

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**THIAGO REGAL DA SILVA**

**PROPOSTA DE ONTOLOGIA PARA INTEGRAÇÃO DE  
SISTEMAS DE MANUTENÇÃO INTELIGENTES E CADEIAS  
DE SUPRIMENTO DE PEÇAS DE REPOSIÇÃO**

Porto Alegre

2015

**THIAGO REGAL DA SILVA**

**PROPOSTA DE INTEGRAÇÃO DE SISTEMAS DE  
MANUTENÇÃO INTELIGENTES E CADEIAS DE  
SUPRIMENTO DE PEÇAS DE REPOSIÇÃO**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Área de concentração: Controle e Automação

**ORIENTADOR:** Carlos Eduardo Pereira

Porto Alegre

2015

THIAGO REGAL DA SILVA

**PROPOSTA DE ONTOLOGIA PARA INTEGRAÇÃO DE  
SISTEMAS DE MANUTENÇÃO INTELIGENTES E CADEIAS  
DE SUPRIMENTO DE PEÇAS DE REPOSIÇÃO**

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: \_\_\_\_\_

Prof. Dr. Carlos Eduardo Pereira, UFRGS

Doutor pela Universidade de Stuttgart, Alemanha

Banca Examinadora:

Prof. Dr. José Reinaldo Silva, Dep. Eng. Mecatrônica e Sistemas Mecânicos-USP  
Doutor pela Univ. de São Paulo (USP), São Paulo, Brasil

Prof. Dr. Renato Ventura Bayan Henriques, PPGEE-UFRGS  
Doutor pela Univ. Federal de Minas Gerais (UFMG), Belo Horizonte, Brasil

Prof. Dr. Altamiro Amadeu Susin, PPGEE-UFRGS  
Doutor pelo Institut National Polytechnique de Grenoble, Grenoble, França

Coordenador do PPGEE: \_\_\_\_\_

Prof. Dr. Luís Fernando Alves Pereira

Porto Alegre, abril de 2015.

## **DEDICATÓRIA**

Dedico primeiramente a Deus, aquele que criou e implementou a ontologia mais perfeita e completa – o Universo com tudo que conhecemos e desconhecemos. Também à minha esposa Camilla pelo apoio total, pela vida que compartilhamos e pelo sacrifício da distância nos meses em que estive fazendo o mestrado-sanduíche. Dedico este trabalho à minha mãe pela crença inabalável naquilo que posso fazer e pelas oportunidades que me deu e que jamais poderei agradecer suficientemente. Ao meu avô José, o primeiro engenheiro da família, ainda que sem o estudo formal. Por fim, mas não menos importante, aos amigos Thiago Michelin, Eduardo Israel e Andre Albrecht, que são a família que encontrei na Alemanha.

## **AGRADECIMENTOS**

Ao Programa de Pós-Graduação em Engenharia Elétrica, PPGEE, pela oportunidade de realização de trabalhos em minha área de pesquisa e ao meu orientador, Prof. Carlos Eduardo.

## RESUMO

A competição global e a necessidade de otimização dos sistemas de produção e logística têm levado à necessidade de sistemas de manutenção mais eficientes. A habilidade de prever a ocorrência de falhas, através de Sistemas de Manutenção Inteligentes (*Intelligent Maintenance Systems* - IMS), pode evitar paradas não programadas e constituir uma grande vantagem competitiva. Além disso, um planejamento de demanda mais preciso em Cadeias de Suprimento de Peças de Reposição (*Spare Parts Supply Chain* - SPSC) resulta na disponibilidade de peças e serviços quando eles são necessários no chão de fábrica, evitando interrupções na produção. Uma integração adequada de ambos domínios é de grande importância neste contexto. Alguns dos desafios relativos a esta integração vêm das diferenças semânticas entre as áreas com diferentes vocabulários e conceitos. Este trabalho pretende propor uma modelagem semântica dos domínios através da criação de uma ontologia, provendo um vocabulário comum e uma integração semântica apropriada entre as áreas, como base para a construção futura de sistemas de informação para integrar IMS e SPSC.

**Palavras-chave:** Sistemas de Manutenção Inteligentes, Cadeias de Suprimento de Peças de Reposição, Ontologia.

## **ABSTRACT**

Global competition and the need for logistic and production systems optimization has been leading to the need of more eficiente maintenance systems. Ability to forecast failures, through Intelligent Maintenance Systems (IMS), can avoid breakdowns and be a competitive advantage. Moreover, a more precise demand planning in Spare Parts Supply Chains (SPSC) results in the availability of parts and services when they are needed at shop floor, avoiding production interruptions. A proper integration of both domains is of utmost importance in this context. Some of the challenges related to this integration come from the semantic differences between areas with such a diversity of vocabulary and concepts. This work intends to propose a semantic modelling through the creation of an ontology, providing a common vocabulary and proper semantic integration, as basis for a future implementation of an integration information system between IMS and SPSC.

**Keywords: Intelligent Maintenance System, Spare Parts Supply Chain, Ontology.**

# SUMÁRIO

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>INTRODUÇÃO.....</b>                          | <b>12</b> |
| 1.1      | CONTEXTUALIZAÇÃO .....                          | 12        |
| 1.2      | OBJETIVOS.....                                  | 14        |
| 1.3      | ORGANIZAÇÃO DO TEXTO .....                      | 15        |
| <b>2</b> | <b>REVISÃO DE LITERATURA .....</b>              | <b>16</b> |
| 2.1      | MODELAGEM CONCEITUAL E ONTOLOGIAS .....         | 16        |
| 2.2      | DESENVOLVIMENTO DE ONTOLOGIAS .....             | 17        |
| 2.2.1    | Conceitos.....                                  | 17        |
| 2.2.2    | Linguagens .....                                | 18        |
| 2.2.3    | Princípios de construção .....                  | 19        |
| 2.3      | ONTOLOGIAS EM MANUFATURA .....                  | 20        |
| 2.4      | ONTOLOGIAS EM MANUTENÇÃO.....                   | 24        |
| 2.5      | ONTOLOGIAS EM CADEIAS DE SUPRIMENTO .....       | 27        |
| <b>3</b> | <b>PROPOSTA DE MODELAGEM CONCEITUAL.....</b>    | <b>34</b> |
| 3.1      | INTRODUÇÃO.....                                 | 34        |
| 3.2      | ESTRATÉGIA DE DESENVOLVIMENTO.....              | 34        |
| 3.3      | DESIGN MULTI-CAMADAS.....                       | 35        |
| 3.4      | CAMADA DE META ONTOLOGIA .....                  | 37        |
| 3.5      | CAMADA DE ONTOLOGIA DE DOMÍNIO .....            | 39        |
| 3.5.1    | Cadeia de Suprimentos.....                      | 40        |
| 3.6      | CBM/IMS .....                                   | 46        |
| 3.6.1    | Peça e Dispositivo .....                        | 46        |
| 3.6.2    | Degradação.....                                 | 48        |
| 3.6.3    | Intelligent Maintenance System (IMS) .....      | 48        |
| 3.6.4    | Reparo, Ferramenta, Pessoal e Habilidades ..... | 49        |
| 3.7      | CAMADA DE APLICAÇÃO/INSTANCIAMENTO .....        | 50        |
| 3.8      | INTEGRAÇÃO SEMÂNTICA ENTRE AS CAMADAS.....      | 51        |
| 3.8.1    | Integração com MASON.....                       | 51        |
| 3.8.2    | Integração entre as ontologias de domínio.....  | 54        |

|          |  |           |
|----------|--|-----------|
| 3.8.3    | Integração entre a camada de ontologia de domínio e a camada de aplicação/instanciação ..... | 55        |
| <b>4</b> | <b>IMPLEMENTAÇÃO DA ONTOLOGIA.....</b>   | <b>56</b> |
| 4.1      | ESCOLHA DA LINGUAGEM.....  | 56        |
| 4.1.1    | Classes e Relações Taxonômicas .....   | 56        |
| 4.1.2    | Propriedades .....   | 56        |
| 4.1.3    | Indivíduos.....  | 57        |
| 4.1.4    | Restrições .....   | 57        |
| 4.1.5    | Tipos de propriedades .....  | 58        |
| 4.2      | FERRAMENTA DE DESENVOLVIMENTO .....  | 60        |
| 4.2.1    | Estrutura de arquivos.....   | 61        |
| <b>5</b> | <b>ESTUDOS DE CASO .....</b>   | <b>62</b> |
| 5.1      | ESTUDO DE CASO 1: VÁLVULAS INDUSTRIAIS.....  | 62        |
| 5.1.1    | Válvulas e peças .....   | 62        |
| 5.1.2    | Degradação e conceitos associados.....   | 63        |
| 5.1.3    | Grupos de Dispositivos .....   | 64        |
| 5.2      | ESTUDO DE CASO 2: AI2MS .....  | 64        |
| 5.3      | CONSIDERAÇÕES SOBRE OS ESTUDOS DE CASO.....  | 67        |
| <b>6</b> | <b>USO DA ONTOLOGIA PARA A CRIAÇÃO DE FERRAMENTAS DE INTEGRAÇÃO .....</b>                    | <b>69</b> |
| 6.1      | DETALHAMENTO DA CAMADA DE APLICAÇÃO .....  | 69        |
| 6.2      | MODELAGEM DA ARQUITETURA BASEADA NA ONTOLOGIA.....   | 70        |
| 6.3      | CONSTRUÇÃO DO SISTEMA DE INTEGRAÇÃO .....  | 70        |
| <b>7</b> | <b>CONCLUSÕES E TRABALHOS FUTUROS .....</b>  | <b>72</b> |
|          | <b>REFERÊNCIAS .....</b>   | <b>74</b> |

## LISTA DE ILUSTRAÇÕES

|  |    |
|--|----|
| Figura 1 As áreas temáticas do projeto I2MS2C.....   | 13 |
| Figura 2 Desafios de integração entre as perspectivas técnicas e de gerenciamento de SPSC.....           | 14 |
| Figura 3 Visão geral das classes e propriedades principais do MASON. ....                                | 23 |
| Figura 4 Structural model of FMEA knowledge.....   | 26 |
| Figura 5 Conceitos de ontologia de domínio FMEA.....   | 27 |
| Figura 6 SCO Top level classes and relations.. ....  | 30 |
| Figura 7 <i>Framework</i> para enriquecimento semântico de modelos de referência. ....                   | 31 |
| Figura 8 Design multi-camadas.....   | 36 |
| Figura 9 Hierarquia parcial de MASON .....   | 38 |
| Figura 10 Conceito de peça na hierarquia de MASON .....  | 39 |
| Figura 11 Conceitos principais de SCO.. ....   | 41 |
| Figura 12 Hierarquia de <i>Supply Chain Structure</i> .....  | 42 |
| Figura 13 Hierarquia de <i>Actor</i> .....   | 43 |
| Figura 14 Hierarquia de <i>Role</i> .....  | 43 |
| Figura 15 Hierarquia de <i>Process</i> .....   | 44 |
| Figura 16 Hierarquia de Recurso.....   | 46 |
| Figura 17 Padrão de projeto <i>Composite</i> .. ....   | 47 |
| Figura 18 Visão geral da camada CBM/IMS .....  | 50 |
| Figura 19 Integração semântica MASON-SPSC.....   | 53 |
| Figura 20 Integração semântica entre MASON-IMS Fonte: o autor.....                                       | 54 |
| Figura 21 Interface do Protégé 4.3 .....   | 60 |
| Figura 22 Arquivos na composição da ontologia .....  | 61 |
| Figura 23 Representação parcial da camada de domínio (IMS) e dos agentes AI2MS do estudo de caso 2. .... | 67 |

## LISTA DE TABELAS

|   |    |
|---|----|
| Tabela 1 Análise comparativa dos trabalhos relacionados .....   | 32 |
| Tabela 2 Tabela de modelos de válvula e peças .....   | 63 |
| Tabela 3 Alguns conceitos relacionados à degradação e definidos na camada de aplicação<br>do estudo de caso 1 ..... | 63 |
| Tabela 4 Resumo de Ocorrências .....  | 80 |
| Tabela 5 Resumo por Modelo (referência real omitida).....   | 81 |

## **LISTA DE ABREVIATURAS**

**BRAGECRIM:** Brazilian-German Collaborative Research Initiative in Manufacturing Technology

**FMEA:** Failure Mode and Effects Analysis

**I2MS2C:** Integrating Intelligent Maintenance Systems and Spare Parts Supply Chains

**IMS:** Intelligent Maintenance System

**SPSC:** Spare Parts Supply Chain

**UFRGS:** Universidade Federal do Rio Grande do Sul

**UFSC:** Universidade Federal de Santa Catarina

**WWU:** Westfälische Wilhelms-Universität – University of Münster

**CBM:** Condition-Based Maintenance

# 1 INTRODUÇÃO

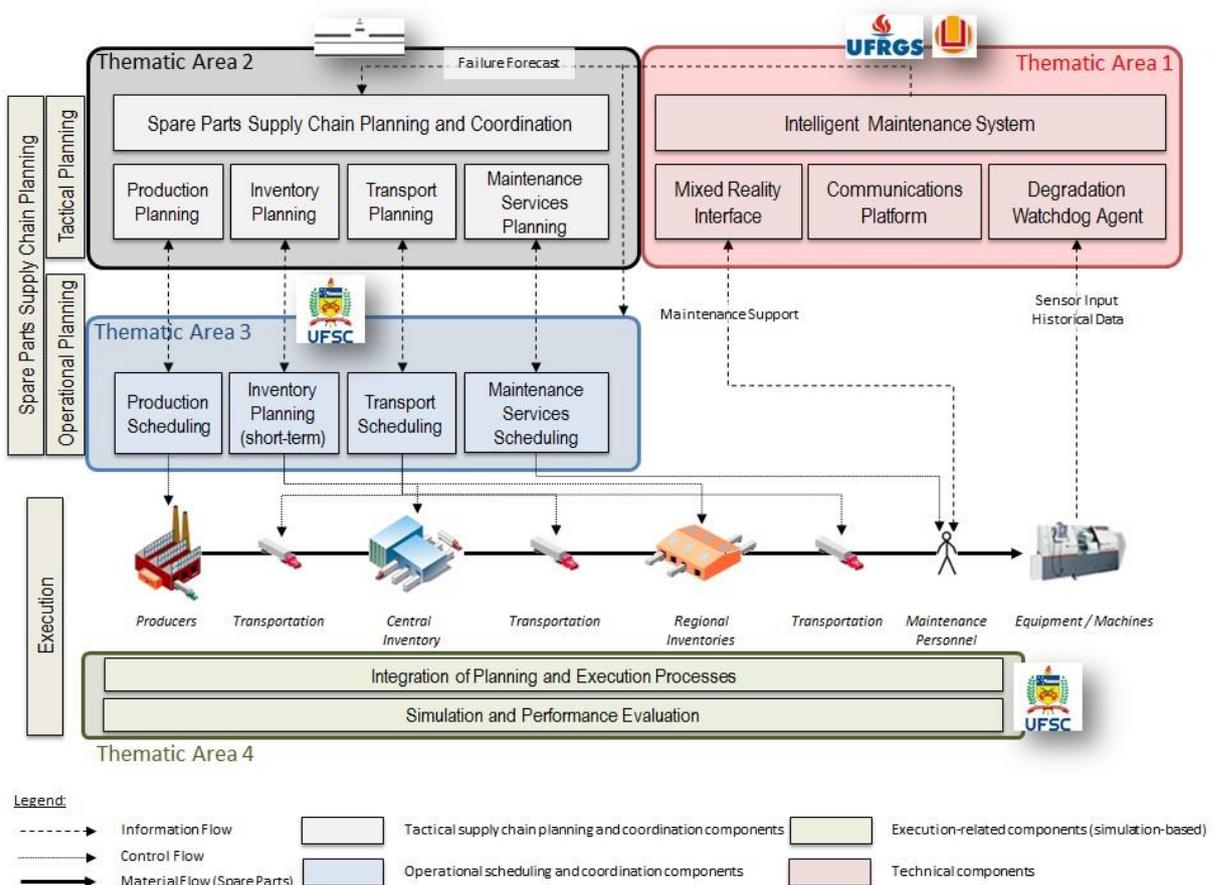
## 1.1 CONTEXTUALIZAÇÃO

A competição global tem levado a sistemas de produção mais complexos, nos quais uma manutenção satisfatória é crucial para a operação. Os efeitos de baixos níveis de manutenção podem ter enorme impacto econômico nos sistemas de produção. Um dos pontos-chave em possuir manutenção de alto nível é, por um lado, estar apto a estimar as necessidades de manutenção antecipadamente, evitando quebras e interrupções inesperadas na produção. Por outro lado, gerenciamento e planejamento nos sistemas de cadeias de suprimento de peças de reposição se tornaram mais importantes, tendo em vista a combinação da necessária disponibilidade de peças com os custos de transporte e estoques.

Para endereçar o primeiro ponto, pesquisas no domínio de Sistemas de Manutenção Inteligentes (*IMS – Intelligent Maintenance Systems*), tem despontado nos últimos anos, provendo meios de prever falhas em dispositivos através da análise de leituras de sensores, resultando na capacidade de prever necessidades de manutenção e de peças de reposição, baseado na análise da degradação do desempenho. Adicionalmente, muitas pesquisas sobre abordagens para previsão de demanda vêm sendo realizados nos últimos anos (Syntetos, Keyes e Babai, 2009), incluindo previsão de demanda através do monitoramento das condições dos dispositivos em chão de fábrica (Espindola *et al.*, 2012; Hellingrath e Cordes, 2013).

O projeto I2MS2C (acrônimo de *Integrating Intelligent Maintenance Systems and Spare Parts Supply Chains*), parte do programa BRAGECRIM (*Brazilian-German Collaborative Research Initiative in Manufacturing Technology* - <http://www.bragecrim.rwth-aachen.de>) objetiva melhorar a efetividade e eficiência de operações de gerenciamento de serviços para sistemas técnicos complexos através da combinação de esforços de pesquisa na área de IMS e gerenciamento e planejamento de Cadeia de Suprimentos de Peças de Reposição (SPSC – Spare Parts Supply Chain). O projeto é organizado em quatro áreas temáticas: Sistemas

de Manutenção Inteligentes; Planejamento e Coordenação Tática de SPSC; Planejamento Operacional de SPSC; e Integração de Processos e Simulação e Avaliação. Cada área temática se relaciona com os diversos atores envolvidos desde o chão de fábrica até os provedores da cadeia de suprimento, cada um com um fluxo próprio de informações e materiais, conforme ilustrado na Figura 1. As responsabilidades por cada área são, respectivamente, da UFRGS, WWU e UFSC para as duas últimas áreas.

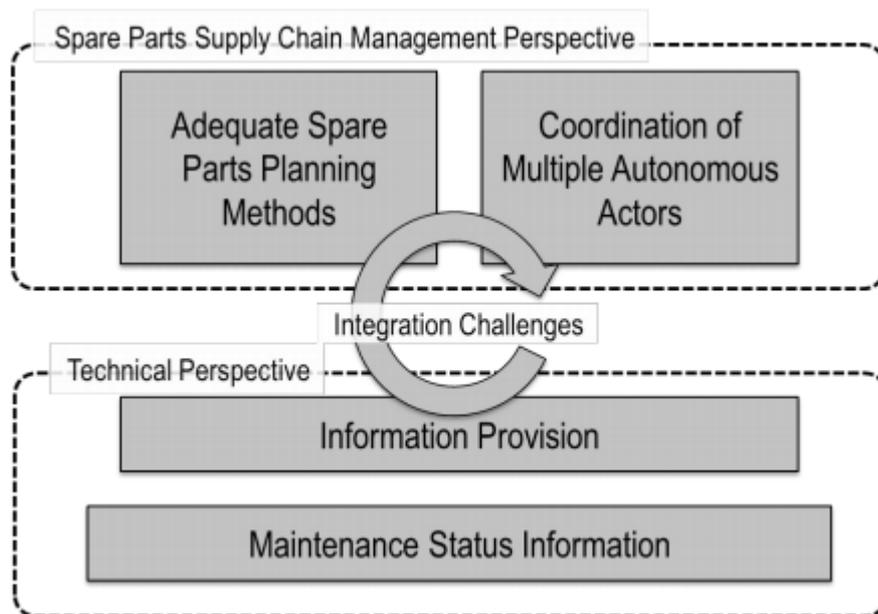


**Figura 1** As áreas temáticas do projeto I2MS2C. Fonte: (Espindola *et al.*, 2012)

Um tópico importante para abordar no contexto do projeto I2MS2C é uma apropriada integração entre as áreas, que possuem diferentes níveis de abstração e com conceitos e embasamentos tecnológicos distintos. Além disso, a integração não deve contemplar apenas a simples troca de mensagens, mas a modelagem dos conceitos envolvidos e suas relações,

proporcionando, assim, um nível avançado de integração e entendimento das diferentes partes envolvidas no projeto.

