



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA E ENGENHARIA
DE PETRÓLEO



Contribuição ao desenvolvimento de ontologias para processos petroquímicos: estudo de caso em uma planta DEA

Anthony Andrey Ramalho Diniz

Orientador: Prof. Dr. Jorge Dantas de Melo

Co-orientador: Prof. Dr. Adrião Duarte Dória Neto

Natal, Setembro de 2010

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA E ENGENHARIA
DE PETRÓLEO



Contribuição ao desenvolvimento de ontologias para processos petroquímicos: estudo de caso em uma planta DEA

Anthony Andrey Ramalho Diniz

Orientador: Prof. Dr. Jorge Dantas de Melo

Co-orientador: Prof. Dr. Adrião Duarte Dória Neto

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência e Engenharia de Petróleo da UFRN (área de concentração: Automação na Indústria de Petróleo e Gás Natural) como parte dos requisitos para obtenção do título de Mestre em Ciências.

Natal, Setembro de 2010

Divisão de Serviços Técnicos
Catalogação da Publicação na Fonte. UFRN / Biblioteca Central Zila Mamede

Diniz, Anthony Andrey Ramalho.

Contribuição ao desenvolvimento de ontologias para processos petroquímicos : estudo de caso em uma planta DEA / Anthony Andrey Ramalho Diniz. – Natal, RN, 2010.

92 f. : il.

Orientador: Jorge Dantas de Melo.

Co-orientador: Adrião Duarte Dória Neto.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Norte. Centro de Tecnologia. Programa de Pós-Graduação em Ciência e Engenharia do Petróleo.

1. Processos petroquímicos – Dissertação. 2. Adoçamento de gás combustível – Dissertação. 3. Planta DEA – Dissertação. 4. Engenharia ontológica – Dissertação. I. Melo, Jorge Dantas de. II. Dória Neto, Adrião Duarte. III. Universidade Federal do Rio Grande do Norte. IV. Título.

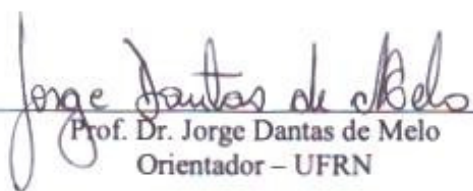
RN/UF/BCZM

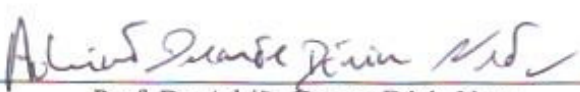
CDU 665.6/.7(043.2)

Contribuição ao desenvolvimento de ontologias para processos petroquímicos: estudo de caso em uma planta DEA


Anthony Andrey Ramalho Diniz

Dissertação de Mestrado aprovada em 03 de Setembro de 2010, pela banca examinadora composta pelos seguintes membros:


Prof. Dr. Jorge Dantas de Melo
Orientador – UFRN


Prof. Dr. Adrião Duarte Dória Neto
Coorientador – UFRN


Prof. Dr. Luiz Affonso Henderson G. de Oliveira
Examinador interno – UFRN


Dr. Vicente Delgado Moreira
Examinador externo – PETROBRAS

*Dedico este trabalho aos meus grandes amigos,
porque foram muito importantes nesta fase
completamente inusitada da minha vida.*

Agradecimentos

Primeiramente a Deus, pelo dom da vida e a saúde para seguir caminhando.

À minha família, pelo apoio.

Aos meus amigos Esdras Matheus e Renata Paes, porque eles me colocaram nessa trilha no momento em que eu precisava escolher novos caminhos.

Àos amigos Eliane Bandeira, Priscila Santos, Erika Mônico, Maurício Mônico e Gabriella Mônico, que estão sempre acompanhando meus passos, me dando forças e torcendo por mim.

Aos professores Jorge Dantas e Adrião Duarte, por terem me acolhido e me orientado durante esta caminhada.

Ao professor Affonso Henderson, por ter me acolhido no projeto que me subsidiou de informações e viabilizou financeiramente esta dissertação.

Ao engenheiro Kaku Saito, por ter fornecido as informações do processo DEA e dirimido as dúvidas surgidas no decorrer da caminhada.

Aos demais professores do PPGCEP, pelos conhecimentos que agregaram à nossa formação.

Aos funcionários do PPGCEP, Viviane Medeiros e Carlos Macêdo, que nos ajudaram sempre que buscamos informações.

Aos amigos que fiz no PPGCEP, em especial a Aneliése Lunguinho, Albérico Canário, Dayse da Mata, Emanuel Gomes, Glauco Braga, Igor Aires, Larissa Araújo, Luciana Frazão, Zildiany Ibiapina, que escolheram seguir o mesmo caminho, compartilhando alegrias, tristezas e realizações.

Aos amigos Armando Lemos, Judson Carlos e Sérgio Massao, que foram grandes parceiros nas empreitadas acadêmicas durante este Mestrado e nos churrascos.

Aos amigos Allan Garcia, Alan Paulo, Ana Maria, André Freitas, Carlos Padilha, Cicilia Maia, Danilo Ichihara, Fabiana Santana, Gláucia Regina, Heliana Bezerra, João Paulo, Keylly Eyglis, Luiz Paulo, Mademerson Leandro, Naiyan Lima e Robinson Alves, que estiveram comigo no laboratório, me fizeram sorrir e deram forças sempre que eu precisei.

Aos amigos Helton Maia, Orlando Almeida e Rafael Marrocos, que me acolheram no apartamento onde eu encontrei paz para realizar meus trabalhos, em especial a Moisés Dantas, que também é um grande filósofo do cotidiano.

Ao meu amigo Luciano Fábio, que me deu apoio aqui em Natal, não deixando que meus finais de semana fossem monótonos.

Ao Dr. Vicente Delgado Moreira, pelas contribuições apresentadas durante a defesa, que ajudaram a enriquecer o conteúdo desta dissertação.

A todas as pessoas que duvidam de meu potencial, porque a dúvida delas me fortalece.

Resumo

Nas últimas décadas, o segmento de óleo & gás e petroquímica tem registrado uma série de grandes acidentes. Influenciadas por esse contexto, as empresas tem sentido a necessidade de se engajar em processos de proteção do ambiente externo, que se traduz na preocupação ecológica. No caso particular da indústria nuclear mundial, a educação sustentável e o treinamento, que dependem muito da qualidade e utilidade da base de conhecimento, têm sido considerados pontos chave para utilização desse tipo de energia com segurança. Dessa forma, a motivação dessa pesquisa foi aplicar o conceito de ontologia como ferramenta para melhorar a gestão do conhecimento em uma refinaria, através da representação de uma planta de adoçamento de gás combustível, condensando os vários tipos informações associados com o seu modo de operação normal. Em termos de metodologia, este estudo pode ser classificado como uma pesquisa aplicada e descritiva, em que foram analisadas, classificadas e interpretadas informações que possibilitaram criar a ontologia descritiva de uma planta real. A modelagem da planta DEA foi realizada de acordo com os fluxogramas de processo, fluxogramas de tubulação e instrumentação, documentos descritivos de seu modo de operação e a relação de alarmes associados, que foram complementadas com uma entrevista não estruturada de um especialista em seu modo de operação. A validação aconteceu através da comparação de grafos montados a partir da ontologia com a documentação original e debatidos com o grupo de trabalho. Os conceitos utilizados nesta pesquisa podem ser expandidos para representar outras plantas da própria refinaria ou mesmo de outras indústrias. A ontologia pode ser considerada uma base de conhecimento, que devido ao seu caráter formal, pode ser aplicada como um dos elementos no desenvolvimento de ferramentas de navegação da planta, simulação de comportamento, diagnóstico de falhas, dentre outras possibilidades.

Palavras-chave: Representação de conhecimento, engenharia ontológica, Protégé, adoçamento de gás combustível.

Abstract

In the last decades, the oil, gas and petrochemical industries have registered a series of huge accidents. Influenced by this context, companies have felt the necessity of engaging themselves in processes to protect the external environment, which can be understood as an ecological concern. In the particular case of the nuclear industry, sustainable education and training, which depend too much on the quality and applicability of the knowledge base, have been considered key points on the safely application of this energy source. As a consequence, this research was motivated by the use of the ontology concept as a tool to improve the knowledge management in a refinery, through the representation of a fuel gas sweetening plant, mixing many pieces of information associated with its normal operation mode. In terms of methodology, this research can be classified as an applied and descriptive research, where many pieces of information were analysed, classified and interpreted to create the ontology of a real plant. The DEA plant modeling was performed according to its process flow diagram, piping and instrumentation diagrams, descriptive documents of its normal operation mode, and the list of all the alarms associated to the instruments, which were complemented by a non-structured interview with a specialist in that plant operation. The ontology was verified by comparing its descriptive diagrams with the original plant documents and discussing with other members of the researchers group. All the concepts applied in this research can be expanded to represent other plants in the same refinery or even in other kind of industry. An ontology can be considered a knowledge base that, because of its formal representation nature, can be applied as one of the elements to develop tools to navigate through the plant, simulate its behavior, diagnose faults, among other possibilities.

Keywords: Knowledge representation, ontological engineering, Protégé, gas sweetening.

Sumário

| | |
|--|------------|
| Lista de Figuras..... | iv |
| Lista de Tabelas | vi |
| Lista de Símbolos e Abreviaturas..... | vii |
| 1. Introdução Geral..... | 1 |
| 1.1 Introdução | 2 |
| 1.2 Apresentação do problema | 3 |
| 1.3 Objetivos..... | 4 |
| 1.4 Metodologia..... | 5 |
| 1.5 Organização do texto..... | 6 |
| 2. Representação do Conhecimento Usando Ontologias..... | 8 |
| 2.1 Introdução | 9 |
| 2.2 O conhecimento e sua representação..... | 9 |
| 2.3 Representação do conhecimento através das ontologias | 12 |
| 2.3.1 Características e tipos de ontologias..... | 13 |
| 2.3.2 Linguagens para criação de ontologias..... | 18 |
| 2.3.2.1 A linguagem OWL | 20 |
| 2.3.2.2 O editor de ontologias Protégé | 24 |
| 2.3.2.3 Metodologias para construção das ontologias..... | 26 |
| 3. Planta DEA..... | 33 |
| 3.1 Introdução | 34 |
| 3.2 REPLAN – Refinaria do Planalto Paulista | 35 |
| 3.3 Descrição da planta DEA – adoçamento de gás combustível..... | 37 |
| 3.4 A planta DEA da REPLAN | 41 |
| 3.5 Análise da planta DEA - conclusão..... | 43 |

| | |
|---|-----------|
| 4. Ontologia do Processo DEA..... | 44 |
| 4.1 Introdução | 45 |
| 4.2 Representação da planta DEA através de ontologia..... | 45 |
| 4.2.1 Determinação do escopo | 45 |
| 4.2.2 Consideração de reuso | 47 |
| 4.2.3 Enumeração dos termos | 47 |
| 4.2.4 Definição das classes e taxonomia | 48 |
| 4.2.5 Definição das propriedades | 50 |
| 4.2.6 Definição das facetas | 53 |
| 4.2.7 Definição das instâncias..... | 54 |
| 4.2.8 Checagem de inconsistências..... | 57 |
| 4.3 Desenvolvimento da ontologia - conclusão..... | 58 |
| | |
| 5. Análise de Resultados | 60 |
| 5.1 Introdução | 61 |
| 5.2 O <i>plugin</i> Jambalaya..... | 61 |
| 5.3 Representação do fluxo dos produtos..... | 63 |
| 5.3.1 Representação do fluxo de gás combustível com sulfeto de hidrogênio | 66 |
| 5.3.2 Representação do fluxo de gás combustível | 66 |
| 5.3.3 Representação do fluxo de DEA rica com gás combustível | 67 |
| 5.3.4 Representação do fluxo de DEA rica..... | 69 |
| 5.3.5 Representação do fluxo de vapor | 70 |
| 5.3.6 Representação do fluxo de sulfeto de hidrogênio com DEA rica | 71 |
| 5.3.7 Representação do fluxo de sulfeto de hidrogênio..... | 72 |
| 5.3.8 Representação do fluxo de DEA pobre..... | 72 |
| 5.4 Representação dos pontos de conexão de instrumentos | 73 |
| 5.4.1 Representação dos pontos de conexão dos sensores | 74 |
| 5.4.2 Representação dos elementos de controle..... | 76 |
| 5.5 Indicação dos alarmes..... | 77 |
| 5.6 Análise da ontologia - conclusão | 78 |
| | |
| 6. Conclusões e Sugestões de Trabalhos Futuros..... | 80 |
| 6.1 Conclusão | 81 |
| 6.2 Sugestão de trabalhos futuros | 83 |

| | |
|---|-----------|
| Referências Bibliográficas | 84 |
|---|-----------|

Lista de Figuras

| | |
|---|----|
| 2.1. Tipos de ontologias, de acordo com o nível de dependência em uma tarefa particular ou ponto de vista, onde as setas representam uma especialização de relações | 15 |
| 2.2. Sublinguagens OWL | 22 |
| 2.3. Trecho de código em OWL | 22 |
| 2.4. Primitivas da representação do conhecimento através do OWL-Lite e OWL-DL | 23 |
| 2.5. Tela principal do Protégé para edição de ontologias em OWL | 26 |
| | |
| 3.1. Esquema de refino composto por uma unidade de destilação | 35 |
| 3.2. Esquema de refino orientado para a produção de combustíveis e aromáticos | 35 |
| 3.3. REPLAN – Refinaria do Planalto Paulista | 36 |
| 3.4. Esquema simplificado da REPLAN | 37 |
| 3.5. Reação de absorção de H ₂ S por dietanolamina | 38 |
| 3.6. Esquema do tratamento do GLP e do gás combustível com DEA | 39 |
| | |
| 4.1. Simplificação da planta através da ilustração dos principais fluxos considerados | 46 |
| 4.2. Classes e taxonomia definidas para a ontologia | 51 |
| 4.3. Relação de todas as propriedades diretamente no Protégé | 56 |
| 4.4. Trecho da planta DEA em que está situado o controlador LIC-98851 | 57 |
| 4.5. Procedimento de cadastramento de instâncias na classe “Controlador” | 58 |
| 4.6a. Tela do Protégé confirmando que a ontologia não possui inconsistências | 59 |
| 4.6b. Tela indicando a existência de inconsistências | 59 |
| | |
| 5.1. Visualização gerada no plugin Jambalaya de todos os elementos de tubulação e processos da planta DEA, inclusive sensores conectados | 62 |
| 5.2. Menu de geração de visualizações no <i>plugin</i> Jambalaya | 63 |
| 5.3. Visualização de instâncias cujo nome inicia com um número | 64 |
| 5.4. Representação da ligação entre todos os elementos da ontologia presentes na planta DEA, indicando o fluxo geral de todos os produtos | 65 |
| 5.5. Representação do trecho em que fluem gás combustível e sulfeto de hidrogênio juntos. | 66 |

| | |
|---|----|
| 5.6. Representação dos trechos em que flui apenas gás combustível | 67 |
| 5.7. Representação do trecho da planta em que flui DEA rica com gás combustível | 68 |
| 5.8. Representação do trecho da planta em que flui apenas DEA rica | 69 |
| 5.9. Representação do trecho da planta em que flui apenas vapor | 70 |
| 5.10. Representação do trecho da planta em que fluem sulfeto de hidrogênio e DEA rica | 71 |
| 5.11. Representação do trecho da planta em que flui apenas sulfeto de hidrogênio | 72 |
| 5.12. Representação dos trechos da planta em que flui apenas DEA pobre | 74 |
| 5.13. Representação dos sensores no trecho de fluxo da DEA rica | 75 |
| 5.14. Representação dos elementos de controle no trecho de fluxo da DEA rica | 77 |
| 5.15. Representação dos elementos de controle no trecho de fluxo da DEA rica, inclusive dos alarmes associados aos sensores e controladores | 78 |

