* e *EPC Class 1*. Essa camada executa um mecanismo cuja responsabilidade é processar dados lidos de acordo com o tipo de protocolo usado.

A camada de processamento de dados implementa regras para o processamento de dados brutos vindos da camada de protocolos. Tal processamento é responsável por remover leituras duplicadas e também por verificar etiquetas lidas. Qualquer tipo de anomalia na operação desta camada é manipulado através de diferentes formatos de alertas, como e-mails, mensagens e outros formatos definidos pelos usuários.

A camada do framework *XML* é responsável por disponibilizar dados processados e formatados na camada de processamento de dados, disponibilizando-os através de um formato *XML* para que possa ser usado por diferentes aplicações. Já a camada de apresentação de dados é uma camada de aplicação que usa dados disponibilizados pela camada do framework *XML* para visualização e tomada de decisão.

WinRFID foi projetado usando o framework de desenvolvimento *.NET*, o qual facilita a adição de funcionalidades para aplicações através de *plug-ins* em tempo de execução. Com isso, *WinRFID* tem um mecanismo de regras de negócio que permite que usuários possam criar e programar suas próprias regras no formato de novos *plug-ins*.

3.2.2.3 SUN JAVA SYSTEM RFID SOFTWARE

SUN Java System RFID Software [SUN11] foi uma das primeiras plataformas de integração RFID focada no desenvolvimento de aplicações de larga escala. Sua arquitetura orientada a serviços provê serviços de redes para aplicações através de várias interfaces e protocolos padrões.

O sistema consiste em dois componentes principais: o gerenciador de eventos, que filtra e agrega dados RFID; e o serviço de informação, que provê acesso a eventos de negócio gerados pelo gerenciador de eventos e que serve também como camada de integração. A arquitetura de software da plataforma está baseada no *Java Enterprise System*, o qual permite que o sistema compartilhe uma série de componentes e serviços que possibilitam uma melhor integração e manutenção. Além disso, tal funcionalidade faz com que a aplicação herde alguns requisitos de segurança inerentes à tecnologia *JAVA* e as instancie na plataforma RFID proposta.

3.2.2.4 BEA WEBLOGIC RFID

BEA WEBLogic RFID [BEA06] é um sistema que apresenta uma plataforma RFID baseada em padrões e que foi projetada para automatizar novos processos de negócio RFID. *BEA* é composto por três produtos principais: *Edge Server*, *Enterprise Server* e *Compliance Express*. *Edge Server* é um módulo de software que pode ser instanciado em diversos ambientes e que oferece funcionalidades de filtragem, integração e disseminação de dados RFID, assim como monitoração e configuração de leitores RFID. O *Enterprise Server* realiza um gerenciamento centralizado de dados coletados pelo *Edge Server*. Já o *Compliance Express* foi projetado para suportar desafios atuais de compatibilidade e conformidade com padrões de integração, enquanto estabelece uma fundação para futuras expansões.

3.2.2.5 ACCADA

Accada, o qual recentemente foi renomeado para *Fosstrak* [FOS11], é uma plataforma de sistema RFID *open-source* que segue as especificações da *EPCglobal* e que provê uma infraestrutura RFID baseada na implementação de componentes de baixo nível para comunicação com elementos do *EPC Network*.

Fosstrak é composto por três módulos *principais*: leitura, middleware para coleta e filtragem de dados RFID e serviço de informações *EPC (EPCIS)*. O módulo de leitura provê filtragem, disseminação de dados síncrona e assíncrona (conforme especificado pelo protocolo de leitura da *EPCglobal*) e gerenciamento de dispositivos de leitura.

Uma vez que os dispositivos de leitura capturam dados relevantes, eles notificam a camada de middleware, a qual combina os dados vindos de diferentes leitores num relatório que é enviado para a aplicação de acordo com determinadas pré-configurações. A interface entre middleware (filtro e coleta) e aplicação é baseada na especificação de eventos de aplicação (ALE) segundo a *EPCglobal*. Uma vez que ALE provê funcionalidades de agregação de dados, é possível omitir eventos de leitura redundantes de diferentes dispositivos de leitura.

O Serviço de Informação (*EPCIS*) é um componente responsável pelo recebimento de dados RFID originados do middleware e que traduz estes dados em eventos de negócio

significativos disponibilizando esses dados para a aplicação. O *EPCIS* é composto por três módulos: repositório, módulo de captura (que recebe os dados RFID vindos do middleware) e módulo de consulta (que provê funcionalidades de busca de eventos desejados direto no repositório).

3.2.3 ANÁLISE COMPARATIVA E QUESTÕES EM ABERTO

Em relação aos trabalhos apresentados sobre gerenciamento de dados RFID (Subseção 3.2.1), é possível perceber, através da tabela 3.1, que a maioria dos projetos avaliados oferecem suporte tanto ao tratamento de dados volumosos quanto ao tratamento de dados imprecisos. Entretanto, percebe-se também que poucos projetos tratam questões relacionadas com modelagem de dados temporais e espaciais, referências implícitas e expectativa de vida limitada. Apenas o projeto Siemens RFID Middleware oferece suporte a tais requisitos.

Tabela 3.1 – Abordagens sobre processamento de dados RFID.

Plataforma	Imprecisão	Grandes fluxos de dados	Dado temporal	Dado espacial	Referências implícitas	Expectativa de vida limitada
Siemens RFID Middleware	Tratamento parcial no módulo de filtragem de dados no gerenciador de eventos (filtragem de eventos duplicados e erros de detecção)	Armazenamento baseado em particionamento de dados	Relacionamento dinâmico baseado em ER (modelo DRER)	Transformação de dados baseado em regras (modelo DRER)	Considerado no particionamento de dados (relacionamento baseado em estados e combinação de atributos)	Armazenamento baseado em particionamento de dados - modelo ER
MDI-SMURF RFID Middleware	Limpeza de dados adaptativa e declarativa baseado em abordagens estatísticas	Não aplicado	Não aplicado	Não aplicado	Não aplicado	Não aplicado
EPC-bitmap	Não aplicado	Coleção de EPCs baseado no tipo de dado EPC-bitmap	Não aplicado	Não aplicado	Não aplicado	Não aplicado
UIUC's RFID Warehousing	Não aplicado	Estrutura de dados baseado no RFID-cuboid	Tabela estática	Tabela estática	Não aplicado	Não aplicado
REFILL	Mecanismo com implementação do protocolo de leitura EPC - EPC RP	Não aplicado explicitamente	Não aplicado explicitamente	Não aplicado explicitamente	Não aplicado	Não aplicado
LIT Middleware	Implementação do protocolo de leitura EPC (EPC RP)	Modelo de execução baseado em estados	Não aplicado explicitamente	Não aplicado explicitamente	Não aplicado	Não aplicado
RF²ID	Leitores virtuais	Gerenciamento de caminho	Não aplicado explicitamente	Não aplicado explicitamente	Não aplicado	Não aplicado
MARM	SMURF (Técnica estatística de limpeza de dados RFID não confiáveis)	Sistema multiagentes	Modelo temporal	Não aplicado explicitamente	Não aplicado	Não aplicado

Muitos sistemas ainda estão focados em mecanismos que melhoram a confiabilidade de dados de eventos de leitura e acabam não tratando problemas relacionados com capacidade de abstração e elevação semântica de informações RFID.

Embora os esforços dedicados à tecnologia RFID tenham gerado bons resultados na área de gerenciamento de dados, as tecnologias existentes ainda são imaturas e necessitam de um

empenho significativo para tentar superar alguns desafios que se mantêm não resolvidos, tais como: melhor confiabilidade e exatidão no fornecimento de dados, abordagens genéricas para gerenciamento de dados RFID volumosos e mecanismos mais inteligentes e flexíveis para transformação de dados RFID.

Dados RFID brutos têm pouco valor operacional enquanto não são transformados em uma forma mais significativa e amigável para interações no nível de aplicação. Além disso, dados RFID apresentam significados implícitos e relacionamentos associativos com outros dados onde inferências importantes podem ser aplicadas. Nesse sentido, diversos estudos estão sendo feitos rumo à adoção de tecnologias de processamento de eventos complexos (CEP) na integração de dados RFID temporais com processos de negócio.

Tabela 3.2 – Plataformas para integração de dados RFID.

Plataforma	Escalabilidade	Heterogeneidade	Capacidade de administração e manutenção	Flexibilidade	Openness
WinRFID	Módulos de middleware distribuídos e auto-contidos	Módulos abstratos de protocolos e framework XML para representação de dados	Console de gerenciamento WinRFID	Mecanismo baseado em regras e plugins WinRFID.	Frameworks baseado em padrões (Web services e XML)
SAP All	Controladores de dispositivos distribuídos e nodos Auto-ID	Interface de dispositivo independente de HW e nodos Auto-ID distribuídos	Administrador Auto-ID, ferramentas de simulação e teste de mensagens e leitores	Mecanismo baseado em regras customizadas	Compatível com padrões <i>EPCglobal</i>
Sun Java System RFID Software	Arquitetura distribuída do gerenciador de eventos	Adaptadores de dispositivo extensível	Interface baseada em navegador web para monitoramento centralizado e gerenciamento de serviços e dispositivos	Não aplicado	Arquiteturas orientadas a serviços (SOA) e baseada em padrões
BEA WebLogic RFID	Arquitetura distribuída baseada em servidores RFID (Edge Servers)	Suporte (<i>out-of-box</i>) para a maioria dos leitores e interfaces SOA	Console administrativo, monitoramento e agente de gerenciamento e simulador de leitores	Não aplicado	Padrões compatíveis (<i>EPCglobal</i> e padrões ISO)
Accada	EPCIS distribuído e serviço de descoberta centralizado	Dispositivos compatíveis com EPC	Não aplicado explicitamente	Não aplicado	Compatível com padrões <i>EPCglobal</i> propostos

Na tabela 3.2, diversas plataformas para integração de dados foram comparadas usando os requisitos de integração apresentados na subseção 3.1.2.2. Através deste estudo, é possível perceber que a maioria das plataformas para integração de aplicações RFID abrange em seus projetos, principalmente, questões como escalabilidade, heterogeneidade e *openness*. No entanto, elas oferecem um suporte muito limitado para requisitos como flexibilidade e capacidade de administração e manutenção.

Algumas arquiteturas provêm funcionalidades distribuídas, e poucas integram dados de eventos RFID com processos de negócio através de mecanismo baseado em regras. Além disso, nenhum destes sistemas trata o problema da falta de mecanismos para compartilhamento de dados baseado em eventos e somente o Fosstrak [FLO07] (antigo *Accada*) apresenta alguma solução para o *gap* do ALE quanto a falta de escalabilidade e flexibilidade da aplicação de captura proposta pela *EPCglobal* na especificação da *EPC Network*.

Embora as atuais soluções fundamentem a criação de aplicações RFID, inúmeras questões de pesquisa ainda precisam ser solucionadas quanto a capacidade de integração, principalmente

questões como: melhor suporte a aplicações de larga escala e melhor capacidade de integração com sistemas legados e sensores de ambientes.

Suporte à integração com sistemas legados é uma questão importante para plataformas de integração RFID. Muitas empresas procuram meios de maximizar investimentos existentes e preservar conhecimentos valiosos de negócio enquanto se adaptam a novas evoluções tecnológicas. Nesse sentido, novas plataformas de middleware RFID vêm sendo desenvolvidas para integrar novos sistemas RFID com infraestruturas já existentes. Além disso, muitos pesquisadores acreditam que integrar etiquetas RFID e sensores de ambiente é uma alternativa interessante e ideal no sentido de suprir a necessidade de obtenção de informações do mundo físico. Infelizmente, RFID e sensores apresentam características de dados diferentes. Tais características devem ser representadas e os problemas inerentes resolvidos, visando facilitar a integração destas duas tecnologias.

3.3 PROCESSAMENTO SEMÂNTICO DE DADOS RFID

Como já foi mencionado na seção 3.2, o gerenciamento de dados de eventos RFID é um desafio em constante evolução. Isso ocorre, pois, o volume e a velocidade de crescimento destes dados em ambientes operacionais muitas vezes excede a capacidade infra estrutural de armazenamento existente na maioria das empresas o que pode desencadear inúmeros problemas.

Ter capacidade para capturar grandes volumes de dados e em alta velocidade é um dos primeiros passos, no entanto, não é suficiente quando se trata da tecnologia RFID. Não basta o sistema ser capaz de capturar e armazenar dados, mas sim conseguir capturar a informação contida nestes dados e que agregue valor operacional de negócio aos processos envolvidos.

3.3.1 PRINCÍPIOS DE UM PROCESSAMENTO SEMÂNTICO DE DADOS RFID

Para se obter vantagens com a implantação de sistemas RFID e conseguir agregar valor de negócio aos dados de eventos lidos, é necessário um gerenciamento em tempo real dos fluxos de dados, além de estar preparado para utilizar estes dados ou estas informações de maneira correta e na hora correta.

Em [PAL05], o autor sugere alguns princípios fundamentais para gerenciamento efetivo de dados RFID. Estes princípios indicam inúmeras melhorias em sistemas de middleware RFID as quais foram adotadas em diversos sistemas existentes, inclusive no framework RFID proposto nesta tese de doutorado. Dentre os princípios propostos, é relevante destacar as seguintes ideias:

A. Tratar/digerir dados RFID perto das fontes de dados

As fontes de dados de um sistema RFID são os próprios dispositivos de leitura os quais geram grandes volumes de dados durante suas operações. Segundo o autor [PAL05], uma abordagem positiva para lidar com esta situação é tratar e digerir dados lidos durante o tráfego de eventos o mais próximo possível da fonte de dados e repassar para a aplicação apenas dados de eventos mais significativos. Tal abordagem ajuda a diminuir o fluxo de dados e abstrai da aplicação a responsabilidade de tratar dados lidos.

Esse tratamento de dados é mais eficiente que uma filtragem básica a qual muitas vezes pode ser realizada pelo próprio dispositivo de leitura. Como esse processo de tratamento é realizado por um mecanismo de software que interage diretamente com dispositivos de leitura (Middleware RFID), ele é considerado uma atividade de limpeza, consolidação e agrupamento de dados, ou até mesmo uma compensação para a falta de confiabilidade dos ambientes operacionais e leitores RFID. Tratar dados RFID próximo das fontes de dados permite um gerenciamento de dados mais efetivo, melhorando a confiabilidade de dados lidos e permitindo uma melhor escalabilidade e integração com aplicações de negócio.

B. Agregar significado a eventos

Eventos RFID são tipos de eventos formados por dados simples provenientes de sistemas de middleware RFID e que não apresentam informações explícitas de negócio. Como estes dados precisam ser capturados e transformados em eventos de negócio para serem armazenados no repositório de dados do EPCIS, a tecnologia CEP vem sendo utilizado como tecnologia capaz de realizar esta atividade de processamento, suprimindo funcionalidades e deficiências da aplicação de captura definida para o *EPC Network* (Capítulo 2).

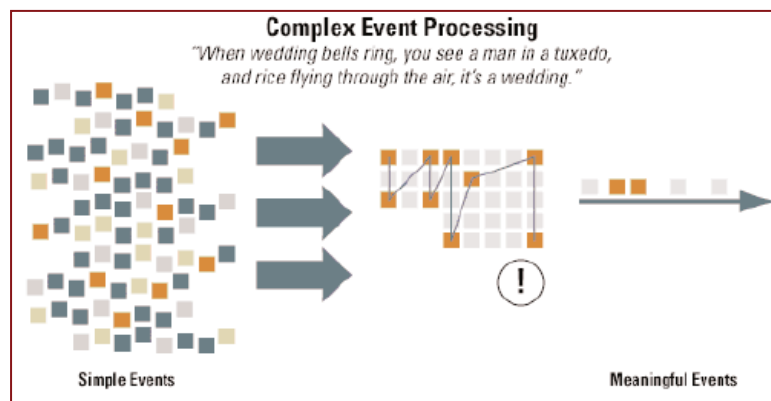


Figura 3.6 – Exemplo de uso da tecnologia CEP [PAL05].

CEP ou processamento de eventos complexos é um campo da área de gerenciamento de dados que lida com tarefas relacionadas ao processamento de múltiplos fluxos de eventos simples com objetivo de identificar eventos complexos e que agreguem informações de contexto aos eventos ocorridos (Figura 3.6). O princípio da tecnologia CEP é transformar eventos simples em eventos significativos que derivem conhecimento efetivo para tomada de decisão. CEP ajuda a descobrir eventos complexos através da análise de ocorrência de outros eventos.

A utilização de CEP em sistemas RFID vem sendo proposta por diversos autores como uma abordagem de processamento de dados de eventos mais efetivo, ou seja, uma abordagem que transforme eventos RFID em eventos de negócio agregando semântica de negócio aos dados lidos.

C. Criar armazenamentos temporários de fluxos de eventos

Uma maneira de melhorar o tratamento de grandes fluxos de dados RFID é a criação de concentradores de dados próximo das fontes de dados. Concentrador de dados é um mecanismo

de software que permite armazenar temporariamente o fluxo de eventos lidos pelos dispositivos de leitura para que tais eventos possam ser processados de maneira mais eficiente.

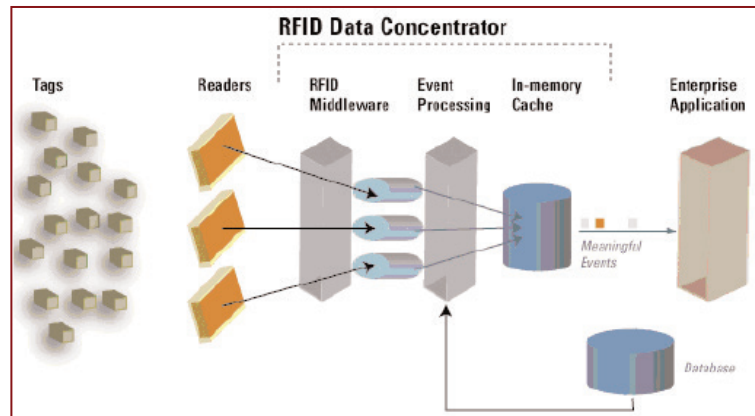


Figura 3.7 – Concentrador de dados RFID [PAL05].

Um concentrador de dados RFID pode ser composto por três elementos principais (Figura 3.7): middleware RFID, processador de eventos e mecanismo de *cache*. O mecanismo de *cache* deve permitir que o concentrador de dados explore características de processamento em tempo real. Técnicas de gerenciamento de dados em tempo real são cruciais para tratar a natureza de tempo real da tecnologia RFID, onde o uso de memória *cache* provê níveis semelhantes aos de banco de dados em termos de confiabilidade, disponibilidade e tolerância a falhas.

D. Utilizar dados de referência para contextualização da informação

Determinar eventos complexos a partir de dados de eventos simples requer o uso de informações de contexto (informações complementares), as quais geralmente são conseguidas através de dados de referência. O contexto da informação de um produto pode ser adquirido, por exemplo, através de informações contidas em uma nota fiscal ou através da união de uma nota fiscal com algum outro documento que agregue informação ao contexto do produto. O contexto pode ser criado também, através de dados de eventos EPCIS, como por exemplo, dados que determinam a origem, o destino e o fabricante de um produto. Além disso, dados de referência podem ser originados também a partir de outros sistemas, como sistemas legados de uma empresa ou qualquer outra aplicação distribuída que esteja integrada a infraestrutura RFID utilizada.