Portanto, este trabalho visa endereçar este desafio de integração, conforme mostrado na Figura 2, através de um esforço para analisar e descrever quais são os conceitos dos diferentes domínios envolvidos no projeto e como eles se relacionam uns com os outros, apresentando uma modelagem conceitual que pode servir de base para a definição de um vocabulário comum e para a derivação de sistemas tecnológicos para integração entre as diferentes áreas.



**Figura 2 Desafios de integração entre as perspectivas técnicas e de gerenciamento de SPSC. Fonte: (Espindola *et al.*, 2012).**

## 1.2 OBJETIVOS

O objetivo deste trabalho é descrever o conhecimento e as relações entre os diferentes domínios envolvidos no contexto da integração entre IMS e SPSC, através da proposição de uma modelagem conceitual. Os conceitos principais serão descritos semanticamente através do desenvolvimento de uma ontologia, como base para futuras arquiteturas de integração.

### 1.3 ORGANIZAÇÃO DO TEXTO

O texto está organizado da seguinte forma: o Capítulo 1 contextualiza o trabalho, descrevendo o problema que se quer resolver e os objetivos presentes. No Capítulo 2 é examinado o estado da arte na modelagem conceitual e no desenvolvimento de ontologias nos diferentes domínios de que trata este trabalho. O Capítulo 3 apresenta a proposta de modelagem conceitual, descrevendo os conceitos, relações e camadas envolvidas, enquanto o Capítulo 4 descreve o desenvolvimento da ontologia utilizando ferramenta e formalismos semânticos próprios. O Capítulo 5 apresenta um estudo de caso utilizado para validar os conceitos. O Capítulo 6 apresenta uma breve abordagem sobre o uso da ontologia para criação de sistemas de integração. Finalmente, o Capítulo 7 apresenta uma conclusão e proposta de trabalhos futuros.

## 2 REVISÃO DE LITERATURA

### 2.1 MODELAGEM CONCEITUAL E ONTOLOGIAS

Modelagem conceitual é a atividade de construção de uma representação de um domínio do mundo real, como um passo para o desenvolvimento de um sistema de informação. Ela provê a base para as outras atividades do processo de desenvolvimento. A construção de modelos conceituais de alta qualidade é um objetivo crítico (Weber, 2003). Existem várias maneiras de construir e representar um modelo conceitual. Uma delas é através do uso do formalismo semântico das ontologias. Várias tentativas foram feitas para estabelecer uma base teórica para a modelagem conceitual baseada em ontologias (Wand *et al.*, 1995; Weber, 2003; Wussusek, 2006; Guizzardi e Halpin, 2008). O uso de “teorias de ontologia provê a chave para o avanço do conhecimento e da prática na modelagem conceitual”<sup>1</sup> (Weber, 2003).

De acordo com (Guizzardi, 2005), o termo ontologia “foi cunhado no século XVII em paralelo pelos filósofos Rudolf Göckel, no seu *Lexicon Philosophicum*, e por Jacob Lorhard, no seu *Ogdoas Scholastica*. O termo *Ontologia*, entretanto, foi popularizado nos círculos filosóficos apenas no século XVIII, pela publicação em 1730 de *Philosophia prima sive Ontologia* por Christian Wolff”<sup>2</sup>. Ontologias, portanto, tiveram origem no campo do Filosofia, onde significam uma explicação sistemática da existência (Gómez-Pérez e Benjamins, 1999) ou uma teoria da existência (Mizoguchi, 2003). No campo dos Sistemas de Informação, uma

---

<sup>1</sup> Original em inglês.

<sup>2</sup> Original em inglês.

ontologia pode ser descrita como uma “especificação de um vocabulário representacional para um domínio compartilhado de discurso – definições de classes, relações, funções e outros objetos” (Gruber, 1993)<sup>3</sup>. Ela define um “vocabulário comum para pesquisadores que precisam compartilhar informação em um domínio”<sup>4</sup> (Noy e McGuinness, 2001). Muitas disciplinas desenvolveram ontologias para descrever o conhecimento em diferentes campos, tais como biologia (Ashburner *et al.*, 2000), medicina (Luciano *et al.*, 2011), manufatura (Borgo e Leitão, 2007) e cadeia de suprimentos (Grubic e Fan, 2010). Como indicado por (Noy e McGuinness, 2001), algumas das razões para o crescimento no uso de ontologias neste campo residem em alguns importantes benefícios obtidos pelo seu uso, tais como:

- Compartilhar um entendimento comum da estrutura da informação entre pessoas ou agentes de software;
- Permitir o reuso do conhecimento do domínio para fazer hipóteses explícitas;
- Separar o conhecimento do domínio do conhecimento da operação.

## 2.2 DESENVOLVIMENTO DE ONTOLOGIAS

### 2.2.1 Conceitos

Ontologias são classificados em muitos tipos (Gómez-Pérez e Benjamins, 1999), mas por uma questão de clareza e concisão, neste trabalho considera-se apenas dois tipos: *Upper Ontologies*, como aqueles que descrevem conceitos genéricos sobre os quais ontologias mais

---

<sup>3</sup> Original em inglês.

<sup>4</sup> Original em inglês.

especializadas são construídas e *Domain Ontologies*, que descrevem o conhecimento em um domínio específico. Uma *Upper Ontology* é geralmente aplicável a vários domínios.

Os seguintes conceitos são parte de uma ontologia (Gómez-Pérez e Benjamins, 1999):

- **Classes ou conceitos:** *representam qualquer entidade sobre a qual algo possa ser dito, de uma maneira ampla, significando funções, tarefas, ações e assim por diante;*
- **Relações:** *representam o tipo de relações que os conceitos possuem em um domínio. Por exemplo: “é subclasse de”, “faz parte de”;*
- **Funções:** *caso especial de relações na qual o enésimo elemento é único em a todos os elementos de um tipo;*
- **Instâncias ou indivíduos:** *são a realização dos conceitos, de forma similar a que um objeto é, relacionado a uma classe, no paradigma de orientação a objetos.*

### 2.2.2 Linguagens

Ao longo dos anos, muitas linguagens foram desenvolvidas para representar ontologias de uma maneira formal (Corcho e Gómez-Pérez, 2000). Entre as ditas linguagens “tradicionais”, ou seja, as primeiras a serem desenvolvidas estão Ontolingua (Farquhar, Fikes e Rice, 1997), talvez a mais representativa entre elas, OKBC (Chaudhri *et al.*, 1997), OCML (Motta, 1999), FLogic (Kifer, Lausen e Wu, 1995), LOOM (Macgregor, 1991). Também várias das ontologias que serão analisadas neste trabalho utilizam linguagens como UML (Garetti e Fumagalli, 2012), (Zhao e Zhu, 2010), ou até implementação em linguagem C++ (Fadel, Fox e Gruninger, 1994).

Entretanto, com a necessidade de cooperação entre agentes distintos e o intercâmbio de ontologias em rede, um novo conjunto de linguagens passou a ser desenvolvido, baseadas especialmente em padrões “amigáveis” à rede, como XML e RDF. Mais recentemente, o W3C (*World Wide Web Consortium*) adotou como padrão a linguagem OWL (*Web Ontology Language*) (Mcguinness e Van Harmelen, 2004; Bechhofer, 2009), baseada em XML, e com

estrutura que permite o processamento dos conceitos, relações e outros elementos da ontologia por máquina. Um dos seus propósitos é servir de linguagem formal de ontologia para a web semântica, facilitando o processamento automático e a integração da informação disponível na web. Além de ser um padrão W3C, também possui atualmente uma ferramenta de desenvolvimento (Protégé) que a suporta com alto grau de maturidade (Stanford, 2013).

### 2.2.3 Princípios de construção

Alguns dos princípios para a construção de ontologias foram compilados por (Gómez-Pérez e Benjamins, 1999):

- **Clareza e objetividade** (Gruber, 1995): *significando que os conceitos e relações descritos pela ontologia devem ser inequívocos e claramente compreensíveis, também incluindo documentação em linguagem natural;*
- **Completude** (Gruber, 1995): *a definição das entidades deve ser feita pelo uso de condições suficientes e necessárias, ao invés de apenas condições necessárias;*
- **Coerência** (Gruber, 1995): *as ontologias devem ter consistência lógica, para permitir inferências;*
- **Máxima extensibilidade monotônica** (Gruber, 1995): *trocias ou inclusões de novos termos não devem gerar alterações nos termos existentes;*
- **Mínimo comprometimento ontológico** (Gruber, 1995): *fazer poucas afirmações sobre o mundo a ser modelado, para manter a liberdade de outros especializarem a ontologia;*
- **Princípio da Distinção Ontológica** (Borgo, Guarino e Masolo, 1996): *todas as classes devem ser disjuntas;*
- **Diversificação de hierarquias** (Azpírez et al., 1998): *para melhorar a capacidade oferecida por múltiplos mecanismos de herança;*

- **Modularidade** (Bernaras, Laresgoiti e Corera, 1996 apud Gruber 1995): *minimizar acoplamento entre as entidades;*
- **Minimização da distância semântica entre conceitos irmãos** (Azpírez et al., 1998): *conceitos similares devem ser representados no mesmo grupo;*
- **Padronização de nomes sempre que possível** (Azpírez et al., 1998).

Outro importante conceito é o *reasoning*, que significa as inferências de relações e conhecimento feitas baseadas nas relações e conhecimentos pré-existentes. O uso de uma ferramenta de *reasoning* (um *reasoner*) ajuda a manter a consistência e a validar uma ontologia durante o seu desenvolvimento.

### 2.3 ONTOLOGIAS EM MANUFATURA

Muitos trabalhos foram desenvolvidos propondo uma ontologia no domínio da manufatura. O trabalho de (Pouchard, Ivezic e Schlenoff, 2000) propõe uma abordagem baseada em ontologia para colaboração distribuída na manufatura, objetivando interoperabilidade e mecanismos de tradução representando conceitos de manufatura. Estes conceitos são representados através de uma ontologia composta por três entidades básicas: *activity*, *object* e *timepoint*; e quatro relações básicas: *participates\_in*, *before*, *begin\_of* and *end\_of*.

Uma ontologia de manufatura foi proposta por (Zhou e Dieng-Kuntz, 2004), como forma de apoiar a *Excellent Manufacture*, na qual o conhecimento de manufatura é compartilhado através da internet. Ele define o modelo organizacional no conhecimento de manufatura como baseado em material, custo e fluxo de informação, e os termos de mais alto nível na ontologia como sendo produto, atividade, ator, instalação, método e valor.

O trabalho de (Borgo e Leitão, 2007) detalha o uso de ontologias no domínio da manufatura, com o propósito de controle na manufatura. Para isto, a arquitetura ADACOR

(*ADaptive holonic COntrol aRchitecture*) é utilizada. ADACOR tem a sua ontologia própria, proprietária, expressada através de um *framework* orientado a objetos, para suportar interoperabilidade entre holons ADACOR. É fortemente acoplada a uma arquitetura de controle holônica, o que restringe o seu uso e não está disponível em outro formalismo semântico mais geral.

A ontologia P-PSO (*Politecnico di Milano–Production Systems Ontology*) (Garetti e Fumagalli, 2012) é uma representação estruturada do domínio da manufatura, baseado em UML para prover sua representação semântica. Os aspectos abordados estão organizados da seguinte forma:

- **Físicos:** *definição material de sistema, incluindo equipamento, trabalhadores e instalações;*
- **Tecnológicos:** *definem a visão transformacional do sistema, incluindo o processo transformacional que acontece no sistema de manufatura;*
- **Controle:** *define o ciclo de gerenciamento, incluindo planejamento, agendamento e atividades de controle.*

Os autores sugerem alguns exemplos de uso, como troca de informação, atividades de *design* e controle. Uma validação foi feita através do seu uso em robótica para ambientes de manufatura automotiva.

MSDL (*Manufacturing Service Description Language*) (Ameri, Urbanovsky e McArthur, 2012) é uma ontologia desenvolvida para representar serviços de manufatura, de maneira formal, focados especialmente no domínio de maquinaria mecânica. Também descreve uma extensão do MSDL para incluir fundição de metal e uma interessante metodologia geral para extensão de ontologias. Seu objetivo original era “servir como linguagem ontológica para

um framework baseado em agentes para implantação de cadeia de suprimentos”<sup>5</sup>. É decomposto em cinco níveis de abstração, nomeadamente: *supplier-level*, *shop-level*, *machine-level*, *device-level* e *process-level*. *Supplier-level* descreve as capacidades de um fornecedor rodando uma instalação de manufatura, tais como expertise, habilidades, foco industrial e de produto. *Shop-level* “descreve as capacidades em nível de sistema de um sistema de manufatura possuído por um fornecedor e descreve o sistema através do seu layout e sistema de manuseamento de materiais e outros sistemas de suporte, tais como planejamento de produção e controle de estoque”<sup>6</sup>. *Machine-level* caracteriza máquina envolvidas na transformação de matéria-prima em bens. *Device-level* lida com a caracterização de dispositivos, que são considerados o nível mais baixo na hierarquia de recursos físicos no sistema de manufatura. Habilidades de máquinas envolvidas em manufatura podem ser inferidas através da agregação de serviços. *Process-level* descreve os processos de manufatura. Algumas das classes principais propostas no MSDL são *Service* para descrever serviços fornecidos por provedores de manufatura e consumidos por consumidores, *SupplierProfile* e *RFQ (Request for Proposal)*, representando demanda e oferta, com *SupplierProfile* tendo dois componentes principais: *Supplier* e *ManufacturingServices*, que os fornecedores provêem. É um foco interessante, devido ao caráter orientado a serviço do modelo. MSDL utiliza SWRL (*Semantic Web Rule Language*) para limitar conceitos através da criação de regras complexas que, em combinação com OWL, provêem um poderoso e atualizado formalismo semântico.

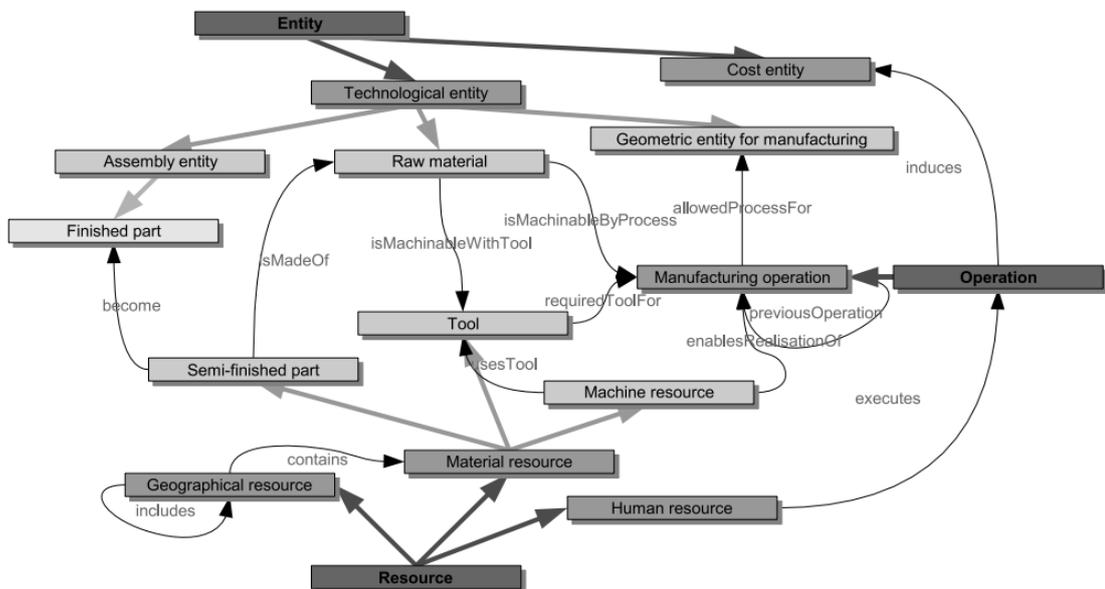
---

<sup>5</sup> Original em inglês.

<sup>6</sup> Original em inglês.

MASON (*Manufacturing Semantic ONtology*) foi proposta por (Lemaignan *et al.*, 2006) como uma *upper-ontology* para o domínio de manufatura, para ser usada como base para ontologias mais específicas por domínio. Este trabalho está relacionado à proposta de (Pouchard, Ivezic e Schlenoff, 2000), mas é baseado em formalismos mais atuais (OWL). Também utiliza SWRL, que provê meios para especificar regras mais complexas. É construído com base em três conceitos:

- **Entities:** *provê conceitos para especificar o produto, dando uma visão abstrata sobre ele, como características geométricas, material e custos;*
- **Operations:** *usada para descrever processos ligados a manufatura, como operações humanas e de lançamento;*
- **Resources:** *representa recursos usados e ligados a manufatura. Por exemplo, ferramentas de máquinas, ferramentas, recursos humanos e geográficos, como plantas fabris, oficinas, depósitos e assim por diante.*



**Figura 3** Visão geral das classes e propriedades principais do MASON. Fonte: (Lemaignan *et al.*, 2006)

Um ponto chave de MASON é que foi idealizada para ser uma *upper-ontology*, não ligada especificamente a nenhum domínio em manufatura. Entretanto, sua expressividade é limitada quanto à modelagem de serviços.

#### 2.4 ONTOLOGIAS EM MANUTENÇÃO

Alguns esforços foram feitos para formalizar a manutenção na área de desenvolvimento de *software*, mas há poucos trabalhos significativos sobre manutenção na área de manufatura. Entretanto, nas últimas décadas, pesquisas na área de *Condition-Based Maintenance* (CBM), têm sido realizadas. CBM se baseia na análise do ambiente e do dispositivo para avaliar a deterioração em uma máquina e realizar rotinas de manutenção baseadas nesta análise. CBM se diferencia da manutenção tradicional pois “ações de manutenção são baseadas na necessidade do maquinário”<sup>7</sup> (Fumagalli *et al.*, 2010). Sob esta perspectiva, o trabalho de (Emmanouilidis *et al.*, 2010) ressalta a importância de desenvolver ontologias de domínio no contexto de CBM, para estruturar o conhecimento e os dados relevantes na tarefa de diagnóstico, mas sem fazer a proposição de uma ontologia.

Entretanto, um conceito importante a ser considerado é o de falha. Em última análise, uma falha ou a sua possibilidade é o que causa as ações de manutenção desde o chão de fábrica até a cadeia de suprimentos de peças de reposição. Seus efeitos e análise, assim como uma própria rastreabilidade para peças e causas-raiz são de alta importância no contexto de uma apropriada integração entre todas as entidades envolvidas neste trabalho. Uma técnica

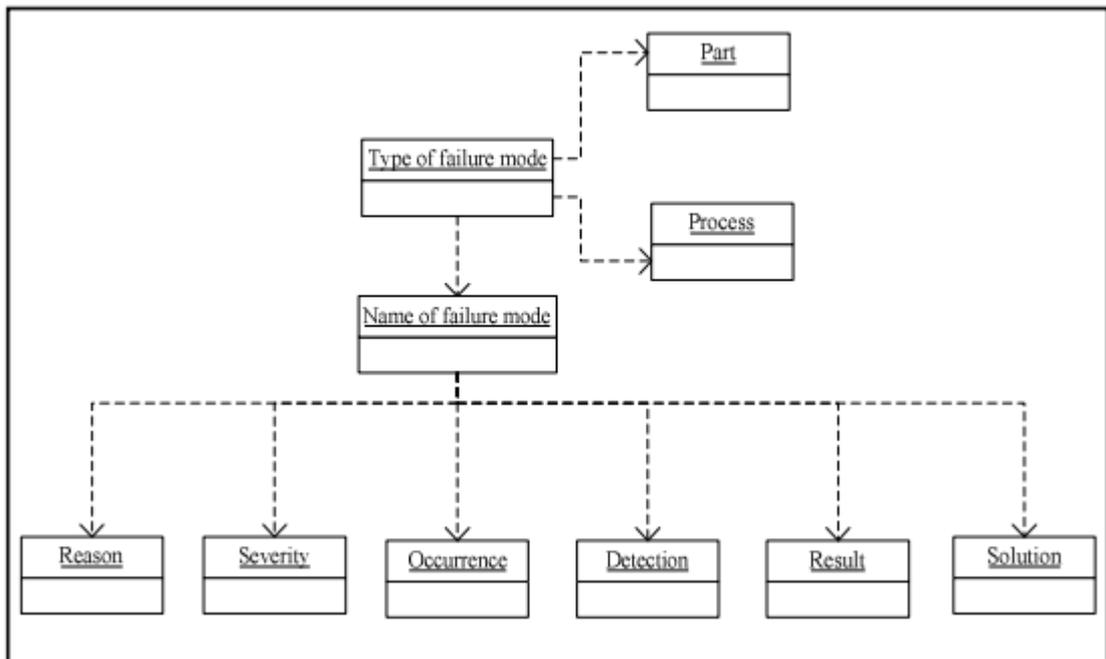
---

<sup>7</sup> Original em inglês.

sistemática para análise de falhas chamada FMEA (*Failure Mode and Effects Analysis*) foi proposta primeiramente pela NASA (Ebrahimipour, Rezaie e Shokravi, 2010) para analisar a segurança e a confiabilidade de sistemas de forma sistemática. FMEA provê um *framework* para entender e clarificar muitas características de eventos de falha, incluindo todos os sistemas, subsistemas e peças envolvidas, severidade, efeitos e causa-raiz. Provê um conjunto de conceitos que cobre uma série de características úteis para análise de eventos de falha. Alguns destes conceitos são (Langford, 1995):

- ***Failure***: a perda de uma função;
- ***Failure mode***: a forma específica como a falha ocorre. Por exemplo, a perda de um contato elétrico;
- ***Failure cause***: a(s) causa(s) que levam à falha. Pode ser, por exemplo, o mau ajuste de um contato elétrico;
- ***Failure effect***: as consequências de uma falha na operação;
- ***Severidade***: depende das últimas consequências de uma falha;
- ***Remarks/actions***: proposta de mitigação para diminuir ou justificar um nível de risco ou cenário.