Lista de Tabelas

| | |
|---|----|
| 2.1a. Linguagens para a construção de ontologias | 18 |
| 2.1b. Linguagens para a construção de ontologias | 19 |
| 2.2. Metodologia de desenvolvimento de ontologias “ <i>ontology development 101</i> ” | 28 |
| 2.3. Metodologias para construção de ontologias | 32 |
| | |
| 3.1. Perfil de produção da REPLAN no final da década de 1990 | 36 |
| | |
| 4.1. Relação de propriedades básicas definidas para a ontologia | 53 |
| 4.2. Relação de subpropriedades definidas para a propriedade <i>hasOutput</i> | 54 |
| 4.3. Relação das propriedades definidas para a ontologia e suas respectivas características relacionais | 55 |

Lista de Símbolos e Abreviaturas

API: Application Programming Interface

CC: Ciência da Computação

CNPQ: Conselho Nacional de Desenvolvimento Científico e Tecnológico

DEA: Dietanolamina

DL: *Description logics*

FCC: Craqueamento Catalítico Fluido

GLP: Gás Liquefeito de Petróleo

HAZOP: *Hazard and Operability Study*

HDT: Hidrotratamento

IA: Inteligência Artificial

MEROX: *Mercaptans Oxidation*

MTBE: Metil-Terc-Butil-Éter

OKBC: *Open Knowledge Base Connectivity*

OWL: *Web Ontology Language*

P&I: *Piping and instrumentation*

QAV: Querosene de Aviação

RC: Representação do Conhecimento

RDF: *Resource Description Framework*

RDFS ou RDF-S: RDF Schema

RDF(S): Combinação de RDF e RDFS

SBC: Sistema Baseado em Conhecimento

XML: *Extensible Markup Language*

W3C: *World Wide Web Consortium*

Capítulo 1

Introdução Geral

1.1 Introdução

Nas últimas décadas, o segmento de óleo, gás e petroquímica tem registrado uma série de grandes acidentes, dentre os quais estão o de San Juan Ixhuapetec (México-1984), Bophal (Índia-1984), plataforma Piper-Alpha (Mar do Norte-1988), Phillips 66 (Texas-1989), Exxon-Valdez (Alasca-1989), Baía de Guanabara (Brasil-2000), plataforma P-36 (Brasil-2001) (THEOBALD; GILSON, 2005). Ainda este ano, ocorreu um acidente no Golfo do México, protagonizado pela British Petroleum, que tem permanecido na mídia devido à dificuldade de ser sanado.

Através dos registros disponíveis na literatura, constata-se que as investigações de acidentes realizadas pelas empresas ou por organismos independentes apontam a “falha humana” com uma das principais causas de acidentes (THEOBALD; GILSON, 2005).

O próprio fato do ser humano estar no centro do processo de trabalho, torna o erro uma possibilidade em todas as atividades em que ele atua, desde o projeto e fabricação de equipamentos, até o desenvolvimento de novas tecnologias, novos métodos de trabalho, normas técnicas, programas de segurança, etc. (BARBOSA, 2009).

Uma questão que tem obtido destaque é a crescente necessidade das empresas se engajarem em processos de proteção do ambiente externo, que se traduz na preocupação ecológica. Esse fato tem obrigado as organizações a desenvolver uma nova cultura para a inovação e aprendizagem, fazendo-as rever suas estruturas de práticas de organização do trabalho, através de interações contínuas e recursivas entre as dimensões do homem, empresa e novos conhecimentos adquiridos durante o processo (PEREIRA; DONDONI, 2008).

Em segmentos que exploram recursos ambientais, tal qual a indústria petrolífera, a dinâmica do processo de aprendizagem organizacional vem se tornando de fundamental relevância para a existência sustentada, uma vez que o processo de transformação de recursos naturais, como é a operação de refino, envolve alto risco ao meio ambiente (PEREIRA; DONDONI, 2008).

Laallam e Sellami (2007) afirmam que a atividade de supervisão e controle de processos industriais é uma atividade muito complexa. Ela requer muita experiência, que tanto pode ser perdida quando um funcionário se desliga, como pode ser mantida dispersa entre vários especialistas. Em uma refinaria, a supervisão e controle de seus vários processos constituintes é uma atividade crítica para a segurança, pois a ocorrência de um sinistro pode resultar em grave impacto ambiental para a vida presente na localidade em que ela está instalada.

A indústria nuclear mundial tem um histórico de alta segurança, tendo demonstrado um aperfeiçoamento substancial nas últimas décadas. Segundo Taniguchi e Lederman (2004), a chave para a utilização da energia nuclear com segurança é a educação sustentável e o treinamento, e depende muito da qualidade e utilidade da base de conhecimento.

Para Meireles (2009),

A preocupação com o conhecimento não é algo novo. Platão, Sócrates, dentre outros já se preocupavam com este tema. Porém as abordagens sobre gestão do conhecimento nas organizações iniciam-se na década de 1990, pois se percebeu a importância de se desenvolver ferramentas para tornar explícito e disponível o conhecimento existente na empresa bem como a importância da troca e compartilhamento do conhecimento das pessoas.

Motivando-se por esse contexto, esta pesquisa aplicou o conceito de ontologias como ferramenta de gestão do conhecimento, formalizando o conhecimento disponibilizado acerca de uma planta petroquímica, que pode servir como base de conhecimento para o desenvolvimento de ferramentas utilizadas para aperfeiçoar a segurança da planta analisada.

1.2 Apresentação do problema

Quando adquire uma planta industrial, o cliente recebe vários documentos técnicos, dentre os quais estão os memoriais descritivos, fluxograma de processo, fluxograma de engenharia (*P&I* - tubulação e instrumentação), especificações técnicas, folhas de dados, listas de equipamentos, plantas e isométricos das instalações, etc. (FREIRE, 2009).

Em termos de conhecimento sobre os processos industriais, as duas principais fontes acabam sendo o pessoal da planta e os dados operacionais, com suporte de fontes secundárias tais como os documentos listados anteriormente, os procedimentos documentados e as teorias de operação (BROWNE *et al.*, 2005).

No caso das fontes secundárias, os dados sobre as plantas normalmente ficam dispersos em muitos documentos, esquemáticos e diagramas, que nem sempre são de fácil interpretação por operadores iniciantes ou por pessoas que não trabalham no dia a dia desses sistemas. Quando se toma como referência a planta DEA, é possível verificar que existem vários caminhos entre os processos (torres, bombas, filtros, vasos, etc.), mas apenas consultando os descritivos é possível deduzir que alguns desses trechos são utilizados durante o modo de operação normal e outros apenas em situações especiais.

Ao serem designados para trabalhar em processos industriais, é comum que os colaboradores recebam treinamentos para capacitá-los a desempenharem suas atribuições e

permissão para acessar os documentos das instalações, mas geralmente leva um certo tempo para que eles atinjam a maturidade necessária ao manuseio desses materiais com habilidade.

Por outro lado, devido ao formato dessas informações, mesmo elas estando normalmente digitalizadas, o desenvolvimento de sistemas que consigam utilizá-las em seu formato original não é uma tarefa simples.

A proposta desta dissertação foi aplicar as ontologias como uma forma de suprir essas lacunas, motivados pelo fato de que, segundo Freitas (2003, p.4), “as ontologias servem como ferramenta para organização, reuso e disseminação de conhecimento já especificado”, tendo tornado possível, neste caso, condensar a informação disponibilizada em vários documentos acerca da planta analisada, formalizando o conhecimento sobre este domínio de interesse.

Esta dissertação faz parte do conjunto de iniciativas do projeto de um “sistema inteligente para suporte à decisão para análise e operação de plantas industriais”, atendendo ao objetivo de servir como uma base de conhecimento que descreve a operação do sistema, integrando informações de alarmes e variáveis contínuas e discretas do processo, que poderá ser utilizada para o desenvolvimento de um sistema de auxílio ao diagnóstico e à tomada de decisão para o processo petroquímico em estudo.

A escolha do processo DEA como foco do estudo ocorreu porque a Petrobras informou que haveria maior facilidade em disponibilizar informações sobre essa planta, o que facilitou o andamento dos trabalhos.

1.3 Objetivos

Foram os objetivos desta dissertação:

- Aplicar os conceitos de representação do conhecimento através de ontologias ao processo de adoçamento de gás combustível de uma refinaria;
- Representar na ontologia os principais fluxos de produtos da planta;
- Contemplar a representação da localização dos instrumentos de monitoramento e controle na ontologia;
- Incluir na ontologia a associação dos alarmes indicados para os instrumentos da planta;
- Construir automaticamente grafos que propiciem avaliar a representatividade da ontologia.

Como produto desta pesquisa, foi disponibilizada uma descrição formal (ontologia) da planta DEA analisada, que sintetiza tanto as informações dos fluxogramas de engenharia

como das listas de alarmes da planta tomada como estudo de caso. Os conceitos apresentados nesta dissertação podem ser expandidos para a descrição de outras plantas da refinaria, ou mesmo aplicados em outros segmentos industriais.

Em termos de contribuições, essa ontologia pode ser utilizada para o desenvolvimento de sistemas baseados em conhecimento (SBCs), pois através desse modelo formal é possível, por exemplo, conhecer os elementos da planta que estão conectados, os instrumentos instalados, bem como os alarmes associados com cada grandeza. O modelo também indica a forma como o fluxo de produtos acontece no processo e ainda os trechos em que circulam na planta. Esse conjunto de informações pode ser utilizado em um sistema de diagnóstico, ou mesmo para auxiliar os operadores em análises da planta, uma vez que condensam um conjunto de dados importantes do sistema, o que pode contribuir para o aumento do grau de segurança dessa indústria.

Em face dos graves acidentes que a indústria petrolífera tem enfrentado nos últimos anos, qualquer ferramenta que possa aumentar esse grau de segurança na atividade trará benefícios para toda a sociedade, levando em conta a importância econômica que o petróleo tem para o país e o impacto ambiental que um acidente pode provocar.

Para a UFRN e a Petrobras, os frutos dessa pesquisa servem para intensificar os laços de parceria mantidos por essas instituições.

1.4 Metodologia

Segundo Rodrigues (2007, p.3), “a pesquisa é uma atividade voltada para a solução de problemas, através do emprego de processos científicos”. Nesta pesquisa foi criada uma ontologia para representar o comportamento de uma das plantas de uma refinaria, como forma de facilitar o acesso a esse conhecimento, bem como disponibilizá-lo na forma de uma base de conhecimento (ontologia¹) para futuras aplicações.

Quanto à finalidade, este trabalho pode ser enquadrado no conceito de “pesquisa aplicada”, em que, de acordo com Rodrigues (2007, p.4), “os conhecimentos adquiridos são utilizados para aplicação prática voltados para a solução de problemas concretos da vida moderna”, pelo fato do conceito de ontologia ter sido aplicado para a modelagem de uma planta petroquímica real, condensando o conhecimento disperso em vários documentos.

Quanto ao tipo de pesquisa, esta dissertação pode ser classificada como descritiva, em que fatos são analisados, classificados e interpretados, sem a interferência do pesquisador,

¹ O termo “ontologia” será definido no item 2.3.

pois a ontologia foi desenvolvida como fruto da análise dos descritivos da planta e informações obtidas em entrevista não estruturada com um especialista no assunto (RODRIGUES, 2007). Essas informações foram classificadas e interpretadas com a finalidade de subsidiar a modelagem da planta.

Em síntese, esta pesquisa teve início com a análise de uma série de documentos descritivos da planta DEA, que incluíam o fluxograma de processo, os fluxogramas de tubulação e instrumentação (*P&Is*), e descrições da forma de operação e a relação de alarmes, que foram complementadas com uma entrevista não estruturada a um especialista na operação do processo.

Após a coleta e análise de todos esses dados, foram construídas as primeiras versões da ontologia da planta, que foi sendo gradativamente aperfeiçoada através de debates junto ao grupo de pesquisa.

A validação ocorreu através da construção de grafos descritivos e comparação com as informações coletadas, de forma que foi possível concluir pela representatividade do modelo e verificar que ele cobria o escopo definido.

1.5 Organização do texto

O restante do texto está dividido da seguinte maneira:

- Capítulo 2 – apresenta a fundamentação teórica para a pesquisa, abordando temas como o conhecimento e sua representação através de ontologias, os tipos de ontologias e suas características, metodologias e linguagens para criação, e o editor escolhido para modelagem da planta DEA;
- Capítulo 3 – aborda aspectos gerais do refino do petróleo, inserindo a planta de adoçamento de gás combustível no contexto; também apresenta a refinaria e o processo petroquímico escolhido como estudo de caso, com foco em aspectos gerais do funcionamento de uma planta DEA;
- Capítulo 4 – enfoca a criação da ontologia que descreve a planta, seguindo o processo de desenvolvimento passo a passo;
- Capítulo 5 – apresenta as análises realizadas, com o objetivo de comprovar a representatividade do modelo. O capítulo inicia com a apresentação do *plugin* utilizado para visualização de grafos descritivos da ontologia, que serviu como base para a avaliação do modelo; exibe a representação dos elementos físicos constituintes da planta e suas respectivas conexões, que terminam por indicar o fluxo dos produtos

e seus desdobramentos em subtrecos. Neste capítulo também é mostrada a representação dos instrumentos que estão conectados em cada trecho e, por fim, a associação dos alarmes configurados para cada um desses instrumentos.

- Capítulo 6 – aborda as conclusões obtidas ao final desta pesquisa e as perspectivas de trabalhos futuros que podem ser desenvolvidos tomando os resultados e as conclusões apresentadas como base.

Capítulo 2

Representação do Conhecimento Usando Ontologias

2.1 Introdução

Neste capítulo será apresentado o aporte teórico que serviu de alicerce para a modelagem da planta DEA utilizando ontologias.