E. Distribuição federada da informação em tempo real

No item A (*tratar/digerir dados RFID perto das fontes de dados*) foi apresentada a ideia de manter mecanismos de gerenciamento de dados próximo das fontes de dados para evitar que aplicações tenham que prover funções específicas para processamento e integração de eventos.

Embora este requisito beneficie a aplicação, a distribuição física dos mecanismos concentradores de dados muitas vezes dificulta a manutenção e o agrupamento de dados de um domínio. Sistemas de middleware distribuídos facilitam o compartilhamento de dados, permitindo que eventos sejam disseminados entre concentradores envolvidos.

Os princípios de um sistema de distribuição de informação em tempo real baseiam-se na junção dos itens já apresentados, tais como: *processamento de dados perto das fontes de dados*, *utilização de concentradores de dados*, armazenamento temporário de fluxos de dados, e principalmente, o *repasse de eventos significativos entre membros de uma aliança*. Através do conceito de distribuição federada da informação, é possível melhorar a escalabilidade global de sistemas RFID deixando-os com funções de processamento o mais perto do tempo real possível.

F. Envelhecimento gracioso de dados RFID

Mesmo que se tenha uma grande capacidade de armazenamento de dados, muitas vezes se torna desnecessário armazenar enormes quantidades diárias de dados para acomodar informações RFID de uma empresa. Além disso, é desaconselhável armazenar dados operacionais para sempre, algo que pode comprometer o desempenho e a escalabilidade do sistema. Com o contínuo “envelhecimento” de dados de eventos RFID (atividade de gerenciamento da vida útil de um dado), é possível reduzir o conjunto de dados de trabalho e também a própria carga dos fluxos de eventos, mantendo conjuntos de dados gerenciáveis e altamente contextualizados.

3.3.2 ABORDAGENS PARA PROCESSAMENTO SEMÂNTICO DE DADOS RFID

Seguindo os princípios propostos em [PAL05 – Subseção 3.2.1], esta subseção apresenta uma revisão dos principais trabalhos relacionados com processamento semântico de dados de eventos RFID. Muitos dos trabalhos aqui selecionados seguem os princípios de [PAL05] e os complementam usando diferentes abordagens de processamento e integração de dados. A avaliação destas abordagens permite mapear o estado da arte da área de processamento semântico de dados de eventos RFID e também identificar possíveis questões em aberto e trabalhos futuros.

Ao todo foram avaliados 12 artigos que abordam os seguintes temas: (A) técnicas para processamento semântico de dados de eventos RFID, (B) middleware RFID de larga escala e de alto desempenho e (C) frameworks para gerenciamento de eventos RFID. A seguir, uma breve análise dos artigos relacionados a cada um dos temas identificados será apresentada.

A. Técnicas para processamento semântico de dados de eventos RFID.

O processamento de dados de eventos é uma atividade indispensável para qualquer aplicação RFID sendo considerada uma responsabilidade de sistemas de middleware RFID. Com a evolução dos requisitos de aplicações RFID, um middleware RFID deve ser capaz de processar dados de eventos complexos e com isso extrair informações significativas relacionadas a eventos de negócio. Tal requisito é uma carência das especificações do ALE. Além disso, middleware deve suportar o processamento de eventos temporais.

- **Principais problemas e soluções relacionadas:**

- Segundo os autores [BER02] [WAL07], muitas aplicações industriais baseiam seus sistemas de processamento de dados em mecanismos de programação de regras de negócio dirigidos a padrões de eventos, os quais na maioria das vezes estão baseados

em algoritmos RETE⁶. Como RETE não oferece suporte ao tratamento de eventos temporais, os autores propõem extensões do algoritmo RETE para o processamento de eventos baseado em regras com requisitos temporais visando a criação de contextos temporais que elevem a habilidade do sistema em detectar padrões de eventos.

- Dados RFID brutos (sem prévio tratamento) não apresentam nenhum significado de negócio explícito o que é um desafio para sistemas de middleware RFID. Dentro desse contexto, os autores [WU07] [WAN09] [YAO10] apresentam as seguintes propostas de solução:
 - [WU07] propõe um mecanismo de processamento de dados baseado em regras ECA (evento-condição-ação) através da tecnologia *Drools* para estender as funcionalidades de um middleware RFID. Os autores afirmam ter melhorado a modularidade, a capacidade de manutenção, a extensibilidade e o desempenho de sistemas de middleware, uma vez que a abordagem usada permite reduzir o grande volume de dados processados em conjuntos precisos de informações significativas. Os autores afirmam também, que através do uso da tecnologia *Drools*, é possível definir regras e as colocar diretamente no sistema em operação sem a necessidade de compilação da aplicação. Esta atividade é automaticamente abstraída pelo mecanismo de regras sem necessidade de reiniciar o sistema;
 - [WAN09] propõe um mecanismo de processamento de eventos complexos (CEP) baseado em regras de negócio para processamento de eventos de um middleware RFID. Tal abordagem baseia-se em modelar e tratar a lógica da aplicação RFID através de eventos complexos usando um formalismo para representar a especificação semântica e as regras de relacionamento de eventos. Os autores afirmam que a abordagem declarativa baseada em eventos simplifica, de maneira efetiva, tanto o trabalho relacionado ao processamento quanto ao processo de integração de dados. Os autores afirmam também que o framework proposto já foi integrado na tecnologia de desenvolvimento RFID da *Siemens (Siemens RFID Middleware – subseção 3.2.1)* para prover uma solução integrada para aplicações de negócio RFID;
 - [YAO10] propõe o uso de CEP para contextualização de eventos baseados em dados brutos e dados de referência integrados ao EPCIS. Os autores afirmam que existem alguns desafios e trabalhos futuros importantes como: melhorar a eficiência de mecanismos de CEP para atuar sobre dados RFID, determinar a melhor linguagem para processamento de eventos visando a integração de dados RFID com aplicações de negócio e pesquisar/analisar o uso de uma linguagem de definição de regras de negócio para diferentes tipos de aplicações visando detectar novas oportunidades de negócio e riscos inerentes;
- Em [WAN08], os autores afirmam que o processamento de dados é a questão mais importante no projeto de um middleware RFID e o resultado deste processamento não

⁶ **RETE**: é um dos primeiros algoritmos criados para detecção de eventos complexos baseado em regras de *evento-condição-ação (ECA)*. É um mecanismo muito utilizado e já foi estendido para suportar requisitos de diversas aplicações, inclusive RFID.

consegue ser genérico o suficiente para suprir os objetivos/requisitos de qualquer aplicação. Nesse trabalho, os autores apresentam a proposta de uma arquitetura de software para processamento de dados RFID baseada em CEP e regras de negócio para ser aplicado em sistemas de middleware RFID visando generalizar os requisitos de aplicação. Através dessa abordagem de processamento de eventos é possível que: (1) o contexto de negócio das aplicações seja abstraído e integrado ao sistema através de regras de negócio; (2) informações redundantes e maciças podem ser filtradas, reduzindo a carga de comunicação entre sistemas; e (3) eventos simples podem ser agregados em eventos complexos.

- Por fim, em [SU07], os autores afirmam que ALE não oferece suporte ao processamento de eventos complexos (CEP). Desta forma, no trabalho proposto pelos autores, é apresentado um mecanismo para processamento e integração de dados em tempo real com funcionalidades diferentes da especificação do ALE. Tal abordagem processa e entrega dados requisitados pela aplicação através de regras definidas nas condições de assinatura de serviços. No entanto os autores afirmam que sua proposta carece de um melhor desempenho.

B. Middleware RFID de larga escala e de alto desempenho:

Independente das características de um middleware RFID, este deve manter níveis aceitáveis de escalabilidade e desempenho para suportar os atuais requisitos das áreas de negócio. Middleware RFID com arquiteturas que ofereçam funcionalidades distribuídas é uma tendência futura pois eleva a eficiência do processamento em tempo real de grandes volumes de dados, permite que eventos complexos sejam processados de maneira distribuída, permite que dados sejam tratados perto das fontes de dados (diminuir o custo de comunicação) e eleva a confiabilidade do sistema com tolerância a falhas.

▪ Principais problemas e soluções relacionadas:

- Em [KU08] os autores afirmam que a tecnologia CEP como uma arquitetura centralizada requer um grande poder tanto computacional quanto de comunicação/rede (robustez) e apresenta certa carência em termos de escalabilidade devido ao fato de ser uma arquitetura vulnerável a falhas ou pontos de defeito. No trabalho proposto pelos autores, é apresentada uma proposta de CEP distribuído para aplicações RFID de larga escala. Os autores descrevem esta arquitetura como sendo capaz de propagar tarefas centralizadas entre múltiplos nodos/estações envolvidos(as) através de uma arquitetura de mensagens distribuídas baseado em redes *JINI*. Um mecanismo para detecção de eventos complexos distribuídos baseado na arquitetura mestre-escravo também é proposto. Os resultados dos experimentos demonstram que o desempenho da abordagem proposta é excelente para aplicações de larga escala.
- Sobrecarga no processamento de dados é um problema em sistemas de middleware RFID que possuem arquitetura do ALE centralizada (falta de robustez, flexibilidade, disponibilidade e escalabilidade) [LIU08]. Como solução, os autores apresentam uma proposta de ALE distribuída e que está dividida em implementações paralelas do ALE e mecanismos de balanceamento de carga que sirvam como coordenadores de tarefas

entre elemento ALE distribuídos. Segundo os autores, uma das vantagens desta abordagem distribuída é garantir um funcionamento operacional esperado, evitando que que sobrecargas no processamento centralizado de eventos seja um gargalo no desempenho do middleware.

- Segundo os autores em [ZIE09], existe um gap tecnológico nos padrões de serviços de integração de dados especificados pela *EPCglobal* por não suportarem esquemas de interação baseados em eventos assim como a detecção de eventos complexos distribuídos. Como solução para este problema, em [ZIE09] é apresentada a arquitetura de um middleware para redes RFID baseado em comunicação *peer-to-peer* que permite uma interação entre nodos distribuídos e que melhora a capacidade de processamento interno da rede trabalhando acima das especificações existentes (estende os conceitos do *EPC Network*).

C. Frameworks para gerenciamento de eventos RFID:

Visando uma melhor integração com aplicações de negócio e também facilitar o desenvolvimento de aplicações RFID é necessário que sistemas de middleware RFID sejam capazes de extrair regras de negócio da lógica de negócio das aplicações, suportar configurações dinâmicas de acordo com alterações na própria lógica de negócio e utilizar regras de negócio para processamento de eventos complexos.

▪ Principais problemas e soluções relacionadas:

- Segundo os autores em [KIM07], sistemas RFID devem implementar funções para processar grandes volumes de dados de eventos e suportar configurações dinâmicas de acordo com mudanças de negócio (contexto do negócio). Embora aplicações RFID consumam e ajam diretamente sobre eventos, extrair regras de negócio da lógica de negócio da aplicação conduz a uma melhor modularização do sistema, o que conseqüentemente eleva a capacidade de manutenção da aplicação. Nesse trabalho os autores apresentam um framework RFID ciente do contexto do negócio e que envolve a integração de técnicas para processamento de eventos complexos (CEP) com instâncias do *EPC Network* (modelo de dados baseado em eventos de negócio (*BESpec* e *BEReports*)). Como trabalho futuro, os autores pretendem estender o framework para contextualizar informações processadas de acordo com a situação do ambiente de aplicação. A intenção dos autores é que o framework tenha capacidade para reconhecer, dinamicamente, várias mudanças de contexto, incluindo RFID e também vários outros tipos de sensores, tais como temperatura, umidade, localização, entre outros, além de combinar sensores com a lógica de negócio.
- Segundo os autores em [WEL08], um sistema de processamento de eventos deve fornecer uma infraestrutura que permita especificação, extração e gerenciamento de eventos significativos a partir de dados RFID brutos. Os autores apresentam um sistema para desenvolvimento de aplicações baseadas no processamento de eventos complexos e que provê três serviços principais: (1) permite que desenvolvedores de aplicações e também usuários do sistema possam especificar eventos de interesse através de uma linguagem declarativa de definição de eventos ou usando uma interface

gráfica com uma linguagem visual, (2) provê uma API que facilita o desenvolvimento de aplicações RFID baseada em eventos, e (3) provê um mecanismo para processamento e integração de dados.

3.4 RESUMO E TENDÊNCIAS FUTURAS

Nesse capítulo foram apresentados diversos trabalhos relacionados com a tecnologia de sistemas de software RFID. Mais especificamente, trabalhos focados no gerenciamento e integração de dados RFID. O objetivo deste esforço foi apresentar o estado da arte dos conceitos e sistemas relacionados e com isso poder estabelecer um embasamento técnico que ajude a organizar não apenas os problemas da área e suas reais soluções, mas também conseguir expor ideias que sirvam de fomento para trabalhos futuros e que venham a mitigar desafios ainda existentes.

Os trabalhos apresentados foram divididos em dois grandes grupos: “principais tecnologias RFID existentes (Seção 3.2)” e “abordagens para processamento semântico de dados RFID (Seção 3.3)”. No primeiro grupo foram apresentados trabalhos relacionados tanto com gerenciamento quanto com integração de dados RFID, onde estão presentes trabalhos acadêmicos e produtos comerciais. Já o segundo grupo concentra-se em abordagens acadêmicas focadas em técnicas e métodos para processamento semântico de dados RFID o que muitas vezes envolve também atividades de integração de sistemas.

Dos requisitos avaliados na análise dos trabalhos apresentados, existem aqueles que são contemplados pela maioria dos sistemas/trabalhos existentes (antigos desafios) e também aqueles que nunca são considerados ou são considerados em poucos trabalhos (novos desafios). E é nestes novos requisitos e desafios que estamos interessados nesta tese de doutorado, pois embasam as novas ideias propostas neste trabalho.

Dentro do contexto de novos desafios, é possível resumir os principais requisitos em: *(1) capacidade de gerenciamento inteligente e flexível de dados RFID baseado no processamento de eventos complexos e temporais (CEP), (2) suporte a contextualização de eventos baseado em dados de referência, (3) middleware RFID com funcionalidades distribuídas que eleve a confiabilidade, o desempenho e a escalabilidade do sistema, (4) configurações dinâmicas dos mecanismos de gerenciamento e integração de dados de acordo com regras de negócio da aplicação e (5) elevação da capacidade de manutenção e administração da infraestrutura RFID para um melhor suporte à integração de sistemas RFID.*

4. ABORDAGEM DE TRABALHO

Este capítulo apresenta o plano de solução e as abordagens científicas usadas nesta tese para construção do framework RFID proposto. O plano de solução sintetiza, através de um conjunto de definições, quais são as abordagens usadas neste trabalho para tentar mitigar os problemas apresentados e que estão relacionados a limitações e deficiências funcionais de alguns serviços especificados para o *EPC Network* segundo a *EPCglobal*. Detalhes de como a solução foi implementada seguindo estas definições serão abordados no capítulo 5.

4.1 PLANO DE SOLUÇÃO

Neste trabalho é apresentada a proposta de um **framework de software RFID**. O objetivo deste framework é elevar a capacidade de integração interorganizacional de sistemas de middleware RFID. A tecnologia CEP é integrada a um sistema de middleware RFID já existente (*CloudRFID*) [AMA09a] [AMA09b] e que implementa as especificações do ALE segundo a *EPCglobal* (Capítulo 5).

O framework baseia-se no conceito de CEPs cooperantes para processamento e notificação de eventos complexos. Cada nodo de processamento envolvido tem seu próprio mecanismo de CEP e a cooperação entre nodos provê um sistema de regras de negócio global que permite a decomposição e execução de regras de negócio em diferentes organizações [DUT07].

Para elevar o nível de abstração da integração de sistemas proposta neste framework, estende-se as abordagens usadas em [ZIE09] através de serviços de notificação de eventos para mitigar os problemas de integração de eventos RFID. Com isso, é possível definir a decomposição de uma regra de negócio entre CEPs de diversas organizações e assim facilitar a identificação de eventos complexos distribuídos.

Através do serviço de notificação de eventos se reduz a sobrecarga de comunicação provocada pelas atividades de consulta de eventos no processamento de eventos distribuídos. Essa redução na comunicação ocorre pois as abordagens usadas no framework substituem as consultas de eventos (realizadas pela aplicação em nodos distribuídos) por técnicas de notificação de eventos pré-configurados. Desta maneira, aplicações são notificadas apenas sobre eventos de interesse.

A Figura 4.1 apresenta uma visão geral da arquitetura proposta. A responsabilidade da aplicação nesta arquitetura é utilizar as interfaces de integração oferecidas pelo mecanismo de CEP para integrar objetivos e processos de negócio com eventos RFID de interesse. Essa integração ocorre através da associação entre eventos RFID e regras de negócio visando mapear a inter-relação entre lógica de negócio e eventos complexos [KIM07].

O framework permite configurações baseadas no contexto do negócio, ou seja, através da decomposição e mapeamento da lógica e processos de negócio em regras para identificação de eventos complexos, mecanismos de suporte permitem a configuração da estrutura física (middleware e dispositivos de leitura RFID) e lógica do sistema, assim como o relacionamento entre eles.

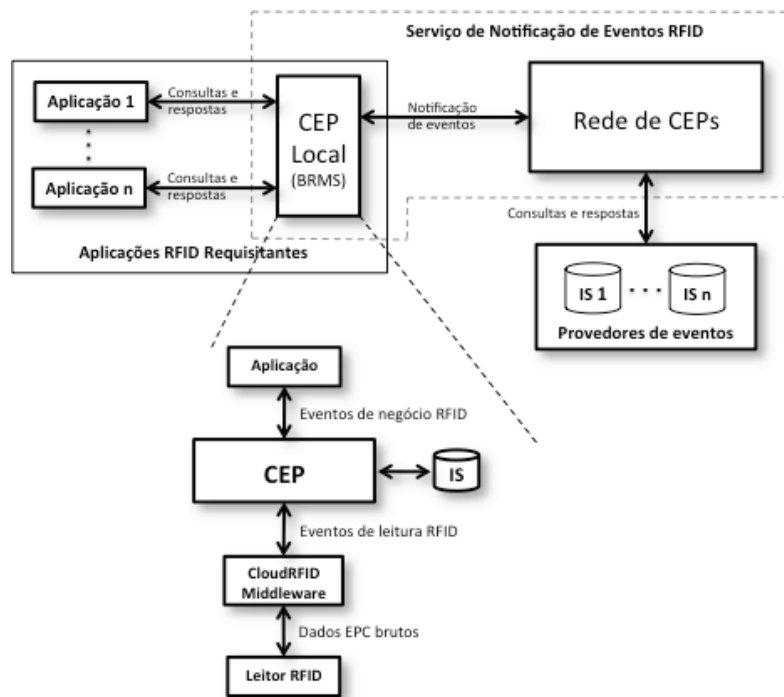


Figura 4.1 – Arquitetura do framework desenvolvido.