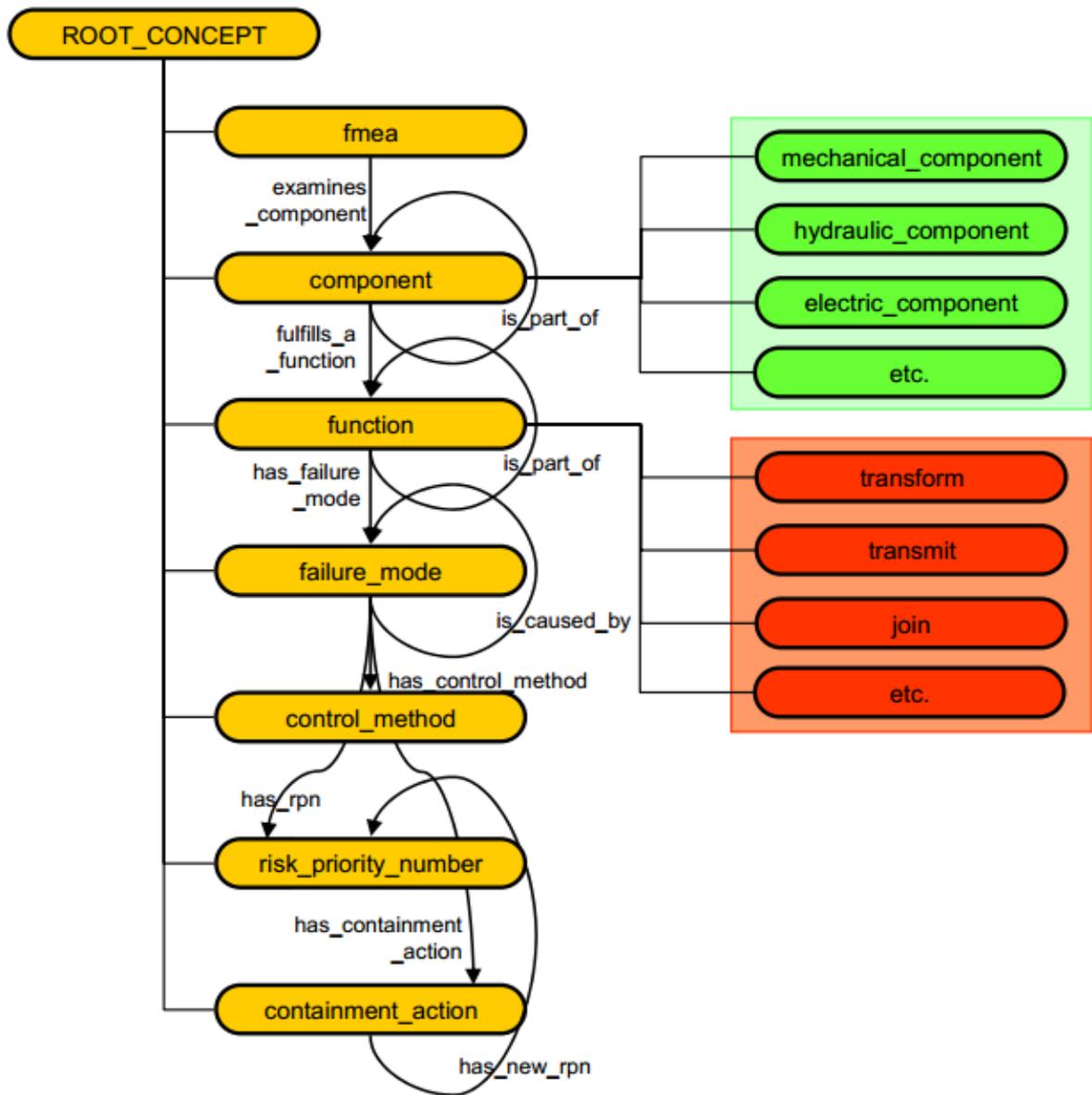
Alguns trabalhos foram propostos para descrever o conhecimento de FMEA usando ontologias. (Zhao e Zhu, 2010) apresenta um modelo baseado em UML para representar o conhecimento em FMEA no qual um modo de falha relacionado com uma peça ou um processo possui as seguintes características, baseadas nos conceitos originais de FMEA: *Reason*, *Severity*, *Ocurrence*, *Detection*, *Result*, *Solution*, conforme mostrado na Figura 4.



**Figura 4 Structural model of FMEA knowledge. Source: (Zhao e Zhu, 2010)**

(Koji, Kitamura e Mizoguchi, 2005) propuseram combinar uma ontologia funcional estendida com uma ontologia FMEA para lidar com funções e problemas de projeto inesperados por projetistas de produto. (Lee, 2001) apresenta um *framework* chamado DAEDALUS, proposto para lidar com problemas encontrados na integração de FMEA com modelos de diagnóstico.

O trabalho de (Dittmann, Rademacher e Zelewski, 2004) buscou combinar o conhecimento de FMEA com a engenharia de conhecimento, através do desenvolvimento de ontologias, de forma a facilitar o desenvolvimento de sistemas de informação para implementação de FMEA. A Figura 5 mostra alguns conceitos da ontologia de domínio proposta neste trabalho.



**Figura 5** Conceitos de ontologia de domínio FMEA. Fonte: (Dittmann, Rademacher e Zelewski, 2004)

## 2.5 ONTOLOGIAS EM CADEIAS DE SUPRIMENTO

Algumas ontologias de cadeias de suprimento foram estudadas, começando com aquelas descritas por (Grubic e Fan, 2010). EO (*Enterprise Ontology*) é uma tentativa de modelar o conhecimento de uma empresa, sendo a base para trabalhos mais especializados. Foi desenvolvida na Universidade de Edinburgo para analisar o uso de ontologias na modelagem

de empresas. Seus objetivos são (i) melhorar a comunicação entre humanos, (ii) prover a base para a especificação de aplicações e (iii) suportar a interoperabilidade. Propõe cinco seções: meta-ontologia e tempo; atividade, plano, capacidade e recurso; organização; estratégia; e *marketing*. Não é relacionada especificamente com a área de cadeia de suprimentos, apesar de alguns conceitos poderem ser adaptados para essa modelagem. Não possui versão disponível em formalismo mais atualizado, como OWL.

As ontologias TOVE (*Toronto Virtual Enterprise*) (Fadel, Fox e Gruninger, 1994) é um conjunto de ontologias contemplando ontologias de recurso, custo, organização, produto, atividade-estado-tempo e gerenciamento da qualidade. Algumas ontologias refletem uma perspectiva mais próxima da cadeia de suprimentos, como recurso, organização e atividade-estado-tempo. Foram implementadas com o uso de uma ferramenta C++ para representação do conhecimento e implementação dos axiomas. Não possui, também, formalismos mais atuais.

IDEON emprega visões representando diferentes conceitos e relações em uma empresa. As quatro visões são: *enterprise context view*, *enterprise organization view*, *process view* e *resource/product view*. É representada por um modelo orientado a objetos usando notação UML.

O trabalho de (Smirnov e Chandra, 2000) discute uma metodologia de desenvolvimento e provê um exemplo da indústria. É focado em cadeias de suprimento cooperativas, na integração e no gerenciamento do conhecimento entre membros e trata os membros da cadeia de suprimentos como fornecedores e consumidores, dependendo das relações de um membro com os outros. O exemplo da indústria é o projeto DESO (*Design of Structured Objects*). Suas unidades básicas são objetos como item, máquina, operação e assim por diante. Não está claro como o formalismo semântico foi representado, apenas que foi usada uma abordagem orientada a objetos.

O modelo SCOR (*Supply Chain Operations Reference*) é usado no trabalho de (Fayez, Rabelo e Mollaghasemi, 2005) para propor uma ontologia para simulação de cadeias de suprimento. De acordo com o modelo, a proposta é organizada nos processos *Plan*, *Source*, *Make*, *Deliver* e *Return*. Os processos de gerenciamento do modelo SCOR são ainda decompostos em três outros níveis de detalhe. O modelo SCOR foi usado como ontologia *core* e duas outras camadas foram adicionadas. A segunda camada foi chamada *middle ontology* e explicita formalmente todos os conceitos de diferentes visões da cadeia de suprimentos. A terceira camada, chamada *dynamic ontology*, provê a capacidade de adaptação para situações específicas e diferentes configurações de cadeias de suprimento. Este trabalho mostra alguns *screenshots* do *software* usado para implementar a ontologia, mas a ontologia propriamente dita não foi encontrada disponível.

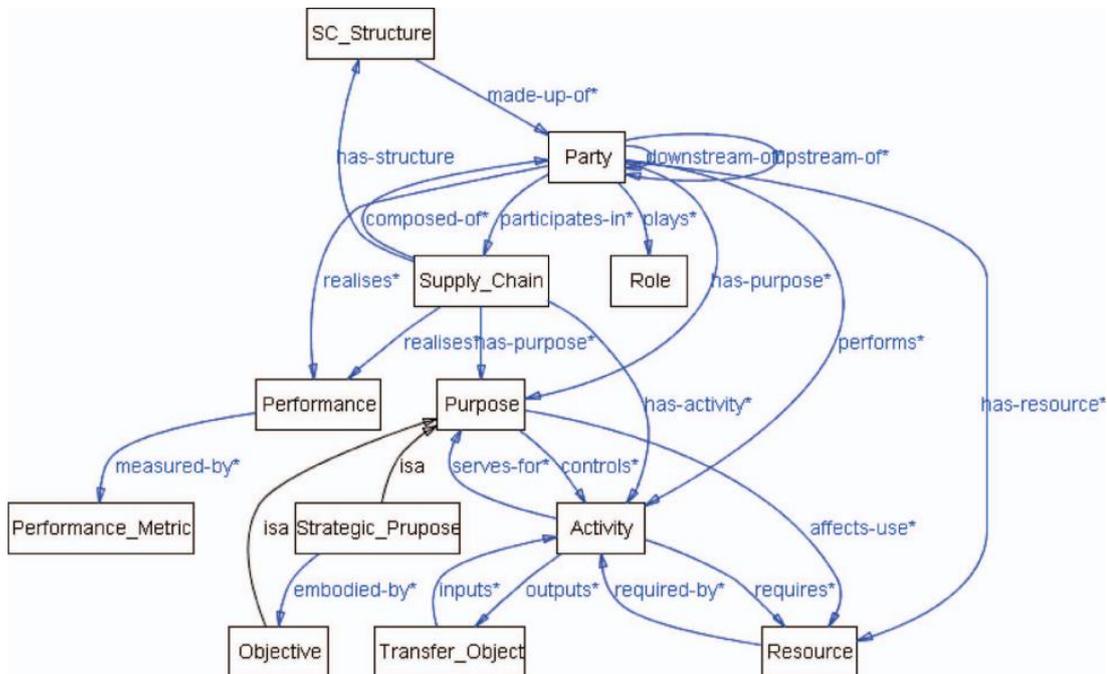
Uma ontologia chamada SCO (*Supply Chain Ontology*) foi proposta por (Ye *et al.*, 2008) com o objetivo de prover a integração semântica e a interoperabilidade entre aplicações dos membros da cadeia de suprimentos, agindo como uma “interlíngua” para a arquitetura de integração da aplicação. O modelo SCOR foi usado como base para a modelagem do conhecimento. O modelo de (Uschold *et al.*, 1998) é seguido para a construção da ontologia, que inclui quatro fases: identificação do propósito; construção da ontologia; avaliação; e documentação. Esta ontologia é extensível, para suportar informação e semântica adicionais para domínios específicos, e é parcialmente mostrada na Figura 6. As classes de alto nível são:

- ***Supply\_Chain***: representa a própria cadeia de suprimentos;
- ***SC\_Structure***: se refere ao conjunto de estruturas que representam as relações consumidor-fornecedor dentro da cadeia de suprimentos;
- ***Party***: é uma entidade legal que é parte da cadeia de suprimentos. Segundo os autores, cada parte é uma entidade superior ou inferior de outra parte na cadeia e

desempenha diferentes papéis, tais como fornecedor, fabricante, vendedor, consumidor, em termos de capacidade e objetivo.

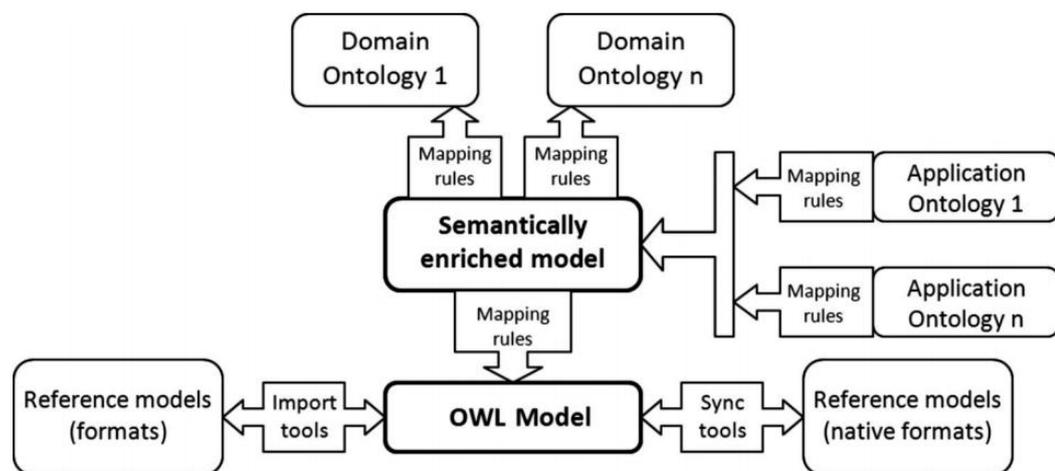
- **Purpose:** se refere ao conhecimento que afeta as decisões de gestão da cadeia de suprimentos e sua configuração e é especializado nas subclasses *Objective* e *Strategy*;
- **Activity:** é algo que precisa ser feito e requer uma quantidade de recursos;
- **Resources:** representam um mecanismo de suporte para atividades;
- **Performance:** representa resultados que as entidades obtém através de atividades;
- **Performance\_Metric:** refere-se à medida dos atributos de performance na cadeia de suprimentos.

A ontologia é parcialmente listada na publicação, mas a listagem não permite o uso da ontologia, que não foi encontrada disponível.



**Figura 6 SCO Top level classes and relations. Source: (Ye *et al.*, 2008)**

Devido a alegadas deficiências do modelo SCOR, (Zdravković *et al.*, 2011) propuseram uma abordagem para formalizar operações na cadeia de suprimentos superando tais problemas pelo uso de ontologias para estender o modelo e abordar os seus principais problemas. A proposta buscou interoperabilidade semântica entre os participantes e contribuir para o aperfeiçoamento do modelo. Esta abordagem foi demonstrada pelo desenvolvimento de modelos semanticamente alinhados do conhecimento implícito sobre as operações da cadeia de suprimentos (modelo SCOR), chamado SCOR-KOS (*SCOR Knowledge Organization System*), domínio do problema, chamado SCOR-Cfg para configuração de processo e, finalmente, uma micro teoria para operações de cadeia de suprimentos, chamada SCOR-Full, que enriquece semanticamente o modelo SCOR. Baseia-se em três das cinco abordagens propostas por (Holsapple e Joshi, 2002): indução, inspiração e síntese, conforme mostrado na Figura 7. As outras são dedução e colaboração. Indução refere-se à fase de enriquecimento semântico de um modelo de referência, melhorando a sua precisão semântica. Inspiração refere-se ao ponto de vista individual sobre as abstrações utilizadas, isto é, o ponto de vista do autor. Síntese é o mapeamento de uma ontologia enriquecido para outras ontologias, possibilitando a interoperabilidade semântica.



**Figura 7 Framework para enriquecimento semântico de modelos de referência. Fonte: (Zdravković *et al.*, 2011)**

Uma abordagem muito interessante foi apresentada por (Jian e Jianyuan, 2011) para a criação de um ontologia hierárquica para cadeia de suprimentos. Começando pelo reuso de ontologias clássicas como TOVE, e considerando o modelo SCOR, o autor propôs uma hierarquia composta por três camadas: meta ontologia, ontologia de domínio e *upper ontology*. A meta ontologia é composta pela ontologia de representação e a *upper ontology*. Ontologia de domínio inclui uma ontologia de empresa genérica, uma ontologia SCOR e uma ontologia de cadeia de suprimentos industrial. No nível mais baixo, conceitos detalhados para uso interno são providos, além de instâncias e um exemplo baseado na indústria de alimentos. A Tabela 1 mostra uma análise comparativa dos trabalhos relacionados.

**Tabela 1 Análise comparativa dos trabalhos relacionados**

| Domínio    | Trabalho                                    | Formalismo |   |
|------------|---|------------|---|
|            |   | semântico  | Características   |
| Manufatura | Pouchard, Ivezic e Schlenoff, 2000          | Não        | Colaboração distribuída, interoperabilidade e mecanismos de tradução.   |
|            | Zhou e Dieng-Kuntz, 2004                    | Não        | Conhecimento da manufatura compartilhado através da internet. Modelo organizacional baseado em material, custo e fluxo de informação.   |
|            | Borgo e Leitão, 2007                        | Não        | Utiliza arquitetura orientada ao uso de holons na manufatura.   |
|            | P-PSO (Garett e Fumagalli, 2012)            | UML        | Uso em troca de informações, atividades de design e controle.   |
|            | MSDL (Ameri, Urbanovsky e Mcarthur, 2012)   | OWL/SWRL   | Representa serviços de manufatura.  |
|            | MASON (Lemaignan et al., 2006)              | OWL        | Upper ontology para manufatura.   |
| Manutenção | FMEA (Ebrahimipour, Rezaie, Shokravi, 2010) | Não        | Framework para entender e clarificar muitas características de eventos de falha, incluindo todos os sistemas, subsistemas e peças envolvidas, severidade, efeitos e causa-raiz. |
|            | Zhao e Zhu, 2010                            | UML        | Baseado em FMEA.  |
|            | Koji, Kitamura e Mizoguchi, 2005            | Não        | Combina uma ontologia funcional estendida com uma ontologia FMEA para lidar com funções e problemas de projeto inesperados por projetistas de produto.                          |
|            | DAEDALUS (Koji, Kitamura e Mizoguchi, 2005) | Não        | FMEA direcionado a projeto de produtos.   |
|            | Dittmann, Rademacher e Zelewski, 2004       | Não        | Foca no desenvolvimento de sistemas baseados em FMEA.   |

## Cadeia de suprimentos

|  |                     |  |
|--|---------------------|--|
| Enterprise Ontology (Grubic e Fan, 2010) | Não                 | Parte de um estudo para analisar o uso de ontologias na modelagem de empresas.                                 |
| Tove (Fadel, Fox e Gruninger, 1994)      | C++                 | Conjunto de ontologias focando em custo, organização, produto, atividade-estado-tempo e qualidade.             |
| IDEON                                    | UML                 | Baseada em visões de uma mesma empresa.  |
| DESON (Smirnov e Chandra, 2000)          | Orientado a objetos | Foco em cadeias de suprimento cooperativas.  |
| Fayez, Rabelo e Mollaghasemi, 2005       | Não                 | Baseado em SCOR.   |
| SCO (Ye et al., 2008)                    | Não                 | Baseado em SCOR.   |
| Zdravković et al., 2011                  | Não                 | Modelo que demonstra o uso da arquitetura em camadas e extensível.   |
| Jian e Jianyuan, 2011                    | Não                 | Também usa arquitetura em camadas, começando por uma upper ontology e uma ontologia intermediária, de domínio. |

### 3 PROPOSTA DE MODELAGEM CONCEITUAL

#### 3.1 INTRODUÇÃO

Esta seção pretende apresentar o trabalho de criação da modelagem conceitual, etapa anterior à implementação propriamente dita da ontologia para integração entre IMS e SPSC. Iniciou-se pela definição de uma estratégia de desenvolvimento, definição da arquitetura e organização da ontologia e, finalmente, passou-se à definição de cada uma de suas partes.