A seguir serão abordados o conhecimento e sua representação através de ontologias, os tipos de ontologias e suas características; as linguagens utilizadas para a criação de ontologias, com maior enfoque na linguagem OWL, que foi escolhida para este estudo de caso; a ferramenta adotada para a criação da ontologia da planta DEA e as metodologias sugeridas na literatura para criar ontologias.

2.2 O conhecimento e sua representação

O que é “conhecimento”? Segundo Brachman e Levesque (2004), essa é uma questão que vem sendo discutida pelos filósofos desde a Grécia antiga, e mesmo assim não está totalmente desmistificada. Para Furgeri (2006, p.29), “o conhecimento já foi concebido como produto da inteligência divina, como uma dádiva de Deus aos homens, como fruto da razão e como produto da experiência”.

Na história da humanidade, ele já foi utilizado como mecanismo de dominação e controle, mas sempre foi visto como um bem precioso. Na atualidade, a sua disseminação tem contribuído cada vez mais para o avanço das pessoas, organizações e Estados. Contudo, sua falta tem produzido diferenças sociais e econômicas extremamente acentuadas, que ficam evidentes quando observamos que as nações mais ricas são grandes detentoras do conhecimento, sobretudo o tecnológico e o científico (FURGERI, 2006).

Representações, segundo Almeida (2005), “são conhecimentos construídos socialmente por uma comunidade ou grupo de sujeitos”. Furgeri (2006, p.41) detalha esse conceito quando afirma que “para se representar algo é necessário que o fenômeno observado e suas representações estejam assentadas na consciência do grupo e pressupõe a utilização de categorias, classes ou modelos definidos pelos integrantes”.

Brachman e Levesque (2004) apresentam algumas considerações informais acerca do assunto, tomando como base um ponto de vista de lógica proposicional, dentro da Ciência da Computação. Eles não procuram desmistificar conceitos sobre o “conhecimento”, mas partem da ideia de que, dentre outras coisas, ele pode ser entendido como uma relação entre um “conhecedor” e uma “proposição”, ou seja, a ideia expressa por uma sentença declarativa simples, tal como “John sabe que Abraham Lincoln foi assassinado”.

Ainda segundo Brachman e Levesque (2004), parte do mistério em volta do conhecimento é devido à natureza das proposições. Embora o que importe, sob a ótica deles, seja o fato dessas proposições se tratarem de entidades abstratas que podem ser verdadeiras ou falsas, corretas ou erradas.

Outros tipos de proposições também podem ser apresentadas, como é o caso de sentenças envolvendo conhecimento que não é explicitamente mencionado, como por exemplo “John sabe quem Mary está levando para a festa”. Por outro lado, quando se diz que alguém tem uma habilidade ou um conhecimento profundo sobre alguém ou algo, não fica claro se qualquer proposição útil está envolvida (BRACHMAN E LEVESQUE, 2004).

Uma noção relacionada que Brachman e Levesque (2004) consideraram importante é o conceito de “crença”. Para esses autores, as sentenças “John acredita que ...” e “John sabe que ...” são claramente relacionadas, embora a primeira delas seja utilizada quando não se deseja enfatizar que o julgamento de John sobre o mundo é necessariamente preciso ou suportado por razões adequadas. Na verdade, pode existir uma faixa de atitudes proposicionais, expressas por sentenças do tipo: “John está absolutamente certo que ...”, “John está convencido que...”, “John tem a opinião que ...”, “John está confiante que ...”, dentre outras. Segundo Brachman e Levesque (2004), o que importa é que todas as proposições compartilham com conhecimento uma ideia muito básica que: John pensa que o mundo seja de uma forma e não de outra.

Para Grigorova e Nikolov (2007), a Representação do Conhecimento (RC) é um elemento chave nos sistemas com inteligência artificial. A RC pode ser definida como algo que substitui o objeto ou fenômeno real, permitindo a uma entidade determinar as consequências de um ato pelo pensamento ao invés de sua realização (DAVIS; SHROBE; SZOLOVITS, 1993). Ela também pode ser definida como o campo de estudo preocupado com o uso de símbolos formais para representar uma coleção de proposições acreditadas por um suposto agente (BRACHMAN E LEVESQUE, 2004).

De acordo com Rezende (2005, p.29), uma RC “pode ser entendida como uma forma sistemática de estruturar e codificar o que se sabe sobre uma determinada aplicação”. Porém, a mesma autora ressalta que essa codificação não pode ser realizada sem quaisquer critérios, destacando como importante o atendimento às seguintes características:

- constituir uma representação compreensível ao ser humano, de tal modo que, caso seja necessário avaliar o estado de conhecimento do sistema, ela permita a sua interpretação;

- abstrair-se dos detalhes de como funciona internamente o processador de conhecimento que será utilizado para sua interpretação;
- ser robusta, isto é, permitir sua utilização mesmo que não aborde todas as situações possíveis;
- ser generalizável, ao contrário do conhecimento em si, que é individual. Pois uma representação necessita de vários pontos de vista do mesmo conhecimento, de tal forma que possa ser atribuída a diversas situações e interpretações.

Desde cedo a representação do conhecimento humano em sistemas computacionais foi um dos objetivos base da Inteligência Artificial (IA), pois ser inteligente é ter a habilidade de manipular conhecimentos; reconhecer símbolos; e chegar a uma solução quando se depara com um problema (GARCÊS et al., 2006).

Nos dias atuais, o escopo do problema de representação do conhecimento é amplo e diz respeito à lógica, à ontologia e às áreas computacionais (GRIGOROVA E NIKOLOV, 2007; GONZALÉZ, 2005). A lógica fornece a estrutura formal e as regras de inferência; enquanto a ontologia define os tipos de coisas que existem na aplicação de domínio; e a computação suporta as aplicações que distinguem a representação do conhecimento da filosofia pura. Assim, a RC pode ser definida como a aplicação da lógica e ontologia para a tarefa de construir modelos computacionais de algum domínio (GONZALÉZ, 2005).

O modelo ideal de representação do conhecimento deve ser suficientemente flexível para apresentar o conhecimento com um escopo diferente e granularidade; deve ter possibilidade de inferência e uso do contexto; deve trabalhar com informação incompleta e incerta; deve observar a perfeição e consistência quando ocorrerem mudanças no domínio e deve ser fácil de utilizar pelos engenheiros de conhecimento, que não necessariamente tenham experiência de programação (GRIGOROVA E NIKOLOV, 2007). Entretanto, Grigorova e Nikolov (2007) admitem que um modelo ideal não existe na realidade, fazendo com que engenheiros e desenvolvedores tenham que se comprometer de acordo com suas metas específicas.

A escolha da técnica de representação do conhecimento vai depender do grau de complexidade do conhecimento que deve ser explicitado e do tipo de aplicação onde será usado (FURGERI, 2006).

Neste trabalho, foi escolhida a representação do conhecimento através das ontologias, uma vez que esse tipo de representação do conhecimento é amplamente utilizado em

engenharia de conhecimento e inteligência artificial, em diferentes aplicações da ciência da computação e em novos campos emergentes como a web semântica² (DOKAS, 2007).

As ontologias são especialmente adequadas para formalizar conhecimento específico de domínio, que uma vez formalizado, pode ser facilmente interconectado com outros formalismos, o que facilita a interoperabilidade entre comunidades independentes e faz das ontologias um dos blocos fundamentais da construção da web semântica (GONZALÉZ, 2005).

Assim, as ontologias baseadas em computador fornecem uma representação formal e estruturada do conhecimento de domínio e são projetadas para servir como matéria prima para inferência e agentes computacionais, pois as ontologias fornecem uma especificação definida formalmente do significado de termos, os quais são utilizados pelos agentes durante sua interação (LACLAVÍK, 2005).

Nos tópicos seguintes apresentaremos as ontologias e os principais conceitos associados com esse tema.

2.3 Representação do conhecimento através das ontologias

A noção de ontologia tem sido motivo de debate dentro da filosofia por muitos anos. O que são as coisas? Qual a essência que permanece no interior das coisas quando elas mudam (mudanças no tamanho, na cor, etc.)? Os conceitos (mesa, casa, árvore, etc.) existem fora de nossa mente? Como podem ser classificadas as entidades do mundo? O estudo da ontologia, ou a teoria do “o que é” (do grego “ontos” = ser), tem sido uma disciplina de direito próprio desde os dias de Aristóteles, que pode ser visto como seu fundador e inspirador (GÓMEZ-PÉREZ; FERNÁNDEZ-LÓPEZ; CORCHO, 2004; SCHREIBER, 2008).

Assim, ao longo da história, muito foi debatido e proposto sobre esse tema dentro do campo filosófico, sendo que no final do Século XX e início do Século XXI, as ontologias emergiram como uma área importante de pesquisa na ciência da computação (GÓMEZ-PÉREZ; FERNÁNDEZ-LÓPEZ; CORCHO, 2004).

Segundo Furgeri (2006, p.113), “o termo ontologia possui diferentes significados, dependendo da área estudada. Em uma mesma área, como ocorre na CC, pode ocorrer variações em sua definição”. Gómez-Pérez, Fernández-López e Corcho (2004) comentam que

² “Web Semântica” é o nome genérico do projeto capitaneado pelo W3C, com o objetivo de embutir inteligência e contexto nos códigos XML utilizados para confecção de páginas da web, de modo a melhorar a maneira com que os programas podem interagir com as páginas e também possibilitar um uso mais intuitivo por parte dos usuários (SOUZA; ALVARENGA, 2004).

encontraram na literatura muitas definições sobre o que é uma ontologia e também tiveram oportunidade de observar como essas definições mudaram e evoluíram ao longo do tempo.

Gómez-Pérez, Fernández-López e Corcho (2004) não propõem uma nova definição para o tema, mas afirmam, de maneira conclusiva, que o objetivo das ontologias é capturar conhecimento consensual de forma genérica, de maneira que possa ser reutilizada e compartilhada através de aplicações de *software* e por grupos de pessoas, sendo geralmente construídas cooperativamente por diferentes grupos de pessoas em diferentes localizações.

Dentre as fontes consultadas nesse trabalho, a definição mais citada é a de Borst (1997, p.12): “*an ontology is a formal specification of a shared conceptualization*”, que Almeida e Bax (2003) adotam em seu trabalho e interpretam por partes: “formal” em referência ao fato de ser legível para computadores; “especificação explícita” em referência à definição explícita de conceitos, propriedades, relações, funções, restrições, axiomas; “compartilhado” pelo fato de ser um conhecimento consensual; e “conceitualização” por ser um modelo abstrato de algum fenômeno do mundo real.

No decorrer deste trabalho adotamos as ontologias segundo a definição proposta por Borst (1997), mas com o maior foco na finalidade das ontologias segundo Manica, Dantas e Todesco (2008, p.152):

Ontologias são usadas para modelar de maneira precisa o conhecimento de uma comunidade ou grupo de especialistas para que possa ser posteriormente consultado ou até mesmo reutilizado em novas aplicações. Ontologia é uma técnica importante para a área da Ciência da Computação, que viabiliza soluções para problemas como a falta de padronização, interoperabilidade, recuperação de informação, reuso, confusões terminológicas, problemas de troca de informações entre agentes de software, dentre muitos outros.

A seguir faremos uma breve apresentação de alguns tipos de ontologias e alguns comentários sobre suas características.

2.3.1 Características e tipos de ontologias

Segundo Almeida e Bax (2003, p.9), “as ontologias não apresentam sempre a mesma estrutura, mas existem características e componentes básicos comuns e presentes em grande parte delas”.

De uma forma mais didática, podemos dizer que as ontologias são compostas basicamente de **classes**, que são conceitos organizados em uma taxonomia; de **relações**, que representam o tipo de interação entre os conceitos de um domínio; de **axiomas**, que são

utilizados para modelar sentenças que são sempre verdadeiras; e de **instâncias**, que representam os elementos específicos de cada classe (ALMEIDA E BAX, 2003).

Para ilustrar esses conceitos, podemos pensar inicialmente em dois conjuntos, sendo um deles de pessoas e outro de veículos que elas possuem. A esses dois conjuntos poderiam ser atribuídas duas classes distintas, onde poderiam ser listadas todas as pessoas e veículos, sendo que cada elemento se constituiria em uma instância de sua respectiva classe. Como cada pessoa possui um veículo, essas duas classes podem ser relacionadas por uma propriedade que expresse essa posse e conecte uma instância da classe de pessoas ao respectivo veículo (instância da classe de veículos) que possui.

Em relação aos tipos, as ontologias podem ser classificadas de várias maneiras.

Mizoguchi, Vanwelkenhuysen e Ikeda (1995) apresentam uma classificação para as ontologias em três tipos, levando em consideração a sua conceitualização:

- ontologias de domínio: são reutilizáveis no domínio e fornecem um vocabulário sobre conceitos desse domínio, seus relacionamentos e sobre as atividades e regras que os governam;
- ontologias de tarefa: fornecem um vocabulário sistematizado de termos, especificando tarefas que podem estar ou não no mesmo domínio;
- ontologias genéricas: incluem um vocabulário relacionado a coisas, eventos, espaço, comportamento, causalidade, funções, etc.

Ushold e Gruninger (1996) apresentam uma classificação em quatro níveis para as ontologias, levando em consideração o grau de formalidade pelo qual o vocabulário é criado e seu significado é especificado:

- ontologia altamente informal: expressa livremente em linguagem natural;
- ontologia semi-informal: expressa em uma forma de linguagem natural restrita e estruturada;
- ontologia semi-formal: expressa em uma linguagem artificial definida formalmente;
- ontologia rigorosamente formal: expressa através de termos definidos meticulosamente com semântica formal, teoremas e provas de propriedades, tais como solidez e completude.

Para Heijst, Schreiber e Wielinga (1997), as ontologias podem ser classificadas segundo duas dimensões: a quantidade e tipo de estrutura da conceitualização, e o tema da conceitualização. Assim, quanto à primeira dimensão, os referidos autores apresentam três possibilidades de classificação:

- ontologias terminológicas: especificam os termos que são utilizados para representar o conhecimento no domínio de discurso, por exemplo, os léxicos;
- ontologias de informação: especificam a estrutura de registro de dados. Um exemplo são os esquemas de bancos de dados;
- ontologias de modelagem de conhecimento: especificam conceituações do conhecimento. Em comparação com as ontologias de informação, as ontologias de modelagem de conhecimento geralmente possuem uma estrutura interna mais rica;

Quanto à segunda dimensão, os mesmos autores apresentam quatro possibilidades:

- ontologias de aplicação: contém todas as definições que são necessárias à modelagem do conhecimento e que serão utilizadas em uma determinada aplicação. Esse tipo de ontologia pode ser obtido a partir da seleção de teorias de uma livraria de ontologias, que são então ajustadas para uma aplicação particular;
- ontologias de domínio: expressam conceituações que são específicas de um domínio particular de conhecimento;
- ontologias genéricas: são semelhantes às ontologias de domínio, mas os conceitos que elas definem são considerados genéricos e comuns a vários campos;
- ontologias de representação: explicam as conceituações que suportam os formalismos de representação do conhecimento. Ontologias de domínio e ontologias genéricas são descritas utilizando as primitivas disponibilizadas pelas ontologias de representação.