O framework também oferece funções tanto para transformação semântica de dados em eventos significativos, quanto para monitoramento e execução de regras de negócio tendo autonomia para automatizar a invocação de processos de negócio quando necessário. Estas funcionalidades permitem que o mecanismo de CEP realize o papel da aplicação de captura [FOS11], provendo meios de especificar e executar regras de negócio e interpretar e executar a lógica do negócio automaticamente.

Quanto à identificação de eventos distribuídos, o framework oferece uma forma simples de integração entre processamento de eventos complexos intraorganizacional e interorganizacional. Sempre que uma regra de negócio é definida e decomposta entre nodos internos da organização, tal regra não é registrada em outros CEPs e o processamento é local e hierárquico entre os nodos envolvidos [DUT07]. No entanto, quando uma regra necessita de eventos interorganizacionais, a regra é declarada no CEP local e repassada para os nodos CEPs envolvidos. Assim, cada nodo sabe quais eventos devem ser identificados e também quais ações de cooperação devem ser executadas (como a notificação de um evento interorganizacional ao nodo requisitante).

4.2 EMBASAMENTO TEÓRICO

O embasamento teórico permite identificar conceitos e abordagens científicas que fundamentam o plano de solução apresentado. As abordagens científicas estão classificadas de acordo com os requisitos funcionais identificados na seção 1.1 e apresentam um pouco mais de detalhes de alguns dos trabalhos já apresentados no capítulo 3.

(Requisito Funcional 1) “Mitigar os gaps existentes na especificação do ALE tanto para limitações na capacidade de processamento de eventos RFID complexos (gap no processamento semântico de dados) quanto na falta de contextualização de dados para integração com processos de negócio (gap de integração de dados entre middleware e

repositório de dados EPCIS), provendo gerenciamento inteligente e flexível de dados RFID baseado em regras de negócio para contextualização de eventos e em dados de referência”.

De acordo com padrões existentes e também com o que foi apresentado no capítulo 2, o principal meio de integração de dados entre middleware RFID e processos de negócio de uma aplicação é através de uma aplicação de captura de eventos. Uma aplicação de captura de eventos deve receber como entrada (*input*), relatórios (*ALE ECR*Reports) contendo resultados de ciclos de eventos processados por algum sistema de middleware RFID e produzir como saída (*output*), eventos EPCIS (*EPCIS Events*) que são armazenados no repositório de dados do serviço de informação EPCIS. Tal transformação de dados requer funções específicas para tratamento de eventos por parte da aplicação de captura, visto que é necessário incorporar informações da situação e do contexto em que eventos são capturados e dar a eles uma semântica de negócio adicional.

A Figura 4.2 exemplifica a importância de uma aplicação de captura de eventos no contexto do *EPC Network*. Neste exemplo, a *aplicação de captura (Palletizer Capture App)* acrescenta semântica de negócio ao dado EPC capturado e agrega informações através da interpretação da situação e do contexto operacional do ambiente de negócio. Tal contexto deve ser baseado na lógica de negócio da aplicação e deve embasar algum tipo de regra que identifique os eventos ocorridos e os interprete como eventos de negócio.

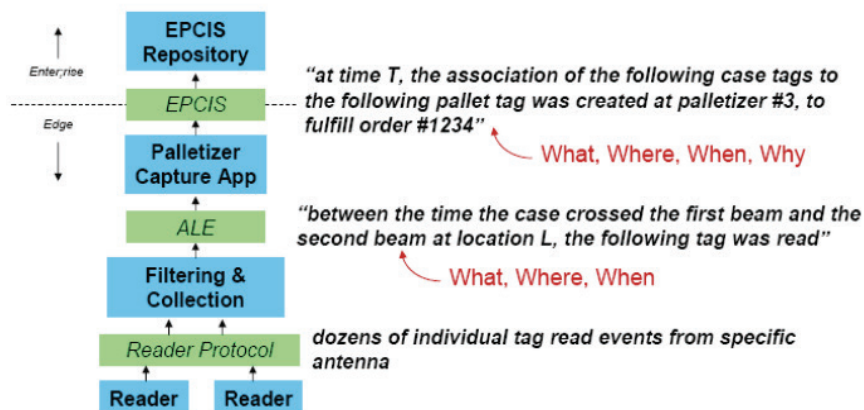


Figura 4.2 – Exemplo de uma aplicação de captura contextualizando dados RFID.

A *EPCglobal* não considerou a aplicação de captura na especificação inicial do *EPC Network*, e não definiu nenhum módulo ou conceito que facilite ou acelere a velocidade de integração de sistemas RFID com processos de negócio de aplicações. No entanto, em sua última versão a especificação do *EPC Network* foi atualizada e passou a prover uma especificação simples para a aplicação de captura a qual foi implementada em [FLO07].

A solução inicialmente proposta em [FLO07] baseia-se em transformar relatórios ALE (*ALE ECR*Reports) em eventos do tipo EPCIS (*EPCIS Events*). Esta solução não é muito viável pois requer a implementação das integrações e transformações desejadas via programação da aplicação de captura, o que dificulta o processo de criação (instanciação) de novas aplicações de captura. Isso ocorre, porque a lógica de negócio tem que ser reprogramada para cada novo cenário de negócio a ser implantado.

Em [FOS11] os autores estenderam as abordagens usadas anteriormente em [FLO07] através do uso da tecnologia CEP e propuseram melhorias para o desenvolvimento de novas aplicações de captura de eventos RFID. A tecnologia CEP vem sendo aplicada em RFID para melhorar as funcionalidades de mecanismos de gerenciamento semântico de dados RFID onde abordagens de processamento de dados baseadas em regras de eventos declarativos simplificam atividades de processamento e reduzem o custo com atividades de integração de dados.

Embora a nova abordagem [FOS11] seja mais eficiente em relação a abordagem anterior [FLO07], ela apresenta algumas restrições funcionais devido a falta de ferramentas (classes e métodos específicos) que auxiliem no projeto e desenvolvimento de novas aplicações de captura. Isso ocorre mesmo Fosstrak permitindo integração com motor de regras CEP pois como atividades de captura e contextualização de eventos RFID estão baseadas em regras de negócio, a criação destas regras necessita de um alto nível de conhecimento de eventos e objetos disponíveis no ambiente de programação do sistema. Além disso, esta relação do Fosstrak com motores de regras CEP não permite integração e configuração de diversas regras de negócio (diversos contextos de negócio) com elementos do sistema e carece de uma inteligência de negócio para transformação semântica de diversos tipos de dados, sejam estes, dados de eventos RFID (relatórios ALE) ou eventos já contextualizados (eventos EPCIS) oriundos de outros repositórios EPCIS.

Compartilhar eventos de negócio (eventos EPCIS) ao invés de dados de eventos RFID é o limiar nos estudos de integração da tecnologia CEP com aplicações RFID. Isso ocorre, pois, quanto mais se eleva o nível de abstração funcional de sistemas RFID (Figura 4.3), maior é a capacidade de abstração de dados e maior é o poder de integração destes dados com aplicações de negócio, permitindo uma diminuição das responsabilidades da aplicação. Esse aumento na abstração de dados aumenta também o significado semântico das informações relacionadas com processos de negócio elevando a capacidade de automação dos sistemas.

CEP pode desempenhar o papel da aplicação de captura através da contextualização de eventos RFID em eventos EPCIS baseados em regras de negócio pré-estabelecidas. O tratamento semântico de informações RFID se dá através do contexto de negócio, o qual integra e transforma a lógica de negócio em eventos complexos. Com isso, através da detecção de eventos complexos, é possível automatizar a lógica de negócio envolvida (Figura 4.3).

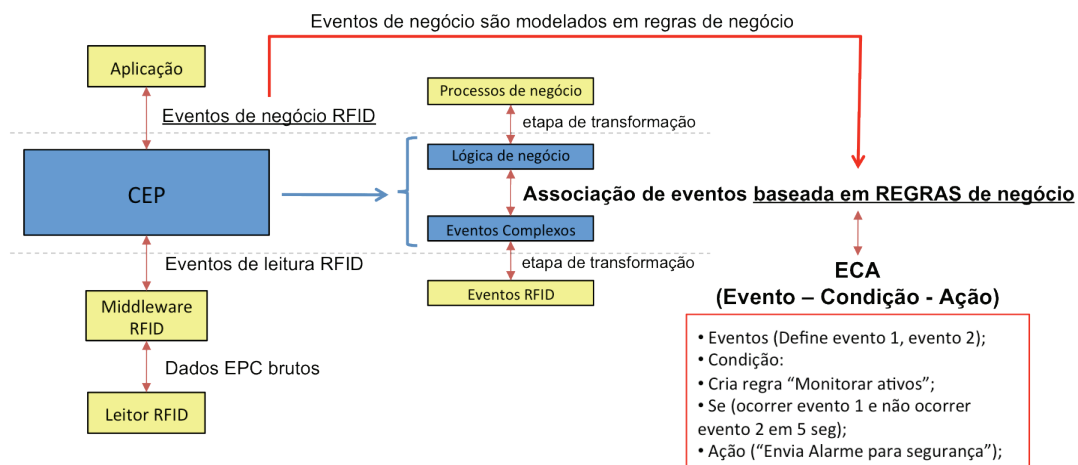


Figura 4.3 – Exemplo de regra de negócio para elevar o nível de abstração de dados RFID.

(Requisito Funcional 2) “Suportar configurações dos mecanismos de gerenciamento e integração de dados de acordo com regras de negócio da aplicação, elevando a capacidade de manutenção e administração do sistema”.

Recentemente alguns trabalhos foram propostos com intuito de desenvolver sistemas cientes do contexto de negócio da aplicação. No entanto, muitos destes trabalhos relacionam-se com dispositivos sensores em geral e não suportam a tecnologia RFID.

Um dos trabalhos mais relevantes para integração do contexto de negócio com aplicações RFID é [KIM07]. Neste trabalho os autores afirmam que para se obter o máximo de benefícios da tecnologia RFID, um sistema RFID deve ser capaz de prover funções para processar grandes volumes de dados e também poder ser configurado dinamicamente de acordo com mudanças nos processos de negócio da aplicação. Como a maioria das aplicações RFID consomem eventos e agem diretamente sobre eventos, extrair regras de negócio da lógica de negócio conduz a uma melhor separação (desacoplamento) dos módulos do sistema, o que conseqüentemente eleva a capacidade de manutenção da aplicação.

Em [KIM07] é apresentado um framework de software RFID ciente do contexto de negócio que é aplicado diretamente a processos de negócio e se integra a instâncias do *EPC Network*. Embora as abordagens utilizadas pelos autores sejam eficientes na contextualização de dados e na automação de processos de negócio, o mecanismo de CEP proposto não permite o processamento de eventos complexos distribuídos entre organizações, o que é uma funcionalidade importante para os atuais requisitos de escalabilidade da tecnologia RFID.

Além disso, [KIM07] também apresenta algumas das deficiências existentes no sistema Fosstrak. Estas deficiências não permitem integração e configuração de diversas regras de negócio com elementos do sistema e também carecem de uma inteligência de negócio para transformação semântica de diversos tipos de dados, sejam estes, dados de eventos RFID (relatórios ALE) ou eventos já contextualizados (eventos EPCIS) oriundos de outros repositórios EPCIS.

(Requisito Funcional 3) “Mitigar o gap existente na especificação do EPC Network nas limitações dos serviços de compartilhamento de dados RFID entre organizações (ONS, DS e EPCIS), os quais não permitem esquemas de interação indireta de eventos e dificultam a detecção de eventos complexos distribuídos”.

Integrar dados de eventos RFID a partir de diversas fontes de informação em uma rede de sistemas RFID integrados é um requisito importante para a tecnologia RFID. Para facilitar essa integração, a *EPCglobal* vem desenvolvendo uma série de especificações de serviços para compartilhamento de informações RFID entre organizações. No entanto, os serviços especificados não suportam esquemas de comunicação baseados em eventos, assim como a detecção de eventos complexos distribuídos, o que é considerado um *gap* tecnológico para os padrões atuais da tecnologia RFID.

Em [ZIE09] os autores apresentam a arquitetura de um middleware para redes RFID. Esta abordagem permite interações entre nodos distribuídos através de eventos e estende os conceitos das especificações existentes para o *EPC Network*. Os autores chamam a proposta de “serviços

de notificação”, os quais facilitam a interação de eventos em cenários de compartilhamento de informações RFID entre organizações.

Dentro deste contexto de compartilhamento de informações RFID, o EPCIS foi especificado para suportar apenas endereçamento direto de recursos o que necessita uma constante adaptação dos links de comunicação entre aplicações requisitantes e fontes de informação de eventos RFID. Esse modelo de comunicação exige um fluxo de informação para cada relação entre produtor e consumidor de eventos degradando o sistema com a grande quantidade de comunicação necessária. Essa característica é um contraste em relação a sistemas que não precisam de interação direta onde o padrão de comunicação não exige que as entidades envolvidas se conheçam.

No compartilhamento de dados usando ONS (Figura 4.4 - esquerda), quando determinado servidor ONS é consultado sobre a existência de um código *EPC*, este servidor responde indicando o endereço do primeiro servidor EPCIS na cadeia de distribuição o qual irá reportar o evento correspondente e indicar o próximo EPCIS envolvido e assim por diante. Nesta abordagem, cada elemento da cadeia de distribuição precisa suportar este fluxo de informação para que o compartilhamento de informações seja realizado. O DS (Serviço de Descoberta) foi proposto para superar estas limitações, ou seja, foi projetado para descobrir fontes de informação de dados RFID sem a necessidade de consultá-las individualmente.

A Figura 4.4 (centro) ilustra uma atividade de consulta de eventos RFID usando o Serviço de Descoberta. Nesse modelo de comunicação o ONS quando consultado por determinado código *EPC* responde indicando qual DS possui conhecimento sobre servidores *EPCISs* que detenham informações sobre o *EPC* consultado. Desta forma, a aplicação pode consultar o DS diretamente, o qual irá retornar uma lista com todos os *EPCISs* envolvidos com o *EPC* consultado. Através desta abordagem é possível que a aplicação solicite eventos relacionados aos *EPCISs* envolvidos de uma única vez.

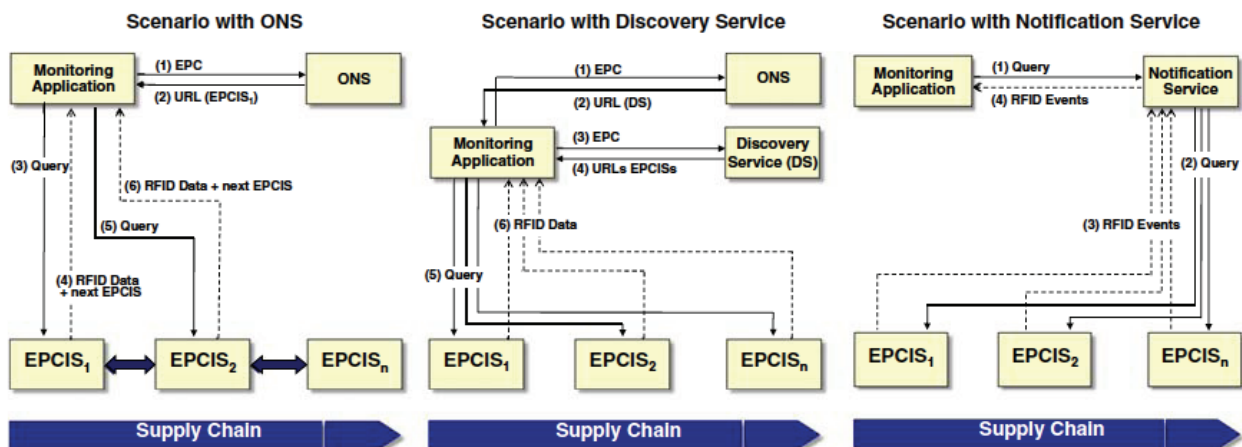


Figura 4.4 – Modelos de integração de dados entre organizações [ZIE09].

Os modelos de interação definidos pelo *EPC Network* podem ser classificados em “endereçamento direto e indireto” e “interações iniciadas pelo requisitante ou pelo provedor”. O EPCIS suporta quase todos os tipos de modelos de interação para descobrimento de dados, exceto o modelo de interação indireta [ZIE09]. Usando ONS e DS o requisitante recebe informações de localização das fontes de dados sem o endereçamento direto destas fontes. Falta

suporte a um modelo de interação baseado em eventos e que combine endereçamento indireto com interações iniciadas pelo provedor. Essa abordagem é uma das recomendações de [ZIE09] onde os autores propõem uma arquitetura de consulta e monitoramento flexível (Serviço de Descoberta flexível) que trabalhe acima dos modelos de interação especificados para o EPCIS.

A falta de modelos de interação baseados em eventos impõe desafios na flexibilidade da aplicação. Qualquer alteração na estrutura física dos elementos requer adaptação nos canais de comunicação. Já nos modelos de interação baseado em eventos, serviços de notificação admitem um endereçamento indireto das fontes de informação. Tal característica permite o desacoplamento entre aplicações de monitoramento e fontes de dados permitindo uma abstração das mudanças na estrutura da cadeia de suprimento.

Uma arquitetura baseada em Serviços de Notificação de eventos reduz a complexidade da aplicação de monitoramento de dados RFID [ZIE09]. Esta abordagem permite a criação de um canal de comunicação com todas as fontes de informação reduzindo a carga de trabalho e de comunicação entre produtores e consumidores de eventos RFID (Figura 4.4 - direita). Tal modelo de comunicação baseia-se em algumas características importantes, tais como: escalabilidade, flexibilidade, descentralização, compatibilidade com padrões existentes e suporte a consultas de eventos complexos.

Embora as abordagens propostas em [ZIE09] permitam compartilhamento de informações RFID entre diferentes organizações, tais propostas não especificam a integração do mecanismo de CEP distribuído com o processamento de eventos locais e hierárquico em uma organização, permitindo integração de informações RFID tanto intraorganizacional quanto interorganizacional.

(Requisito Funcional 4) “Prover uma arquitetura colaborativa que eleve a escalabilidade do sistema na integração de eventos”.

Além da abordagem apresentada no item anterior, em [DUT07] é apresentada uma arquitetura de sistema RFID com funcionalidades distribuídas e que é relevante para os objetivos desta tese. Os autores afirmam que extrair informações significativas de grandes volumes de dados RFID brutos e em tempo real é uma tarefa complicada e exige um alto desempenho das atividades de processamento. Mecanismos de CEP centralizados podem ser o gargalo responsável pela falta de desempenho no processamento visto que quanto maior o número de eventos a serem monitorados, maior também é o número de regras necessárias para tal monitoramento.

Em [DUT07] os autores propõem um middleware distribuído para tratamento de eventos complexos através da execução de regras de negócio em tempo real. Nessa arquitetura, cada elemento de software envolvido tem seu próprio mecanismo de CEP. Cada nodo realiza seu processamento local e coopera com outros nodos envolvidos para implementar um sistema de regras de negócio global (CEP Global).