#### 3.2 ESTRATÉGIA DE DESENVOLVIMENTO

O trabalho de (Holsapple e Joshi, 2002), já mencionado, propôs cinco abordagens para desenvolvimento de ontologias, as quais são citadas a seguir:

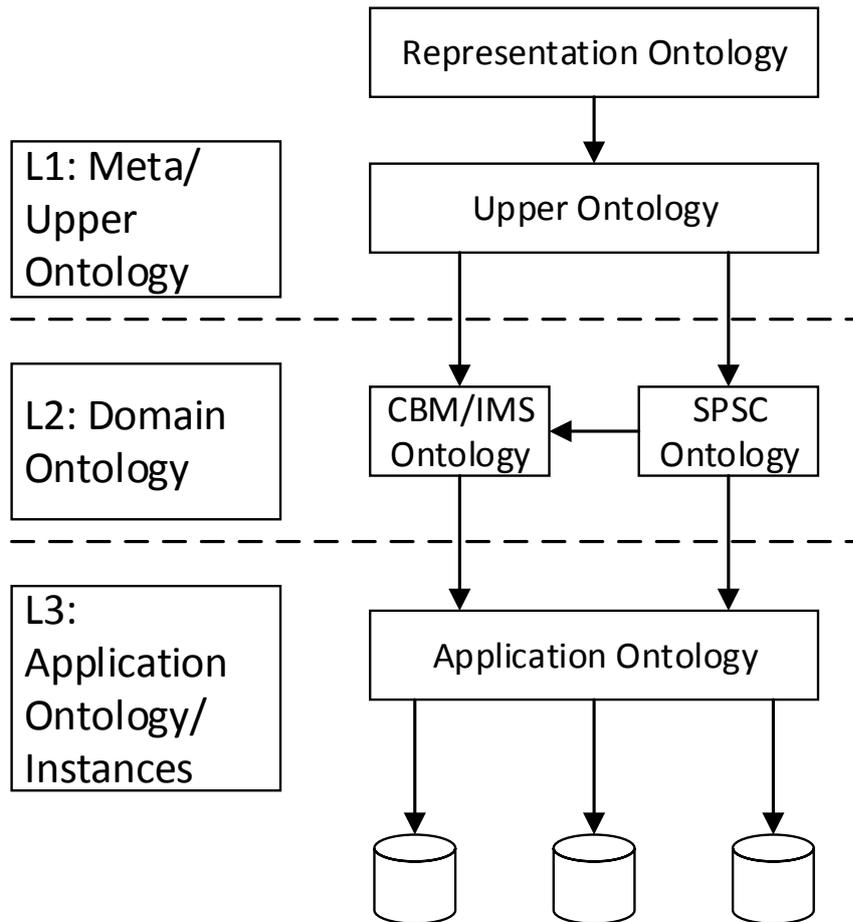
- **Abordagem inspiracional:** o desenvolvedor da ontologia inicia com o motivo pelo qual a ontologia é necessária e utiliza o seu conhecimento, criatividade e visão pessoal sobre o domínio de interesse para o desenvolvimento;
- **Abordagem indutiva:** a ontologia é desenvolvida pela observação de um caso específico no domínio de interesse e sua generalização para outros casos no mesmo domínio;
- **Abordagem dedutiva:** alguns princípios gerais são adotados e adaptativamente aplicados para construir uma ontologia orientada a um caso específico;
- **Abordagem sintética:** o desenvolvedor identifica um conjunto de ontologias que servem como base para uma ontologia resultante unificada;
- **Abordagem colaborativa:** o desenvolvimento é um esforço conjunto de uma equipe de pessoas que compartilham diferentes visões e opiniões para cooperar na construção da ontologia.

No contexto deste trabalho, foram escolhidas as abordagens inspiracional, sintética e colaborativa. Inspiracional pois uma significativa parte do trabalho, especialmente suas

primeiras versões, foram desenvolvidas através do trabalho e pesquisa individuais do autor. Sintética por causa da estratégia de construir uma ontologia baseada em trabalhos anteriores, como será melhor detalhado nas próximas seções. Finalmente, colaborativa devido ao contexto sobre o qual este trabalho foi desenvolvido, em um projeto internacional, com parceiros e especialistas de diferentes áreas avaliando e contribuindo com o resultado.

### 3.3 DESIGN MULTI-CAMADAS

Baseado na abordagem de (Jian e Jianyuan, 2011), foi proposta uma ontologia composta por múltiplas camadas, ou múltiplas ontologias. Entre os benefícios buscados através desta abordagem estão o reuso de *upper ontologies* e ontologias de domínio existentes, consistentes e largamente aceitas como válidas, aumentando a adoptabilidade do modelo que está sendo proposto. Além disso, uma flexibilidade maior é obtida através da definição de uma camada final onde conceitos específicos para cada caso poderão ser definidos. O *design* multi-camadas é mostrado na Figura 8.



**Figura 8 Design multi-camadas. Baseado em: (Jian e Jianyuan, 2011)**

A primeira camada, Meta Ontologia, inclui ontologia de representação e *upper ontology*. Representação descreve o que a ontologia é, seus atributos e relações. A *upper ontology* representa conhecimento de senso comum, não relacionado com um domínio específico.

A segunda camada, ontologia de domínio, descreve termos profissionais para cadeia de suprimentos e CBM e IMS, particularmente. Seu propósito é especificar com mais detalhes o que é conceitualizado pela *upper ontology*, baseado nos domínios considerados. Neste caso, consideramos IMS, CBM e SPSC como os domínios a serem modelados.

Finalmente, a camada de aplicação/instância, contém conceitos relacionados a casos de negócio específicos, assim como as instâncias de conceitos relacionados a tais casos. É

importante ressaltar que o nível de abstração diminui da camada mais alta para a mais baixa, de forma que não é necessário, e isto seria impraticável, modelar todas as possibilidades de integração entre IMS e SPSC em todos os tipos de negócio e contemplando todas as possíveis configurações, tecnologias e abordagens para este assunto. Ao contrário, esta abordagem permite flexibilidade suficiente para adaptar o modelo às especificidades de cada tipo de empreendimento.

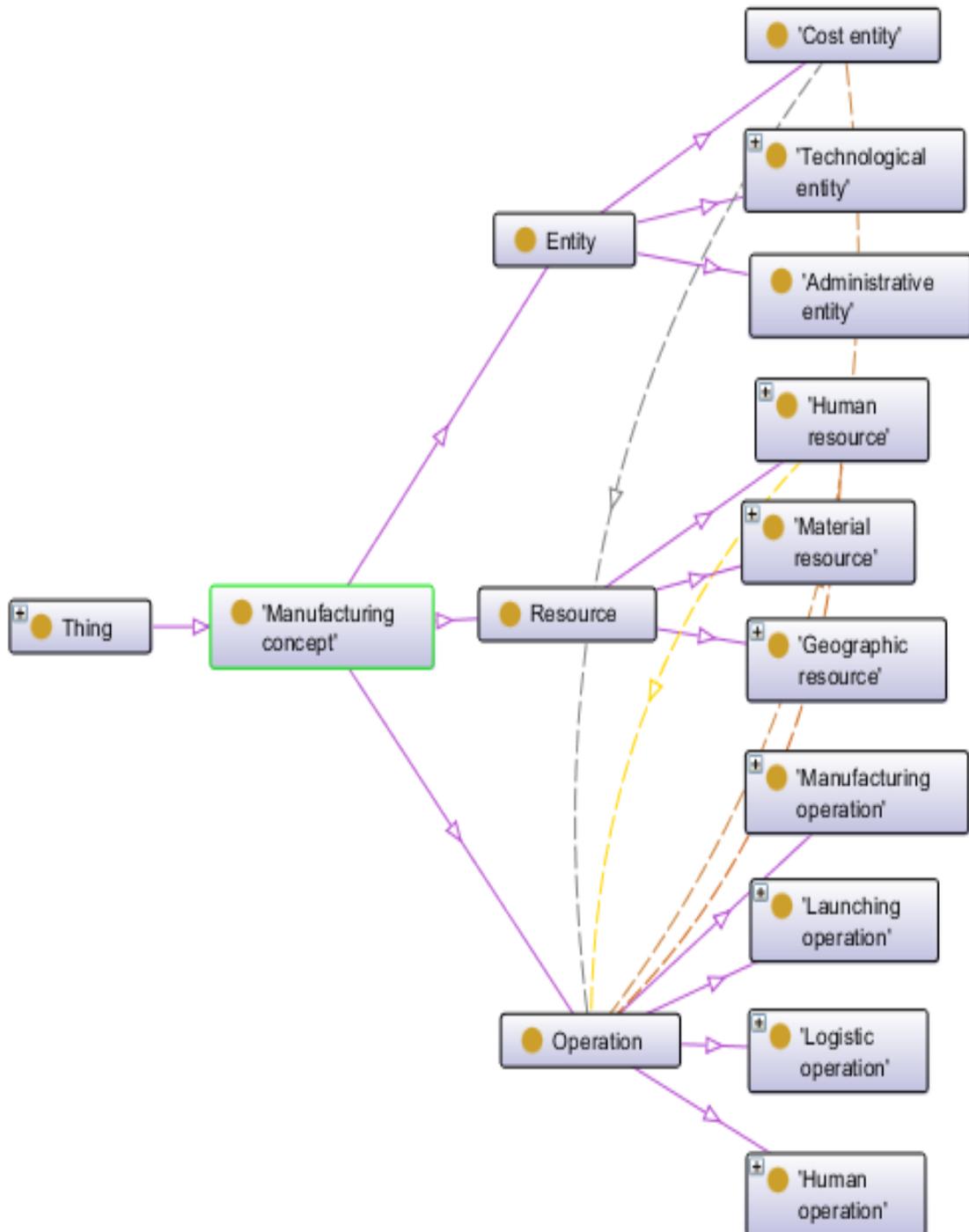
### 3.4 CAMADA DE META ONTOLOGIA

Dentre as inúmeras possibilidades entre os trabalhos já existentes, foram utilizados critérios como a facilidade de integração semântica, disponibilidade da ontologia em formalismo semântico atualizado e expressividade suficiente para embasar o desenvolvimento das próximas camadas. MASON foi, então, escolhida como *upper ontology* para este trabalho. É mais fácil de integrar com os conceitos de SPSC e CBM/IMS, devido à existência de conceitos como entidades, operações e recursos, que são comuns (semanticamente, não necessariamente sintaticamente) a todos os domínios que se quer modelar no contexto desta proposta. Apesar de alguma falta de expressividade no conceito de serviço, acredita-se que esta ausência não impacta a validade do modelo proposto.

Poder-se-ia argumentar, contudo, que MASON não é, em um sentido estrito, uma *upper ontology*, devido ao seu foco em um domínio, o de manufatura. Entretanto, no contexto deste trabalho, considera-se o MASON como um modelo genérico da área de manufatura, cujos domínios se referem a conceitos bem mais específicos do que aqueles propostos por MASON, tal como é a integração entre IMS e SPSC.

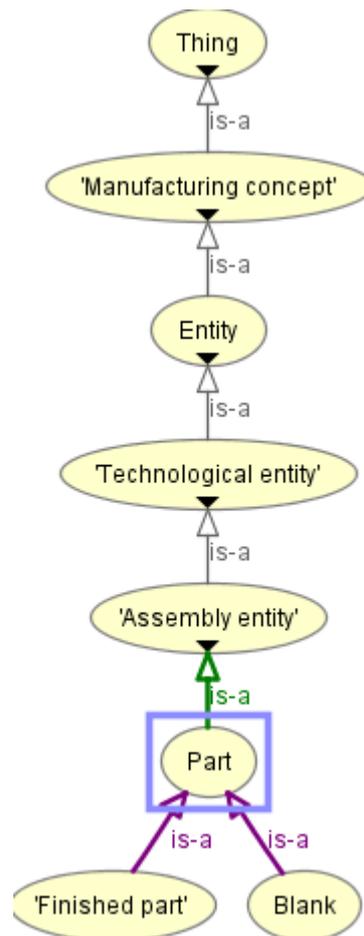
Como pode-se observar pela Figura 9, MASON possui três conceitos principais, herdados do conceito **Conceito de Manufatura** (*Manufacturing Concept*). **Entidade** é detalhada em entidades administrativa, de custos e tecnológica. Estes conceitos se referem a

características e componentes relacionados à existência, propósito e utilidade do que existe no contexto de uma empresa. **Operação** se refere a atividades executadas por recursos para atingir um objetivo. **Recurso** se refere aos participantes de processos e operações, tanto executores quanto habilitadores de tais processos, como recursos humanos ou ferramentas de máquinas.



**Figura 9 Hierarquia parcial de MASON**

Da perspectiva de CBM/IMS, o conceito de **Peça** é de grande importância. Apesar do fato de MASON prover esse conceito, como mostrado na Figura 10, o mesmo será amplamente aperfeiçoado nas próximas camadas, como será mostrado nas próximas seções.



**Figura 10** Conceito de peça na hierarquia de MASON

### 3.5 CAMADA DE ONTOLOGIA DE DOMÍNIO

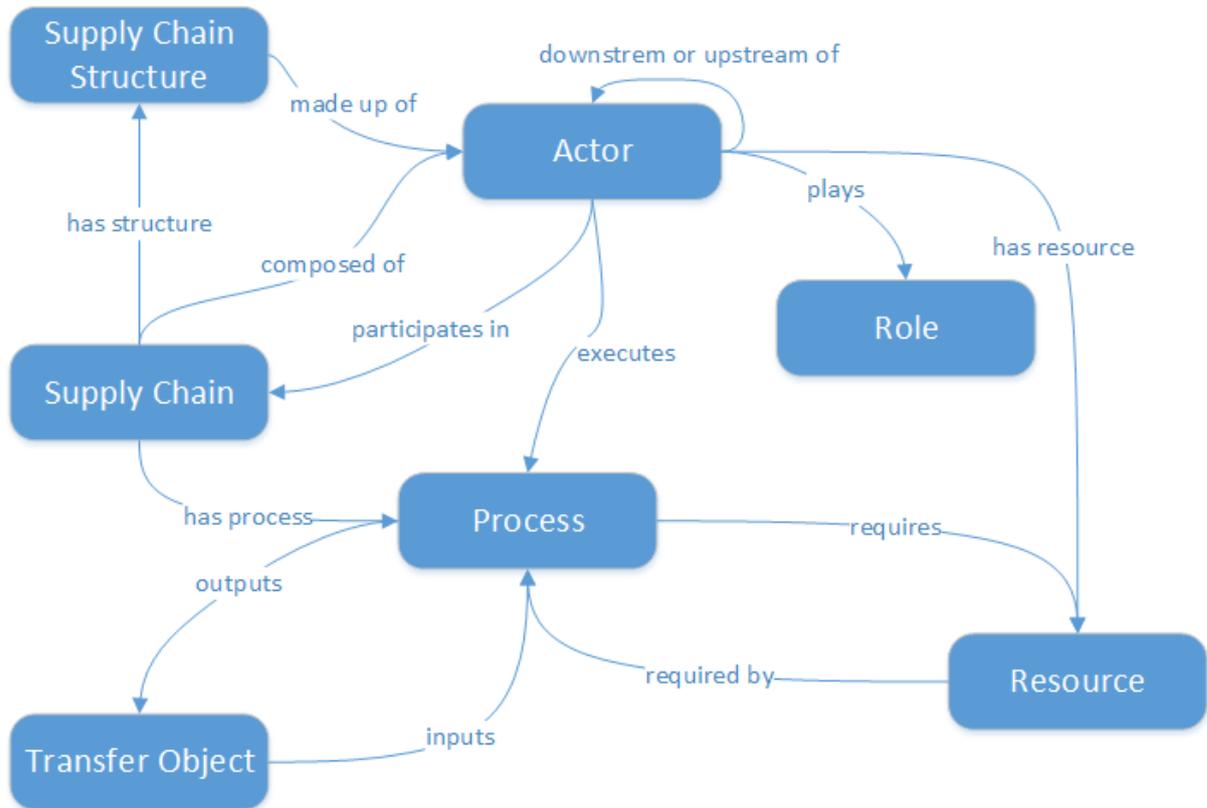
Esta camada descreve, a partir dos conceitos da camada superior, e derivando-os, os conceitos existentes nos domínios de SPSC e CBM/IMS. Para tanto, utiliza-se o conceito de herança a partir dos elementos da camada superior (por exemplo, pela Figura 10, percebe-se que *Part* é uma *Assembly Entity*, que é uma *Technological Entity* e assim por diante). Ou seja, os conceitos de MASON serão especializados para descrever conceitos da camada de domínios.

Para tanto, a camada de domínio será organizada em duas sub-camadas, hierarquicamente no mesmo nível, que serão detalhadas a seguir. Considerando a dificuldade de encontrar ontologias existentes disponíveis em formato utilizável para implementação, para ambas sub-camadas foram escolhidas ontologias existentes (pelo menos a representação) e conceitos existentes dos quais se derivou cada uma das sub-camadas, ou ontologias de domínio.

### 3.5.1 Cadeia de Suprimentos

SCO provê o detalhamento de sete conceitos principais, cuja análise semântica demonstra relativa facilidade de síntese com os conceitos propostos pela camada superior, razão que motivou a escolha desta ontologia como base para a sub-camada da ontologia de domínio. Além disso, ela também suporta a inserção de conceitos relativos a domínios específicos, o que possibilita a construção da camada de aplicação/instanciação.

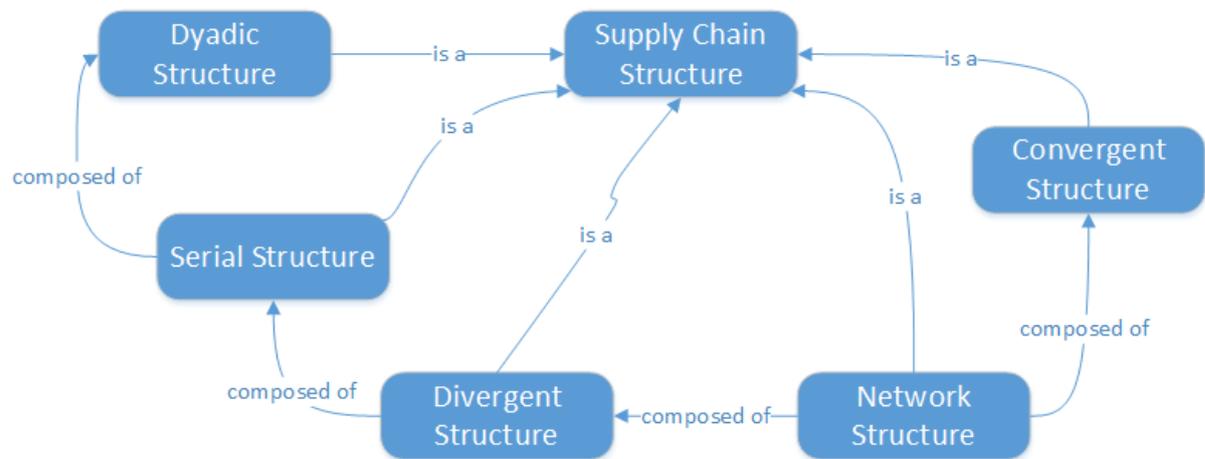
A Figura 11, mostra os conceitos principais de SCO, que serão detalhados para construção da sub-camada, ou ontologia de domínio de SPSC. **Cadeia de Suprimentos** (*Supply chain*) refere-se ao conceito da própria cadeia de suprimentos, e não será detalhado.