Guarino (1998) também propôs uma classificação, onde ele agrupa as ontologias com base no nível de generalidade:

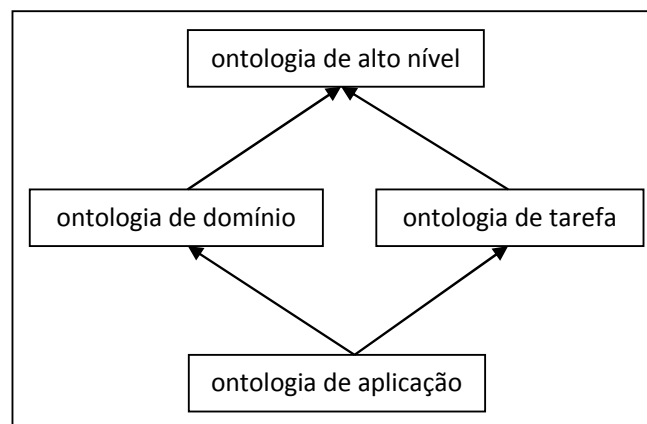


Figura 2.1. Tipos de ontologias, de acordo com seu nível de dependência em uma tarefa particular ou ponto de vista, onde as setas representam uma especialização das relações.

Fonte: Adaptado de Guarino (1998).

- ontologias de alto-nível: descrevem conceitos muito gerais, tais como espaço, tempo, matéria, objeto, evento, ação, etc., que são independentes de um problema particular ou domínio. Ter ontologias de alto nível unificadas parece razoável, pelo menos em teoria, para grandes comunidades de usuários;
- ontologias de domínio e ontologias de tarefas: descrevem, respectivamente, o vocabulário relacionado à um domínio genérico (tal como medicina, ou automóveis) ou uma tarefa genérica ou atividade (como diagnosticar ou vender), especializando os termos introduzidos em uma ontologia de alto nível;
- ontologias de aplicação: descrevem conceitos dependendo de um domínio particular e tarefa, que são geralmente especializações dessas duas ontologias relacionadas. Esses conceitos geralmente correspondem a papéis desempenhados por entidades do domínio enquanto se realiza uma certa atividade, como substituição de uma unidade ou de uma peça sobressalente.

Segundo Uschold e Jasper (1999), as ontologias são utilizadas para melhorar a comunicação entre humanos ou computadores. Tomando como base essa ideia, eles afirmam que as ontologias podem ser agrupadas em três áreas:

- ontologias para auxiliar na comunicação entre agentes humanos (pessoas): nesse caso uma ontologia informal mas inequívoca pode ser suficiente;
- ontologias para assegurar a interoperabilidade: a interoperabilidade entre sistemas de computador pode ser obtida pela tradução entre diferentes métodos de modelagem, paradigmas, linguagens e ferramentas de *software*; nesse caso a ontologia é utilizada como um formato de intercâmbio;
- ontologias para melhorar o processo e/ou qualidade de sistemas de engenharia de *software*: em que as ontologias podem ser utilizadas particularmente para:
 - *Reusabilidade* – a ontologia é a base para uma codificação formal das entidades, atributos e processos mais importantes, e suas interrelações no domínio de interesse. Essa representação formal pode ser (ou tornar-se por tradução automática) um componente reutilizável e/ou compartilhado em um sistema de *software*;
 - *Busca* – uma ontologia pode ser utilizada como metadados servindo como um índice em um repositório de informações;
 - *Confiabilidade* – uma representação formal também possibilita a automação da verificação de consistência, resultando em um *software* mais confiável;

- *Especificação* – a ontologia pode auxiliar o processo de identificação de requisitos e definir uma especificação para um sistema de TI (baseado em conhecimento ou o contrário);
- *Manutenção* – o uso de ontologias no desenvolvimento de sistemas, ou como parte de uma aplicação final, pode tornar a manutenção mais fácil de muitas formas, uma vez que sistemas que são construídos utilizando-se ontologias explícitas podem ter a sua documentação melhorada, o que reduz custos de manutenção;
- *Aquisição de conhecimento* – a velocidade e confiabilidade podem ser melhoradas através do uso de uma ontologia existente como ponto de partida e base para guiar a aquisição de conhecimento no momento de construção de sistemas baseados em conhecimento.

Haav e Lubi (2001) apresentam uma outra classificação para as ontologias, levando em consideração os tipos de classes presentes e sua estrutura, que guarda muita semelhança com a classificação proposta por Guarino (1998), mas não contempla o subitem “ontologia de aplicação”.

Enfim, sobre as várias classificações existentes para os tipos de ontologias, Almeida (2006) critica que não existe um consenso e as categorias propostas pelos autores muitas vezes se sobrepõem, inclusive em termos de nomenclaturas, que se repetem com significados distintos, o que nos leva a concluir que a classificação das ontologias não converge para uma classificação única.

A ontologia desenvolvida neste trabalho buscou representar os conceitos associados com uma planta petroquímica específica, seus relacionamentos e as regras que a governam. Dessa forma, uma iniciativa de classificar a ontologia sob esses vários pontos de vista pode gerar o seguinte resultado:

- Na classificação proposta por Mizoguchi, Vanwelkenhuysen e Ikeda (1995), pode ser classificada como uma ontologia de domínio, por fornecer um vocabulário sobre conceitos do domínio da planta DEA, seus relacionamentos e sobre as regras que a governam;
- Na classificação proposta por Uschold e Gruninger (1996), a classificação da ontologia não fica clara, pois poderia ser interpretada como uma ontologia semi-informal ou como uma ontologia semi-formal;
- Na classificação proposta por Heijst, Schreiber e Wielinga (1997), na primeira dimensão, como uma ontologia de modelagem de conhecimento, uma vez que

especificam conceituações de conhecimento associado à planta. Na segunda dimensão, novamente como uma ontologia de domínio, por, nessa nova ótica, expressar conceituações que são específicas do domínio particular de conhecimento estudado, a planta DEA;

- Na classificação proposta por Guarino (1998), a ontologia está mais próxima de uma ontologia de aplicação, por descrever conceitos que dependem de um domínio particular;
- Na classificação proposta por Uschold e Jasper (1998), não ficou evidente o enquadramento da ontologia, mas a objetivo de utilizar a ontologia como base de conhecimento para sistemas especialistas faz a classificação ficar mais próxima de uma ontologia para assegurar a interoperabilidade.

Nos próximos tópicos, a classificação adotada para se referir à ontologia foi a proposta por Mizoguchi, Vanwelkenhuysen e Ikeda (1995), ou seja, a ontologia será referenciada como de domínio.

2.3.2 Linguagens para criação de ontologias

Para a criação das ontologias, a literatura indica muitas linguagens. Dentre as possibilidades disponíveis, podemos verificar que elas diferem quanto ao nível de expressividade e às habilidades para inferência (SCHAUER E SCHAUER, 2008).

Em sua tese, Almeida (2006) apresenta uma coletânea das principais linguagens utilizadas e uma breve descrição de cada uma delas, que reproduzimos na tabela 2.1.

Tabela 2.1a. Linguagens para a construção de ontologias.

| Linguagens | Breve descrição |
|----------------------------------|--|
| CycL | Linguagem formal que expressa conhecimento através de um vocabulário de termos (constantes semânticas, variáveis, números, sequências de caracteres, etc.) os quais são combinados em expressões, sentenças e finalmente em bases de conhecimento. |
| Flogic (<i>Frame Logic</i>) | Integra <i>frames</i> e lógica de primeira ordem. Trata de forma declarativa os aspectos estruturais das linguagens baseadas em <i>frames</i> e orientadas a objeto (objetos, herança, tipos polimórficos, métodos de consulta, encapsulamento, etc.). Permite a representação de conceitos, taxonomias, relações binárias, funções, instâncias, axiomas e regras. |
| LOOM | Descendente da família KL-ONE (<i>Knowledge Language One</i>), é baseada em lógica descritiva e regras de produção. Permite a representação de conceitos, taxonomias, relações n-árias, funções, axiomas e regras de produção. |

Fonte: Almeida (2006, p. 128-129).

Tabela 2.1b. Linguagens para a construção de ontologias.

| | |
|--|--|
| CARIN | Trata-se de uma combinação da <i>Datalog</i> (linguagem baseada em regras) e lógica descritiva ALN. Uma ontologia CARIN é construída por dois componentes terminológicos: um conjunto de conceitos com declarações de inclusão e um conjunto de regras que usam os conceitos. |
| GRAIL | Linguagem que especifica uma ontologia de domínio médico (GALEN). É baseada em lógica descritiva terminologicamente limitada, que permite a construção de hierarquias de primitivas e axiomas de inclusão de conceitos. |
| Ontolingua | Combina paradigmas das linguagens baseadas em <i>frames</i> e lógica de primeira ordem. Permite a representação de conceitos, taxonomias de conceitos, relações n-árias, funções, axiomas, instâncias e procedimentos. Sua alta expressividade causa problemas na construção de mecanismos de inferência. |
| OCML | Permite a especificação de classes, relações, instâncias e regras; utilizada em aplicações em gestão do conhecimento, desenvolvimento de ontologias, comércio eletrônico e sistemas baseados em conhecimento; e em domínios como medicina, ciências sociais, memória corporativa, engenharia, portais da Web, etc. |
| OML (<i>Ontology Markup Language</i>) | Linguagem baseada em lógica descritiva e grafos conceituais, que permite a representação de conceitos organizados em taxonomias, relações e axiomas. |
| RDF (<i>Resource Description Framework</i>)/RDFS (<i>RDF Schema</i>) | Desenvolvidos pelo W3 Consortium, tem por objetivo a representação de conhecimento através de redes semânticas. São linguagens que permitem a representação de conceitos, taxonomias de conceitos e relações binárias. |
| NKRL (<i>Narrative Knowledge Representation Language</i>) | Linguagem de representação baseada em <i>frames</i> , especialmente desenvolvida para descrever modelos semânticos de documentos multimídia. |
| SHOE (<i>Simple HTML Ontology Extensions</i>) | Utiliza extensões ao HTML, adicionando marcações para inserir metadados em páginas Web. As marcações podem ser utilizadas para a construção de ontologias e para anotações em documentos da Web. |
| XOL | É uma linguagem que especifica conceitos, taxonomias e relações binárias. Não possui mecanismos de inferência e foi projetada para intercâmbio de ontologias no domínio da biomédica. |
| OIL (<i>Ontology Interchange Language</i>) | Precursor do DAML+OIL e linguagem base para a Web Semântica. Combina primitivas de modelagem das linguagens baseadas em frames com a semântica formal e serviços de inferência da lógica descritiva. |
| DAML (<i>DARPA Agent Markup Language</i>) + OIL | DAML+OIL é uma linguagem de marcação semântica para a Web que apresenta extensões a linguagens como o DAML, RDF e RDFS, através de primitivas de modelagem baseadas em linguagens lógicas. |
| FOML (<i>Formal Ontology Markup Language</i>) | Trata-se de uma linguagem de marcação, baseada em XML, que conecta documentos da Web a ontologias formais. O objetivo é a aquisição automática de conhecimento de domínios específicos. |

Fonte: Almeida (2006, p. 128-129).

Uma linguagem importante que Almeida (2006) não cita em sua pesquisa é a OWL (*Web Ontology Language* ou linguagem de ontologias para a Web), que é uma linguagem nova, desenvolvida pelo *World Wide Web Consortium* (W3C). Esta linguagem deriva de um consenso entre duas propostas, a europeia OIL (linguagem de intercâmbio entre ontologias) e a DAML (linguagem de anotação de agentes do Departamento de Defesa dos Estados Unidos)

(DE FREITAS, 2003; VIINIKKALA, 2004; HORROCKS; PATEL-SCHNEIDER; VAN HARMELEN, 2003).

A OWL foi a linguagem escolhida para aplicação nesta pesquisa e é por isso que será apresentada mais detalhadamente a seguir.

2.3.2.1 A linguagem OWL

A linguagem OWL (*Web Ontology Language*) é uma linguagem para a Web Semântica, desenvolvida pelo W3C, inicialmente projetada para representar informações sobre categorias de objetos e como eles estão interrelacionados, podendo representar também informações sobre os próprios objetos (HORROCKS; PATEL-SCHNEIDER; VAN HARMELEN, 2003).

Segundo o W3C (2004a), a linguagem OWL foi projetada para uso em aplicações que necessitam processar o conteúdo de informação, ao invés de apenas apresentar informações para os seres humanos. Para eles, esta linguagem facilita a interpretabilidade de máquina do conteúdo da Web mais do que as linguagens XML, RDF, e RDFS, por fornecer um vocabulário adicional juntamente com uma semântica formal.

Houve muitas influências no projeto da linguagem OWL. Como esta linguagem é fruto de um esforço do W3C, ela tinha que se ajustar à visão da Web Semântica de uma pilha de linguagens incluindo XML e RDF, inclusive como já havia muitas linguagens de ontologia projetadas para uso na Web, ela também tinha que manter o máximo de compatibilidade possível com as linguagens já existentes, incluindo SHOE, OIL e DAML+OIL (HORROCKS; PATEL-SCHNEIDER; VAN HARMELEN, 2003).

Todas essas influências criaram um conflito entre a expressividade da linguagem e a escalabilidade da inferência, o que resultou na divisão do OWL em três diferentes sublinguagens, refletindo diferentes graus de expressividade: OWL-Lite, OWL-DL e OWL-Full (GRIMM; HITZLER; ABECKER, 2007; GÓMEZ-PÉREZ; FERNÁNDEZ-LÓPEZ; CORCHO, 2004).