A arquitetura proposta em [DUT07] baseia-se basicamente em (Figura 4.5):

- Eventos complexos precisam do monitoramento e da combinação de eventos básicos de outros CEPs envolvidos (outras fontes de informação). Desta forma é preciso consolidar eventos de múltiplas fontes de informação.

- É preciso determinar nodos primários (CEPs primários) para cada evento de interesse global seguindo a organização hierárquica da rede RFID relacionada (ex. região – estado – cidade).
- Eventos de interesse (ou regras de negócio) são especificados em qualquer nodo distribuído.
- Um evento de interesse de determinado nodo pode ter atributos específicos para indicar o nível hierárquico estipulado.
- Quando um evento de interesse é definido em um nodo, é possível definir também em que nodos os eventos básicos que compõem esse evento de interesse devem ser identificados. Com isso, os eventos básicos relacionados devem ser propagados para os demais nodos (CEPs) envolvidos na hierarquia.
- Quando os eventos básicos que compõem a regra de negócio (ou o evento de interesse) são identificados nos nodos definidos, eles são automaticamente repassados para o nodo primário (nodo CEP onde a regra foi definida).
- As regras de eventos (eventos de interesse) são ativadas (aplicar as condições de identificação e as ações a serem tomadas) somente no nodo primário (nodo CEP primário).

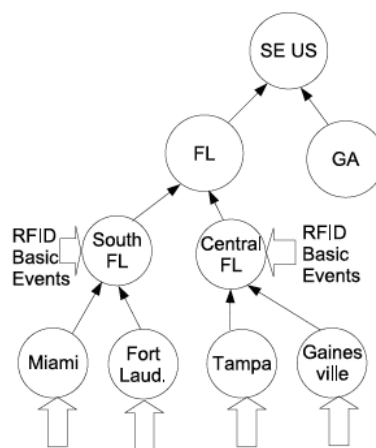


Figura 4.5 – Exemplo de uma arquitetura de CEP distribuída [DUT07].

As ideias propostas em [DUT07] não especificam como o mecanismo de CEP monitora a execução de eventos complexos entre os nodos distribuídos. Além disso, os autores afirmam que as ideias propostas não foram implementadas.

Além de [DUT07], outros dois trabalhos também são relevantes para o embasamento teórico do framework proposto [KU08] [LIU08] e já foram apresentados no capítulo 3.

5. DESCRIÇÃO DO TRABALHO

O trabalho desenvolvido nesta tese concentra atividades em três etapas principais: (1) definição e criação da **plataforma de referência**, (2) criação de soluções e melhorias relacionadas ao **gerenciamento de dados de eventos RFID** e (3) criação de soluções e melhorias relacionadas a **integração e ao compartilhamento de dados de eventos RFID**. Detalhes sobre etapas envolvidas e como as soluções propostas no framework foram implementadas serão apresentados nas seções a seguir.

A Seção 5.1 apresenta a plataforma de referência (etapa 1). Enquanto o framework para processamento e notificação de eventos complexos é apresentado na seção 5.2 (etapas 2 e 3).

5.1 PLATAFORMA DE REFERÊNCIA

5.1.1 MIDDLEWARE RFID (CLOUDRFID)

CloudRFID é um sistema de middleware RFID composto pelos seguintes módulos (Figura 5.1): **Interface de Aplicação (ALE)**, **Camada de Eventos** e **Gerenciador de Dispositivos** (Interfaces de dispositivos e Plugins de dispositivos). O middleware foi desenvolvido em JAVA e pode ser implantado como um serviço rodando em um servidor de aplicação web ou como uma API.

A **Interface de Aplicação** é um webservice que provê meios para que aplicações clientes busquem dados e executem operações em um ou mais dispositivos de leitura RFID. Isso é possível devido a implementação da interface ALE definida pela *EPCglobal*. Como é necessário o acesso a dados de eventos filtrados pelo middleware por mais de uma aplicação cliente (disseminação de dados e suporte a múltiplas aplicações), *CloudRFID* suporta diferentes métodos para troca de mensagens, como SOAP/XML sobre JMS, TCP e HTTP.

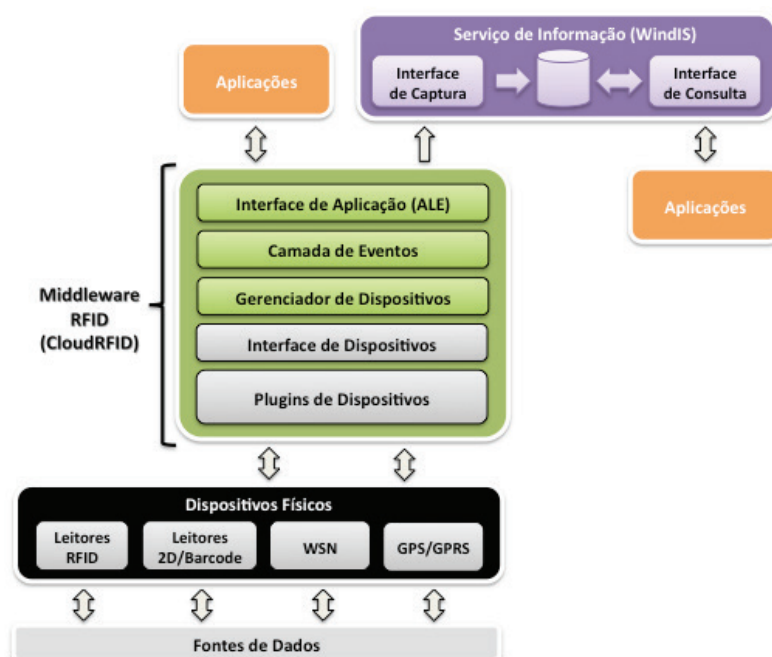


Figura 5.1 – Módulos do Middleware RFID (*CloudRFID*).

A **Camada de Eventos** é encarregada por administrar ciclos de eventos de leitura e escrita (suporte a leitura e escrita em etiquetas). Ela é uma camada de software que realiza a disseminação de dados capturados para aplicações requisitantes. Tal disseminação pode ser feita tanto de forma síncrona quanto assíncrona. Além disso, ciclos de eventos podem apresentar diferentes latências para envio de relatórios contendo dados de eventos ocorridos pois aplicações podem ter diferentes necessidades quanto à frequência que desejam receber estes dados.

Devido à grande quantidade de etiquetas RFID lidas durante um ciclo de leitura, realiza-se a filtragem e agregação de dados (suporte a filtragem e agregação de dados) pois nem todos os eventos gerados pelos dispositivos de leitura são de interesse da aplicação. A filtragem de eventos RFID é realizada através da eliminação de etiquetas duplicadas. Enquanto que a agregação é feita através do uso de expressões regulares que separam etiquetas de acordo com máscaras de filtragem de dados passadas durante a configuração do ciclo de evento, como foi previsto na especificação ALE. Além disso, é possível também configurar o relatório do ciclo de eventos através da parametrização do formato dos dados e de quais dados serão relevantes para a aplicação (suporte a interpretação de dados).

A aplicação pode determinar a duração do ciclo de eventos de leitura, sua latência, o que deve ser reportado, de onde buscar dados e quando o ciclo deve ser iniciado e terminado. Para isso, basta configurar o middleware RFID através da interface de parametrização do ciclo de leitura. Ao terminar o ciclo de leitura, um relatório deve ser enviado para a aplicação contendo as informações lidas das etiquetas RFID e também os problemas ocorridos no processo de identificação (suporte a confiabilidade).

O **Gerenciador de Dispositivos** é um módulo que gerencia e controla dispositivos RFID. Ele é responsável pela configuração e coordenação das atividades operacionais de um ou mais dispositivos de leitura e pela verificação dos seus status. Ele também informa para a camada de eventos quais são as etiquetas lidas e os problemas ocorridos durante a leitura. Etiquetas que não foram corretamente lidas não serão reportadas.

A **Interface de Dispositivos** é responsável pela abstração lógica que provê suporte a novos dispositivos de leitura RFID (suporte a novos dispositivos e protocolos de leitura). Esta abstração lógica permite que a aplicação, de forma transparente, interaja com diferentes dispositivos, sendo que cada dispositivo implementa uma interface própria para integrar com o gerenciador de dispositivos. Esta característica permite que novos elementos sejam acoplados ao *CloudRFID* como novos plugins (camada de **Plugins de Dispositivos**).

5.1.2 SERVIÇO DE INFORMAÇÃO DE EVENTOS EPCIS (*WINDIS*)

O *WindIS* é um Serviço de Informação EPC que está dividido em três partes funcionais principais: **Repositório de Dados de Eventos**, **Consulta** e **Captura de Eventos** (Figura 5.2). O *WindIS* foi desenvolvido em JAVA e deve ser implantado como um serviço em um servidor de aplicação web.

Na arquitetura do *WindIS* (Figura 5.2), é possível identificar dois *webservices*, um para integração com o **módulo de consulta de eventos** e outro para integração com o **módulo de captura de eventos**. Na prática, os módulos de consulta e captura de eventos são duas interfaces que podem ser instanciadas através de *webservices* onde cada interface é proposta pelo EPCIS

da *EPCglobal*. Já o **repositório de dados de eventos** é um conjunto de especificações de objetos e parentesco entre classes de objetos que servem de modelo de dados para captura e consulta de eventos EPCIS (conforme definições da seção 2.1.6). O *WindIS* utiliza *XML* como formato padrão de dados e suporta protocolos padrões de comunicação (*HTTP*, *TCP* e *WSDL/SOAP*).

É possível considerar o repositório como um conjunto de objetos lógicos relacionados aos processos de negócio da aplicação e que podem ser consultados e adicionados em repositórios via *webservice*. O módulo de repositório tem como função básica definir quais são os objetos que serão persistidos no banco de dados. Estes objetos interessam à aplicação e devem ser armazenados para que posteriormente sejam recuperados. O *WindIS* é compatível com diferentes sistemas de gerenciamento de bancos de dados, como *Oracle*, *PostgreSQL*, *MySQL*, etc.

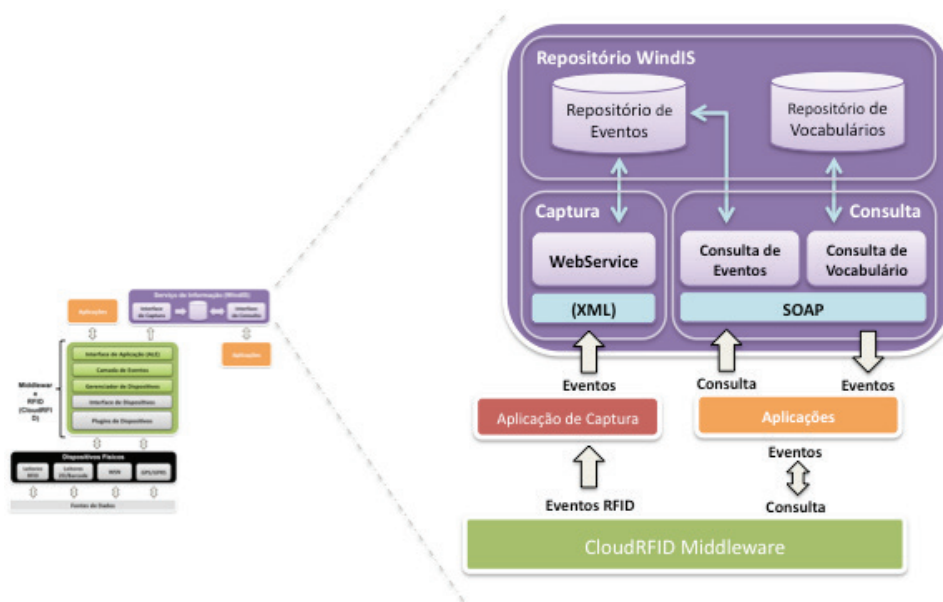


Figura 5.2 – Módulo do Serviço de Informação EPCIS (*WindIS*).

O módulo de consulta de eventos fornece uma interface de abstração para a aplicação. A aplicação terá que instanciar um *webservice* e a partir deste projeto, realizar consultas sobre dados armazenados no repositório. Estas consultas, por sua vez, não se restringem a informações brutas provenientes do *middleware* RFID. As consultas realizadas pela aplicação são direcionadas ao seu contexto de negócio sem a necessidade de tratar algum detalhe técnico sobre os dados apresentados. Através das respostas geradas, a aplicação tem capacidade de tomar decisões conforme o plano de negócio estabelecido.

O módulo de captura permite que a aplicação possa usufruir do Serviço de informação EPC (*EPCIS*) sendo necessário que o repositório esteja sempre atualizado com registros de eventos e transações do mundo real provindos pelo *middleware* RFID (*CloudRFID*). Esta atividade é responsabilidade do componente de captura (aplicação de captura) que faz a interface de comunicação entre o IS e o *middleware* RFID.

5.2 FRAMEWORK PARA PROCESSAMENTO E NOTIFICAÇÃO DE EVENTOS COMPLEXOS

5.2.1 VISÃO ARQUITETURAL E FUNCIONALIDADES DO FRAMEWORK

O Framework RFID é um pacote de software desenvolvido em JAVA e formado por três sistemas independentes (**Mecanismo de CEP**, **Middleware RFID** e **IS**) que juntos oferecem funcionalidades de **captura, processamento, armazenamento, integração e compartilhamento** de dados de eventos RFID. Assim como nos módulos de Middleware e IS, o Mecanismo de CEP também pode ser implantado como um serviço em um servidor de aplicação web ou como API.

Conforme pode ser visto na figura 5.3, o **módulo de CEP** contém um **motor de regras de negócio** capaz de realizar processamento de eventos complexos, como por exemplo, contextualizar eventos providos do middleware RFID (ALE *ECReports*) em eventos de negócio (EPCIS). O módulo de CEP também é responsável pela detecção de eventos complexos que possam ser gerados a partir da sequência de eventos de outras fontes de informação (outros CEPs, por exemplo). O motor de regras está baseado no *Drools*, um sistema gerenciador de regras de negócio [DRO11]. As regras criadas no módulo de CEP são descritas conforme linguagem de `definição de regras usada pelo *Drools*.

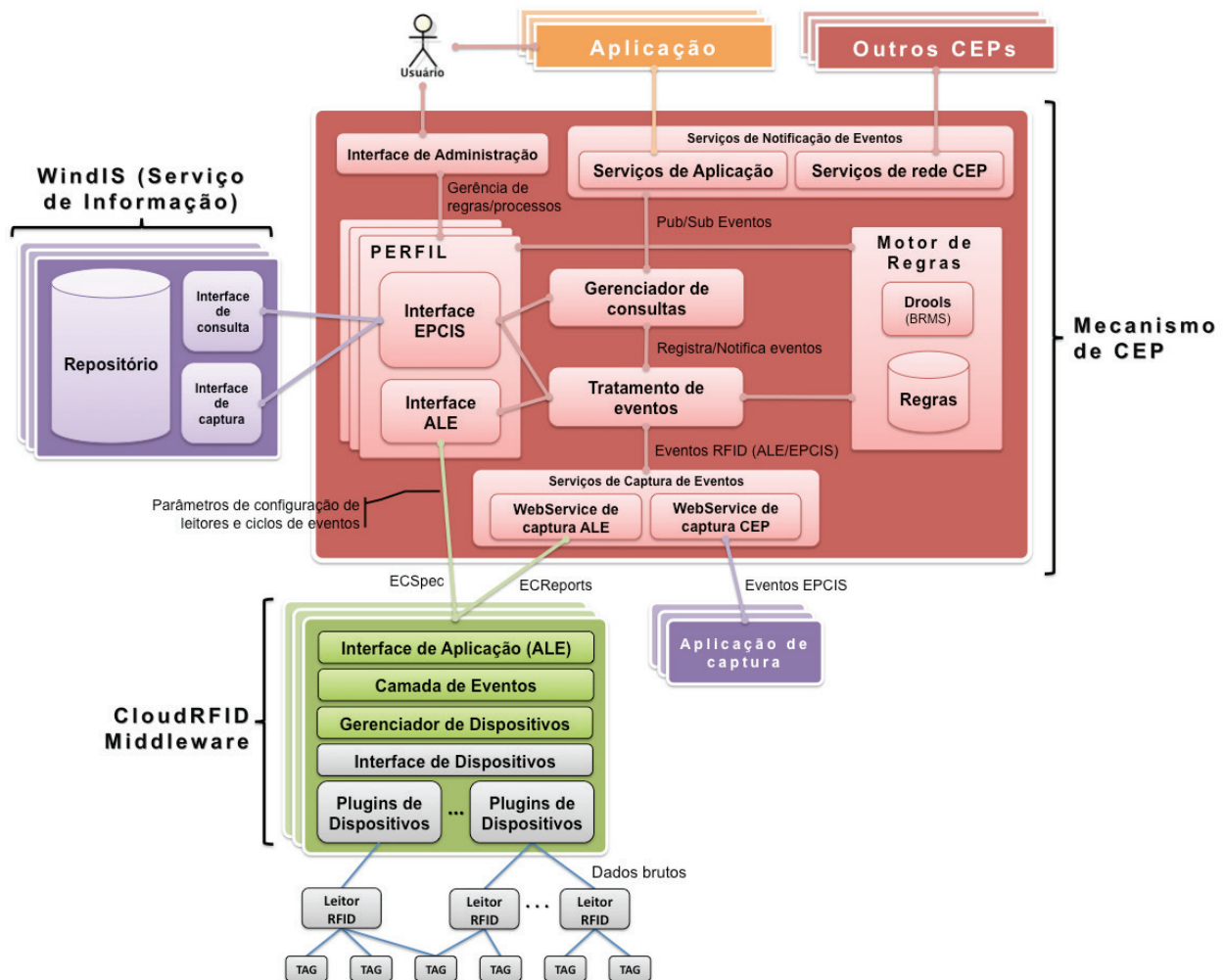


Figura 5.3 – Visão geral do framework RFID.

Eventos RFID (*ECReports*) são capturados através de um **serviço de captura de eventos (WebService de captura ALE)** e contextualizados (transformados) em eventos de negócio (EPCIS) através do motor de regras. Após a contextualização, estes eventos são armazenados no **Serviço de Informação EPC (EPCIS)**.

O mecanismo de CEP também possui um **serviço de captura de eventos EPCIS (WebService de captura CEP)**. Este serviço permite que o mecanismo de CEP receba eventos de outras aplicações de captura (outras fontes de dados ou outros CEPs) para serem processadas localmente, para armazenamento ou até mesmo para gerar novos eventos de negócio.

Eventos capturados são gerenciados através de um **módulo de tratamento de eventos** que é responsável pelo envio de eventos para o motor de regras. Este módulo também se comunica com outras instâncias de middleware (ALE) e serviços de informação EPC (EPCIS) através de **webservices que implementam as interfaces ALE e EPCIS** (interface de captura e consulta), podendo executar comandos de leitura em outros sistemas de middleware ALE e consultas no(s) repositório(s) EPCIS(s) quando uma regra é disparada.