**Figura 11** Conceitos principais de SCO. Baseado em: (Ye *et al.*, 2008)

### 3.5.2 Estrutura da Cadeia de Suprimentos

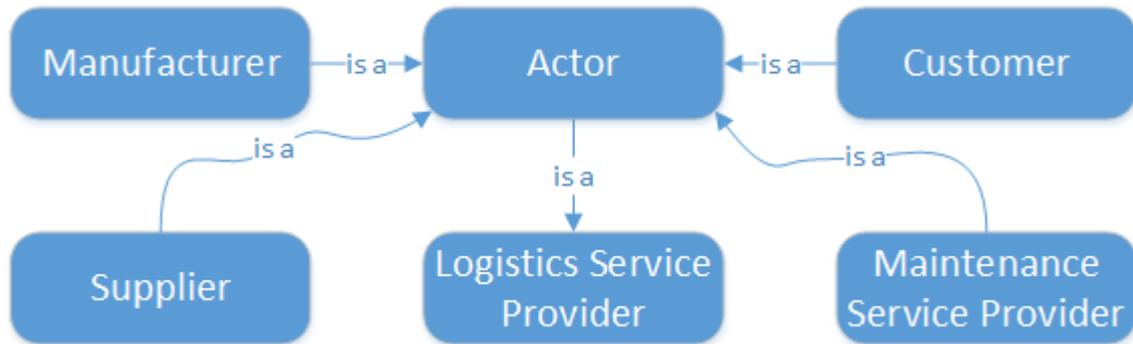
A **Estrutura da Cadeia de Suprimentos** (*Supply Chain Structure*) é determinada pela topologia e funções dos seus atores. Em uma **Estrutura Convergente** (*Convergent Structure*), vários produtos são combinados ou misturados para formar um único produto de saída. Já em uma **Estrutura Divergente** (*Divergent Structure*) a entrada é de um produto simples, que é separado em diversos novos produtos que compõem a saída. Se matérias primas são apenas alteradas em suas características, resultando no produto final, temos uma **Estrutura Serial** (*Serial Structure*). Uma **Estrutura de Rede** (*Network Structure*) descreve o fluxo de materiais para entidades que podem ser convergentes, divergentes, seriais ou uma mistura das três. Por fim, uma **Estrutura Diádica** (*Dyadic Structure*) se caracteriza por uma relação ponto-a-ponto entre dois atores de uma cadeia de suprimentos. A Figura 12, mostra os conceitos e relacionamentos entre as estruturas das cadeias de suprimento.



**Figura 12 Hierarquia de *Supply Chain Structure***

### 3.5.2.1 Ator

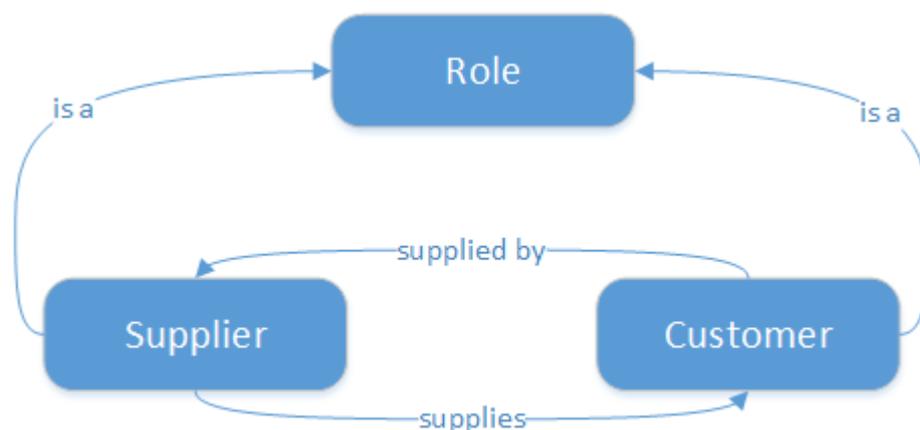
Os **Atores** (*Actor*) na cadeia de suprimentos são as diferentes entidades responsáveis por alguma função específica, cuja relação, baseada no fluxo de materiais, compõe a estrutura da cadeia. O **Cliente** (*Customer*) é aquele que, em última análise, gera a demanda e recebe produtos e serviços gerados na cadeia de suprimentos. No contexto deste projeto, são as organizações que utilizam IMS para prever a demanda e programar suas ações de manutenção. **Fabricante** (*Manufacturer*) é o ator responsável por receber produtos e matérias primas e transformá-los em novos produtos. **Provedor Logístico** (*Logistics Service Provider*) representa o ator que provê serviços de transporte e armazenamento na cadeia de suprimentos, enquanto o **Provedor de Manutenção** (*Maintenance Service Provider*) realiza os reparos, fornece pessoal técnico e ferramentas para ações de manutenção em equipamentos. Por fim, **Fornecedor** (*Supplier*) é responsável apenas pelo fornecimento de produtos e matérias primas, sem que exista, ou seja relevante, a ação de transformação, característica do fabricante.



**Figura 13 Hierarquia de Actor**

### 3.5.2.2 Papel

O conceito de **Papel** (*Role*) se diferencia do conceito de ator pois refere-se às relações entre os atores, e não à sua atividade propriamente dita. Desta forma, um ator pode ser tanto um **Fornecedor** (*Supplier*) quando estiver fornecendo produtos e serviços a outros atores da cadeia, quanto um **Cliente** (*Customer*), quando estiver adquirindo produtos de um ator na cadeia.

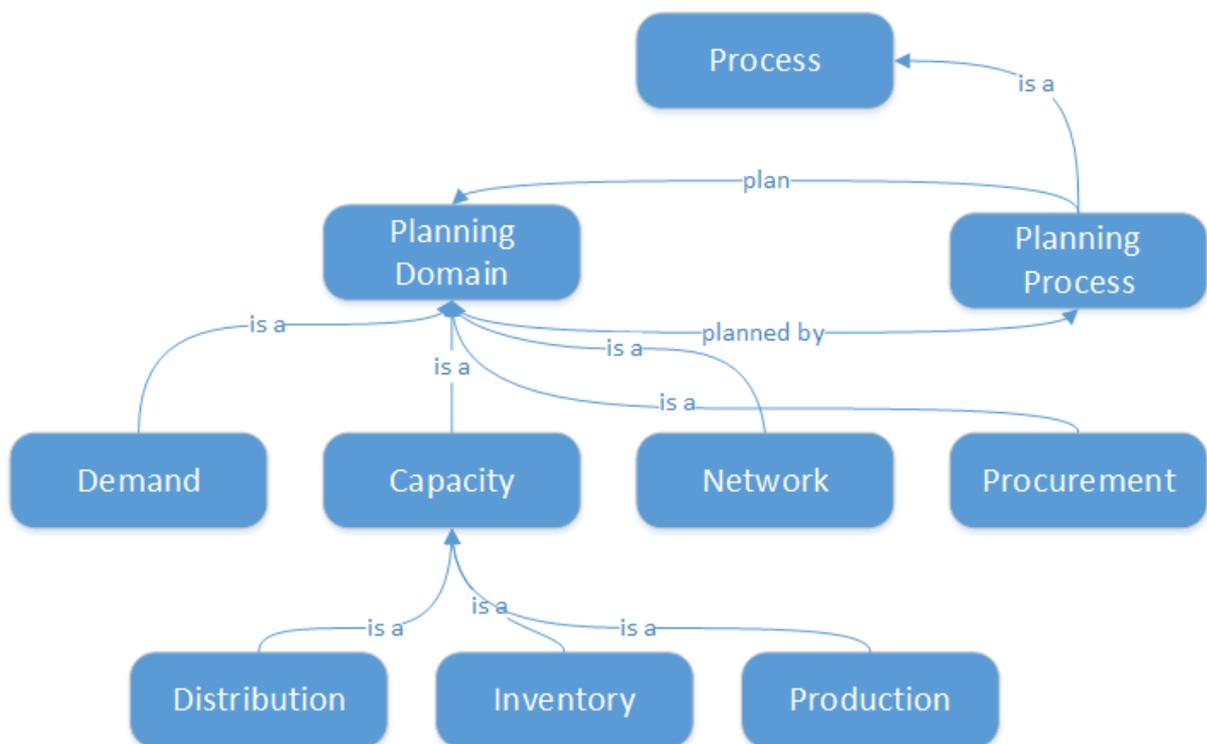


**Figura 14 Hierarquia de Role**

### 3.5.2.3 Processo

No contexto deste trabalho, limitou-se a definição dos processos àqueles que se referem ao planejamento, de acordo com o modelo proposto por (Hompele e Hellingrath, 2007), por causa da necessidade de utilizar informações de previsão de demanda obtidas por IMS para

o planejamento dos processos da cadeia de suprimentos. Desta forma, **Processo** (*Process*) refere-se a processos de planejamento que auxiliam na atividade de planejar os diferentes **Domínios de Planejamento** (*Planning Domain*), que são parte da cadeia de suprimentos e os processos de planejamento relacionados que estão sob o controle de uma organização (Stadtler e Kilger, 2008). Neste caso, temos o planejamento de **Demanda** (*Demand*), diretamente afetada pela avaliação de degradação de componentes por sistemas IMS, **Rede** (*Network*), relacionada à escolha de fornecedores e provedores de serviço que atendam à demanda, **Legais** (*Procurement*), relacionado a acordos, licenças, autorizações e aquisições necessárias à operação. Por fim, o planejamento de **Capacidade** (*Capacities*), que se desdobram no planejamento das capacidades de **Produção** (*Production*), **Distribuição** (*Distribution*) e **Estoque** (*Inventory*).



**Figura 15** Hierarquia de *Process*

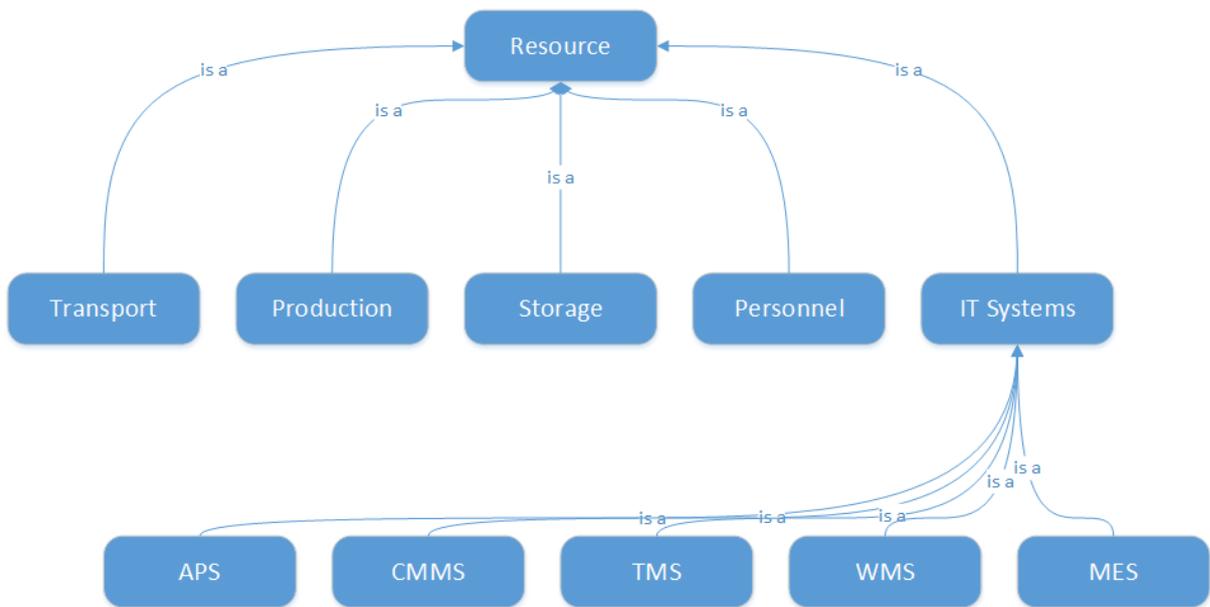
#### 3.5.2.4 Recurso

**Recurso** (*Resource*) refere-se ao conceito que representa os recursos de suporte aos atores da cadeia de suprimentos para desempenhar as suas atividades e garantir o fluxo de materiais e informações. O recurso **Pessoal** (*Personnel*) refere-se às pessoas nos diferentes papéis, desde operadores de produção, passando por pessoal de apoio até os técnicos de serviço ou manutenção. **Produção** (*Production*) representa os recursos necessários à produção, desde maquinários e equipamentos, até instalações e espaços de apoio. Recursos de **Transporte** (*Transport*) são aqueles necessários para garantir o fluxo de materiais entre os diferentes atores da cadeia de suprimentos, enquanto **Armazenamento** (*Storage*) refere-se aos recursos de armazenamento de materiais e produtos. Por fim, **Sistemas de Informação** (*IT Systems*) são os sistemas de apoio ao funcionamento da cadeia, cada um com um papel específico, assim organizados:

- **APS (Advanced Planning System):** *é um tipo de sistema que provê funcionalidades para coordenação dos fluxos de materiais e informações na cadeia de suprimentos, buscando o cumprimento de pedidos e o funcionamento geral da cadeia “on time”;*
- **CMMS (Computerized Maintenance Management System):** *sistema auxilia na gestão de operações de manutenção, colaborando no agendamento, decisões de compra ou reparo e no controle das ações;*
- **TMS (Transportation Management System):** *auxilia no planejamento, tomada de decisão, execução, acompanhamento e medição das ações de transporte de materiais;*
- **WMS (Warehouse Management System):** *é um sistema de apoio para controle do movimento e armazenamento de materiais em locais de armazenamento, além de outras tarefas associadas, como remessa, recebimentos e coletas;*

- **MES (Manufacturing Execution System):** sistema que auxilia as tomadas de decisão em chão de fábrica através da análise de múltiplos elementos do processo produtivo para traçar cenários de otimização da produção.

A Figura 16 mostra a hierarquia de recursos proposta nesta camada.



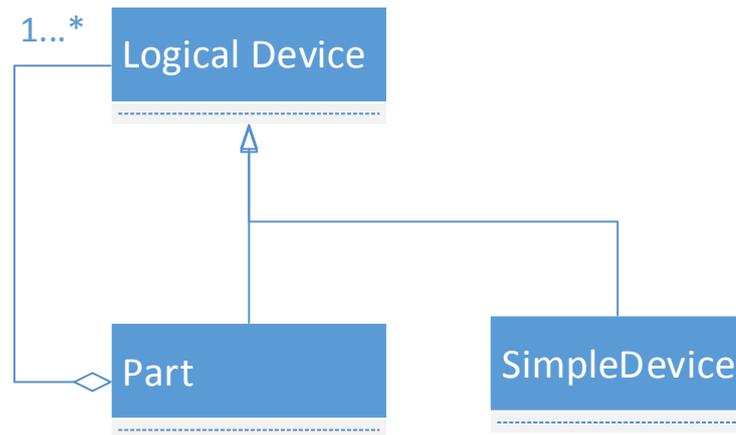
**Figura 16 Hierarquia de Recurso**

### 3.6 CBM/IMS

#### 3.6.1 Peça e Dispositivo

Um dos conceitos centrais nesta parte da ontologia é o de **Peça (Part)**. Trata-se de qualquer parte que componha um **Dispositivo (Device)** e que possua variáveis físicas, que podem ser monitoradas e a partir das quais a degradação do dispositivo será prevista. **Peças** serão definidas dentro de um nível de granularidade a ser definido, dependendo de cada contexto. Essa definição ocorrerá na camada de aplicação/instanciação. **Dispositivos** são elementos compostos de uma ou mais peças ou dispositivos que, em conjunto, realizam operações úteis no contexto de manufatura e cuja parada não-programada se deseja evitar.

**Peças e Dispositivos** possuem uma relação definida à partir do *design pattern Composite* (Gamma *et al.*, 1995), conforme a Figura 17. Desta forma, um dispositivo pode ser composto por uma ou mais peças, além de outros dispositivos simples.



**Figura 17 Padrão de projeto Composite. Baseado em: (Gamma *et al.*, 1995)**

Peças possuem, ainda, outros conceitos associados. Cada uma delas possui uma **Função** (*Function*), que representa o propósito da existência da peça na composição do dispositivo. A composição de várias funções das diferentes peças em um dispositivo resultará na função final do dispositivo. Além de propriedades físicas e industriais, como forma geométrica, massa, material, fabricante, código de produto e de fabricação, as peças possuem também **Variáveis Físicas** (*Physical Variables*) associadas a elas. É o monitoramento das variáveis físicas e seu comportamento ao longo do tempo que possibilita a avaliação do estado de degradação geral do dispositivo e de cada uma das peças para programação das ações de manutenção. O dispositivo mais alto em uma hierarquia possui todas as variáveis físicas dos dispositivos/peças que o compõem.

**Grupos de Dispositivos** são conjuntos de dispositivos que compartilham características que possibilitem a inferência dos seus estados de degradação. Por exemplo, seria razoável supor que um conjunto de válvulas do mesmo modelo e do mesmo lote de fabricação,

sendo utilizadas em condições semelhantes, poderão ter, em média, uma degradação semelhante. Neste caso, pode-se monitorar apenas uma válvula e inferir o estado de degradação de outras válvulas instaladas posteriormente.

### 3.6.2 Degradação

O conceito de **Degradação** (*Degradation*) é modelado baseado no conceito de falha de FMEA, com a diferença de que, no âmbito desta proposta, falha é o estado em que o **Nível de Degradação** é tal que existe a perda de função. Portanto, uma peça ou dispositivo possui degradação, medida através da alteração das variáveis físicas a eles associadas e, em último grau, o avanço do nível de degradação leva à falha. A determinação do nível considerado falha depende de cada caso específico.

Baseado em FMEA e no trabalho de (Dittmann, Rademacher e Zelewski, 2004), os outros conceitos associados à degradação são **Dispositivo/Peça** e **Função**, já descritos, **Modo de Degradação** (*Degradation Mode*), que representa a causa da degradação. **Severidade** refere-se ao nível de gravidade da possível perda de função gerada pela degradação. **Método de Monitoramento** (*Monitoring Method*) associado também ao conceito de IMS, além de **Ação de Contenção** (*Containment Action*) e **Técnico Responsável** pela manutenção (*Responsible Personnel*).

### 3.6.3 Intelligent Maintenance System (IMS)

O conceito de **IMS** descreve a entidade que é capaz de monitorar variáveis físicas de uma peça ou dispositivo, buscando determinar um nível de degradação que possa levar a uma perda de função, propiciando ações programadas de manutenção. Pode ser tanto um dispositivo embarcado quanto um sistema que realiza esse monitoramento de forma remota, recebendo dados através de uma rede de comunicação, por exemplo. Um IMS utiliza um **Método de Monitoramento**, que pode ser, por exemplo, análise visual ou de vibração, e um **Algoritmo de Monitoramento** (*Monitoring Algorithm*), que gera uma **Previsão de Degradação**

(*Degradation Forecast*), que possui uma **Data Provável de Falha** (*Failure Probable Date*) e que, junto com o nível de degradação, fornece informações sobre o estado atual de degradação da peça e o prognóstico de estado futuro, permitindo programar as ações de **Reparo** (*Repair*).

#### 3.6.4 Reparo, Ferramenta, Pessoal e Habilidades

O **Reparo** é uma ação que altera a degradação de uma peça, mudando o seu nível e prognóstico. Deve ser realizado por um ou mais técnicos responsáveis, que possui **Habilidades** (*Skill*) específicas e faz uso de **Ferramentas** (*Tool*). Estes conceitos são os mais próximos, semanticamente falando, do conjunto de conceitos de SPSC nesta camada. A Figura 18 mostra uma visão geral dos conceitos desta camada.



provavelmente, na modelagem de uma fábrica de roupas. Mas são válidos no contexto de uma refinaria de petróleo ou de uma infinidade de outros negócios. Por outro lado, após a conclusão da conceituação, ou seja, da definição do conceito de motor de passo, na instanciação passamos a representar não apenas o conceito, mas a existência física de um motor de passo, como representação de um elemento físico real.

Por conta das características específicas, um melhor entendimento desta camada se dará após a leitura do capítulo 5, que trata da criação de conceitos relacionados aos estudos de caso e a instanciação dos seus conceitos para validação da ontologia.

### 3.8 INTEGRAÇÃO SEMÂNTICA ENTRE AS CAMADAS

Nas seções anteriores descreveu-se a modelagem de conceitos e relações das ontologias dos dois domínios de integração (SPSC e IMS), ambas localizadas, segundo o *design* proposto, na camada de ontologia de domínio. Entretanto, para que este modelo realmente represente, ainda que parcialmente, os domínios no mundo real, é necessário demonstrar as relações com a *upper ontology*, que descreve os conceitos mais básicos, dos quais os outros conceitos herdam ou se especializam, e também as relações horizontais entre os conceitos de SPSC e IMS. É importante frisar que estas relações foram levadas em conta desde o início do desenvolvimento desta proposta. Entretanto, para fins de clareza e entendimento do texto, deixou-se a descrição de tais relações para depois da descrição dos conceitos das ontologias de domínio. Além disso, as definições apresentadas nesta seção podem não ser as únicas soluções semânticas para integração dos conceitos. Outras podem existir e ser igualmente válidas.