De acordo com o W3C (2004a), essas três sublinguagens foram projetadas para uso por comunidades específicas de implementadores e usuários, com as seguintes características:

- *OWL-Lite*: estende as linguagens RDF e RDFS e captura as características mais comuns do OWL, por isso é destinada aos usuários que precisam apenas criar taxonomias de classes e restrições simples. Por exemplo, ela permite valores de cardinalidade apenas 0 ou 1. É mais simples fornecer ferramentas de suporte para OWL-Lite do que para as outras duas sublinguagens mais expressivas. O OWL-Lite

também tem uma complexidade formal mais baixa que o OWL-DL. A vantagem de se optar por esta sublinguagem é eficiência no que diz respeito à inferência, tanto para usuários como para desenvolvedores de ferramentas, mas, por outro lado, perde-se muita expressividade (W3C, 2004a; GÓMEZ-PÉREZ; FERNÁNDEZ-LÓPEZ; CORCHO, 2004; CARDOSO, 2006);

- *OWL-DL*: é de longe o dialeto mais popular dos tipos de OWL e inclui o vocabulário OWL completo, embora ele possa ser utilizado apenas sob certas restrições (por exemplo, uma classe pode ser uma subclasse de muitas classes, mas uma classe não pode ser uma instância de uma outra classe). Suporta aqueles usuários que querem o máximo de expressividade mantendo a integridade computacional (a garantia de que todas as conclusões são computáveis) e decidibilidade (todos os cálculos serem completados em tempo finito). Na nomenclatura OWL-DL, o DL refere-se à lógica de descrição (*description logics*), que é um campo de pesquisa relacionado a um fragmento particular da lógica de primeira ordem decidível. A vantagem de se optar por esta sublinguagem é que o OWL-DL permite uma resposta mais rápida do motor de inferência e o motor de inferência é mais fácil de ser construído, mas, por outro lado, a desvantagem é que não se pode ter a expressividade ou a conveniência disponibilizada pelo OWL-Full (POLLOCK, 2009; CARDOSO, 2006; GÓMEZ-PÉREZ; FERNÁNDEZ-LÓPEZ; CORCHO, 2004; W3C, 2004a);
- *OWL-Full*: destina-se aos usuários que querem a expressividade máxima e a liberdade sintática do RDF, mas sem garantias computacionais. Por exemplo, em OWL-Full uma classe pode ser tratada simultaneamente como uma coleção de indivíduos e como um indivíduo em seu próprio direito. Esta sublinguagem permite que uma ontologia aumente o significado de um vocabulário pré-definido (RDF ou OWL), por outro lado é improvável que qualquer *software* de inferência seja capaz de fornecer suporte completo para cada recurso do OWL-Full. A vantagem do OWL é óbvia, ou seja, tem-se tudo à disposição e se pode usufruir de uma expressividade muito conveniente durante o desenvolvimento de uma ontologia, mas a desvantagem é que a ontologia pode se tornar muito poderosa e computacionalmente cara para que seja possível fornecer um suporte completo à inferência, ou seja, a eficiência é outro fator a ser considerado (CARDOSO, 2006; W3C, 2004a).

De acordo com Cardoso (2006), toda ontologia OWL-Lite é uma ontologia OWL-DL, mas não o inverso, bem como para o OWL-DL e OWL-Full, conforme mostrado na figura 2.2:

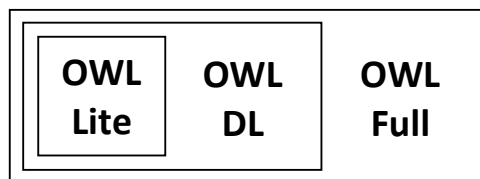


Figura 2.2. Sublinguagens OWL

Na sublinguagem de representação de conhecimento OWL-DL existem 40 primitivas (16 classes e 40 propriedades), sendo que algumas primitivas das RDF(S) podem ser utilizadas em todas as versões do OWL (Lite e DL), e as primitivas da OWL-Lite podem ser utilizadas em OWL-DL. Todas as primitivas de representação do conhecimento são listadas na figura 2.4, lembrando que as primitivas das sublinguagens OWL-DL e OWL-Full são as mesmas, conforme anteriormente comentado (GÓMEZ-PÉREZ; FERNÁNDEZ-LÓPEZ; CORCHO, 2004)

Na figura 2.4, as primitivas que são classes estão realçadas em azul, e como a criação da linguagem OWL foi influenciada pela DAML+OIL, algumas primitivas possuem equivalentes nessa outra linguagem, que são citadas entre parênteses. É um exemplo a primitiva OWL *owl:allValuesFrom*, que corresponde à primitiva DAML+OIL *daml:toClass* (GÓMEZ-PÉREZ; FERNÁNDEZ-LÓPEZ; CORCHO, 2004).

O W3C (2004b) disponibiliza um guia para uso da linguagem, de onde podem ser extraídos vários exemplos do uso da linguagem, como o mostrado na figura 2.3, onde é definida uma classe *WineColor*, cujos membros são os indivíduos *White*, *Rose* e *Red*.

```

<owl:Class rdf:ID="WineColor">
  <rdfs:subclassOf rdf:resource="#WineDescriptor"/>
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#White"/>
    <owl:Thing rdf:about="#Rose"/>
    <owl:Thing rdf:about="#Red"/>
  </owl:oneOf>
</owl:Class>

```

Figura 2.3. Trecho de código em OWL

Fonte: W3C (2004b)

| | |
|---|---|
| OWL DL | |
| Expressões de classe permitidas em: | rdfs:domain, rdfs: range, rdfs:subClassOf owl:intersectionOf, owl: equivalentClass, owl: allValuesFrom, owl: someValuesFrom |
| Valores não são restritos (0..N) em: | owl:minCardinality, owl:maxCardinality, owl:cardinality |
| owl:DataRange, rdf:List, rdf:first, rdf:rest, rdf:nil | |
| owl:hasValue (<i>daml:hasValue</i>) owl:oneOf (<i>daml:one of</i>) owl:unionOf (<i>daml:unionOf</i>), owl:complementOf (<i>daml:complementOf</i>) owl:disjointWith (<i>daml:disjointWith</i>) | |
| OWL Lite | |
| owl:Ontology (<i>daml:Ontology</i>), owl:versionInfo (<i>daml:versionInfo</i>), owl:imports (<i>daml:imports</i>), owl:backwardCompatibleWith, owl:incompatibleWith, owl:priorVersion, owl:DeprecatedClass, owl:DeprecatedProperty | |
| owl:Class (<i>daml:Class</i>), owl:Restriction (<i>daml:Restriction</i>), owl:onProperty (<i>daml:onProperty</i>), owl:allValuesFrom (<i>daml:toClass</i>) (apenas com identificadores de classe e tipos de dados nomeados), owl:someValuesFrom (<i>daml:hasClass</i>) (apenas com identificadores de classe e tipos de dados nomeados), owl:minCardinality (<i>daml:minCardinality</i> ; restrito a {0,1}), owl:maxCardinality (<i>daml:maxCardinality</i> ; restrito a {0,1}), owl:cardinality (<i>daml:cardinality</i> ; restrito a {0,1}) | |
| owl:intersectionOf (apenas com identificadores de classes e restrições de propriedade) | |
| owl:ObjectProperty (<i>daml:ObjectProperty</i>), owl:DatatypeProperty (<i>daml:DatatypeProperty</i>), owl:TransitiveProperty (<i>daml:TransitiveProperty</i>), owl:SymmetricProperty, owl:FunctionalProperty (<i>daml:UniqueProperty</i>), owl:InverseFunctionalProperty (<i>daml:UnambiguousProperty</i>), owl:AnnotationProperty | |
| owl:Thing (<i>daml:Thing</i>) owl:Nothing (<i>daml:Nothing</i>) | |
| owl:inverseOf (<i>daml:inverseOf</i>), owl:equivalentClass (<i>daml:sameClassAs</i>) (apenas com identificadores de classe e restrições de propriedade), owl:equivalentProperty (<i>daml:samePropertyAs</i>), owl:sameAs (<i>daml:equivalentTo</i>), owl:sameIndividualAs, owl:differentFrom (<i>daml:differentIndividualFrom</i>), owl:AllDifferent, owl:distinctMembers | |
| RDF(S) | |
| rdf:Property rdfs:subPropertyOf rdfs:domain rdfs:range (apenas com identificadores de classe e tipos de dados nomeados) rdfs:comment, rdfs:label, rdfs:seeAlso, rdfs:isDefinedBy rdfs:subClassOf (apenas com identificadores de classe e restrições de propriedade) | |

Figura 2.4. Primitivas da representação do conhecimento através do OWL-Lite e OWL-DL

Fonte: Adaptado de Gómez-Pérez, Fernández-López e Corcho (2004)

Segundo Kim e Kim (2008), a linguagem OWL faz uma grande contribuição para a engenharia ontológica, mas ela tem uma sintaxe difícil e complexa, com muitas etiquetas (*tags*) para compreensão pela máquina. Assim, uma ontologia pode ser desenvolvida através de uma descrição em texto ou utilizando um editor de ontologias.

Existem várias ferramentas que podem ser utilizadas para o desenvolvimento de ontologias sem que o usuário precise entrar em contato direto com a sintaxe da linguagem OWL. Dentre essas ferramentas, destaca-se o *software* Protégé, que foi desenvolvido pelo Centro de Pesquisa de Informática Biomédica da Universidade de Stanford, em sua faculdade de medicina, que por ter sido utilizado neste trabalho será apresentado a seguir.

2.3.2.2 O editor de ontologias Protégé

O Protégé é um editor de ontologias popular, desenvolvido em código aberto e um *framework* baseado em conhecimento (ERIKSSON, 2003; KASSAHUN; SCHOLTEN, 2006). Conforme anteriormente citado, foi desenvolvido pelo Centro de Pesquisa de Informática Biomédica da Universidade de Stanford, em sua faculdade de medicina, por um grupo liderado por Mark Musen, implementado em Java, e com apoio de várias empresas e agências governamentais americanas (STANFORD, 2010; LICHTENSTEIN; SIGULEM, 2008).

O projeto Protégé teve início em 1987, com a criação de uma metaferramenta para sistemas baseados em conhecimentos na área médica, originalmente com o propósito de construir ferramentas de aquisição de conhecimento para alguns programas especializados em planejamento médico (LICHTENSTEIN; SIGULEM, 2008).

Ele foi gradativamente modernizado para acompanhar a evolução da tecnologia de SBCs (Sistemas Baseados em Conhecimento), com o objetivo de servir para aquisição de conhecimentos diretamente de especialistas de domínios, com menor dependência de engenheiros do conhecimento, bem como permitir diversos formalismos e estratégias de inferência, integrar tarefas em um mesmo ambiente (aquisição de ontologias e instâncias no mesmo ambiente, testes com inferência, etc.), criar automaticamente formulários para entrada do conhecimento, acessar ontologias via protocolo OKBC (*Open Knowledge Base Connectivity*) e combinar ontologias (SEMPREBOM; CAMADA; MENDONÇA, 2007).

Segundo Stanford (2010), a plataforma Protégé suporta duas maneiras principais de modelagem de ontologias:

- *Editor Protégé-Frames*: permite ao usuário construir e popular ontologias que são baseadas em *frames*, em conformidade com o protocolo OKBC. Nesse modelo, uma ontologia consiste de um conjunto de classes organizadas em uma hierarquia para representar um domínio de conceitos relevantes, um conjunto de propriedades associadas às classes para descrevê-las e indicar seus relacionamentos, e um conjunto de instâncias dessas classes (indivíduos que exemplificam esses conceitos e que possuem valores específicos para as propriedades);
- *Editor Protégé-OWL*: permite ao usuário construir ontologias para a web semântica, na linguagem OWL padronizada pelo W3C. Uma ontologia OWL pode incluir descrições de classes, propriedades e suas instâncias. Através de uma ontologia, a semântica formal OWL especifica como derivar consequências lógicas, ou seja, fatos que não estão literalmente presentes na ontologia, mas decorrentes da semântica. Essas conclusões podem ser baseadas em um documento simples ou em múltiplos documentos distribuídos que tenham sido combinados utilizando mecanismos OWL definidos (W3C, 2004; STANFORD, 2010).

Como o Protégé foi escrito em java e com a opção por trabalhar com código aberto, usuários podem desenvolver novos *plugins* utilizando as APIs do Protégé, que podem ser avaliadas por outros usuários. Um exemplo dessas contribuições é o *plugin* Jambalaya, que é um utilitário com animação e vários outros recursos para visualização de dados (KIM E KIM, 2008; SEMPREBOM; CAMADA; MENDONÇA, 2007).

Para que se possa ter um ideia da interface do Protégé, a figura 2.5 apresenta a tela inicial do *software*, em modo de edição OWL, que foi utilizado para a edição das ontologias durante este trabalho.

A ferramenta Protégé possibilita ao usuário a edição das ontologias em linguagem OWL, utilizando uma interface gráfica amigável e sem a preocupação específica de aplicar corretamente a sintaxe da linguagem para definição das classes, relações, restrições, instâncias, verificação de consistência das ontologias (inferência), etc., o que tornaria esta tarefa mais complexa e mais sujeita a erros. Neste trabalho foi utilizada a versão 3.4.4 do *software*, que foi lançada em 8 de março de 2010.

Segundo Kim e Kim (2008), o Protégé é, atualmente, o editor mais famoso de ontologias, sendo utilizado por muitos pesquisadores e desenvolvedores.

Almeida (2006) também optou por adotar essa ferramenta em sua pesquisa, à época em sua versão Protégé 2000, tendo justificado sua escolha com base nas seguintes constatações: (a) está disponível para uso imediato; (b) não exige grandes recursos de *hardware* nem

licenças; (c) conta com o envolvimento de uma grande comunidade de pesquisa e de usuários; (d) foi concebida há mais de dez anos; (e) é frequentemente atualizada; (f) possui interface amigável e documentação; (g) possui grande número de funcionalidades que permitem representar peculiaridades de um domínio organizacional.

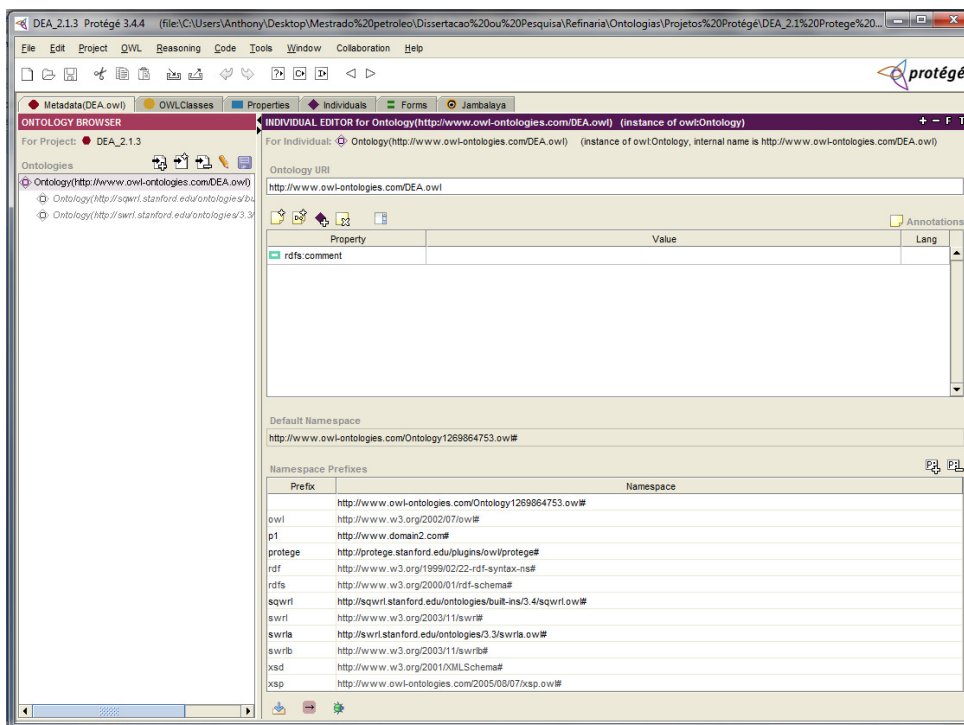


Figura 2.5. Tela principal do Protégé para edição de ontologias em OWL.