A Interface ALE é responsável pela integração entre o mecanismo de CEP e sistemas de middleware RFID (ALE), sendo capaz de definir e executar ciclos de leitura, bem como gerenciar dispositivos lógicos de leitura contidos no middleware. A Interface ALE permite incluir, remover e modificar dispositivos lógicos.

A Interface EPCIS é responsável pela comunicação com os serviços de consulta e captura de eventos do repositório EPCIS. O serviço de consulta possibilita que CEP utilize os registros contidos no repositório para detecção de eventos complexos (através de uma linguagem específica de consulta de eventos similar ao SQL e especificada pelo *EPC Network*), bem como realizar pesquisas de informações de interesse de uma aplicação.

CEP pode conter uma ou mais interfaces ALE e EPCIS ambas sendo gerenciadas pela **interface de administração**. A interface de administração é responsável pela gerência dos processos do mecanismo de CEP (regras e processos de negócio), atividade que é realizada através de **perfis organizacionais** que definem a lógica do negócio descrita através de regras. Cada perfil contém um conjunto de interfaces (EPCIS e ALE) e regras pertencentes a um ou mais processos de negócio.

Com a utilização de perfis, torna-se possível definir a decomposição de uma regra de negócio entre diversas organizações e assim identificar eventos complexos não somente em uma organização, mas entre todas as organizações envolvidas no negócio. A criação e decomposição de regras é feita de forma manual diretamente no perfil dos nodos CEPs envolvidos. Não existe um módulo global para gerenciamento de regras que abstraia a criação, implantação e integração de regras nos sistemas envolvidos.

O CEP possui um conjunto de **serviços para notificação de eventos** que são divididos em **Serviço de Aplicação** e **Serviços de Rede CEP**. Estes serviços existem para notificar eventos às entidades interessadas (aplicação ou outros CEPs), pesquisar informações de eventos (eventos EPCIS), entre outras funcionalidades.

5.2.2 DROOLS COMO MOTOR DE REGRAS

O sistema *Drools* [DRO11] foi escolhido como motor de regras do mecanismo de CEP proposto neste trabalho. *Drools* suporta processamento de eventos complexos a partir de múltiplos fluxos de eventos para detecção, correlação e abstração de dados.

Drools apresenta diversas características que o tornam uma ferramenta importante para este trabalho. Primeira, ele suporta múltiplos fluxos assíncronos de dados por onde eventos são recebidos oriundos de diversas fontes de informação. Segunda, como raciocínio temporal é uma característica essencial da tecnologia CEP, *Drools* oferece um conjunto de operadores temporais que permite modelagem e raciocínio sobre relacionamentos temporais entre eventos. Terceira, além de raciocínio temporal, *Drools* suporta também raciocínio sobre ausência de eventos (ex. Eventos de negação).

Drools está baseado em um mecanismo de inferência que usa uma implementação do algoritmo *Rete*. Este algoritmo é usado em sistemas especialistas e é baseado em raciocínios lógicos combinando conjuntos de fatos com conjuntos de regras de inferência. Por padrão, RETE não suporta operadores temporais os quais são supridos pelos operadores e construtores de eventos do *Drools*.

Para suportar restrições temporais em regras de negócio, eventos podem ser modelados como fatos baseados em seus tempos de ocorrência (*timestamps*). Estes fatos são inseridos na memória de trabalho do mecanismo de regras em tempo de execução. Além disso, eventos precisam ser descartados quando não forem mais de interesse do mecanismo de execução ou quando não contribuírem mais para detecção de eventos complexos.

5.2.3 EVENTOS E CONSTRUTORES DE EVENTOS

As subseções a seguir formalizam a definição de alguns conceitos, como eventos, construtores de eventos e regras CEP. A relação entre estes conceitos é ilustrada na figura 5.4.

Um **evento** pode ser definido como o registro de atividade de um sistema dentro do contexto computacional de processamento de dados, ou a ocorrência de algum acontecimento no decorrer do tempo [WAN09]. Em geral, eventos podem ser categorizados em **eventos básicos** e **eventos complexos** (ou **eventos compostos**). Neste trabalho, são usadas letras maiúsculas e minúsculas (como “*E*” e “*e*”) para representar, respectivamente, tipos de eventos (E) e instâncias de eventos (e).

Definição 1 (Evento Básico): Um evento básico pode ser representado como $E = E(id, a, t)$, onde *id* é o identificador único do evento, $a = \{a_1, a_2, \dots, a_n\}$ $n > 0$ é o conjunto de atributos do evento e *t* é o tempo de ocorrência do evento. Um evento básico é atômico, indivisível e ocorre em um determinado tempo.

Definição 2 (Evento RFID): Um evento RFID pode ser denotado como $E = e(id, r, t)$, onde *id* é o identificador único da etiqueta RFID, *r* é o identificador da leitora RFID que captura a etiqueta e *t* representa o instante no tempo que o evento de leitura ocorreu (*timestamp*). Embora o tempo da atividade de leitura da etiqueta possa ser inferior ao tempo da captura do evento pelo sistema, neste trabalho é assumido que esta diferença é muito pequena para ser reconhecida. Um evento RFID é considerado um evento básico.

Definição 3 (Evento Complexo): Um evento complexo pode ser definido como $E = E(id, a, c, t_i, t_f)$ $t_f \geq t_i$, onde id é o identificador único do evento, $a = \{a_1, a_2, \dots, a_n\}$ $n > 0$ é o conjunto de atributos do evento, $c = \{e_1, e_2, \dots, e_n\}$ $n > 0$ é o vetor que contém os eventos básicos e/ou eventos complexos que causam a ocorrência do evento E , e t_i e t_f são os tempos iniciais e finais de ocorrência do evento complexo. Um evento complexo pode ocorrer durante um período de tempo (ex. pode ocorrer entre os tempos t_i e t_f).

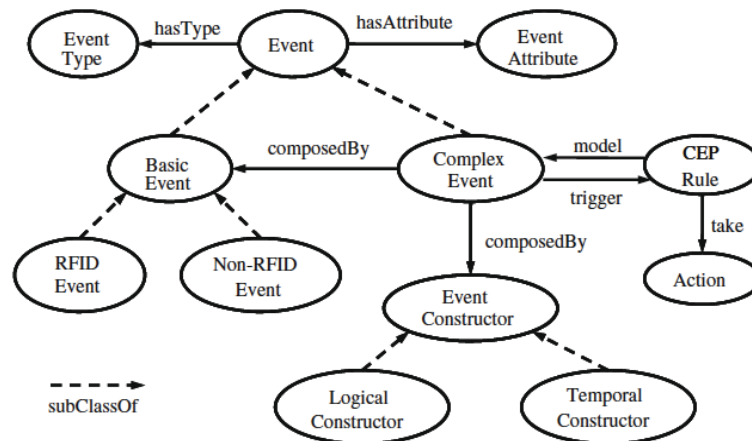


Figura 5.4 – Conceitos relacionados com CEP.

Um evento complexo é uma agregação de eventos básicos e/ou complexos que usa um conjunto específico de construtores de eventos, tais como *disjunção*, *conjunção* e *sequência* para a definição do relacionamento entre eventos. Esta agregação faz referência a um conjunto de eventos para indicar a situação de negócio descrita no cenário de aplicação. Eventos complexos têm maior significado semântico e são mais úteis e aplicáveis nas atividades de tomada de decisão em aplicações de negócio.

Construtores ou operadores de eventos são usados para expressar relacionamentos entre eventos (e seus eventos correlacionados) na formação de eventos complexos. A Tabela 5.1 apresenta uma lista de operadores lógicos e temporais segundo as funcionalidades do *Drools* [DRO11]. Estes construtores de eventos podem ser usados para definir padrões de eventos que capturam informações significativas de fluxos de dados de tempo real.

Tabela 5.1 – Construtores de Eventos do Drools.

Tipo	Operador	Expressão	Significado
Evento Lógico	and	E1 and E2	Conjunção entre E1 e E2
	or	E1 or E2	Disjunção entre E1 e E2
	Not	Not E1	Negação de E1
	Forall	forall(E1)	Cada ocorrência de E1
	Eval	eval(expression)	Avalia uma expressão booleana
Evento Temporal	window ()	window(E1, t) window(E1, n)	E1 ocorre por um período de tempo t E1 ocorre n vezes (n>0)
	coincides	E1 coincides(t1, t2) E2	E1 ocorre ao mesmo tempo que E2
	before	E1 before(t1, t2) E2	E1 ocorre antes de E2
	after	E1 after(t1, t2) E2	E1 ocorre após E2
	meets	E1 meets(t1) E2	E1 termina quando E2 começa
	overlaps	E1 overlaps(t1,) E2	E1 termina após E2 começar e E2 termina após E1 terminar
	includes	E1 includes(t1, t2) E2	E2 ocorre durante E1
	during	E1 during(t1, t2) E2	E1 ocorre durante E2
	starts	E1 starts(t1) E2	E1 começa junto com E2
	finishes	E1 finishes(t1) E2	E1 termina junto com E2

5.2.4 REGRAS CEP

Regras estabelecem condições a serem avaliadas pelas atividades de processamento de eventos para que ações apropriadas sejam adotadas. O mecanismo de CEP proposto neste trabalho baseia a definição de regras na formalização de eventos e construtores de eventos descritos nas seções anteriores.

Uma regra é uma inferência lógica pré-definida ou um padrão para detecção de eventos complexos. Existem estudos que descrevem diferentes sintaxes para regras CEP. No entanto, neste trabalho é utilizada uma linguagem de expressões de regras baseada no padrão *ECA* (*Event-Condition-Action*) a qual é uma linguagem declarativa que descreve modelos de eventos.

A representação genérica de uma regra *ECA* pode ser expressada como:

```
Rule rule_id, rule_name
  ON event
  IF condition
  THEN action 1, action 2, ... action n
END
```

Os elementos *rule_id* e *rule_name* são únicos para cada regra e representam o identificador e o nome da regra. O elemento *event* especifica o evento de interesse da regra. O elemento *condition* é uma combinação booleana de funções definidas pelo usuário. Os elementos *action* definem os procedimentos definidos pelos usuários, como por exemplo, o disparar de um alarme ou a atualização do status de um evento ocorrido. Através das regras CEP é possível ter um suporte eficiente para processamento de eventos RFID e outros tipos de dados de eventos.

5.2.5 MODELANDO EVENTOS RFID

Neste trabalho regras de negócio são criadas através da linguagem declarativa de eventos provida pelo *Drools*. Um exemplo de regra de negócio pode ser a criação de um evento EPCIS (a contextualização de um evento RFID em evento EPCIS) baseado no recebimento de relatórios provindos de um sistema de middleware RFID contendo dados de eventos de leitura ocorridos (*ALE ECREports*).

O primeiro exemplo de regra de negócio a ser apresentado (*rule "Create EPCIS event from ECREports"*) cria um evento EPCIS que é usado pela segunda regra (*rule "Cast to transaction EPCIS event"*) para contextualização de eventos RFID. Esta regra pode ser representada como:

```
rule "Create EPCIS event from ECREports" dialect "java" //Define o nome da regra;
when //Condição da regra;
  $reports : ECREports($specName : specName) //Referencia $reports ao nome da especificação;
  $sepcs : LinkedList(size > 0) from collect (EPC() from getReadInformation($reports)) //Cria uma lista
  com os EPCs existentes no relatório;
  $readPoint : ReadPoint(name == $specName) //Define o nome do ponto físico de leitura;
then //Ação da regra;
  EPCISEventType event = createEpcisEvent($readPoint, $sepcs); //Cria evento EPCIS a partir da lista
  de eventos criada na condição da regra;
  insert(event); //Armazena eventos;
  retract($reports); //Descarta relatório criado;
end
```

A segunda regra (*rule "Cast to transaction EPCIS event"*) recebe um evento do middleware RFID e o contextualiza em um evento EPCIS do tipo transação (Seção 2.1.6 – Eventos EPCIS). Uma validação é realizada antes da criação do evento. Este evento de transação tem uma nota fiscal que é usada para verificar a existência de objetos relacionados com o passo de negócio do evento corrente. Esta verificação é representada pela terceira regra de negócio (*rule "Check items against invoice"*).

```
rule "Cast to transaction EPCIS event" dialect "java" //Define o nome da regra;
when //Condição da regra;
    $event : EPCISEventType() //Define o tipo de evento EPCIS que será avaliado nessa regra;
    eval($event instanceof TransactionEventType) //Verifica se o evento é um evento do tipo transação;
then //Ação da regra;
    TransactionEventType transactionEvent=(TransactionEventType) $event; //Cria evento de transação;
    insert(transactionEvent); //Armazena evento criado;
    retract($event); //Descarta relatório criado;
end
```

```
rule "Check items against invoice" dialect "java" //Define o nome da regra;
when //Condição da regra;
    $event : TransactionEventType($parentID : parentID) //Define o tipo de transação;
    $invoice : EletronicInvoice(id == $parentID not(eval(check($event, $invoice))) //Verifica
    inconsistência entre nota e itens do evento;
then //Ação da regra;
    TransactionEventypepcisEvent=createTransactionFailed($event, $invoice); //Criar evento de falha
    de transação;
    captureEvent("Retailer", epcisEvent); //Armazena evento inconsistente;
    insert(epcisEvent); //Cria evento EPCIS;
    retract($event); //Descarta evento;
    retract($invoice); //Descarta nota fiscal;
end
```

A quarta regra (*rule "Failed delivery"*) é um exemplo de uso de operadores temporais. Esta regra estabelece uma falha de transação em um cenário de negócio onde uma ação de evento é criada se determinado evento de transação não ocorre após certo período de tempo (ex. após 1 hora e 15 minutos).

```
rule "Failed delivery" dialect "java" //Define o nome da regra;
when //Condição da regra;
    $invoice : EletronicInvoice(id == $parentID) //Define o ID da nota fiscal;
    not($event : TransactionEventType($parentID : parentID, this after[0s,1h30m] $invoice)) //Verifica
    se transação de evento determinado não acontece;
then //Ação da regra;
    TransactionEventType epcisEvent=createTransactionFailed($event, $invoice); //Cria falha de
    transação;
end
```

6. EXPERIMENTOS REALIZADOS

Este capítulo apresenta os experimentos realizados com os artefatos de software do framework RFID proposto. Duas abordagens de validação foram empregadas. A primeira **concentra-se em aplicar alguns dos artefatos criados em aplicações de casos reais**. Enquanto a segunda **concentra-se em buscar limitações do framework através de um plano de testes de desempenho do mecanismo de CEP criado**. Optou-se por estas duas abordagens de validação para seguir a ordem das atividades de desenvolvimento do sistema (conforme descrito no capítulo 5), ou seja, primeiro se testa a aplicabilidade do middleware RFID e depois a capacidade de detecção de eventos do mecanismo de CEP.

Com os resultados conseguidos é possível demonstrar que as ideias e abordagens usadas melhoram atividades de projeto e desenvolvimento de aplicações RFID quando comparados com arquiteturas de referência. Além disso, o framework agrega novas funcionalidades para integração, gerenciamento e compartilhamento de informações de eventos RFID.

6.1 APLICAÇÃO DA PLATAFORMA DE REFERÊNCIA EM CASOS REAIS

Através da plataforma de referência (Seção 5.1) foi possível observar na prática alguns dos problemas e limitações dos sistemas de software RFID identificados por outros autores e que foram documentados neste trabalho por meio do estado da arte (Capítulos 3 e 4). Com os testes da plataforma de referência em aplicações de casos reais, identificou-se novos requisitos e funcionalidades que contribuíram para a definição da abordagem de trabalho (Capítulo 4).

Os experimentos realizados são descritos através de aplicações desenvolvidas utilizando os módulos de software da plataforma de referência. O Framework RFID como um todo ainda não foi testado em aplicações de casos reais, somente em ambiente simulado (Seção 6.2).

A subseção 6.1.1 apresenta um exemplo de aplicação desenvolvida para um cenário de negócio real cuja área de aplicação são processos de controle de abastecimento de frotas de ônibus [AMA09a] [AMA09b]. Para este estudo de caso será apresentada uma breve descrição do propósito da aplicação, como ela foi desenvolvida e os principais resultados obtidos.

6.1.1 CONTROLE DE ABASTECIMENTO DE FROTAS DE ÔNIBUS

O **objetivo** desta aplicação é fornecer uma infraestrutura baseada em RFID que automatize o processo atual de fornecimento de combustível de uma empresa visando reduzir custos e evitar a utilização indevida de combustível (recorrentes por falhas humanas) e uma gestão deficiente de processos.

As funcionalidades do sistema criado são: fornecer controle de acesso automático para as bombas de combustível e uma melhor integração de dados. Além disso, o sistema visa diminuir a quantidade de tarefas propensas a erros e que possam comprometer as operações diárias de abastecimento, diminuindo vulnerabilidades no processo operacional.

O **problema** pode ser caracterizado como "falta de confiabilidade no processo existente para abastecimento de combustível", e pode ser resumido da seguinte forma:

- As atividades do processo de abastecimento de combustível são basicamente manuais e sujeitos a falhas (por exemplo, erros humanos maliciosos ou não);
- Ausência de controle automatizado para o processo de abastecimento;
- Baixo nível de integração entre o processo de abastecimento e os sistemas legados da empresa (sistemas de back-end);
- Baixo nível de confiabilidade nos dados coletados que possam ser utilizados para tomada de decisão.

Uma **operação de abastecimento de combustível consiste** na aproximação do veículo junto à bomba de combustível onde um profissional é designado para estacionar o veículo, introduzir a quantidade de litros planejada e controlar manualmente a atividade de abastecimento. Este procedimento deve ser repetido para uma frota de aproximadamente 80 veículos, uma ou duas vezes por dia, durante trinta dias por mês.



Figura 6.1 – Visão geral do ambiente operacional do sistema.

A infraestrutura do sistema implantado na empresa é composta por três bombas de combustível, um leitor RFID, três antenas RFID, um módulo de hardware proprietário para controlar as bombas, um PC/host para hospedar várias aplicações e etiquetas RFID (uma para cada veículo utilizado) (Figura 6.1).

Com a implantação do novo sistema, algumas atividades do processo de abastecimento foram modificadas e automatizadas, como por exemplo, uma nova rotina para autorizar automaticamente o bloqueio e desbloqueio do controle de bombas de combustível pelo sistema RFID. Além disso, outras mudanças foram feitas na forma em que as operações são armazenadas e integradas com os sistemas legados existentes.

Usando a versão da aplicação com o middleware *CloudRFID* foi possível analisar o middleware realizando coleta e filtragem de dados até o envio da informação aos sistemas legados da empresa. Todos os módulos envolvidos tiveram suas funções centralizadas no middleware *CloudRFID* agregando vantagens no processo de concepção e desenvolvimento, bem como no ciclo de vida do sistema e do processo relacionado.