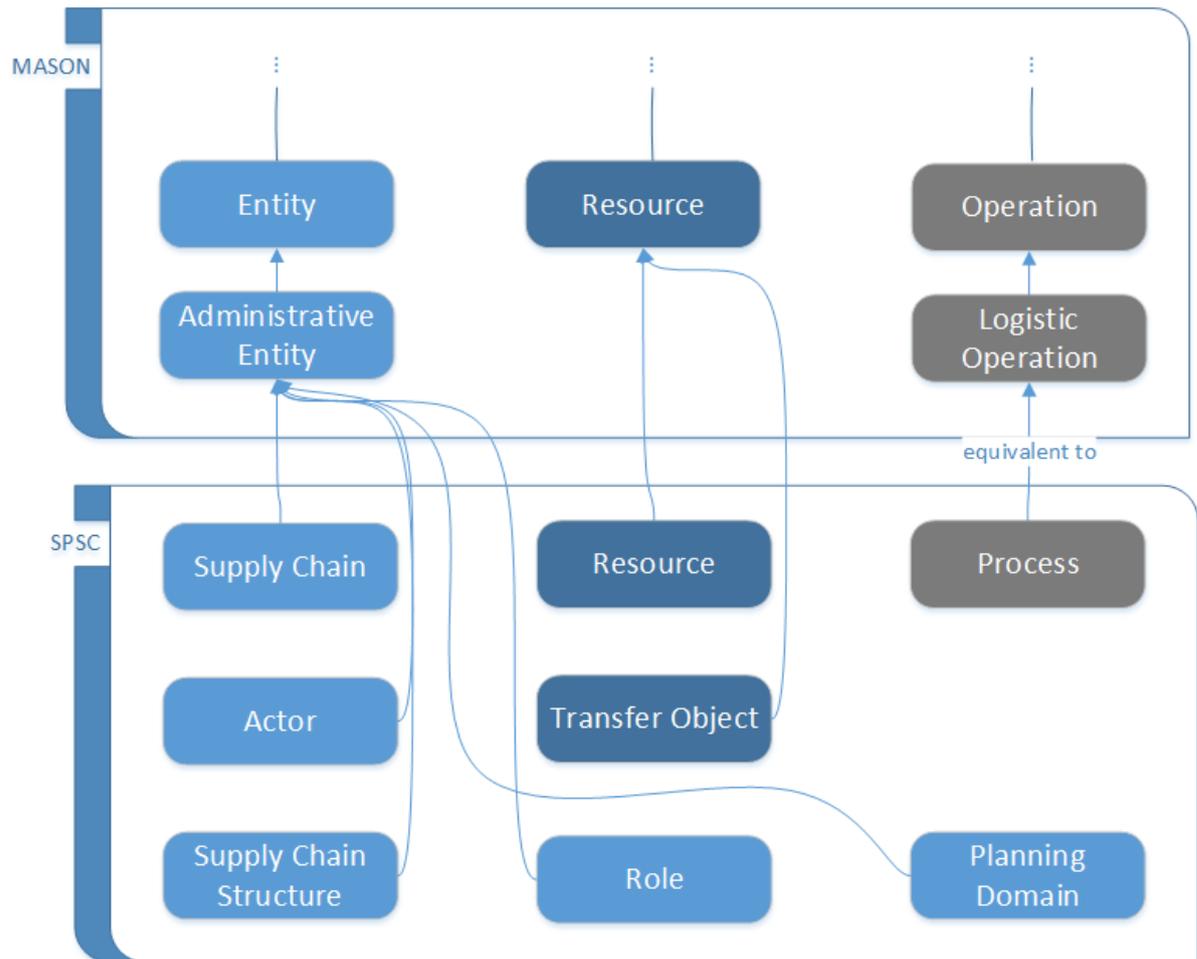
#### 3.8.1 Integração com MASON

Como *upper ontology* escolhida para esta proposta, foi necessário derivar os novos conceitos a partir dos conceitos originais de MASON. Nem todos os seus conceitos foram

utilizados, mas sempre que possível, os conceitos originais foram aproveitados. Segundo (Lemaignan *et al.*, 2006), MASON possui três conceitos principais:

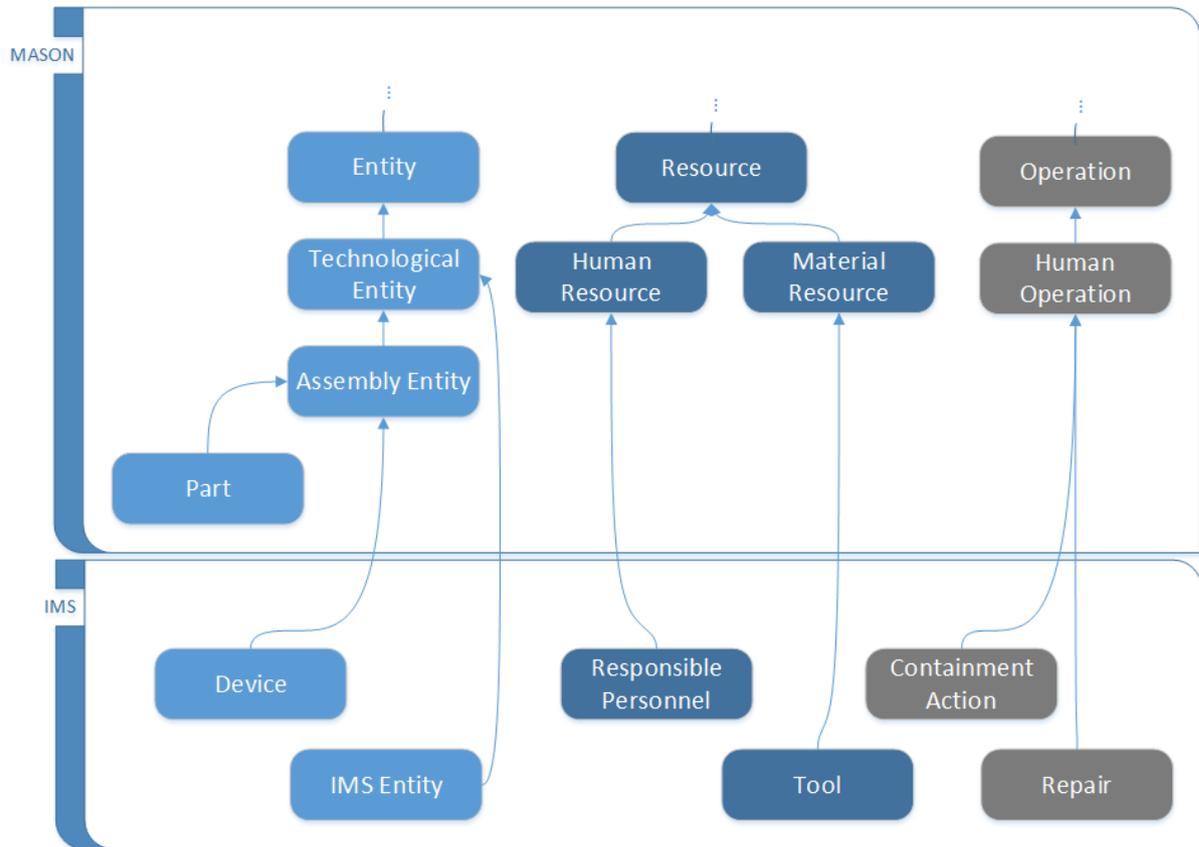
- **Entities:** *são todos os conceitos auxiliares comuns, cujo objetivo é ajudar a especificar, por exemplo, produtos, provendo uma visão abstrata dos mesmos. Possui subconceitos como technological entities, administrative entities e cost entities;*
- **Operations:** *cobre todos os processos ligados à manufatura, em um sentido amplo. Entre alguns dos seus conceitos principais estão manufacturing, human e logistic operations;*
- **Resources:** *inclui o conjunto de recursos ligados a manufatura, tais como tools, machine, geographic (plants, por exemplo) e human resources.*

Primeiramente, o conceito de **administrative entity** serviu como conceito pai para os principais conceitos da camada de SPSC. O próprio conceito de **Cadeia de Suprimentos** foi definido como “filho” de **administrative entity**, junto com **Estrutura da Cadeia de Suprimentos**, **Ator**, **Papel** e **Domínio de Planejamento**. **Processo** foi definido como equivalente à **Operação Logística**, abaixo do conceito **Operations** de MASON. Por fim, **Objeto de Transferência** foi alocado abaixo de **Resource**, enquanto foi criado um subconceito de **Resource** chamado **Recurso Logístico (Logistic Resource)** para abrigar os subconceitos de **Recurso** (da camada de SPSC). A Figura 19 mostra graficamente os locais de integração semântica entre MASON e a ontologia de SPSC.



**Figura 19** Integração semântica MASON-SPSC

Com relação ao domínio de IMS, o conceito de **Peça** já existia na ontologia MASON como subconceito de *Technological Entity* e *Assembly Entity* e foi reaproveitado. **Dispositivo** foi inserido no mesmo local de hierarquia. Um novo subconceito de *technological entity* foi criado com o nome **Entidade IMS** (*IMS Entity*) para abrigar os conceitos de **Degradação**, **Previsão de Degradação**, **Nível de Degradação**, **Modo de Degradação**, **Data Provável de Falha**, **Função**, **IMS**, **Algoritmo de Monitoramento**, **Método de Monitoramento**, **Variável Física**, **Severidade** e **Habilidade**. Já os conceitos de **Ação de Contenção** e **Reparo** foram inseridos como subconceitos de *Operation* e *Human Operation* da ontologia MASON. A Figura 20 mostra graficamente os locais de integração semântica entre MASON e a ontologia de SPSC.



**Figura 20 Integração semântica entre MASON-IMS Fonte: o autor**

### 3.8.2 Integração entre as ontologias de domínio

A integração entre as ontologias de domínio diz respeito ao estabelecimento das relações entre os conceitos de ambos domínios SPSC e IMS. O estabelecimento destas relações é necessário para o entendimento dos efeitos que os conceitos podem causar em outros, nos diferentes domínios. Não é intenção, como de resto em toda a proposta, definir exaustivamente todos as possíveis relações, mas estabelecer alguns deles e permitir o enriquecimento semântico destas relações ligado a cada área de negócio, através do uso da camada de aplicação/instanciação, que será detalhado adiante.

A relação mais básica que se pode inferir entre os domínios de SPSC e IMS é aquela entre a degradação de um dispositivo e a necessidade de peças de reposição, serviços de reparo

e pessoal técnico. Portanto, a primeira relação foi definida entre **Degradação e Demanda**, afirmando genericamente que a primeira “afeta” a segunda. Também é importante definir que uma ação de **Reparo** depende da **Capacidade**, ou seja, da disponibilidade de peças, ferramentas, pessoal técnico (com as habilidades necessárias), dentro da **Data Provável de Falha**. O **Reparo** também afeta os **Recursos** disponíveis, fazendo com que uma parte deles deixe de estar disponíveis (sejam utilizados) para a realização desta ação. Podemos dizer, então, que a disponibilidade de recursos “habilita” a realização do reparo e este, por sua vez, “utiliza” os recursos. Uma série de outras relações pode ser inferida dentro deste contexto, mas no âmbito desta proposta não pretende-se apresentar todas as possíveis definições, uma vez que considera-se, também, que uma parte significativa delas depende do contexto específico de cada área de negócio, que é modelado caso a caso na camada de aplicação/instanciação.

### 3.8.3 Integração entre a camada de ontologia de domínio e a camada de aplicação/instanciação

Esta integração dependerá de cada caso, uma vez que o uso da camada de aplicação/instanciação serve para especificar conceitos específicos de diferentes áreas de negócio. Entretanto, os conceitos criados nesta camada deverão ser semanticamente integrados com as camadas superiores. Assim, o conceito de **Peça**, por exemplo, poderá ser estendido na camada de aplicação como **Display Multi-toque**, **Placa Controladora** e **Carcaça**, por exemplo. Assim, estes conceitos combinados podem criar uma peça chamada **IHM** (Interface Homem-Máquina), que por sua vez pode ser parte de um dispositivo chamado **Torno CNC**. Da mesma forma, os diferentes atores da cadeia de suprimentos e suas relações específicas serão definidas a partir dos conceitos já expostos.

## 4 IMPLEMENTAÇÃO DA ONTOLOGIA

Este capítulo descreve o processo de representação dos conceitos definidos nos capítulos anteriores em uma linguagem formal, estruturada e utilizável para seu futuro uso no desenvolvimento de sistemas de integração entre SPSC e IMS.

### 4.1 ESCOLHA DA LINGUAGEM

Desde 2004, o W3C recomenda o uso de OWL (Mcguinness e Van Harmelen, 2004) para representação de ontologias. Além de ser um padrão, é voltada para utilização em web semântica, mas provê um conjunto completo de características necessárias para a descrição de ontologias. Ontologias descritas em OWL podem incluir os conceitos descritos a seguir.

#### 4.1.1 Classes e Relações Taxonômicas

São os conceitos relacionados a um domínio, e sobre os quais constrói-se toda a rede de relações e propriedades que descrevem aquele domínio. Em OWL, todas as classes são instâncias de uma metaclassa (`owl:Class`), que por sua vez ainda herda de `rdfs:Class`. Além disso, as relações de herança são definidas pelo elemento `rdfs:subClassOf`. Desta forma, a definição do conceito **Válvula** (*Valve*), subconceito de **Peça** (*Part*), poderia ser descrito da seguinte forma:

```
<owl:Class rdf:ID="Valve">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Part"/>
  </rdfs:subClassOf>
</owl:Class>
```

#### 4.1.2 Propriedades

As propriedades podem ser do tipo propriedade de objeto (`owl:ObjectProperty`) ou propriedades de tipo de dados (`owl:DatatypeProperty`). A primeira relaciona objetos a outros objetos, enquanto a segunda relaciona objetos a tipos de dados. Por exemplo, o objeto

**Motor de Passo** pode ter uma propriedade *velocidade máxima* (propriedade de tipo de dado) e uma propriedade *isPartOf*, relacionando este objeto com o objeto **Válvula**. Isto seria representado da seguinte forma:

```
<owl:DatatypeProperty rdf:ID="max_speed">
  <rdfs:domain rdf:resource="#StepMotor"/>
  <rdfs:range rdf:resource="xsd:int"/>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="isPartOf">
  <rdfs:domain rdf:resource="#Valve"/>
  <rdfs:range rdf:resource="#StepMotor"/>
</owl:ObjectProperty>
```

#### 4.1.3 Indivíduos

Indivíduos são as instâncias, ou a realização de um conceito. Utilizando o exemplo do paradigma de orientação a objetos, os conceitos são equivalentes às classes, enquanto os indivíduos são equivalentes aos objetos. A representação de um indivíduo é feita da seguinte forma:

```
<owl:NamedIndividual rdf:about="#Valve_MD120_1">
  <rdfs:type rdf:resource="#Valve_MD120"/>
</owl:NamedIndividual>
```