Fonte: Pesquisa direta (2010).

Para completar a abordagem sobre ontologias, serão apresentadas algumas metodologias para construção das ontologias que são citadas na literatura sobre o tema.

2.3.2.3 Metodologias para construção das ontologias

A literatura apresenta várias possibilidades para o desenvolvimento de ontologias. Para se ter uma ideia da abrangência do tema, Cristani e Cuel (2005) pesquisaram as principais metodologias existentes e afirmaram que seu estudo não pôde ser exaustivo pelas seguintes razões: (1) na prática das aplicações da web semântica e em muitos casos reais dos sistemas inteligentes de informação, as pessoas desenvolvem suas próprias metodologias, o que torna quase impossível uma revisão completa; (2) muitas metodologias têm sido desenvolvidas como uma ilustração dos princípios que inspiraram as ferramentas e as linguagens, mas são

essencialmente equivalentes a outras metodologias; e (3) muitos casos de metodologias implícitas existem; em particular, no contexto específico de linguagens ou ferramentas.

Quando se pensa nas várias metodologias existentes é inevitável se perguntar: Será que existe uma metodologia mais correta? Segundo Noy e McGuinness (2001), nem existe uma única metodologia correta para o desenvolvimento de ontologias, e nem é esse o objetivo desses autores ao proporem uma metodologia própria para desenvolvimento de ontologias que é muito citada na literatura. Para eles, esse processo pode ser iterativo, de tal forma que se faça uma primeira passagem na ontologia, depois se revise e refine a ontologia em desenvolvimento, preenchendo os detalhes. Esses autores entendem que existem algumas regras fundamentais no projeto de ontologias que podem ajudar a tomar decisões em muitos casos:

1. Não há nenhuma forma correta de modelar um domínio, mas sempre há alternativas viáveis. A melhor solução quase sempre depende da aplicação que você tem em mente e as extensões que você antecipa;
2. O desenvolvimento de uma ontologia é necessariamente um processo iterativo;
3. Os conceitos em uma ontologia devem ser próximos dos objetos (físicos ou lógicos) e relacionamentos em seu domínio de interesse. Eles são mais susceptíveis de serem substantivos (objetos) ou verbos (relacionamentos) em sentenças que descrevem seu domínio.

Noy e McGuinness (2001) ainda recomendam que entre as muitas alternativas viáveis é preciso determinar a que trabalhará melhor para a tarefa projetada, que será mais intuitiva, mais extensível e de melhor manutenção, lembrando que uma ontologia é um modelo da realidade do mundo e os conceitos na ontologia devem refletir essa realidade.

O fruto desse processo será a primeira versão da ontologia, que precisará ser testada e depurada através de seu uso em aplicações, métodos de resolução de problemas ou ainda em debates com especialistas no assunto, o que certamente irá resultar na necessidade de ajustar a ontologia inicial, em um processo iterativo que vai se repetir continuamente durante o ciclo de vida da ontologia (NOY E MCGUINNESS, 2001).

Segundo Cristani e Cuel (2005), a metodologia desenvolvida por Noy e McGuinness (2001) em seu manual "*Ontology Development 101*" é um guia muito simples, baseado em um projeto iterativo que auxilia desenvolvedores, mesmo os que não são especialistas, a criar uma ontologia utilizando algumas ferramentas, dentre as quais está o Protégé-2000 (versão mais antiga do Protégé), seguindo a sequência de passos apresentada na tabela 2.2.

Tabela 2.2. Metodologia de desenvolvimento de ontologias “*ontology development 101*”.

| Nome da fase | Entrada | Descrição da fase | Saída |
|---|--|--|---|
| Determinar o domínio e o escopo da ontologia | Nada. É o primeiro passo | Definição de: - Qual o domínio que a ontologia irá cobrir, - Qual ontologia será utilizada, - A que tipos de questões as ontologias devem fornecer respostas (questões de competência são muito importantes nesse domínio, porque permitem ao projetista entender quando a ontologia contém informação suficiente e quando ele alcança o nível adequado de detalhe ou representação), - Quem utilizará e fará manutenção na ontologia. | O documento resultante pode mudar durante todo o processo, mas a qualquer tempo essa documentação ajuda a limitar o escopo do modelo. |
| Considerar a reutilização de ontologias existentes | Documentos com o domínio e o escopo da ontologia | Procurar por outras ontologias que estão definindo o domínio. Existem livrarias de ontologias reusáveis na web e na literatura (por exemplo, livraria de ontologias Ontolingua, livraria de ontologias DAML, UNSPSC, RosettaNet, e DMOZ) | Uma ou mais ontologias de domínio, ou parte delas com sua descrição. |
| Enumerar termos importantes na ontologia | Documentos com o domínio, o escopo da ontologia, e as livrarias no domínio | Escrever uma lista de todos os termos utilizados na ontologia, descrevendo esses termos, seus significados, e suas propriedades | Termos e aspectos importantes para modelar na ontologia |
| Definir as classes e a hierarquia das classes | Termos importantes na ontologia, domínio, e descrição do escopo | Existem muitas aproximações possíveis no desenvolvimento de um hierarquia de classes: - o processo de desenvolvimento de cima para baixo (<i>top-down</i>) se inicia com a definição dos conceitos mais gerais do domínio e a subsequente especialização dos conceitos; - o processo de desenvolvimento de baixo para cima (<i>bottom-up</i>) vai na direção oposta; - o processo de desenvolvimento combinado é uma mistura das duas aproximações anteriormente citadas. | Classes e hierarquia das classes |
| Definir as propriedades de classes-fresta (<i>slot</i>) | A taxonomia, o domínio e a descrição de escopo | Acrescenta todas as propriedades necessárias e informação que permita à ontologia responder as questões de competência | Classes, hierarquia das classes, e propriedades |
| Definir as facetas das frestas (<i>slots</i>) | Frestas (<i>slots</i>) e classes | Existem diferentes facetas descrevendo o tipo de valor, valores admissíveis, o número de valores, e outras características que as frestas podem assumir: cardinalidade da fresta, tipo de valor da fresta, domínio, e faixa | Ontologia |
| Criar as instâncias | A ontologia | Criar as instâncias de classes na hierarquia, o que significa escolher uma classe, criar um indivíduo da instância da classe, e preencher os valores das frestas. | Ontologia e o domínio modelado |

Fonte: Adaptado de Cristani e Cuel (2005).

Ao analisar a metodologia proposta por Noy e McGuinness (2001), Antoniou e Harmelen (2004) distinguem oito passos principais no processo de desenvolvimento de uma ontologia, ressaltando que como qualquer processo de desenvolvimento, na prática, não se constituem em um processo linear, mas em passos iterativos, que poderão ser retornados a estágios anteriores sempre que for necessário em qualquer ponto do processo. Os oito passos indicados por Antoniou e Harmelen (2004) são os seguintes:

1. Determinar o escopo: desenvolver uma ontologia é o mesmo que definir um conjunto de dados e sua estrutura para outros programas utilizarem, ou seja, uma ontologia é um modelo de um domínio particular, construído para um propósito particular. Perguntas básicas a serem respondidas nesta etapa: Qual é o domínio que a ontologia deverá cobrir? Para que objetivo a ontologia deverá ser utilizada? Quais os tipos de questão que a ontologia deverá responder? Quem irá utilizar e manter a ontologia?
2. Considerar o reuso: com a disseminação da web semântica, mais ontologias serão disponibilizadas, fazendo com que raramente um projeto seja iniciado sem que haja pelo menos uma ontologia de um terceiro que sirva como ponto de partida.
3. Enumerar termos: um primeiro passo em direção à definição da ontologia é escrever uma lista não estruturada de todos os termos relevantes que se espera utilizar na ontologia, observando que é comum os substantivos formarem a base para os nomes das classes, e os verbos formarem a base para os nomes das propriedades (por exemplo, *is part of*, *has component*).
4. Definir taxonomia: após identificar os termos relevantes, esses termos devem ser organizados em uma hierarquia taxonômica, para garantir que seja respeitada a semântica construída de primitivas como *owl:subClassOf*.
5. Definir propriedades: normalmente as propriedades que conectam as classes são organizadas ao mesmo tempo em que são organizadas as classes em uma hierarquia. Também já se pode verificar os domínios e as faixas de valores que essas propriedades podem ter enquanto se está conectando propriedades às classes.
6. Definir as facetas das propriedades: até o passo anterior foi utilizada apenas a expressividade disponibilizada pelas primitivas RDFS, sendo que a partir desse passo passam a ser utilizadas as primitivas em OWL, através do enriquecimento das propriedades definidas previamente com as facetas:
 - Cardinalidade – especifica para as propriedades se é permitido ou necessário que elas tenham um certo número de valores diferentes. Os casos mais comuns são

“no mínimo um valor” (*at least one value*) e “no máximo um valor” (*at most one value*).

- Valores necessários – as classes são geralmente definidas por força de uma certa propriedade assumindo valores particulares, sendo que esses valores podem ser especificados em OWL, através do uso de *owl:hasValue*. Por outro lado, às vezes os requisitos são menos rigorosos e uma propriedade pode assumir valores de uma determinada classe (não necessariamente um valor específico, *owl:someValuesFrom*).
- Características relacionais – a família final das facetadas diz respeito às características relacionais das propriedades, que são a simetria, transitividade, propriedades inversas, e valores funcionais.

A partir da conclusão deste passo será possível checar inconsistências internas na ontologia. Antes não seria possível porque a linguagem RDFS não é suficientemente rica para expressar inconsistências.

7. Definir instâncias: geralmente as ontologias são desenvolvidas para organizar um conjunto de instâncias, que tipicamente são em número muito maior que o número de classes de uma ontologia. Por esse motivo, muitas vezes esse processo não é feito manualmente, mas mediante busca em fontes como bancos de dados ou extraídos automaticamente do corpo de um texto.
8. Checagem de inconsistências³: uma vantagem do uso do OWL é a possibilidade de detectar inconsistências na própria ontologia ou em um conjunto de instâncias que foram definidas para popular a referida ontologia. Alguns exemplos de inconsistências comuns nas propriedades são a incompatibilidade entre as definições do domínio e imagem, transitividade, simetria, inversibilidade, ou cardinalidade. Outra possibilidade pode ser conflito entre os requisitos para os valores das propriedades e as restrições para os domínios e faixa de valores admissíveis.

As características relacionais citadas no passo 6 podem ser consideradas especializações das propriedades das classes, com os seguintes significados (ANTONIOU; HARMELEN, 2004; WAGNER FILHO; LÓSCIO, 2009):

- Transitividade (*TransitiveProperty*) – se um par de instâncias (x,y) se relaciona através de uma propriedade transitiva P , e o par (y,z) também se relaciona por P , é possível concluir que o par (x,z) também se relaciona por P ;

³ Passo adicional proposto por Antoniou e Harmelen (2004), em complemento à metodologia proposta por Noy e McGuinness (2001).

- Simetria (*SymmetricProperty*) – se duas propriedades forem declaradas simétricas, sabendo que a propriedade P relaciona o par (x,y) , é possível concluir que P também relaciona o par (y,x) ;
- Funcionalidade (*FunctionalProperty*) – se uma propriedade é dita funcional, é possível afirmar que ela tem restrição de mínima cardinalidade igual a 0 e máxima cardinalidade igual a 1;
- Inversa – se uma propriedade P é dita inversa de Q , sabendo-se que (x,y) se relacionam através da propriedade P , é possível concluir que a propriedade Q relaciona o par (y,x) .
- Funcional e inversa (*InverseFunctionalProperty*) – indica que uma propriedade é inversa de uma propriedade funcional.

Conforme comentado no início desta seção, existem muitas outras metodologias para desenvolvimento de ontologias, além da metodologia proposta por Noy e McGuinness (2001) e desses oito passos indicados por Antoniou e Harmelen (2004) (baseados na metodologia anterior).

Segundo Almeida e Bax (2003, p.9), “existem metodologias para a construção de ontologias, para construção de ontologias em grupo, para aprendizado sobre a estrutura de ontologias e para a integração de ontologias”. As metodologias apresentadas anteriormente nesta seção são aplicáveis ao desenvolvimento de ontologias de domínio, que foi o tipo de ontologia utilizado nesta pesquisa.

A tabela 2.3, extraída de Almeida e Bax (2003) indica outras possibilidades existentes na literatura, em termos das metodologias para desenvolvimento de ontologias.

De acordo com Cristani e Cuel (2005), na Ciência da Computação existem muitas linguagens e ferramentas para ajudar o usuário final e os desenvolvedores de sistemas na criação de ontologias boas e efetivas, sendo que várias dessas ferramentas ajudam as pessoas a criarem, tanto manualmente como semiautomaticamente, categorias, partonomias⁴, taxonomias, e outros níveis organizacionais das ontologias.

Em termos de ferramenta para criação das ontologias, neste estudo optou-se por desenvolver as ontologias utilizando o *software* Protégé, na versão 3.4, que é uma ferramenta popular para a criação de ontologias e, dentre as vantagens citadas por Almeida (2006), chamou a atenção o fato de se tratar de uma ferramenta desenvolvida em código aberto e

⁴ De acordo com Mastella *et al.* (2005, p.635), “Partonomia é uma organização de domínio baseada em uma relação de *parte-de*, em vez de em uma relação de *é-um*”.

contar com o envolvimento de uma grande comunidade de pesquisa e de usuários, tendo sido mencionada em vários artigos encontrados na literatura.