Em relação **aos resultados e melhorias conseguidas** para a empresa, é possível dizer que todos os requisitos do sistema e os problemas mencionados anteriormente foram minimizados/resolvidos, e os seguintes objetivos foram satisfeitos:

- As atividades do processo de abastecimento de combustível que antes da implantação do sistema eram basicamente manuais e sujeitas a falhas, agora estão quase todas automatizadas, reduzindo as chances de fracasso e corrigindo a ausência de um controle de automação;
- O problema de falta de integração entre o processo de abastecimento e os sistemas legados da empresa também foram resolvidos e esta unificação elevou o nível de confiabilidade dos dados coletados;
- Em termos de coleta de dados para apoio à tomada de decisões, as melhorias de confiabilidade permitiram a criação de um novo sistema, um sistema de bonificação, o qual é uma espécie de aplicação de incentivo para premiar os motoristas de ônibus que economizarem combustível durante suas operações mensais. Esta iniciativa ajudou a empresa a reduzir custos com combustível e criou uma nova atitude nos condutores para uma melhor utilização dos recursos disponíveis na empresa.

6.2 TESTES DO MECANISMO DE CEP

A área de aplicação utilizada neste trabalho como cenário de testes é uma loja inteligente. Implementou-se um **protótipo de aplicação RFID** utilizando os artefatos de software do framework RFID. O protótipo foi simulado em laboratório através de conjuntos sintéticos de dados de testes cujos resultados foram comparados com arquiteturas e propostas semelhantes [YAO10] [ZAN08]. Para isso, dados de eventos RFID (*ALE ECR*reports) foram gerados a partir da simulação de eventos de leitura de etiquetas RFID.

6.2.1 OBJETIVOS DOS TESTES

Os testes realizados têm dois objetivos principais. O primeiro é **identificar pontos de vulnerabilidade no desempenho do mecanismo de CEP proposto**. E o segundo é **avaliar os artefatos e mecanismos que compõem o framework aplicados a atividades de projeto e desenvolvimento de aplicações RFID**.

As avaliações foram feitas através de testes do protótipo e comparações funcionais com arquiteturas de referência, como o projeto Fosstrak [FOS11] por exemplo, o qual é um projeto de middleware *open-source* mundialmente difundido e que serve de base para muitos sistemas.

6.2.2 METODOLOGIA DOS TESTES

A metodologia de teste se resume na criação de dois cenários de testes pertencentes ao mesmo protótipo de aplicação. O primeiro cenário é estimulado por um conjunto menor de regras de negócio formado por regras mais simples e que não envolvem muitos operadores temporais e relacionais. Já o segundo cenário de teste é mais complexo, estimulado por um conjunto maior de regras e que contém regras mais complexas.

As regras criadas no segundo cenário envolvem inter-relação entre dados de eventos RFID de diferentes organizações, dados de eventos EPCIS (*Master Data* e *Event Data*) e dados de referência (notas fiscais por exemplo).

Para evitar *overheads* de comunicação nas etapas de geração de dados de teste (atrasos provocados pela rede), esta etapa foi realizada utilizando diretamente a API do framework (funções do Serviço de Captura de Eventos, por exemplo) através da submissão de arquivos contendo dados de eventos RFID simulados.

O protótipo de aplicação RFID é composto por um armazém e duas lojas (Figura 6.2). A seta 1 representa produtos enviados para armazenamento no armazém, enquanto as setas 2 e 3 representam as requisições de envio de produtos para reabastecimento de estoque das lojas 1 e 2. As setas 4 e 5 representam produtos que saem das lojas.

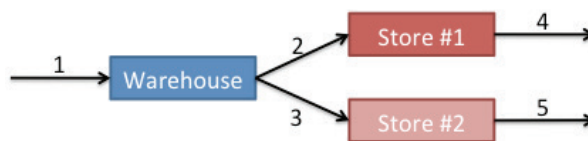


Figura 6.2 – Fluxo de dados do protótipo de teste.

O modelo de geração de dados de teste foi definido da seguinte maneira:

- A partir da chegada de produtos no armazém, estes produtos passam por diversas etapas do processo de negócio envolvido (Figura 6.3), desde o recebimento, inspeção, armazenamento e envio para lojas. Na inspeção, 10% dos produtos tem algum problema e são retirados do modelo.
- Após a chegada nas lojas (Figure 6.4) os produtos são armazenados em estoques internos e de acordo com a quantidade de produtos expostos à venda, eles saem do estoque para as prateleiras. Destes produtos, em torno de 80% vão para provedores, 15% são comprados diretamente e menos de 5% são tentativas de roubo.
- Quando um ou mais produtos entram no provedor é realizada uma consulta no sistema de informação da loja para mostrar sugestões de produtos que foram comprados em conjunto com alguns dos produtos que foram detectados no provedor. Quando uma compra é efetuada, o caixa desativa as etiquetas e o produto é removido da simulação.

Através das figuras 6.3 e 6.4 é possível visualizar processos internos do armazém e das lojas. Na figura 6.3, cada ponto de leitura (*Read Point - RP*) representa um portal (*Door Way*) por onde os produtos passam. Cada sala (*Business Location*) do armazém representa um processo de negócio. Os passos de 0 a 5 demonstram um fluxo básico de um produto no armazém, desde sua chegada, inspeção, armazenamento e envio.

A Figura 6.4 contém o esquema da estrutura das lojas. Os pontos RP:R, RP:S e RP:E representam portais de passagem. Enquanto os pontos RP:P representam prateleiras, os pontos RP:D provedores e os pontos RP:C caixas (POS – Pontos de Venda). Os passos de 5 a 9 representam um fluxo básico de um produto desde sua chegada na loja, seu armazenamento em estoque, sua disposição a venda na loja até sua compra. Quando um produto é vendido ele deixa de existir no modelo.

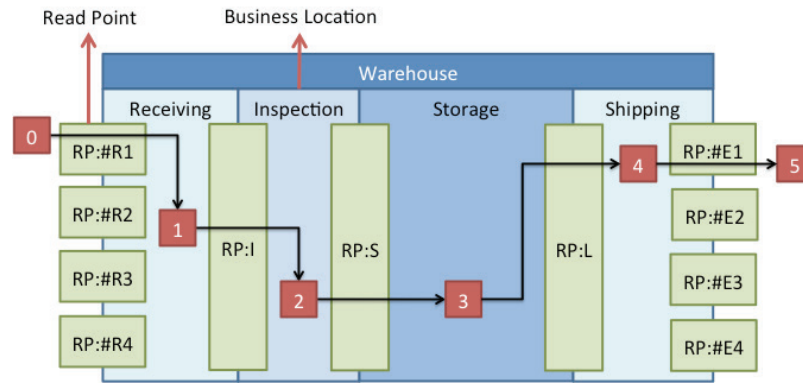


Figura 6.3 – Etapas do processo do armazém.

O primeiro cenário de teste está relacionado apenas aos processos de negócio do armazém (Figura 6.3) cujas regras envolvidas dedicam-se a ocorrência eventos de entrega de produtos mal sucedidos. Enquanto o segundo cenário está relacionado a todos os processos do protótipo criado (um armazém e duas lojas, conforme figura 6.2).

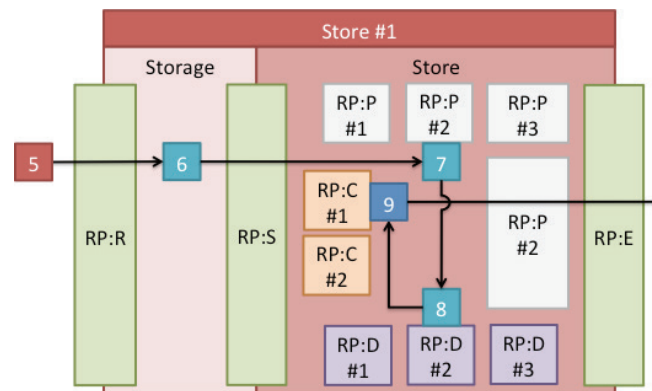


Figura 6.4 – Etapas do processo das lojas.

No segundo cenário de teste, as regras de negócio contêm expressões para detectar eventos complexos e eventos que contenham correlações com eventos passados, como por exemplo, um histórico de compras que contenha certo produto que foi detectado no provador. As expressões usadas para detecção de eventos complexos contêm pelo menos um operador lógico e um operador temporal, podendo também haver correlações com outros eventos (Detalhes dos operadores suportados podem ser consultados na subseção 5.2.3).

A Tabela 6.1 apresenta um resumo das principais regras usadas nos cenários de teste. A coluna “condição” contém os eventos necessários para disparar a ação das regras. Enquanto a coluna “ação” possui as ações ou eventos complexos relacionados.

Os eventos gerados a partir das regras criadas são armazenados no serviço de informação EPCIS e podem ser utilizados tanto para rastrear um produto quanto para serem aplicados em regras que utilizam algum tipo de histórico de eventos, como por exemplo, sugerir novos produtos para compra.

Tabela 6.1 – Resumo das principais regras do cenário de teste.		
Local	Condição	Ação
Armazém	Recebe pedido de produto X de uma loja. Quantidade de produto X está em falta.	Enviar notificação de produto 'X' em falta para a loja.
	Recebe pedido de produto X de uma loja. Quantidade de produto X não está em falta.	Cria nota fiscal de envio.
	Recebe um aviso de falha na entrega. Existe uma notificação de envio.	Reenvia produto para a loja em um tempo determinado.
	Produto recebido. Passou na inspeção.	Transação bem sucedida. Enviar para armazenamento.
	Produto Recebido. Não passou na inspeção.	Falha na transação de inspeção. Notifica setor de qualidade (aplicação externa).
Lojas	Quantidade de produto X está baixa.	Requisitar produtos ao armazém.
	Não recebe um produto em um tempo estipulado. Existe uma notificação de envio.	Criar evento de falha na transação. Envia notificação de falha ao armazém.
	Recebe um pedido. Existe uma notificação de envio.	Pedido recebido. Enviar para inspeção.
	Recebe um pedido. Não existe uma notificação de envio.	Criar um evento de falha na transação. Notificar sistema de gerência (aplicação externa).
	Produto na inspeção. Passou na inspeção.	Criar evento de transação bem sucedida.
	Produto na inspeção. Não passou na inspeção.	Criar evento de falha na transação. Realizar nova requisição.
	Produto detectado nos pontos de leitura de saída. Produto foi vendido.	Desabilitar etiqueta.
	Produto detectado nos pontos de leitura de saída. Produto não foi vendido.	Notificar sistema de alerta (aplicação externa).
	Produto entra no provador.	Sugerir produtos (aplicação externa).
	Quantidade de produtos na prateleira está baixo.	Notificar sistema de gerência (aplicação externa).
	Produto encontra-se durante um tempo 't' em local 'X' mas pertence a 'Y'.	Notificar sistema de gerência (aplicação externa).

A seguir são apresentados alguns exemplos de regras criadas conforme a Tabela 6.1. A primeira regra (*rule "Detect stolen goods"*) é utilizada para detectar produtos que estão sendo roubados. Enquanto a segunda (*rule "Disable RFID tag from sold products"*) é utilizada para desabilitar etiquetas RFID de produtos que foram vendidos e estão sendo retirados da loja. Ambas as regras utilizam operadores lógicos para verificar o status dos produtos avaliados.

```
rule "Detect stolen goods" dialect "java" //Define o nome da regra;
when //Condição da regra;
    $event : ObjectEventType($event.readPoint.id == Store.EXIT) // Define local de leitura do evento;
    not(eval(sold($event))) //Verifica se produto identificado ainda não foi vendido;
then //Ação da regra;
    notifySecurity($event); //Notifica segurança de que o produto está sendo roubado;
    retract($event); //Descarta evento;
end
```

```
rule "Disable RFID tag from sold products" dialect "java" //Define o nome da regra;
when //Condição da regra;
    $event : ObjectEventType($event.readPoint.id == Store.EXIT) //Define local de leitura do evento;
    eval(sold($event)) //Verifica se produto identificado já foi vendido;
then //Ação da regra;
    disableRFIDTag($event); //Desabilita etiqueta RFID lida;
    retract($event); //Descarta evento;
end
```

6.2.3 RESULTADO DOS TESTES

Os elementos de aplicação que compõem o protótipo testado foram implantados em três máquinas separadas onde cada módulo de sistema (um armazém e duas lojas) é composto por “Mecanismo de CEP + Middleware RFID + EPCIS”, conforme a figura 5.3. O número máximo de eventos RFID (eventos simulados de leitura de etiquetas RFID) usado como carga de teste foi 50.000 eventos RFID.

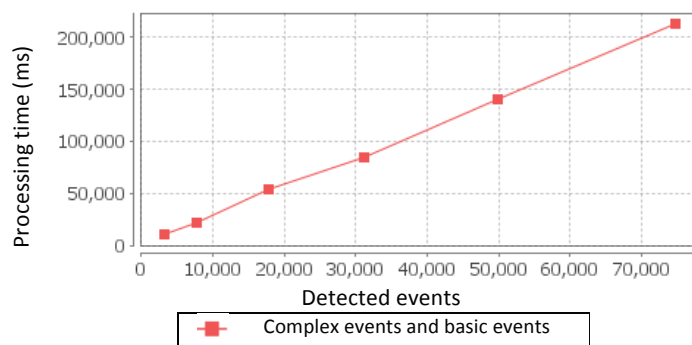


Figura 6.5 – Cenário 1: Tempos de processamento no armazém.

As figuras 6.5 e 6.6 correspondem aos testes feitos com o primeiro cenário de teste (aplicação armazém). Quanto ao tempo de processamento para detecção de eventos (Figura 6.5), foi possível identificar uma média de 390 eventos processados por segundo, incluindo eventos básicos (EPCIS) e eventos complexos.

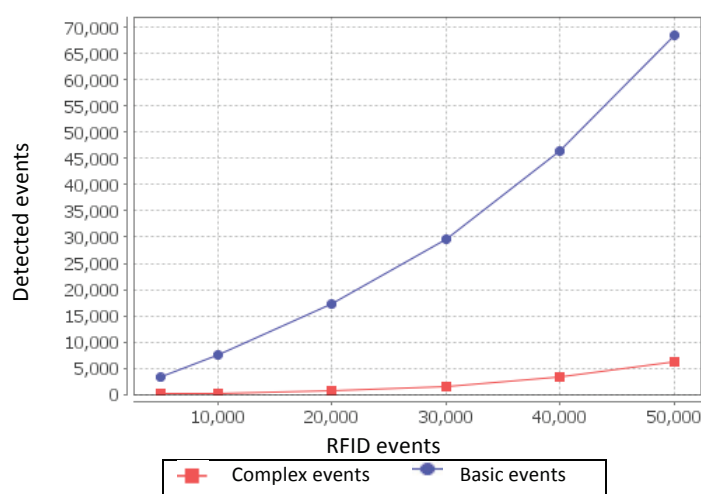


Figura 6.6 – Cenário 1: Eventos detectados.

Na capacidade de detecção de eventos (Figura 6.6), os resultados demonstram que para o primeiro cenário de teste foram detectados em média 1,03 eventos básicos (eventos EPCIS) para cada evento RFID gerado (evento de leitura de uma etiqueta RFID). Já para a capacidade de detecção de eventos complexos, a relação média é de 0,06 eventos complexos detectados para cada evento RFID gerado. Ou seja, menos de um evento complexo detectado para cada 10 eventos RFID gerados. No melhor caso, usando-se uma carga de 50.000 eventos RFID gerados (50.000 eventos de leitura de etiquetas RFID), consegue-se a detecção de 1 evento complexo para cada 10 eventos RFID gerados.

Neste primeiro cenário de teste percebe-se que a aplicação está trabalhando praticamente na contextualização de eventos RFID (evento de leitura de etiquetas RFID) em eventos EPCIS (1,03 eventos EPCIS para cada 1 evento RFID) e muito pouco na detecção de eventos complexos.

As figuras 6.7 à 6.11 correspondem aos testes feitos com o segundo cenário de teste (3 aplicações: um armazém e duas lojas).

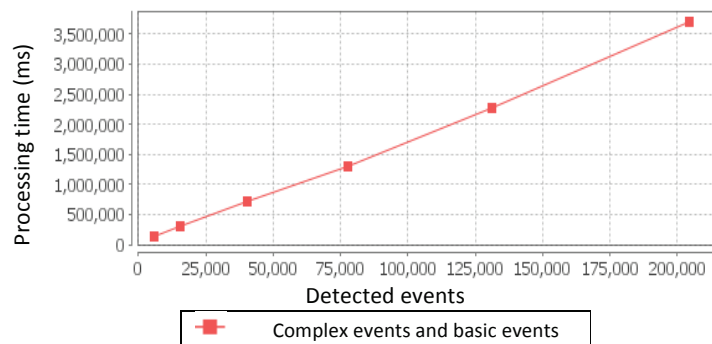


Figura 6.7 – Cenário 2: Tempos de processamento no armazém.

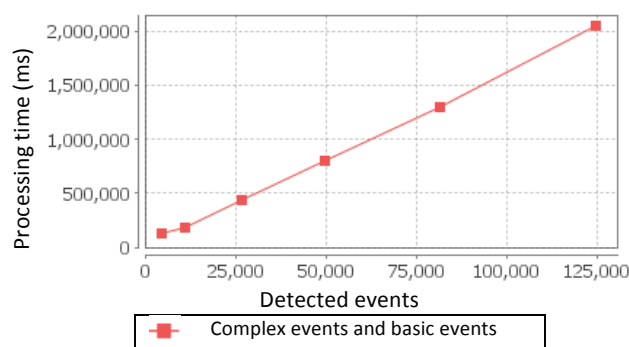


Figura 6.8 – Cenário 2: Tempos de processamento na loja 1.

O tempo médio de processamento de eventos das três aplicações (Figuras 6.7, 6.8 e 6.9) é de 57 eventos processados por segundo (incluindo eventos básicos e eventos complexos). Em relação ao tempo de processamento do primeiro cenário de teste, as aplicações do segundo cenário obtiveram um desempenho inferior o que é justificado pela maior complexidade de regras de processamento envolvidas, conforme descrito na metodologia de teste aplicada.

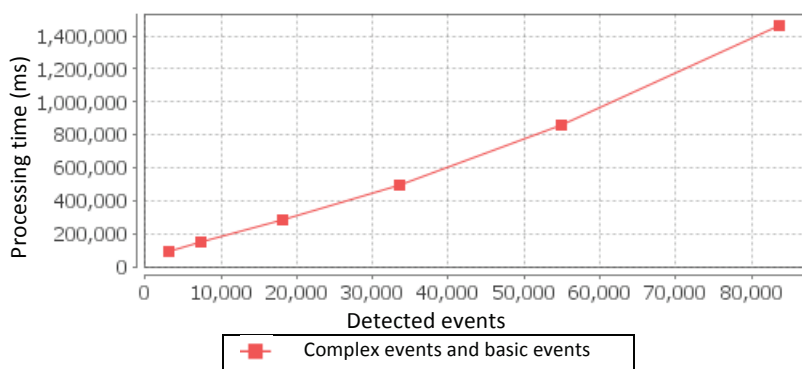


Figura 6.9 – Cenário 2: Tempos de processamento na loja 2.