#### 4.1.4 Restrições

Delimitam certas características das propriedades. São consideradas expressões de classes e restringem as propriedades impedindo que assumam determinados valores, por exemplo. No exemplo abaixo, a restrição de um determinado fabricante de não comprar peças de um grupo empresarial concorrente.

```

<owl:Class rdf:about="#Buyer_1">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:allValuesFrom>
        <owl:Class>
          <owl:complementOf>
            <owl:Class rdf:about="#Competitor_1"/>
          </owl:complementOf>
        </owl:Class>
      </owl:allValuesFrom>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#canBeSuppliedBy"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

As restrições em OWL podem ser classificadas nos seguintes tipos (Horridge *et al.*, 2011):

- **Restrições de quantidade:** pode ainda ser do tipo existencial (*someValuesFrom*), quando os indivíduos devem possuir relação com ao menos um indivíduo através de determinada propriedade. Ou do tipo universal (*allValuesFrom*), quando um indivíduo só possui relações com indivíduos de uma determinada classe, através de uma propriedade específica;
- **Restrições de cardinalidade:** definem a quantidade mínima, máxima ou exata de relações com um determinado indivíduo pode participar através de uma propriedade. Por exemplo, um motor de passo pode ser parte de exatamente uma válvula;
- **Restrições do tipo *hasValue*:** quando define-se um conjunto de valores aceitáveis para uma propriedade de tipo de dados. Por exemplo, o valor máximo de rotação de um modelo de motor deve ser menor ou igual a 3600 RPM.

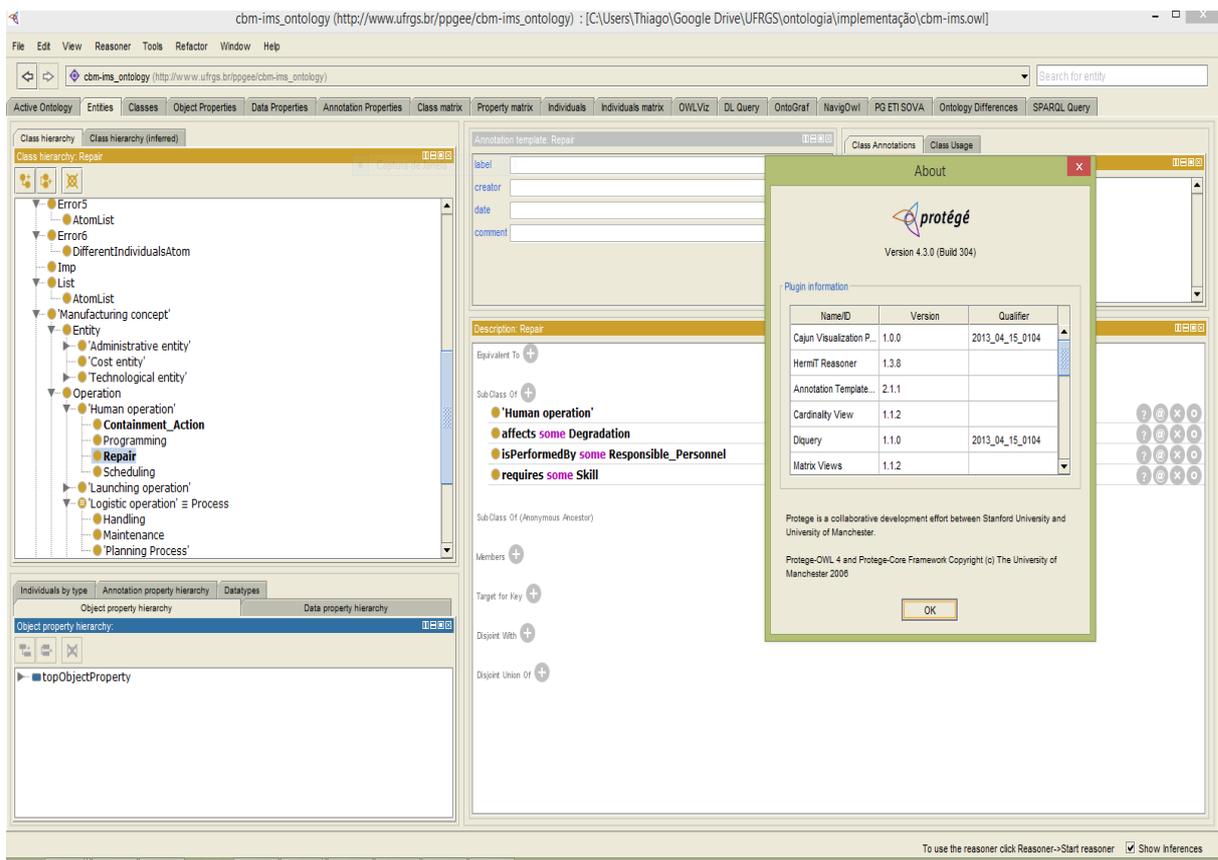
#### 4.1.5 Tipos de propriedades

Propriedades podem ser classificadas nos seguintes tipos (Horridge *et al.*, 2011):

- **Inversa:** se uma propriedade relacionando o indivíduo A ao indivíduo B, então a sua inversa irá relacionar o indivíduo B ao indivíduo A. Por exemplo, a propriedade *supplies* é inversa da propriedade *isSuppliedBy*;
- **Funcional:** neste caso apenas um indivíduo pode se relacionar com outro através desta propriedade. Por exemplo, a propriedade *isPartOf* que relaciona um motor de passo a uma válvula pode ser funcional, ou seja, um motor de passo pode ser parte de apenas uma válvula, e só dela;
- **Funcional inversa:** se uma propriedade é funcional inversa, isso significa que a sua inversa também é funcional. Neste caso, considerando que a propriedade *isPartOf* é funcional, a propriedade *composedBy* é sua funcional inversa;
- **Transitiva:** se a propriedade é transitiva e relaciona o indivíduo A com B e também o indivíduo B com C, então é correto dizer que essa mesma propriedade relaciona o indivíduo A com C. Por exemplo, se o indivíduo **Computador** é relacionado a **Gabinete** com a propriedade *isPartOf*, e **Gabinete** é relacionado a **Fonte** pela mesma propriedade, então podemos inferir que os indivíduos **Computador** e **Fonte** são relacionados por *isPartOf*, considerando que ela seja transitiva;
- **Simétrica:** uma propriedade é simétrica quando, se ela relacionar o indivíduo A com B, podemos inferir que ela também relaciona o indivíduo B com A. Por exemplo, se o indivíduo **Partner 1** se relacionar com **Partner 2** através de *isPartnerOf*, então podemos inferir que **Partner 2** também se relaciona com **Partner 1** através da mesma propriedade, considerando que ela seja simétrica;
- **Assimétrica:** quando uma propriedade que relaciona o indivíduo A com B não pode relacionar ao mesmo tempo o indivíduo B com A.
- **Reflexiva:** quando a propriedade relaciona o indivíduo a ele mesmo;
- **Irreflexiva:** quando a propriedade não pode relacionar o indivíduo a ele mesmo.

## 4.2 FERRAMENTA DE DESENVOLVIMENTO

A ferramenta escolhida para o desenvolvimento chama-se Protégé, versão 4.3 (Stanford, 2013), é desenvolvida em um esforço conjunto entre as universidade de *Manchester* e *Stanford*. É uma ferramenta livre, *open-source*, extensível e desenvolvida em Java, que provê mecanismos de descrição de conhecimento através de uso de OWL, abstraindo a necessidade de edição de código XML para uma interface gráfica. É uma ferramenta suportada por uma comunidade de desenvolvedores e, além da versão *desktop* utilizada neste trabalho, também possui uma versão *web*. Além de OWL, possui compatibilidade com outros formatos, como RDF(S) e XML Schema. A Figura 21 mostra a interface da ferramenta.



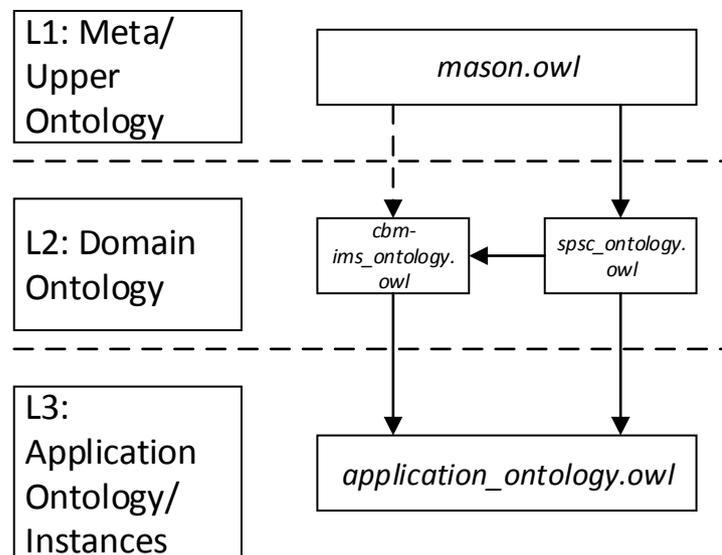
**Figura 21 Interface do Protégé 4.3**

#### 4.2.1 Estrutura de arquivos

Para implementar a estrutura proposta na seção 3.3, foram criados os seguintes arquivos:

- ***SPSC\_ontology.owl***: descreve os conceitos da camada de domínio, ontologia SPSC;
- ***CBM-IMS\_ontology.owl***: descreve os conceitos da camada de domínio, ontologia CBM/IMS;
- ***Application\_ontology.owl***: descreve os conceitos utilizados na camada de aplicação/instanciação. Utilizado para implementação do estudo de caso.

A *upper ontology*, representada na ontologia MASON, foi utilizada através do arquivo *mason.owl*, disponível no site do autor. Desta forma, as ontologias foram importadas, hierarquicamente, para criar o *design* proposto na seção 3.3, conforme mostrado na Figura 22.



**Figura 22** Arquivos na composição da ontologia

## 5 ESTUDOS DE CASO

### 5.1 ESTUDO DE CASO 1: VÁLVULAS INDUSTRIAIS

O desenvolvimento do estudo de caso baseia-se em um exemplo real de uma fabricante de válvulas industriais, com uma ampla gama de produtos e clientes em todo o Brasil. As ocorrências de falha foram levantadas no período de janeiro a julho de 2013. O resumo de ocorrências e as ocorrências por modelo estão descritas no Anexo. Como não é necessário implementar todos os tipos de falha e modelos ocorridos para validar a modelagem, serão agrupados os tipos de falha em três grupos que, no caso real, concentram o maior número de ocorrências: **Firmware**, **Eletrônica Embarcada** e **Estrutura Mecânica**. Apesar da simplificação, não haverá prejuízos na validação do conceito. Da mesma forma, as peças que compõem cada válvula seguirão a mesma classificação. Serão ainda utilizados três modelos de válvulas. Quanto à modelagem da SPSC, apesar do exemplo real utilizar apenas um grande estoque local de peças e enviá-las comumente por correio ou via aérea, incluiremos ainda um estoque intermediário. Também consideraremos que exista ao menos um provedor de serviços de manutenção, apesar de, no exemplo real, este serviço ser realizado internamente pelo cliente.

#### 5.1.1 Válvulas e peças

Primeiramente, foram definidas as características do produto. Cada válvula possuirá três tipos de peça: **Firmware**, **Eletrônica Embarcada** (*Embedded Electronics*) e **Estrutura Mecânica** (*Mechanical Structure*). Cada componente está associado com variáveis físicas, no nosso caso, temperatura e vibração, exceto no caso do **Firmware**, onde serão utilizadas as mesmas variáveis das outras partes, uma vez que um erro neste componente pode alterar as variáveis dos outros componentes. Os modelos e peças utilizados são descritos na Tabela 2.

**Tabela 2 Tabela de modelos de válvula e peças**

| <b>Modelo</b> | <b>Peças</b>   |
|---------------|----------------|
| Vlv_M1        | Firmware_M1    |
|               | Electronics_M1 |
|               | Body_M1        |
| Vlv_M2        | Firmware_M2    |
|               | Electronics_M2 |
|               | Body_M2        |
| Vlv_M3        | Firmware_M3    |
|               | Electronics_M3 |
|               | Body_M3        |

### 5.1.2 Degradação e conceitos associados

Degradação é um conceito associado a um dispositivo e possui um conjunto de informações baseadas em FMEA. Alguns, como severidade, possuem valores padrão e são definidos na camada CBM/IMS. Outros dependem de conhecimento específico de negócio, como modo de falha, causa, efeito e habilidades necessários para execução de um reparo. Neste estudo de caso estas informações foram definidas conforme mostrada na Tabela 3, implementando estes conceitos na camada de aplicação.

**Tabela 3 Alguns conceitos relacionados à degradação e definidos na camada de aplicação do estudo de caso 1**

| <b>Failure Mode</b>     | <b>Failure Cause</b> | <b>Failure Effect</b> | <b>Skills</b>         |
|-------------------------|----------------------|-----------------------|-----------------------|
| Overheating             | Bad contact          | Interruption          | Assembly and electric |
| Fuse                    | Over use             | Interruption          | Electric              |
| Unintended interruption | Firmware bug         | Interruption          | Informatics           |
| Short circuit           | Seal failure         | Interruption          | Assembly and electric |
| Short circuit           | Assembly failure     | Interruption          | Assembly and electric |

Criamos algumas instâncias de degradação, com informações diferentes relacionadas a diferentes tipos de válvulas. Na implementação de sistemas reais, um IMS geraria estas

instâncias baseado no estado de cada dispositivo e, conforme o seu valor e critérios pré-definidos, poderia enviar estas informações a um sistema APS na camada de cadeia de suprimentos.

### 5.1.3 Grupos de Dispositivos

Neste caso, também criamos na camada de aplicação o conceito de “grupos de dispositivos”. Este conceito descreve conjuntos de dispositivos que compartilham alguma característica comum, como localização ou modelo. Baseado nestas características comuns, um IMS poderia monitorar apenas uma amostragem de tais dispositivos e, a partir destes, o comportamento de degradação pode ser extrapolado para toda a população de dispositivos dentro de um conjunto. Esta abordagem é particularmente interessante para casos em que existam limitações técnicas, geográficas ou de custo para que vários dispositivos do mesmo tipo possuam um IMS acoplado a eles.

## 5.2 ESTUDO DE CASO 2: AI2MS

O trabalho de (Zuccolotto *et al.*, 2015) propõe uma arquitetura para um Sistema de Manutenção Inteligente distribuído, que utiliza sistemas imunológicos artificiais (ou em inglês *Artificial Immune Systems - AIS*), que são uma série de algoritmos inspirados em sistemas bio-imunes, para implementar as funcionalidades de IMS em sistemas de produção de larga escala e geograficamente distribuídos. Esta proposta é chamada *Artificial Immune Intelligent Maintenance System (AI2MS)*.

Uma abordagem multi-agentes foi escolhida para implementar a arquitetura. Os agentes foram classificados em 3 camadas:

- **Camada de dispositivo:** camada básica composta por agentes responsáveis por monitorar uma máquina ou um dispositivo. Composta por *Sensor Agents (AS)*, *Sensor*

*Diagnostic Agents (SDA), Fault Detection Agents (FDA), New Fault Detection Agents (NFDA) e Device Health Assessment Agent (DHAA).*

- **Camada Colaborativa:** *agentes que promovem a colaboração entre categorias de dispositivos através da troca de informações sobre condições de operação e novos modos de falha encontrados no sistema. É composta pelo Cooperative Detection Agent (CDA), que interage com o NFDA para avaliar padrões em vários dispositivos e, assim, detectar novos modos de falha; e pelo Evolution Agent (EA), responsável por promover um NFDA a FDA.*
- **Camada de Planta:** *camada de gerenciamento de manutenção de uma planta, composta pelo Plant Health Assessment Agent (PHAA), responsável por avaliar a “saúde”, de uma planta e comunicar a um sistema de supervisão de produção, e o Failure Mode Update Agent (FMUA), que compartilha dados em caso de conexão ineficiente, além do Update Training Agent (UTA), para coletar dados de treinamento no sistema.*

AS são agentes conectados a um sensor físico que monitoram uma ou mais peças de um dispositivo. Este agente provê dados para o NFDA, DHAA e UTA. É sempre fonte de dados não processados (*raw data*), mas também pode provêr a capacidade de extração, como coeficientes de energia de pacotes *wavelet*, coeficientes de transformada de *Fourier* valor RMS e outros.

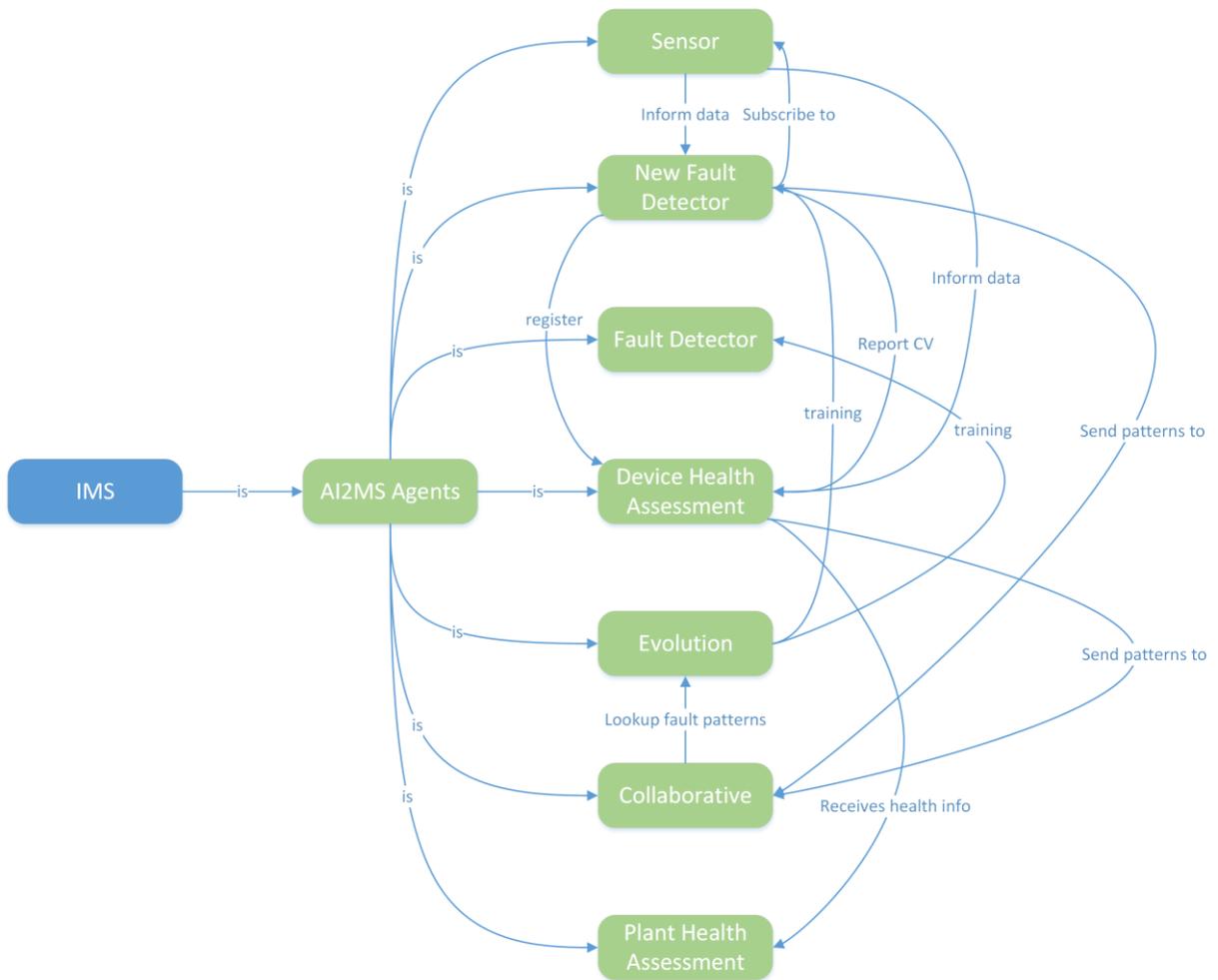
SDA são conectados a AS para supervisioná-los e, em caso de mau funcionamento, informar para o NFDA anexo que os dados devem ser corrigidos ou descartados, além de solicitar manutenção ao DHAA.

FDA e NFDA são responsáveis pela detecção de falhas e diagnóstico de peças específicas, através da avaliação de uma métrica chamada *Confidence Value (CV)*, que pode ter valores entre 0 (peça danificada) e 1 (peça nova, em perfeito estado). NFDA se conecta a SA e, caso CV caia a um determinado valor, indicando falha, este agente reporta o fato a DHAA

e requisita uma ação de manutenção. FDA procura por padrões de falha conhecidos, enquanto NFDA procura padrões desconhecidos em uma única máquina. Já CDA trabalha em máquina diferentes.

DHAA é responsável pela avaliação da saúde do dispositivo, prevendo a vida útil restante e requisitando manutenção quando necessário. Também pode solicitar mudanças no modo de operação do dispositivo para evitar quebra ou estender a vida útil, caso a manutenção não possa ser providenciada em tempo hábil. Cada dispositivo possui um DHAA, que interage com o PHAA, responsável por gerenciar a manutenção da planta.

Como era de se esperar, a representação de AI2MS usando a corrente ontologia, se dá através da criação de conceitos e relações na camada de aplicação. Neste caso, os agentes AI2MS são considerados uma derivação do conceito IMS – *AI2MS Agent* – e, à partir deste, são derivados os agente seguintes e suas relações. A Figura 23 Representação parcial da camada de domínio (IMS) e dos agentes AI2MS do estudo de caso 2. Figura 23 mostra a representação parcial da camada de domínio (IMS), e da camada de aplicação, com a derivação do conceito de IMS em *AI2MS Agents* e nos demais conceitos e relações.



**Figura 23** Representação parcial da camada de domínio (IMS) e dos agentes AI2MS do estudo de caso 2.

### 5.3 CONSIDERAÇÕES SOBRE OS ESTUDOS DE CASO

É importante notar as características dos estudos de caso apresentados. No Estudo de Caso 1, o conjunto de conceitos adicionados possuem um nível de abstração muito mais alto – são definidos tipos de dispositivos e suas relações e elementos da cadeia de suprimentos, ao passo que características mais específicas dos próprios IMS, formas de integração e, até mesmo quais são os sistemas envolvidos nesta integração, não são mostrados.

Já no Estudo de Caso 2, o enfoque é bastante diferente: quer-se modelar uma abordagem distribuída para a implementação de um IMS, ao passo que nenhum dos elementos relacionados à cadeia de suprimentos é modelado.

Isto demonstra um caráter importante do presente trabalho, que é a flexibilidade com que os conceitos e relações podem ser utilizados em diferentes áreas, mantendo a compatibilidade geral na implementação prática dos conceitos. Ou seja, um provedor de serviços de manutenção e logística, pode estar interessado em modelar apenas aquilo que refere-se à sua área de atuação, sem maiores preocupações com a definição de características técnicas de IMS sem violar a integridade semântica da ontologia. Aliás, uma das grandes vantagens do uso desta ontologia é que áreas tão diversas como IMS e SPSC podem ser modeladas por equipes diferentes, com enfoques diferentes, dentro de um mesmo conjunto de conceitos e um vocabulário comum, permitindo a efetiva integração destas distintas áreas.

Os estudos de caso apresentados podem garantir a completude e consistência desta proposta? Primeiramente, o item mais importante a ser garantido é a consistência da ontologia. Foram escolhidos estudos de caso de áreas bastante diferentes, com abordagens e preocupações distintas e foi possível modelar ambas com sucesso. Isto não garante a consistência para todos os possíveis casos, mas é um indicativo forte da mesma. Esta consistência será melhor provada quanto mais desenvolvimentos de novos modelos para usos específicos forem realizados, tendo como base a proposta aqui apresentada.

Quanto à completude da ontologia, considera-se que esta qualidade deve consistir na presença de todos os elementos básicos, e não na existência de todos os possíveis conceitos relacionados a todas as situações de utilização possíveis. Sob este aspecto, seria impraticável modelar uma ontologia “completa” (todos as situações e conceitos possíveis). Uma abordagem mais adequada é fazer o modelo ser extensível, da maneira como desenvolvemos esta proposta.

## **6 USO DA ONTOLOGIA PARA A CRIAÇÃO DE FERRAMENTAS DE INTEGRAÇÃO**

A presente ontologia proposta pretende constituir-se como possibilidade de modelagem semântica da integração entre IMS e SPSC, de modo a estabelecer um domínio de discurso comum para facilitar a criação de sistemas de informação que realizem tal integração. Uma das vantagens de tal abordagem é que, independente da abordagem tecnológica escolhida (linguagem, plataforma, hardware), uma vez que dois sistemas diferentes se baseiem na mesma ontologia, ambos serão semanticamente compatíveis, o que facilita a interoperabilidade entre eles. Ainda que partes de um sistema integrador entre IMS e SPSC sejam desenvolvidas usando abordagens diferentes, a existência de um sistema heterogêneo é facilitada pelo seguimento dos conceitos comuns propostos pela ontologia.

Dito isto, como poderíamos criar um sistema de integração a partir da presente ontologia? Uma resposta detalhada certamente dependeria de informações específicas relacionadas à tecnologia escolhida. Mas de maneira simplificada, pode-se propor alguns passos comuns, detalhados nos tópicos a seguir.

### **6.1 DETALHAMENTO DA CAMADA DE APLICAÇÃO**

O detalhamento da camada de aplicação significa a modelagem do domínio específico, ou seja, criar os conceitos e relações que são específicos à área que se quer modelar. No estudo de caso 1, a área é relacionada a um fabricante de válvulas industriais, com características construtivas específicas e com uma determinada configuração de cadeia de suprimentos. Aqui, algumas alterações podem ocorrer no sentido de otimização dos diferentes componentes como, por exemplo, a simplificação da cadeia de suprimentos para sua otimização, desde que essa alteração seja refletida na realidade.

É importante, também, criar as instâncias apropriadas, ou seja, a realização dos conceitos modelados. Um exemplo seria a criação de instâncias da válvula VALVE\_ATS, caso elas existam, com as suas informações específicas relacionadas a localização, modelo e outras. Ou seja, ao final desta primeira etapa, toda a realidade existente/conhecida deve estar refletida no detalhamento da ontologia.