Tabela 2.3. Metodologias para construção de ontologias.

| Metodologia | Breve descrição |
|------------------------|---|
| <i>Cyc</i> | Codifica manualmente o conhecimento implícito e explícito das diferentes fontes, e, quando já se tem conhecimento suficiente na ontologia, um novo consenso pode ser obtido por ferramentas que utilizam linguagem natural. |
| USCHOLD e KING | Identifica o propósito, os conceitos e relacionamentos entre os conceitos, além dos termos utilizados para codificar a ontologia e, em seguida, documentá-la. |
| GRÜNINGER e FOX | Método formal que identifica cenários para uso da ontologia, utiliza questões em linguagem natural para determinação do escopo da ontologia, executa a extração sobre os principais conceitos, propriedades, relações e axiomas, definidos em PROLOG. |
| <i>KACTUS</i> | Método recursivo que consiste em uma proposta inicial para uma base de conhecimento; quando é necessária uma nova base em domínio similar, generaliza-se a primeira base em uma ontologia adaptada a ambas aplicações; quanto mais aplicações, mais genérica a ontologia. |
| <i>Methontology</i> | Constrói uma ontologia por reengenharia sobre outra utilizando-se o conhecimento do domínio; as atividades principais são especificação, conceitualização, formalização, implementação e manutenção. |
| <i>Sensus</i> | Constrói ontologias a partir de outras ontologias, identificando os termos relevantes para o domínio e ligando-os à ontologia mais abrangente (<i>Sensus</i> , com 50 mil conceitos); um algoritmo monta a estrutura hierárquica do domínio. |
| <i>On-to-knowledge</i> | Auxilia a administração de conceitos em organizações, identificando metas para as ferramentas de gestão do conhecimento e utilizando cenários e contribuições dos provedores/clientes de informação da organização. |

Fonte: Almeida e Bax (2003, p.14).

Concluída a abordagem essencial sobre as ontologias, o próximo capítulo tratará do processo petroquímico que foi utilizado para este estudo de caso.

Capítulo 3

Planta DEA

3.1 Introdução

A indústria moderna possui uma grande variedade de processos químicos, desenvolvidos com as mais diversas finalidades. Uma planta de refino de petróleo pode ser constituída por vários desses processos, cuja seleção irá depender das características do óleo a ser processado e das necessidades do mercado que irá receber os seus derivados.

Segundo Passos *et al.* (2006, p.16), “o refino de petróleo constitui-se da série de etapas operacionais para obtenção de produtos determinados. Refinar petróleo é, portanto, separar as frações desejadas, processá-las e industrializá-las, transformando-as em produtos vendáveis”.

Na atividade de refino, o petróleo é utilizado como um insumo, que é separado em frações de derivados através de processos físico-químicos, processados em unidades de separação e conversão, até que sejam obtidos os produtos finais (SZKLO, 2005).

Assim, segundo Szklo (2005) os produtos finais de uma refinaria podem ser divididos em três categorias:

- Combustíveis (gasolina, diesel, óleo combustível, GLP, QAV, querosene, coque de petróleo, óleos residuais) representam cerca de 90% dos produtos de refino no mundo;
- Produtos acabados não combustíveis (solventes, lubrificantes, graxas, asfalto e coque);
- Intermediários da indústria química (nafta, etano, propano, butano, etileno, propileno, butilenos, butadieno, BTX).

Portanto, as refinarias de petróleo são sistemas complexos, com múltiplas operações, que dependem das propriedades do insumo (ou *mix* de insumos) e dos produtos desejados, de tal forma que, segundo Szklo (2005, p.19), “não existem duas refinarias iguais no mundo”.

Para exemplificar a complexidade que essas plantas de refino podem assumir, as figuras 3.1 e 3.2 apresentam exemplos de um esquema de refino muito simples, apenas com uma coluna de destilação atmosférica, e um esquema mais complexo, para a produção de combustíveis e aromáticos.

Como o número de etapas possíveis para refino do petróleo é elevado, cada uma delas realizando processos químicos e físicos específicos, com os mais variados graus de complexidade, fez-se necessário limitar o universo da pesquisa. Por essa razão, foi desenvolvido um estudo de caso, com enfoque em uma planta específica da REPLAN (Refinaria do Planalto Paulista), denominada DEA, cuja finalidade e detalhes específicos serão abordados nos tópicos seguintes.

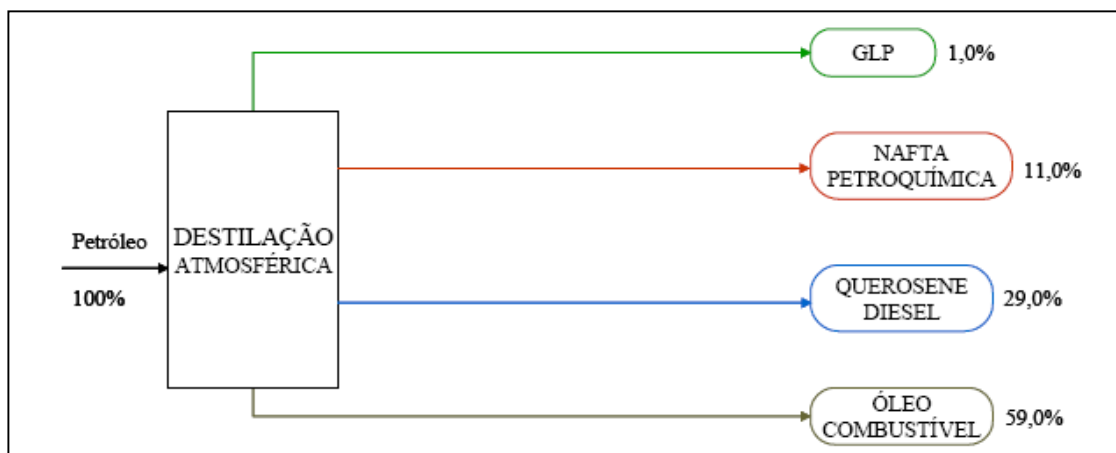


Figura 3.1. Esquema de refino composto por uma unidade de destilação.

Fonte: Passos *et al.* (2006, p.23).

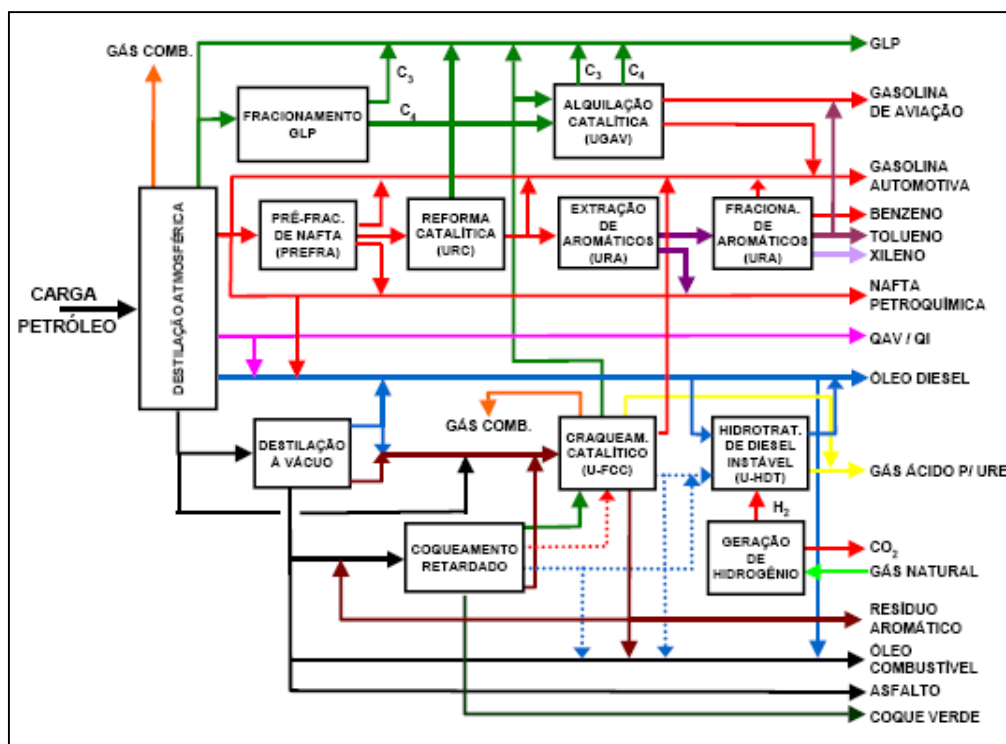


Figura 3.2. Esquema de refino orientado para a produção de combustíveis e aromáticos

Fonte: Passos *et al.* (2006, p.26)

3.2 REPLAN – Refinaria do Planalto Paulista

Esta refinaria (figura 3.3) é a maior do Brasil, responsável pelo refino de 20% de todo o petróleo processado no país, localizada no município de Paulínia, no Estado de São Paulo. A REPLAN entrou em operação em 1972, com capacidade de processamento de 20.000 m³/dia,

contando com unidades de destilação atmosférica e a vácuo e de craqueamento catalítico do tipo FCC (SZKLO, 2005).



Figura 3.3. REPLAN – Refinaria do Planalto Paulista

Fonte: <http://paulinia.net/refinaria-de-paulinia-replan>, [s.d.]

Essa refinaria possui alto rendimento em diesel, mas também produz volumes importantes de nafta petroquímica, para o pólo de São Paulo, e de gasolina. O óleo combustível também é um derivado com considerável produção por essa unidade, cuja produção se concentra no setor industrial de São Paulo, embora este perfil tenha se alterado com a entrada do gás natural neste mercado (SZKLO, 2005).

Segundo Szklo (2005, p.139), como consequência dessa entrada do gás natural no mercado de São Paulo, para a REPLAN, “planeja-se a ampliação de sua capacidade de conversão, através de unidades de fundo de barril (sobretudo, coqueamento retardado) e da sua capacidade de tratamento de produtos finais e intermediários, através de unidades de HDT de diesel e de recuperação de enxofre”.

O perfil de produção da REPLAN, no final da década de 1990, pode ser observado na tabela 3.1 e o esquema simplificado dessa refinaria é apresentado na figura 3.4.

Tabela 3.1. Perfil de produção da REPLAN no final da década de 1990

| Unidade | Produtos |
|---------------------------|---------------------------------|
| Destilação | Gasolina |
| Craqueamento catalítico | GLP |
| Recuperação de enxofre | Diesel/Óleos combustíveis |
| Solventes | Asfalto/Querosene de iluminação |
| MTBE | QAV |
| Tratamento Merox para QAV | Nafta petroquímica |
| | Enxofre |
| | Solventes |
| | MTBE |

Fonte: Tolmasquim et al. (2000) *apud* Szklo (2005, p.139)

Embora o esquema simplificado da REPLAN, apresentado na figura 3.4, não apresente explicitamente a localização da planta DEA, acreditamos que ele esteja contido no bloco de “regeneração de enxofre”.

Conforme anteriormente comentado, o estudo de caso foi desenvolvido na planta DEA da REPLAN, cujo modo de operação será descrito a seguir.

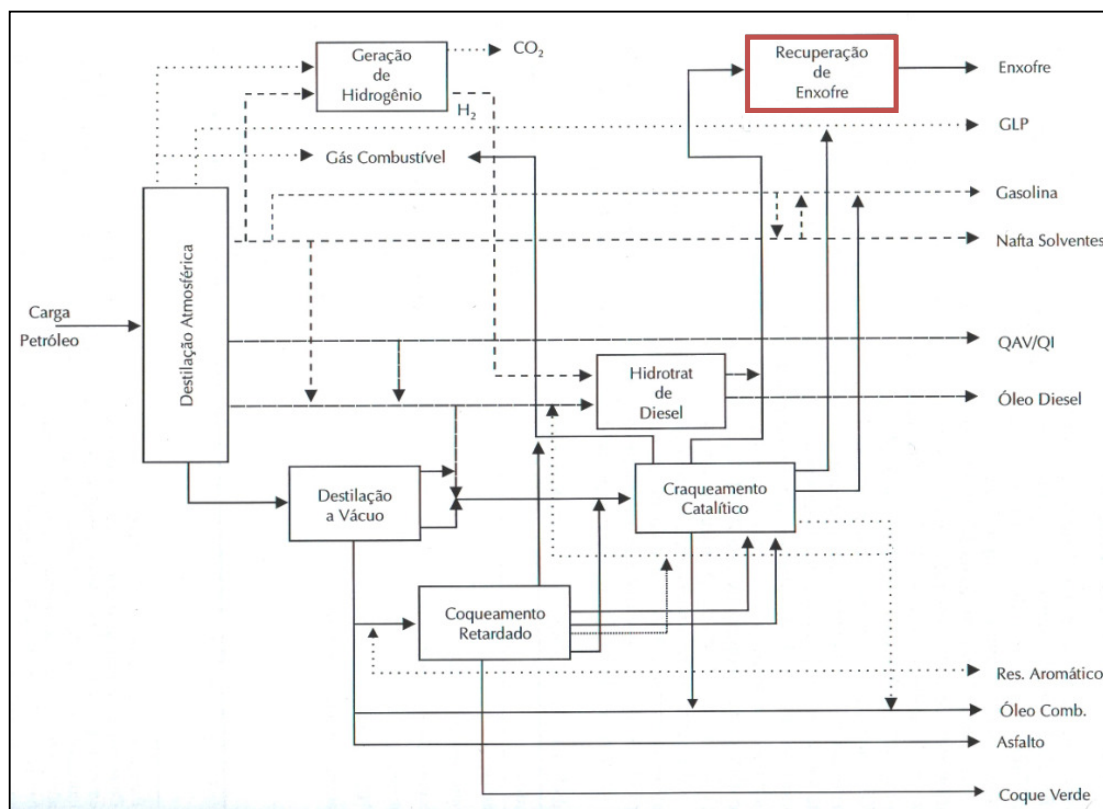


Figura 3.4. Esquema simplificado da REPLAN

Fonte: Szklo (2005, p. 140)

3.3 Descrição da planta DEA – adoçamento de gás combustível

Gases de várias operações em refinarias que processam óleo cru ácido contêm sulfeto de hidrogênio (H₂S) e ocasionalmente sulfeto de carbonila (COS). Até o período entre 1965 e 1970, era uma prática comum simplesmente queimar esse sulfeto de hidrogênio junto com outros gases leves como combustível para refinaria, porque sua remoção dos gases e conversão em enxofre elementar não era economicamente interessante. Porém, em um ambiente com exigências ambientais mais rígidas passou a ser necessário remover a maioria do sulfeto de hidrogênio desses gases e convertê-lo em enxofre elementar (GARY; HANDWERK, 2001).

Quando se trabalha com o gás natural, os compostos de enxofre são considerados contaminantes, pois a sua presença na mistura resulta no aumento das taxas usuais de corrosão dos equipamentos (VAZ; MAIA; SANTOS, 2008).

Para remover esses contaminantes do gás podem ser utilizados processos físicos e químicos. Como somente a absorção química é capaz de reduzir suficientemente o teor dos gases ácidos, alguns processos podem utilizar a combinação de um processo físico seguido de um processo químico para que sejam alcançados os teores admitidos pela Legislação ou pela especificação dos equipamentos (VAZ; MAIA; SANTOS, 2008).