Em relação a capacidade média de detecção de eventos básicos (eventos EPCIS) (Figura 6.10), os testes indicam que foram detectados 2,6 eventos básicos para cada evento RFID gerado (para cada evento de leitura de uma etiqueta RFID). Já para a capacidade média de detecção de

eventos complexos (Figura 6.11), foram detectados 0,3 eventos complexos para cada evento RFID gerado.

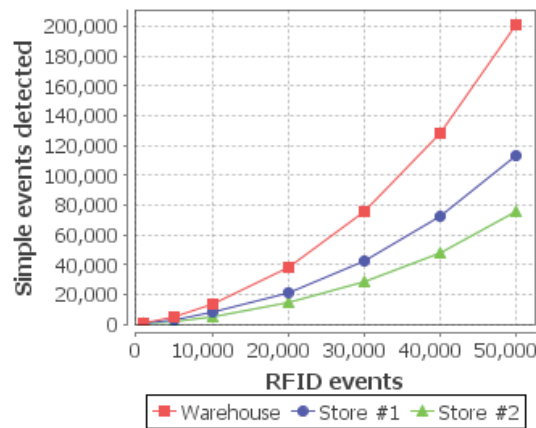


Figura 6.10 – Cenário 2: Eventos básicos detectados

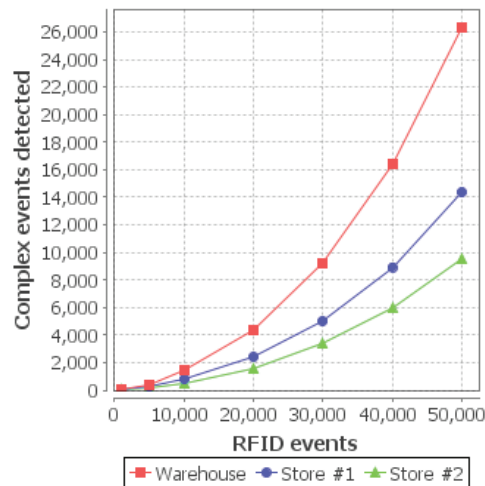


Figura 6.11 – Cenário 2: Eventos complexos detectados.

Percebe-se que as aplicações de teste do segundo cenário estão contextualizando mais eventos RFID em eventos EPCIS do que no primeiro cenário. Além disso, no segundo cenário as aplicações estão detectando um evento complexo a cada 3 eventos RFID gerados, apresentando uma capacidade de detecção de eventos complexos 70% superior em relação ao primeiro cenário.

6.3 ANÁLISE DOS RESULTADOS

6.3.1 ANÁLISE DE DESEMPENHO

Em relação ao desempenho do mecanismo de CEP, os testes revelam que quanto maior o número de dados de teste (eventos de leitura de etiquetas RFID gerados) e maior (e mais complexo) o número de regras de negócio aplicadas no mecanismo de CEP, maior é a capacidade de detecção de eventos tanto eventos básicos quanto complexos. No entanto, o tempo de processamento também se torna superior.

Este acréscimo no tempo de processamento não se deve apenas ao acréscimo do número de regras aplicadas no segundo cenário de teste, mas também aos atrasos de comunicação provocados pela rede durante o compartilhamento de eventos entre as aplicações envolvidas.

Dependendo do tipo de atraso e também das restrições temporais definidas nas regras de negócio, os atrasos de comunicação não tem impacto negativo na capacidade de detecção de eventos. No entanto, ainda são necessário novos testes que exercitem a capacidade de integração de dados em tempo real.

Como atualmente não existe um modelo oficial de avaliação de desempenho (*benchmark*) para a tecnologia CEP, os testes feitos neste trabalho foram comparados com testes de outros sistemas CEP [YAO10] [ZAN08] cujos objetivos são os mesmos, avaliar pontos de vulnerabilidade dos mecanismos de CEP quando aplicados à tecnologia RFID. Estabelecendo uma comparação com estes trabalhos, pode-se dizer que os resultados dos testes são satisfatórios pois, em média, apresentam resultados semelhantes quando mantidas as mesmas características individuais de cada plano de teste utilizado (variação da carga de testes, número de regras e complexidade das regras criadas).

O mecanismo de CEP proposto apresenta um bom desempenho se levarmos em consideração que a carga de teste foi aplicada em máquinas convencionais. Consequentemente, melhores desempenhos podem ser conseguidos se os testes forem aplicados em servidores de alto desempenho e dedicados a serviços específicos.

6.3.2 ANÁLISE DAS CARACTERÍSTICAS ARQUITETURAIS

Em relação a avaliação funcional dos artefatos e mecanismos que compõem o framework RFID, a Tabela 6.2 apresenta uma breve comparação entre o framework e a arquitetura do Fosstrak. Para esta comparação foram utilizadas onze questões relacionadas a atividades de projeto, desenvolvimento e integração de aplicações RFID.

As questões de 1 à 5 são avaliadas através da quantidade mínima de código necessária para implementar a funcionalidade, onde “automático” significa que o sistema implementa a questão sem necessidade de programação. Já as demais questões (de 6 à 11) são avaliadas em termos de complexidade funcional de utilização do recurso.

Tabela 6.2 – Comparativo de funcionalidades arquiteturais.		
Questão	Fosstrak	Framework proposto
1. Criar um objeto EPCIS	37 linhas	3 linhas
2. Capturar eventos EPCIS	4 linhas	Automático
3. Percorrer relatórios ALE	28 linhas	Automático
4. Consultar EPCIS	20 linhas	1 linha
5. Colocar em uma estrutura a hierarquia completa de um item ou pacote	Complexo	1 linha
6. Integração com implementações ALE (middleware) e EPCIS	Alta	Baixo
7. Suporte a processamento de eventos complexos	Sim	Sim
8. Complexidade ao desenvolver novas aplicações de captura	Alta	Media
9. Impacto de mudanças dos processos de negócio	Baixo	Baixo
10. Necessidade de um serviço de descoberta	Alto	Baixo
11. Gerenciamento de consultas EPCIS	Complexo	Baixo

No framework RFID, a integração do mecanismo de CEP entre implementações ALE e EPCIS é considerada de complexidade “baixa”. Esta integração ocorre por meio de arquivos de configuração que são carregados no framework através da Interface de Administração a qual permite que se especifique o tipo de integração desejada para interfaces ALE e EPCIS. No entanto

no Fosstrak, esta integração é considerada de complexidade “alta”, pois é necessária a implementação das integrações desejadas via programação da aplicação de captura.

Em relação ao suporte ao processamento de eventos complexos, tanto o framework quanto o Fosstrak oferecem um mecanismo de CEP para processamento de eventos. No entanto, o Fosstrak apenas usa o motor de regras para capturar e contextualizar eventos RFID, enquanto que no nosso framework, o mecanismo de CEP é utilizado não apenas para capturar e contextualizar eventos RFID, mas para capturar outros tipos de eventos, como eventos EPCIS que são compartilhados entre organizações por meio da Interface de Captura de Eventos e do Serviço de Notificação de Eventos.

Em relação à complexidade do desenvolvimento de novas aplicações de captura de eventos, no Fosstrak essa atividade tem uma complexidade “alta” devido a falta de ferramentas (classes e métodos específicos) que auxiliem na construção de novas aplicações de captura. Isso ocorre mesmo Fosstrak permitindo integração com motor de regras CEP, pois como atividades de captura e contextualização de eventos RFID no Fosstrak estão baseadas em regras de negócio, a criação destas regras necessita de um alto conhecimento dos eventos e objetos disponíveis no ambiente de programação do sistema. Já o framework exige uma complexidade “média” devido à necessidade de um alto conhecimento técnico da linguagem de definição de regras usada pelo mecanismo de CEP, uma vez que atividades de captura e contextualização de eventos RFID ocorrem por meio de regras de negócio que são criadas através da Interface de Administração do framework. Como ambos os sistemas possuem motor de regras CEP, o impacto de mudanças dos processos de negócio tem complexidade “baixa” pois se altera a base de regras ao invés de alterar diretamente o código da aplicação.

O Serviço de Descoberta é necessário quando existem múltiplas instâncias EPCIS envolvidas no negócio. Já o Gerenciador de Consultas é necessário para aprimorar as consultas por eventos EPCIS. No framework, o mecanismo de CEP contém um sistema gerenciador de consultas (complexidade “baixa”). Este mecanismo é responsável pela integração com múltiplos *EPCISs* e também realiza a otimização das consultas nos repositórios de histórico de eventos, diminuindo assim o overhead de comunicação ao realizar consultas. Já o Fosstrak não possui serviço de descoberta, sendo necessária a implementação do serviço diretamente no código da aplicação (complexidade “alta”).

6.3.3 COMPARAÇÃO COM TRABALHOS EXISTENTES

Quando comparado com os principais sistemas RFID existentes e que foram analisados no Capítulo 3 (Tabelas 3.1 e 3.2), o framework RFID apresenta algumas melhorias e cumpre com os principais requisitos, tanto de **gerenciamento** quanto de **integração de dados** de eventos RFID, algo que poucos trabalhos analisados no Capítulo 3 são capazes de satisfazer.

Para **tratamento de dados volumosos** e **limite de vida útil**, o framework utiliza a abordagem que separa o armazenamento de dados ativos de não ativos. Dados ativos podem estar na memória de trabalho do motor de regras e no repositório de eventos EPCIS. No entanto, os dados não ativos, aqueles que já encerraram seu ciclo de vida, são retirados do EPCIS e armazenados em outro repositório de dados, algo que melhora o desempenho do sistema evitando consultas em dados desnecessários.

Quanto ao **tratamento a dados dinâmicos e temporais**, o framework supre estes requisitos no tratamento de eventos através das funcionalidades do mecanismo de CEP, o qual está baseado na capacidade do *Drools* para suportar dados de eventos temporais e espaciais.

Quanto a **capacidade de inferência a dados implícitos**, o framework oferece o conceito de perfil organizacional. Através deste conceito, é possível definir a inter-relação entre “interesses de negócio” e “regras CEP, fontes de informação (leitores lógicos e camada de eventos (middleware RFID)) e repositórios de dados de referência (dados estáticos, dinâmicos e histórico de eventos)”. Além disso, o framework oferece o mecanismo de regras para processamento de eventos complexos e contextualização de dados RFID.

Em relação a **escalabilidade do framework**, as ideias implementadas oferecem uma maneira simples para tentar prover tanto escalabilidade intraorganizacional quanto interorganizacional. No entanto, as ideias não objetivam o tratamento de grandes volumes de dados e arquiteturas distribuídas baseadas em balanceamento de carga. As abordagens usadas se baseiam em CEPs cooperativos que podem se integrar através de regras pré-definidas visando a formação de serviços de notificação de eventos. Através destes serviços o framework consegue suportar uma escalabilidade do negócio, permitindo uma integração entre dados de eventos de diversas organizações.

Quanto a **heterogeneidade de sistemas**, a abordagem usada no framework permite a abstração dos elementos envolvidos e o isolamento em camadas baseado nas especificações e serviços de acordo com o *EPC Network* (especificações de middleware e EPCIS). Além disso, o framework oferece diversos serviços para compartilhamento e integração de dados, como CEPs cooperantes, serviço de notificação de eventos e serviço de captura de eventos, todos provendo interfaces de integração baseadas em web services.

A **capacidade de administração do framework** também está baseada no perfil organizacional. Com a parametrização do perfil, é possível configurar tanto aspectos lógicos do sistema (regras CEP, interfaces para captura e notificação de eventos e a relação do CEP com EPCIS e middleware envolvidos), quanto aspectos físicos relacionados a abstração da infraestrutura de rede (leitores lógicos, etc.). Além disso, o framework permite que a administração das regras de negócio seja realizada de forma independente da aplicação, através do mecanismo de gerenciamento de regras do motor de regras CEP.

A **flexibilidade** refere-se ao **grau de adaptabilidade do sistema para diferentes cenários de negócio**. Este requisito também é suprido no framework através do perfil organizacional. Falta um mecanismo de dependabilidade para prover respostas rápidas a situações anormais.

Por fim, **openness** é um requisito importante para a **heterogeneidade do sistema** e está relacionado com a utilização de padrões de projeto de interfaces e meios de comunicação e integração de sistemas (*XML, Web Services, Java, JPA* e outros). Os serviços propostos no nosso framework usufruem destes requisitos.

7. CONCLUSÃO E TRABALHOS FUTUROS

Este Capítulo apresenta as contribuições da Tese. Inicialmente são apresentadas as contribuições do trabalho desenvolvido e as publicações obtidas. Por fim, são apresentadas as conclusões e os trabalhos futuros.

7.1 CONTRIBUIÇÕES DO TRABALHO

Dentre as contribuições do trabalho apresentado, é possível concentrar esforços em **quatro grandes contribuições**, as quais são apresentadas a seguir.

A criação de uma extensa **revisão do estado da arte** das tecnologias de sistemas de software RFID (Capítulos 3 e 4), mais especificamente, trabalhos focados no gerenciamento e integração de dados de eventos RFID. O objetivo deste esforço é apresentar o estado da arte dos conceitos e abordagens relacionadas com a área de estudo desta tese e com isso estabelecer um embasamento técnico e científico que ajude a organizar não apenas os problemas da área e suas reais soluções, mas também conseguir organizar e expor ideias que sirvam de fomento para trabalhos futuros e que venham a mitigar desafios ainda existentes.

A criação de uma **plataforma de referência** (Seção 5.1) a qual é importante porque serve como base funcional para o framework RFID proposto. Através desta plataforma foi possível dar continuidade aos trabalhos em andamento do autor e estendê-la para prover novas funcionalidades que agreguem valor ao que está sendo proposto (Seção 1.2). A plataforma de referência é um trabalho já publicado [AMA09a] [AMA09b] [AMA10a] que oferece em sua arquitetura um framework modular baseado em ferramentas que facilitam o processo de desenvolvimento de aplicações RFID.

A criação de um **framework de software RFID** cujo projeto aprimora as ideias utilizadas na plataforma de referência (Seções 1.4, 4.1 e 4.2) e que busca melhorias para mecanismos de gerenciamento de dados RFID elevando a capacidade de integração de sistemas de middleware RFID com aplicações de negócio de diferentes organizações.

E a aprovação de **seis artigos científicos** conforme listado na seção 7.2. Ao todo foram seis trabalhos publicados (um periódico internacional, três conferências internacionais, um workshop internacional e um workshop nacional).

7.2 PUBLICAÇÕES

As publicações científicas abaixo contribuíram para definição e desenvolvimento dos trabalhos desenvolvidos nesta tese.

- **14th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2009): “Using CloudRFID middleware for fuel supply control of vehicles fleets”.**
- **IV Simpósio Internacional de Soluções de Negócios em RFID – HP Brasil (2009): “An RFID middleware applied in a fuel supply control process”.**

- **33rd Annual IEEE Software Engineering Workshop (SEW 2009): “An Adaptive Framework Architecture for RFID Applications”.**
- **Elsevier Journal of Network and Computer Applications (JNCA 2010) - Special Issue on RFID Technology, Systems and Applications: “eCloudRFID - A Mobile Software Framework Architecture for Pervasive RFID-Based Applications”.**
- **15th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2010): “Automating the enterprise business processes in a supply chain environment through RFID services based on CEP technology”.**
- **5th Annual IEEE International Conference on RFID (IEEE RFID 2011): “Cooperative CEP-based RFID framework: a notification approach for sharing complex business events among organizations”**

7.3 CONCLUSÕES

Muitos pesquisadores estão engajados em propor e desenvolver soluções para desafios existentes nas tecnologias de software RFID. Embora existam propostas recentes de trabalhos com objetivos semelhantes, muitas destas propostas não cumprem com os atuais requisitos da tecnologia RFID (Capítulo 3) e carecem por trabalhos que analisem a eficiência dos padrões já pré-estabelecidos e que proponham novas abordagens às especificações existentes.

Visando contribuir com a comunidade científica e dar continuidade aos trabalhos realizados pelo autor com arquiteturas de middleware RFID, esta tese de doutorado apresenta um framework de software RFID [AMA10b] [AMA11] cujo objetivo é elevar a capacidade de integração interorganizacional de um middleware RFID através de mecanismos de CEP que cooperam entre si por meio de serviços de notificação de eventos.

A tecnologia CEP é integrada a um middleware RFID já existente (*CloudRFID*) [AMA09a] [AMA09b] [AMA10a] e que implementa as especificações do ALE segundo a *EPCglobal* (Capítulo 5). O framework baseia-se no conceito de CEPs cooperantes para processamento e notificação de eventos complexos distribuídos. Cada nodo de processamento envolvido tem seu próprio mecanismo de CEP e a cooperação entre nodos provê um sistema de regras de negócio global que permite a decomposição e execução de regras de negócio em diferentes organizações.

O trabalho realizado foi dividido em três etapas principais: (1) criação da plataforma de referência (sistema de middleware RFID e serviço de informação EPCIS), (2) criação do módulo de CEP para processamento de eventos RFID complexos e (3) extensão do módulo de CEP para criação do serviço de notificação de eventos RFID.

Na etapa 1, as atividades se concentraram na definição e criação da plataforma de referência. Através da plataforma de referência foi possível observar na prática alguns dos problemas e limitações dos sistemas de software RFID identificados por outros autores e que foram documentados neste trabalho por meio do estado da arte (Capítulos 3 e 4). Com os testes da plataforma de referência em aplicações de casos reais (Capítulo 6), identificou-se novas funcionalidades que originaram as etapas 2 e 3 deste trabalho.

Na etapa 2 as atividades foram concentradas na criação de soluções e melhorias relacionadas ao **gerenciamento de dados de eventos RFID**. Nesta etapa foram agregadas novas funcionalidades à plataforma de referência e criou-se um mecanismo de CEP que está baseado em alguns serviços principais, como: **perfil organizacional**, **serviço único de captura de eventos e tratamento de eventos (Seção 5.2)**.

O conceito de **perfil organizacional** permite definir tanto o contexto de negócio da aplicação quanto o contexto da infraestrutura lógica das fontes de informação usadas no processamento de eventos. No contexto de negócio é possível criar regras que representem a lógica do negócio envolvida no processo de contextualização de eventos RFID. O perfil concentra o armazenamento destas regras e facilita o processo de manutenção da lógica do negócio da aplicação. No contexto da infraestrutura lógica, o perfil permite configurar e definir a inter-relação entre fontes de informação (Middleware e nodos CEP) e repositórios de dados (EPCIS).

No serviço de captura de eventos o mecanismo de CEP realiza o papel da aplicação de captura de eventos evitando que a aplicação tenha que ser reprojeta para prover essa funcionalidade para cada novo cenário de negócio a ser implantado. Para isso, criou-se o **serviço único para captura de eventos** o qual recebe eventos de diversas fontes de informação (N sistemas de middleware e N nodos CEP).

Já o módulo para **tratamento de eventos** está integrado ao **motor de regras** (sistema para gerenciamento de regras de negócio baseado no *Drools*) e juntos realizam o processamento de eventos RFID complexos distribuídos. Através do tratamento de eventos, eventos (RFID ou eventos EPCIS) são capturados pelo serviço de captura e processados pelo motor de regras. O resultado deste processamento pode ser a contextualização de um evento RFID em evento de negócio, a detecção de um evento complexo, o disparar de uma regra de negócio ou notificação de evento, ou o armazenamento de eventos na memória de trabalho ou no repositório de dados para futura consulta.