## 6.2 MODELAGEM DA ARQUITETURA BASEADA NA ONTOLOGIA

Neste passo, a definição dos elementos passa a ser fortemente baseada na tecnologia escolhida. Uma das escolhas a serem feitas é relacionada à natureza dos conceitos que serão implementados. Por exemplo, o IMS pode ser um dispositivo físico independente, acoplado ao dispositivo que se quer monitorar. Ou pode ser um serviço provido por um sistema lógico, ou seja, a funcionalidade IMS pode ser implementada como um software que é alimentado com as informações coletadas por sensores acoplados ao dispositivo que se quer monitorar.

É importante notar que certas definições podem acarretar inclusões de novos conceitos e relações na ontologia. Além disso, nem todos os conceitos existentes na ontologia, existirão na “vida real” na forma de um elemento tecnológico. Ainda que não seja o método mais eficiente, uma requisição de reparo pode, por exemplo, ser feita via comunicação oral sem que isso seja inválido do ponto de vista da ontologia.

## 6.3 CONSTRUÇÃO DO SISTEMA DE INTEGRAÇÃO

Neste passo se dá a implementação real do sistema de integração. Isto inclui quaisquer etapas necessárias de modelagem usando técnicas de diagrama, atividades de análise de sistemas, condificação, testes e implantação. Um aspecto a ser observado é que não é obrigatória a implementação de todos os conceitos e partes da integração. É possível

implementar uma parte dos conceitos e, à medida que se fizer necessário, implementar as partes seguintes.

## 7 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou uma proposta de ontologia para integração de Sistemas de Manutenção Inteligentes (IMS) e Cadeias de Suprimento de Peças de Reposição (SPSC). A relevância deste trabalho está no estabelecimento de uma base semântica comum para integração de domínios tão diferentes, como ponto de partida para a criação de sistemas de integração.

O uso de ontologias existentes, como MASON, além de conceitos consagrados de mercado, como FMEA, garantem consistência com definições em uso. A criação de dois estudos de caso, o primeiro relacionado ao caso real de um fabricante de válvulas industriais, e o segundo relacionado ao uso de sistemas artificiais e multiagentes para implementação de IMS, além de validar a presente proposta, constituem exemplos do uso da ontologia.

Diversas oportunidades de trabalhos futuros se apresentam, conforme listado abaixo:

- **Especialização da ontologia para domínios específicos:** *o detalhamento do presente trabalho visando o uso em domínios ou usos específicos pode ser interessante. Um exemplo é o próprio estudo de caso 2. Diferentes áreas da manufatura poderiam se beneficiar do estabelecimento de um vocabulário ainda mais especializado.*
- **Ferramentas de geração de código a partir da ontologia:** *o uso da ontologia para geração automática de código poderia simplificar os processos de desenvolvimento, além de garantir a consistência semântica de maneira mais simples.*
- **Utilização da ontologia para análise de cenários da cadeia de suprimentos através de ferramentas específicas:** *a modelagem e análise de diferentes cenários pode ajudar a estabelecer configurações otimizadas e prever comportamentos futuros. Linguagens como SWRL podem ser utilizadas para estabelecer regras relevantes para a simulação de cenários, bem como ferramentas específicas podem ser desenvolvidas para este fim.*

- *Melhorias nas ferramentas de desenvolvimento e visualização de ontologias: as ferramentas atuais de desenvolvimento de ontologias carecem da maturidade encontrada em ferramentas equivalentes para desenvolvimento de software. As ferramentas de visualização gráfica das ontologias baseadas em OWL também se beneficiariam de melhorias, uma vez que nenhum dos plug-ins de visualização encontrados foi capaz de mostrar a ontologia com a riqueza de detalhes requerida.*

## REFERÊNCIAS

- AMERI, F.; URBANOVSKY, C.; MCARTHUR, C. A Systematic Approach to Developing Ontologies for Manufacturing Service Modeling. In: INTERNATIONAL CONFERENCE ON FORMAL ONTOLOGY IN INFORMATION SYSTEMS (FOIS), 7., 2012, Graz. **Proceedings...** Amsterdam: IOS Press, 2012. p. 1-13.
- ASHBURNER, M. et al. Gene Ontology: tool for the unification of biology. **Nature genetics**, New York, v. 25, n. 1, p. 25-29, 2000.
- AZPIREZ, J. et al. (ONTO)2 Agent: an Ontology-based WWW broker to select ontologies. In: WORKSHOP ON APPLICATIONS OF ONTOLOGIES AND PROBLEM-SOLVING METHODS, EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE (ECAI), 13., 1998, Brighton. **Proceedings...** Amsterdam: IOS Press, 1998. p. 16-24.
- BECHHOFER, S. OWL: Web ontology language. **Encyclopedia of Database Systems**, Chicago: Springer, pp. 2008-2009, 2009.
- BERNARAS, A.; LARESGOITI, I.; CORERA, J. Building and Reusing Ontologies for Electrical Network Applications. In: PROCEEDINGS OF THE EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE (ECAI), 12., 1996, Budapest. **Proceedings...** Amsterdam: IOS Press, 1996. p. 298-302.
- BORGO, S.; GUARINO, N.; MASOLO, C. Stratified ontologies: the case of physical objects. In: WORKSHOP ON ONTOLOGICAL ENGINEERING, PROCEEDINGS OF THE EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE (ECAI), 12., 1996, Budapest. **Proceedings...** Amsterdam: IOS Press, 1996. p. 5-16.
- BORGO, S.; LEITÃO, P. Foundations for a Core Ontology of Manufacturing. In: SHARMAN, R.; KISHORE, R., et al (Ed.). **Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems**. New York: Springer, 2007. p. 751-775.
- CHAUDHRI, V. K. et al. **The generic frame protocol 2.0**. Stanford: Knowledge Systems Laboratory, 1997.
- CORCHO, O.; GÓMEZ-PÉREZ, A. A roadmap to ontology specification languages. In: KNOWLEDGE ENGINEERING AND KNOWLEDGE MANAGEMENT METHODS, MODELS, AND TOOLS, 12., 2000, Juan-les-Pins. **Proceedings...** Juan-les-Pins: Berlin: Springer, 2000. p.80-96.
- DITTMANN, L.; RADEMACHER, T.; ZELEWSKI, S. Combining knowledge management and quality management systems. In: EUROPEAN ORGANIZATION FOR QUALITY CONGRESS (EOQ), 48., 2004, Moscow. **Proceedings...** Brussels: EOQ, 2004. p. 1-8.

EBRAHIMIPOUR, V.; REZAIIE, K.; SHOKRAVI, S. An ontology approach to support FMEA studies. **Expert Systems with Applications**, New York, v. 37, n. 1, p. 671-677, 2010.

EMMANOUILIDIS, C. et al. Condition monitoring based on incremental learning and domain ontology for condition-based maintenance. In: INTERNATIONAL CONFERENCE ON ADVANCES IN PRODUCTION MANAGEMENT SYSTEMS (APMS), 11., 2010, Cernobbio. **Proceedings...** Berlin: Springer, 2010. p. 11-13.

ESPINDOLA, D. et al. Integrating Intelligent Maintenance Systems and Spare Parts Supply Chains. In: IFAC SYMPOSIUM ON INFORMATION CONTROL PROBLEMS IN MANUFACTURING INFORMATION CONTROL PROBLEMS, 14., 2012, Bucharest. **Proceedings...** Oxford: IFAC Publications, 2012. p.1017-1022.

FADEL, F. G.; FOX, M. S.; GRUNINGER, M. A generic enterprise resource ontology. In: ENABLING TECHNOLOGIES: INFRASTRUCTURE FOR COLLABORATIVE ENTERPRISES, IEEE INTERNATIONAL WORKSHOPS, 3., 1994, Morgantown. **Proceedings...** Morgantown: IEEE Computer Society Press, 1994. p.117-128.

FARQUHAR, A.; FIKES, R.; RICE, J. The ontolingua server: A tool for collaborative ontology construction. **International journal of human-computer studies**, London, v. 46, n. 6, p. 707-727, 1997.

FAYEZ, M.; RABELO, L.; MOLLAGHASEMI, M. Ontologies for supply chain simulation modeling. In: CONFERENCE ON WINTER SIMULATION, 37., 2005, Orlando. **Proceedings...** Orlando: ACM, 2005. p. 2364-2370.

FUMAGALLI, L. et al. Diagnosis for improved maintenance services: Analysis of standards related to Condition Based Maintenance. In: KIRITSIS, D. et al. (Ed.). **Engineering Asset Lifecycle Management**. London: Springer, 2010. p.288-297.

GAMMA, E. et al. **Design patterns: elements of reusable object-oriented software**. Reading: Addison Wesley Publishing Company, 1995.

GARETTI, M.; FUMAGALLI, L. P-PSO Ontology for Manufacturing Systems. **Information Control Problems in Manufacturing**, Oxford, v. 14, n. 1, p. 449-456, 2012.

GÓMEZ-PÉREZ, A.; BENJAMINS, R. Overview of knowledge sharing and reuse components: Ontologies and problem-solving methods. In: IJCAI WORKSHOP ON ONTOLOGIES AND PROBLEM-SOLVING METHODS (KRR5), 1999, Stockholm. **Proceedings...** Amsterdam: CEUR, 1999. p. 1-15.

GRUBER, T. R. A translation approach to portable ontology specifications. **Knowledge acquisition**, London, v. 5, n. 2, p. 199-220, 1993.

GRUBER, T. R. Toward principles for the design of ontologies used for knowledge sharing. **International journal of human-computer studies**, London, v. 43, n. 5, p. 907-928, 1995.

GRUBIC, T.; FAN, I. S. Supply chain ontology: Review, analysis and synthesis. **Computers in Industry**, London, v. 61, n. 8, p. 776-786, 2010.

GUIZZARDI, G. **Ontological foundations for structural conceptual models**. London: Centre for Telematics and Information Technology, 2005.

GUIZZARDI, G.; HALPIN, T. Ontological foundations for conceptual modelling. **Applied Ontology**, Amsterdam, v. 3, n. 1, p. 1-12, 2008.

HELLINGRATH, B.; CORDES, A. K. Approach for integrating condition monitoring information and forecasting methods to enhance spare parts supply chain planning. In: IFAC WORKSHOP ON INTELLIGENT MANUFACTURING SYSTEMS, 11., 2013, São Paulo. **Proceedings...** London: IFAC, 2013. p. 17-22.

HOLSAPPLE, C. W.; JOSHI, K. D. A collaborative approach to ontology design. **Communications of the ACM**, New York, v. 45, n. 2, p. 42-47, 2002.

HOMPEL, M.; HELLINGRATH, B. **Effizienz - Verantwortung - Erfolg**. Berlin: T. Wimmer, 2007.

HORRIDGE, M. et al. **Protégé-OWL tutorial**. Stanford: Stanford, 2011.

JIAN, Y.; JIANYUAN, Y. Towards a hierarchical supply chain ontology. In: INTERNATIONAL CONFERENCE ON SERVICE SYSTEMS AND SERVICE MANAGEMENT (ICSSSM), 8., 2011, Tianjin. **Proceedings...** New York: IEEE, 2011. p.1-6.

KIFER, M.; LAUSEN, G.; WU, J. Logical foundations of object-oriented and frame-based languages. **Journal of the ACM (JACM)**, New York, v. 42, n. 4, p. 741-843, 1995.

KOJI, Y.; KITAMURA, Y.; MIZOGUCHI, R. Ontology-based transformation from an extended functional model to FMEA. In: INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN (ICED), 15., Melbourne, 2005. **Proceedings...** Bristol: The Design Society, 2005. p. 264.81.

LANGFORD, J. W. **Logistics: principles and applications**. New York: McGraw-Hill, 1995.

LEE, B. H. Using FMEA models and ontologies to build diagnostic models. **AI EDAM**, London, v. 15, n. 4, p. 281-293, 2001.

LEMAIGNAN, S. et al. MASON: A proposal for an ontology of manufacturing domain. In: IEEE WORKSHOP ON DISTRIBUTED INTELLIGENT SYSTEMS: COLLECTIVE INTELLIGENCE AND ITS APPLICATIONS (DIS), 2006, Prague. **Proceedings...** Los Alamitos: IEEE, 2006. p.195-200.

LUCIANO, J. S. et al. The Translational Medicine Ontology and Knowledge Base: driving personalized medicine by bridging the gap between bench and bedside. **Journal Biomed Semantics**, London, v. 2, n. Suppl 2, p. S1, 2011.

MACGREGOR, R. M. Inside the LOOM description classifier. **ACM Sigart Bulletin**, New York, v. 2, n. 3, p. 88-92, 1991.

MCGUINNESS, D. L.; VAN HARMELEN, F. OWL web ontology language overview. **W3C recommendation**, Cambridge, v. 10, n. 2004-03, p. 10, 2004.

MIZOGUCHI, R. Part 1: Introduction to ontological engineering. **New Generation Computing**, Tokyo, v. 21, n. 4, p. 365-384, 2003.

MOTTA, E. **Reusable components for knowledge modelling**: Case studies in parametric design problem solving. Amsterdam: IOS press, 1999.

NOY, N. F.; MCGUINNESS, D. L. Ontology development 101: A guide to creating your first ontology. In: STANFORD KNOWLEDGE SYSTEMS LABORATORY. **Technical report KSL-01-05 and Stanford medical informatics technical report SMI-2001-0880**. Stanford, 2001. p. 234-252.

POUCHARD, L.; IVEZIC, N.; SCHLENOFF, C. Ontology engineering for distributed collaboration in manufacturing. In: AIS2000 CONFERENCE, 2000, Long Beach. **Proceedings...** Long Beach: Citeseer, 2000. p. 1-6.

SMIRNOV, A.; CHANDRA, C. Ontology-based knowledge management for co-operative supply chain configuration. IN: AAAI SPRING SYMPOSIUM "BRINGING KNOWLEDGE TO BUSINESS PROCESSES", 2000, Stanford. **Proceedings...** Stanford: AAAI Press, 2000. p.85-92.

STADTLER, H.; KILGER, C. **Supply chain management and advanced planning: concepts, models, software, and case studies**. Berlin: Springer, 2008.

STANFORD. **Protégé**. Stanford Center for Biomedical Informatics Research (BMIR) at the Stanford University School of Medicine. Disponível em: <http://protege.stanford.edu>. Acesso em 23/9/2013.

SYNTETOS, A. A.; KEYES, M.; BABAI, M. Demand categorisation in a European spare parts logistics network. **International Journal of Operations & Production Management**, Bingley, v. 29, n. 3, p. 292-316, 2009.

USCHOLD, M. et al. The enterprise ontology. **The knowledge engineering review**, Cambridge, v. 13, n. 01, p. 31-89, 1998.

WAND, Y. et al. Theoretical foundations for conceptual modelling in information systems development. **Decision Support Systems**, Amsterdam, v. 15, n. 4, p. 285-304, 1995.

WEBER, R. Conceptual modelling and ontology: Possibilities and pitfalls. **Journal of Database Management (JDM)**, Missouri, v. 14, n. 3, p. 1-20, 2003.

WUSSUSEK, B. On ontological foundations of conceptual modelling. **Scandinavian Journal of Information Systems**, Oslo, v. 18, n. 1, p. 63, 2006.

YE, Y. et al. An ontology-based architecture for implementing semantic integration of supply chain management. **International Journal of Computer Integrated Manufacturing**, Abgindon, v. 21, n. 1, p. 1-18, 2008.

ZDRAVKOVIĆ, M. et al. An approach for formalising the supply chain operations. **Enterprise Information Systems**, Abgindon, v. 5, n. 4, p. 401-421, 2011.

ZHAO, X.; ZHU, Y. Research of fmea knowledge sharing method based on ontology and the application in manufacturing process. In: 2ND INTERNATIONAL WORKSHOP ON DATABASE TECHNOLOGY AND APPLICATIONS (DBTA), 2010, Wuhan. **Proceedings...** Wuhan: IEEE, 2010. p. 1-4.

ZHOU, J.; DIENG-KUNTZ, R. Manufacturing ontology analysis and design: towards excellent manufacturing. In: 2ND IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL INFORMATICS, 2004, Berlin. **Proceedings...** Berlin: IEEE, 2004. p. 39-45.

ZUCCOLOTTO, M. et al. A Distributed Intelligent Maintenance Approach Based on Artificial Immune Systems. In: TSE, P. W. et al. **Engineering Asset Management - Systems, Professional Practices and Certification**. Hong Kong: Springer, 2015. cap. 83, p.969-981.

## ANEXO:

Dados utilizados para montagem do estudo de caso.

## ANEXO: DADOS UTILIZADOS PARA A MONTAGEM DO ESTUDO DE CASO

### RESUMO DE OCORRÊNCIAS

**Tabela 4 Resumo de Ocorrências**

| <b>OCORRÊNCIA</b>               | <b>QTD</b> | <b>%</b> |
|---------------------------------|------------|----------|
| CPU/PAINEL                      | 23         | 11,2     |
| FORNECEDOR                      | 9          | 4,4      |
| I/O DIGITAL                     | 1          | 0,5      |
| I/O ANALOGICO                   | 0          | 0,0      |
| CARTÃO PROFIBUS DP              | 7          | 3,4      |
| CARTÃO MODBUS RTU               | 8          | 3,9      |
| CARTÃO NET INTERFACE            | 0          | 0,0      |
| CARTÃO FLEX LC                  | 0          | 0,0      |
| CARTÃO FOUNDATION FIELDBUS      | 0          | 0,0      |
| CARTÃO HART                     | 0          | 0,0      |
| OUTTROS CARTÕES DE REDE         | 0          | 0,0      |
| PLACA DE COMANDO BRUSHLESS      | 9          | 4,4      |
| POTENCIÔMETRO                   | 11         | 5,3      |
| CÉLULA DE CARGA                 | 10         | 4,9      |
| TRANSFORMADOR                   | 3          | 1,5      |
| CONTADORAS                      | 1          | 0,5      |
| MOTOR                           | 2          | 1,0      |
| BARREIRA EXI                    | 16         | 7,8      |
| MONITOR I2C                     | 0          | 0,0      |
| CABOS SATA                      | 1          | 0,5      |
| CABOS DIVERSOS                  | 12         | 5,8      |
| MAU CONTATO                     | 3          | 1,5      |
| FIRMWARE                        | 52         | 25,2     |
| FUSÍVEIS                        | 1          | 0,5      |
| INVERSOR                        | 0          | 0,0      |
| PARAMETRIZAÇÃO                  | 0          | 0,0      |
| CONJUNTO DA MANGA               | 0          | 0,0      |
| SEM FIM                         | 0          | 0,0      |
| SEM FIM DO INDICADOR DE POSIÇÃO | 0          | 0,0      |
| CONJUNTO DO PAINEL              | 2          | 1,0      |
| CONJUNTO DO VOLANTE             | 2          | 1,0      |
| KIT VEDAÇÃO                     | 0          | 0,0      |
| VISORES                         | 0          | 0,0      |
| ROLAMENTOS                      | 0          | 0,0      |
| FALHA POR EXCESSO DE VIBRAÇÃO   | 3          | 1,5      |
| FALHA POR ENTRADA DE ÁGUA       | 12         | 5,8      |
| CALIBRAÇÃO                      | 2          | 1,0      |
| FALHA DE MONTAGEM               | 1          | 0,5      |
| FALHA MECÂNICA                  | 6          | 2,9      |
| CONJUNTO DO POTENCIÔMETRO       | 9          | 4,4      |
| SEM FIM/FUSO DO REDUTOR         | 0          | 0,0      |
| CORO/MANGA DO REDUTOR           | 0          | 0,0      |
| PORCA/SOQUETE DO REDUTOR        | 0          | 0,0      |
| PORCA DO REDUTOR                | 0          | 0,0      |

|              |            |              |
|--------------|------------|--------------|
| <b>TOTAL</b> | <b>206</b> | <b>100,0</b> |
|--------------|------------|--------------|

**Tabela 5 Resumo por Modelo (referência real omitida)**

| <b>MODELO</b> | <b>QT.</b> | <b>%</b>      |
|---------------|------------|---------------|
| M1            | 68         | 33,01         |
| M2            | 37         | 17,96         |
| M3            | 15         | 7,28          |
| M4            | 21         | 10,19         |
| M5            | 17         | 8,25          |
| M6            | 15         | 7,28          |
| M7            | 0          | 0,00          |
| M8            | 18         | 8,74          |
| M9            | 3          | 1,46          |
| M10           | 2          | 0,97          |
| M11           | 0          | 0,00          |
| M12           | 5          | 2,43          |
| M13           | 5          | 2,43          |
| M14           | 0          | 0,00          |
| M15           | 0          | 0,00          |
| M16           | 0          | 0,00          |
| M17           | 0          | 0,00          |
| M18           | 0          | 0,00          |
| M19           | 0          | 0,00          |
| M20           | 0          | 0,00          |
| M21           | 0          | 0,00          |
| M22           | 0          | 0,00          |
| <b>TOTAL</b>  | <b>206</b> | <b>100,00</b> |