Por fim, na etapa 3 as atividades foram concentradas na criação de soluções e melhorias relacionadas a **integração e compartilhamento de dados de eventos RFID**, o que resultou na extensão do mecanismo de CEP para suportar o conceito de **nodos CEPs para notificação de eventos**. Através deste conceito é possível decompor uma regra de negócio entre CEPs de diferentes organizações facilitando a detecção de eventos RFID complexos distribuídos. O serviço de notificação de eventos permite que aplicações compartilhem apenas eventos de interesse sem se preocupar com a busca por eventos ocorridos durante processos de detecção de eventos complexos.

Este trabalho é importante, pois, além de apresentar o estado da arte e identificar limitações e problemas das tecnologias atuais de software RFID e do próprio trabalho desenvolvido, resultados experimentais conseguidos com os testes do framework proposto demonstram que as abordagens usadas na construção do framework melhoram atividades de projeto e desenvolvimento de aplicações RFID mantendo níveis aceitáveis de desempenho do mecanismo de CEP.

Quando comparado com os principais sistemas RFID existentes e que foram analisados no capítulo 3 (Tabelas 3.1 e 3.2), o framework RFID apresenta algumas melhorias e cumpre com os principais requisitos, tanto de **gerenciamento** quanto de **integração de dados** de eventos RFID, algo que poucos trabalhos analisados no capítulo 3 são capazes de cumprir. Além disso, o

framework agrega novas funcionalidades para **integração, gerenciamento e compartilhamento de dados** de eventos RFID.

7.4 TRABALHOS FUTUROS

Como trabalhos futuros, é possível identificar as seguintes possibilidades:

- Em termos de **melhorias para as abordagens científicas utilizadas**:
 - Acredita-se que existe um caminho a ser continuado nos serviços de notificação de eventos, principalmente nos serviços de cooperação entre CEPs. Talvez possam ser exploradas técnicas de comunicação em grupo para definição de um ambiente global que permita uma melhor abordagem tanto do gerenciamento da lógica do negócio quanto das atividades de notificação de eventos.
 - O projeto de uma ferramenta integrada ao framework e que permita modelagem de processos de negócio e configuração do ambiente operacional é outra possibilidade de trabalho futuro. Esta ferramenta seria responsável pela transformação de um modelo de processos em fluxos e regras utilizadas pelo *Drools* e facilitaria o gerenciamento de regras no mecanismo de CEP. Além disso, ela evitaria possíveis falhas humanas causadas pela baixa usabilidade dos recursos de modelagem oferecidos pela linguagem de definição de regras do *Drools*.
 - Uma abordagem de segurança para a integração de sistemas proposta nesta tese também é outra possibilidade de trabalho futuro. Existem muitas pesquisas e propostas de trabalhos para privacidade e segurança de sistemas RFID. No entanto, estes esforços estão concentrados em dispositivos físicos e padrões relacionados e acabam não abrangendo sistemas de software.
 - Além disso, gerenciamento de configuração e falhas de dispositivos e serviços oferecidos por sistemas de middleware também são áreas pouco trabalhadas, onde a inserção ou criação de novos mecanismos de tolerância a falhas podem elevar a confiabilidade de sistemas RFID.
- Em termos de **melhorias para os testes de validação do sistema**, acredita-se que novas abordagens de teste possam ser acrescentadas à metodologia aplicada, tais como: teste do framework em aplicações de casos reais e testes que avaliem os gargalos de rede provocados pela integração dos mecanismos envolvidos.

REFERÊNCIAS BIBLIOGRÁFICAS

- [AHM07] Ahmed, N.; Kumar, R.; French, R. S.; Ramachandaran, U. "RF²ID: A Reliable Middleware Framework for RFID Deployment". In: Proceedings of the IEEE International Parallel and Distributed Processing Symposium (IPDPS'07), 2007, pp. 1-10.
- [ALE02] Alexander, K.; Gilliam, T.; Gramling, K.; Kindy, M.; Mooginame, D.; Schultz, M.; Woods, M. "Focus on the Supply Chain: Applying Auto-ID within the Distribution Center". Technical Report, IBM-AUTOID-BC-002, Auto-ID Center, 2002.
- [ALE11] ALE Standards. "Application Level Events Standard". Capturado em: <http://www.gs1.org/gsmp/kc/epcglobal/ale>, Junho de 2011.
- [AMA09a] Amaral, L. A.; Hessel, F. P.; Bezerra, E. A.; Corrêa, J. C.; Longh, O. B.; Dias, T. F. O. "An Adaptative Framework Architecture for RFID Applications". In: Proceedings of the 33th International Software Engineering Workshop IEEE/NASA (SEW'09), 2009, pp. 15-24.
- [AMA09b] Amaral, L. A.; Hessel, F. P.; Bezerra, E. A.; Corrêa, J. C.; Longh, O. B.; Dias, T. F. O. "Using CloudRFID Middleware for Fuel Supply Control of Vehicles Fleets". In: Proceedings of the 14th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'09), 2009, pp. 1662-1665.
- [AMA10a] Amaral L. A.; Hessel, F. P.; Bezerra, E. A.; Corrêa, J. C.; Longh, O. B.; Dias, T. F. O. "eCloudRFID – A mobile Software Framework Architecture for Pervasive RFID-Based Applications". Elsevier Journal of Network and Computer Application, vol. 34-3, 2010, pp. 972-979.
- [AMA10b] Amaral, L. A.; Hessel, F. P.; Bezerra, E. A.; Corrêa, J. C. "Automating the Enterprise Business Process in a Supply Chain Environment Through RFID Services Based on CEP Technologies". In: Proceedings of the 15th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'10), 2010, pp. 1-4.
- [AMA11] Amaral, L. A.; Hessel, F. P.; Corrêa, J. C. "Cooperative CEP-based RFID Framework: a Notification Approach for Sharing Complex Business Events Among Organizations". In: Proceedings of the 5th IEEE International Conference on RFID (RFID'11), 2011, pp. 215-222.
- [ANA09] Anagnostopoulos, A. P.; Soldatos, J. K.; Michalakos, S. G. "REFiLL: A Lightweight Programmable Middleware Platform for Cost Effective RFID Application Development". Elsevier Journal of Pervasive and Mobile Computing, vol. 5-1, 2009, pp. 49-63.
- [BEA06] BEA Systems. "BEA WebLogic RFID Product Family". Capturado em: <http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products/weblogic/rfid/>, Junho de 2011.
- [BER02] Berstel, B. "Extending the RETE Algorithm for Event Management". In: Proceedings of the 9th International Symposium on Temporal Representation and Reasoning, 2002, pp. 49-51.

- [BO06] Bo, F.; Jin-Ta, L.; Ping, Z.; Jun-Bo, G.; Zhen-Hua, D. "Study of RFID Middleware for Distributed Large-scale Systems". In: Proceedings of the IEEE International Conference on Information and Communication Technology (ICTTA'06), 2006, pp. 2754-2759.
- [BOR04] Bornhövd, C.; Lin, T.; Haller, S.; Schaper, J. "Integrating Automatic Data Acquisition with Business Processes Experiences with SAP's Auto-ID Infrastructure". In: Proceedings of the 30th International Conference on Very Large Data Bases (VLDB'04), 2004, pp. 1182-1188.
- [CAR03] Carney D.; Çetintemel U.; Rasin A.; Zdonik S.; Cherniack M.; Stonebraker M. "Operator Scheduling in a Data Stream Manager". In: Proceedings of the 29th International Conference on Very Large Data Bases (VLDB'03), 2003, pp. 838–849.
- [CHA04] Chawathe, S. S.; Krishnamurthy, V.; Ramachandran, S.; Sarma, S. "Managing RFID Data". In: Proceedings of the 30th International Conference on Very Large Data Bases (VLDB'04), 2004, pp. 1189-1195.
- [CHE04] Chella, A.; Cossentino, M.; Sabatucci, L.; Seidita, V. "From PASSI to Agile PASSI: Tailoring a Design Process to Meet New Needs", In: Proceedings of the 4th International Conference on Intelligent Agent Technology (ITA'04), 2004, pp. 471-474.
- [COL06] Cole, P. H. "Fundamentals in Radio Frequency Identification". Capturado em: <http://autoidlabs.eleceng.adelaide.edu.au/Tutorial/SeattlePaper.doc>, Junho de 2011.
- [DUT07] Dutta, K.; Ramamritham, K.; Karthik, B.; Laddhad, K. "Real-Time Event Handling in an RFID Middleware System". In: Databases in Networked Information Systems (LNCC), vol. 4777, 2007, pp. 232-251.
- [DRO11] Drools. "Drools 5 - The Business Logic integration Platform". Capturado em: <http://www.jboss.org/drools>, Junho de 2011.
- [ENG02] Engels, D. W.; Sarma, S. E. "The Reader Collision Problem". In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC'02), 2002.
- [EPC04] EPC Tag Standard 1.1. "EPC Tag Data Standards Version 1.1". Capturado em: http://www.gs1.org/sites/default/files/docs/tds/tds_1_1_rev_1_27-standard-20050510.pdf, Junho de 2011.
- [EPC11] EPCglobal Standards. "GS1 Standards Knowledge Center". Capturado em: <http://www.gs1.org/gsm/kc/epcglobal/>, Junho de 2011.
- [FIN03] Finkenzeller, K. "RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification". New York: JohnWiley & Sons, 2003, 446p.
- [FLO04] Floerkemeier, C.; Lampe, M. "Issues with RFID Usage in Ubiquitous Computing Applications". In: Proceedings of the 2nd International Conference on Pervasive Computing (Pervasive'04) (LNCC), 2004, pp. 188-193.
- [FLO07] Floerkemeier, C.; Roduner, C.; Lampe, M. "RFID Application Development with the Accada Middleware Platform". In: Journal of IEEE System, vol. 1-2, 2007, pp. 82-94.
- [FOS11] Fosstrak. "Developers Guide - Capture Application", Capturado em: <http://www.fosstrak.org/capturingapp/docs/dev-guide.html>, Junho de 2011.
- [GAR05] Garfinkel, S.; Rosenberg, B. "RFID: Applications, Security, and Privacy". Addison-Wesley Professional, 2005, 608p.
- [GON06a] Gonzalez, H.; Han, J.; Li, X. "FlowCube: Constructing RFID FlowCubes for Multi-dimensional Analysis of Commodity Flows". In: Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB'06), 2006, pp. 834-845.

- [GON06b] Gonzalez, H.; Han, J.; Li, X.; Klabjan, D. "Warehousing and Analyzing Massive RFID Data Sets". In: Proceedings of the 22nd International Conference on Data Engineering (ICDE'06), 2006, pp. 83-93.
- [HU05] Hu, Y.; Sundara, S.; Chorma, T.; Srinivasan, J. "Supporting RFID-Based Item Tracking Applications in Oracle DBMS Using a Bitmap Datatype". In: Proceedings of the 31st International Conference on Very Large Data Bases (VLDB'05), 2005, pp. 1140-1151.
- [JEF06a] Jeffery, S. R.; Garofalakis, M.; Franklin, M. J. "Adaptive Cleaning for RFID Data Streams". In: Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB'06), 2006, pp. 163-174.
- [JEF06b] Jeffery, S. R.; Alonso, G.; Franklin, M. J.; Hong, W.; Widom, J. "A Pipelined Framework for Online Cleaning of Sensor Data Streams". In Proc. of the 22nd International Conference on Data Engineering (ICDE'06), 2006.
- [JUE06] Juels, A. "RFID Security and Privacy: A Research Survey". IEEE Journal on Selected Areas in Communication, vol. 24-2, 2006, pp. 381-394.
- [KAB07] Kabir A., Hong B., Ryu W., Ahn S. "LIT Middleware: Design and Implementation of RFID Middleware Based on the EPC Network Architecture", In: Dynamics in Logistics (LNCC), vol. 7, 2008, pp. 221-229.
- [KIM07] Kim, S.; Moon, M.; Kim, S., Yu, S.; Yeom, K. "RFID Business Aware Framework for Business Process in the EPC Network", In: Proceedings of the 5th International Conference on Software Engineering Research, Management & Applications, 2007, pp. 468-475.
- [KU08] Ku, T.; Zhu, Y.; Hu, K.; Nan, L. "A Novel Distributed Complex Event Processing for RFID Application". In: Proceedings of the 3th International Conference on Convergence and Hybrid Information Technology (ICCIT'08), 2008, pp. 1113-1117.
- [KUU07] Kuo, F.; Fu, C.; Liu, L.; Jin, M. "The Implementation of RFID in Emergency Medicine". In: Proceedings of the 9th International Conference on e-Health Networking, Application and Services, 2007, pp. 125-130.
- [LEO06] Leong, K. S.; Ng, M. L.; Engels, D. W. "EPC Network Architecture". Auto-ID Center, Tech. Rep. AUTOIDLABS-WP-SWNET-012, 2006.
- [LIU08] Liu, F.; Jie, Y.; Hu, W. "Distributed ALE in RFID Middleware". In: Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008, pp. 1-5.
- [MAS09] Massawe, L. V.; Aghdasi, F.; Kinyua, J. "The Development of a Multi-Agent Based Middleware for RFID Asset Management System Using the PASSI Methodology". In: Proceedings of the 6th International Conference on Information Technology: New Generations, 2009, pp. 1042-1048.
- [NGA08] Ngai, E. W. T.; Moon, K. K. L.; Riggins, F. J.; Yi, C. Y. "RFID Research: An Academic Literature Review (1995-2005) and Future Research Directions". In: Elsevier International Journal of Production Economics, Special Section on RFID: Technology, Applications, and Impact on Business Operations, vol. 112-2, 2008, pp. 510-520.
- [OHK05] Ohkubo, M.; Suzuki, K.; Kinoshita, S. "RFID Privacy Issues and Technical Challenges". In: Communications of the ACM, vol. 48-9, 2005, pp. 66-71.
- [PAL05] Palmer, M. "Seven Principles of Effective RFID Data Management". Technical Primer, Progress Software – Real Time Division, 2005.
- [PRA06] Prabhu, B. S.; Su, X.; Ramamurthy, H.; Chu, C.-C.; Gadh, R. "WinRFID - A Middleware for the Enablement of Radio Frequency Identification (RFID)-Based Applications. In: Mobile, Wireless and Sensor Networks: Technology, Applications and Future Directions, 2006, pp. 313-336.

- [QUA08] Quan, Z. S.; Xue, L.; Sherali, Z. "Enabling Next-Generation RFID Applications: Solutions and Challenges", In: IEEE Computer, vol. 41-9, 2008, pp. 21-28.
- [RAO06] Rao, J.; Doraiswamy, S.; Thakkar, H.; Colby, L. A. "Deferred Cleansing Approach for RFID Data Analytics". In: Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB'06), 2006, pp. 175-186.
- [RIE06a] Rieback, M. R.; Crispo, B.; Tanenbaum, A. S. "Is Your Cat Infected with a Computer Virus?" In: Proceedings of the 4th IEEE International Conference on Pervasive Computing and Communications (PerCom'06), 2006, pp. 169-179.
- [RIE06b] Rieback, M. R.; Crispo, B.; Tanenbaum, A. S. "The Evolution of RFID Security. IEEE Pervasive Computing", In: IEEE Pervasive Computing, vol. 7-1, 2006, pp. 62-69.
- [SAP11] SAP. "SAP Auto-ID Infrastructure". Capturado em: http://help.sap.com/content/documentation/sbs/docu_sbs_autoid.htm, Junho de 2011.
- [SAR04] Sarma, S. "Integrating RFID". In: ACM Queue Magazine, vol. 2-7, 2004, pp. 50-57.
- [SU07] Su, X.; Chu, C.; Chu, B. S.; Prabhu, R. G. "On the Utilization and Integration of RFID Data Into Enterprise Information Systems Via WinRFID". In: Proceedings of the Computers in Engineering, 2007, DOI:10.1115/DETC2007-34731.
- [SUN11] Sun Microsystems. "Sun Java System RFID Software". Capturado em: <http://java.sun.com/developer/technicalArticles/Ecommerce/rfid/sjsrfid/RFID.html>, Junho de 2011.
- [THI06] Thiesse, F.; Michahelles, F. "An Overview of EPC Technology". In: Emerald International Journal of Sensor Review, vol. 26-2, 2006, pp. 101-105.
- [UYS08] Uysal, D. D.; Emond, J. P.; Engels, D. W. "Evaluation of RFID Performance for Pharmaceutical Distribution Chain: HF vs. UHF". In: Proceedings of the IEEE International Conference on RFID, 2008, pp. 27-34.
- [WAL07] Walzer, K.; Schill, A.; Loser, A. "Temporal Constraints for Rule-Based Event Processing". In: Proceedings of the ACM first Ph.D. Workshop in the Conference on Information and Knowledge Management (CIKM'07), 2007, pp. 93-100.
- [WAN05] Wang, F.; Liu, P. "Temporal Management of RFID Data". In: Proceedings of the 31st International Conference on Very Large Data Bases (VLDB'05), 2005, pp. 1128-1139.
- [WAN06] Wang, R. "An Introduction to RFID Technology". In: IEEE Pervasive Computing, vol. 7-1, 2006, pp. 25-33.
- [WAN08] Wang, G.; Jin, G. "Research and Design of RFID Data Processing Model Based on Complex Event Processing". In: Proceedings of the International Conference on Computer Science and Software Engineering (CSSE'08), 2008, pp. 1396-1399.
- [WAN09] Wang, F.; Liu, S.; Liu, P. "Complex RFID Event Processing", In: The International Journal on Very Large Data Bases, vol. 18-4, 2009, pp. 913-931.
- [WEI93] Weiser, M. "Ubiquitous Computing", In: Journal of Computer, vol. 26-10, 1993, pp. 71-72;
- [WEL08] Welbourn, E.; Khousainova, N.; Letchner, J.; Li, Y. "Cascadia: A System for Specifying, Detecting, and Managing RFID Events". In: Proceeding of the 6th International Conference on Mobile Systems, Applications and Services (MobiSys'08), 2008, pp. 281-294.
- [WU07] Wu, J.; Wang, D.; Sheng, H. "ECA Rule-Based RFID Data Management", In: Proceedings of the 1st Annual RFID Eurasia, 2007, pp. 1-5.
- [YAO10] Yao, W.; Chu, C.; Li, Z. "Leveraging Complex Event Processing for Smart Hospitals Using RFID", In: Journal of Network and Computer Applications, 2010, doi: 10.1016/j.jnca.2010.04.020.

- [ZAN08] Zang, C.; Fan, Y.; Liu, R. "Architecture, implementation and application of complex event processing in enterprise information systems based on RFID", In: Journal of Information Systems Frontiers, 2008, doi: 10.1007/s10796-008-9109-0.
- [ZIE09] Ziekow, H.; Gunther, O. "Sharing RFID and Complex Event Data Among Organizations", In: Information Systems Frontiers, vol. 12-5, 2009, pp. 541-459.