Dentre as possibilidades existentes estão os processos de absorção química com soluções aquosas de alcanolamina, que podem ser utilizadas para tratamento de fluxos de gás contendo sulfeto de hidrogênio (H_2S) e dióxido de carbono (CO_2), sendo que a seleção da amina apropriada para o atendimento das especificações do gás na saída irá depender da composição do gás combustível de entrada e de suas condições de operação (MOKHATAB; POE; SPEIGHT, 2006).

Nas unidades de tratamento de amina, o sulfeto de hidrogênio é removido de uma corrente de gás através do contato com uma solução aquosa de monoetanolamina (MEA) ou dietanolamina (DEA), de acordo com a seguinte reação química (PARKASH, 2003):

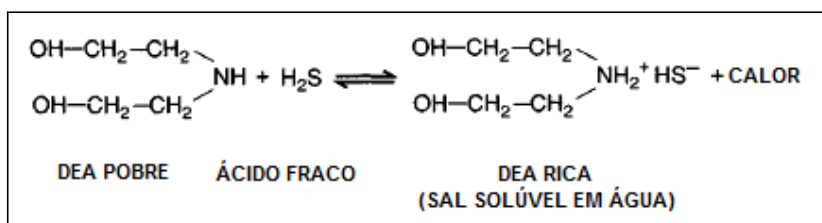


Figura 3.5. Reação de absorção de H_2S por dietanolamina

Fonte: Adaptado de Parkash (2003)

A reação de absorção de sulfeto de hidrogênio é favorecida se as condições de operação do absorvedor forem de baixa temperatura, alta pressão e alta concentração de H_2S (PARKASH, 2003).

Assim, a planta DEA consiste no processo de tratamento de frações leves de petróleo com soluções de dietanolamina (DEA), devido às reações desta solução com os compostos ácidos, tais como H_2S e CO_2 , presentes no GLP ou no gás combustível. Essa amina pode ser regenerada através do aquecimento controlado, que possibilita a remoção do H_2S e CO_2 absorvidos (PASSOS *et al.*, 2006). Dessa forma, a dietanolamina pode ser encontrada referenciada nesse processo sob duas formas: DEA pobre, que é a dietanolamina pronta para

ser utilizada na absorção dos gases ácidos ou regenerada após o uso; ou DEA rica, que é a dietanolamina contaminada com os gases ácidos, após ter sido colocada em contato com eles para absorvê-los.

A planta DEA estudada tem a função de tratar o gás combustível contendo H_2S proveniente da Unidade de Coqueamento Retardado, quando esta processa resíduo de vácuo das unidades de destilação a vácuo (PETROBRAS, [s. d.]).

Embora tenhamos estudado uma planta real, de maior complexidade que o exemplo apresentado por Passos (2006), ilustrado pela figura 3.6 a seguir, a aplicação deste exemplo é de grande relevância para a compreensão do modo de operação desse tipo de unidade. Em geral, essas plantas utilizam uma solução de DEA regenerada e consistem em uma torre extratora de H_2S e CO_2 de GLP (gás liquefeito de petróleo), uma torre absorvedora de H_2S e CO_2 de gás combustível, e uma torre regeneradora de DEA. Segundo Passos (2006, p.97-99), a unidade subdivide-se nas seguintes etapas:

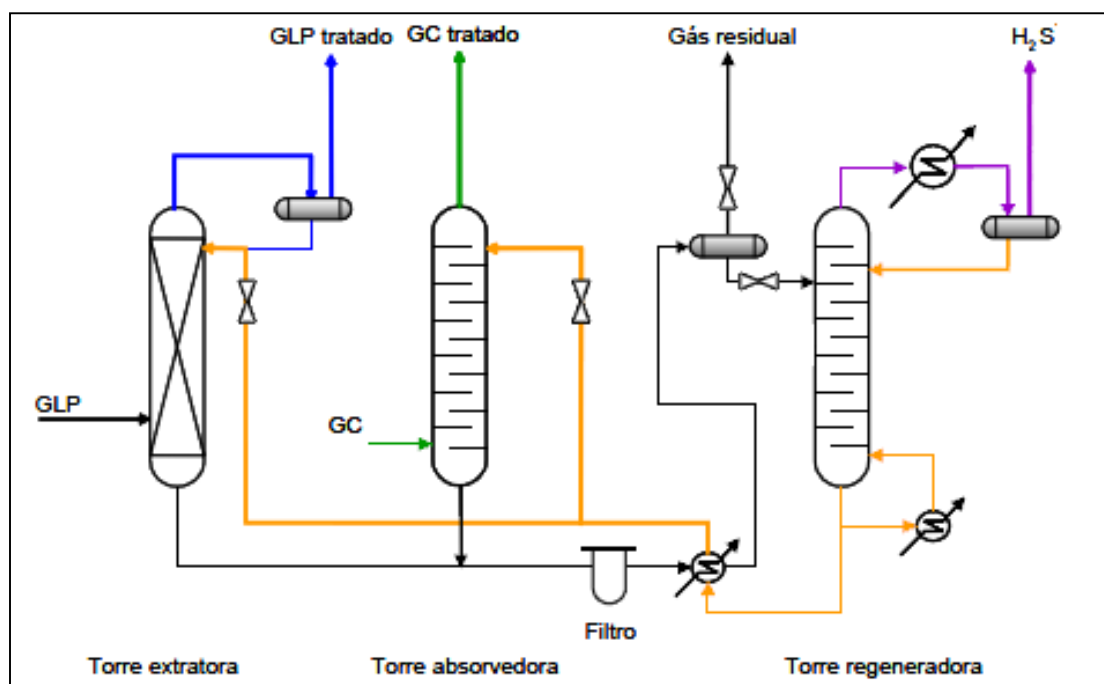


Figura 3.6. Esquema do tratamento do GLP e do gás combustível com DEA.

Fonte: Adaptado de Passos *et al.* (2006).

- Seção de extração – o GLP a ser tratado é encaminhado para a torre extratora, onde entra em contato, em contracorrente, com a solução aquosa de DEA regenerada, denominada de DEA pobre. A torre é constituída de pratos perfurados de aço-carbono ou anéis de polipropileno, sendo a DEA pobre admitida na parte superior e a

DEA rica coletada na parte inferior. No topo da extratora, o GLP separa-se da fase aquosa e segue para um tambor de decantação, onde serão removidos os traços de DEA eventualmente arrastados, indo então para o tratamento cáustico regenerativo (MEROX). A DEA rica retirada do fundo da extratora vai para o sistema de regeneração.

- Seção de absorção – o gás combustível a ser tratado é encaminhado à torre absorvedora, onde é borbulhado, em contracorrente, na solução de DEA. As torres podem ser constituídas por recheio de anéis de polipropileno ou pratos perfurados e o contato entre as fases afeta diretamente a absorção. Da mesma forma que no GLP, a DEA pobre entra pela região superior e a DEA rica sai pela parte inferior, indo para o sistema de recuperação após se juntar com a DEA rica vinda do sistema de extração. O gás retirado do topo da torre é enviado para o sistema de gás combustível.
- Filtração – com o objetivo de retirar os produtos de corrosão (sulfeto de ferro) e de degradação arrastados, antes de ser regenerada, a DEA é filtrada, para que esse material não se deposite nos equipamentos, nem provoque a formação de espuma. O tipo de filtro e a sua posição podem variar nas diversas refinarias. O filtro pode ser constituído de um leito de carvão ativado suportado por uma tela de aço ou, ainda, ser do tipo “cartucho” de aço inox contendo pequenos orifícios, filtros de saco, dentre outros. Nessa etapa também deve ocorrer a expansão ou despressurização da corrente de DEA rica.
- Vaporização de hidrocarbonetos – a DEA rica, que deixa as torres de absorção e extração, é preaquecida em um permutador DEA rica x DEA pobre, passando pelos dutos, e depois é encaminhada para o tambor de vaporização, para separar os hidrocarbonetos arrastados pela DEA. Os hidrocarbonetos são retirados pelo topo do domo no tambor, que é recheado com anéis de carvão, e os traços de H₂S e CO₂ liberados são reabsorvidos pela injeção de uma corrente de DEA no topo do domo. A remoção desses hidrocarbonetos é necessária, porque a sua presença na corrente ácida de H₂S a torna inadequada à produção de enxofre, e eles são escoados sob controle de pressão para o sistema de tocha da refinaria.
- Regeneração – a partir do tambor de vaporização, a DEA rica é enviada à torre de reativação da DEA, passando antes por uma válvula de controle, que vaporiza grande parte dos gases ácidos devido à diferença de pressão nesta válvula. Na torre, a DEA rica entra pela seção de topo e sofre um processo de esgotamento, mediante a geração de vapor d’água da própria solução, por um refeedor existente no sistema

de fundo da torre. Os gases são liberados da solução e sobem em contracorrente com a carga da torre, saindo pelo topo. É interessante manter a pressão da torre o mais baixa possível a fim de se garantir o escoamento dos gases ao destino final. Os gases ácidos seguem para a Unidade de Recuperação de Enxofre (URE) ou são queimados na tocha quando não existe URE. A corrente de DEA pobre, ou seja, reativada, sai pelo fundo da reativadora e preaquece a carga desta torre, sendo, em seguida, bombeada para sofrer um prerresfriamento final com água, a fim de que se atinja a temperatura mais adequada ao processo, antes de retornar às torres de extração e de absorção. A torre regeneradora de DEA é constituída de pratos de aço-carbono, com válvulas de aço inox do tipo AISI 304 e 410.

Embora este modelo teórico auxilie na compreensão do funcionamento desse tipo de processo petroquímico, a planta DEA real instalada na REPLAN é mais complexa e apresenta algumas divergências, conforme será explicado a seguir.

3.4 A planta DEA da REPLAN

Em relação ao modelo teórico apresentado, a planta DEA instalada na REPLAN se constitui em um processo mais complexo e que também apresenta algumas divergências.

Na REPLAN, a planta não é utilizada para remoção de sulfeto de hidrogênio do gás liquefeito de petróleo, dispensando a seção de extração.

Além do sulfeto de hidrogênio, esse tipo de planta também pode ser utilizado para remoção de dióxido de carbono (CO_2), ambos gases ácidos produzidos junto com os hidrocarbonetos, mas a planta real estudada não é utilizada para remoção do dióxido de carbono, porque o teor desse contaminante presente no gás combustível é considerado relativamente baixo, não se constituindo em um inconveniente para o sistema (PETROBRAS, [s.d.]).

Outra diferença observada entre os modelos teórico e a planta real, é a supressão das etapas de bombeamento no fluxograma de processo simplificado da figura 3.3, que é mencionado na etapa de regeneração do modelo teórico, mas não está representado no diagrama. A planta real possui dois sistemas de bombeamento, um deles para bombeamento de DEA pobre, após a etapa de regeneração, e o outro para bombeamento da DEA rica, armazenada no vaso de topo da torre regeneradora, pois um pouco dessa DEA flui junto com o sulfeto de hidrogênio através da saída no topo da torre regeneradora.

A etapa de filtração, mencionada no modelo teórico, está disposta antes da etapa de regeneração, com o objetivo de retirar os produtos de corrosão e degradação arrastados, antes da regeneração da DEA. Na planta real, a mesma etapa está disposta após a etapa de regeneração, mas a documentação da planta não indica as premissas por trás dessa decisão.

A planta real possui um vaso coletor de hidrocarbonetos líquidos, antecedendo a etapa de absorção de H_2S , para que os hidrocarbonetos líquidos fiquem retidos nesse vaso e seja liberado apenas o gás combustível que será tratado, pelo topo desse vaso. Esse vaso não consta no modelo teórico.

Após a saída de topo da torre absorvedora de H_2S da planta real, existe um vaso coletor de DEA arrastada, que não consta no modelo teórico, e serve para reter a DEA que seja eventualmente arrastada pela corrente de gás combustível que deixa a torre.

O modelo teórico não menciona o sistema de tanques de armazenamento de DEA, não detalha o sistema refeedor conectado ao fundo da torre regeneradora, tampouco apresenta indicação do sistema de automação conectado à planta, que são as informações disponibilizadas nos fluxogramas de tubulação e instrumentação.

Assim, a planta real é composta pelos seguintes processos:

- Duas torres, sendo uma delas de absorção de H_2S e a outra de regeneração da DEA rica;
- Quatro vasos de pressão, sendo um coletor de hidrocarbonetos líquidos, um coletor de DEA arrastada, um separador de hidrocarbonetos e um de topo da torre regeneradora;
- Dois sistemas de bombeamento, um deles de circulação de solução DEA e o outro de refluxo da regeneradora;
- Três sistemas trocadores de calor, sendo um permutador DEA pobre x DEA rica, um condensador de topo da regeneradora e um resfriador de DEA pobre;
- Um sistema de filtração para fluido de processo.

Além dos processos, foram catalogados trinta e cinco trechos de tubulação para utilização no desenvolvimento da ontologia, embora existam outros trechos que não foram contemplados no escopo. A planta possui vários elementos de tubulação, como é o caso das válvulas de bloqueio manual, peças figura 8, válvulas PSV, válvulas de retenção, dentre outros.

Em relação à parte de automação, foram catalogados cinquenta e dois instrumentos para utilização na ontologia, sendo:

- Trinta e cinco sensores: dez chaves de nível, quatro sensores de nível, um sensor de pressão diferencial, seis sensores de pressão estática, onze sensores de temperatura e três sensores de vazão;
- Nove controladores: três controladores de nível, três controladores de pressão, dois controladores de temperatura e um controlador de vazão;
- Oito atuadores: três válvulas controladoras de nível, três válvulas controladoras de pressão, uma válvula controladora de temperatura e uma válvula controladora de vazão.

Foram catalogados quarenta e seis alarmes associados aos instrumentos presentes na ontologia, sendo dezoito alarmes associados à medição de nível, um alarme associado à medição de pressão diferencial, dez alarmes associados à medição de pressão estática, treze alarmes associados à medição de temperatura e quatro alarmes associados à medição de vazão.

Conforme mencionado anteriormente, a compreensão da planta DEA da REPLAN demandou a análise de um fluxograma de processo, dois fluxogramas de tubulação e instrumentação (*P&Is*), documentos descritivos do modo de operação da planta e o auxílio de um especialista na operação do processo, que indicou os caminhos que deveriam ser contemplados no escopo da ontologia e esclareceu as dúvidas acerca de seu funcionamento.

3.5 Análise da planta DEA - conclusão

O conhecimento do funcionamento da planta DEA, em particular a instalada na REPLAN, é indispensável para modelar a ontologia descritiva da planta.

A pesquisa bibliográfica possibilitou o acesso a um modelo teórico, que serviu como base para a compreensão da aplicação desse tipo de processo petroquímico e seu modo de funcionamento. Essas informações teóricas foram complementadas com a documentação da planta real e o apoio do especialista, que também definiu o escopo a ser contemplado na modelagem da planta.

A partir desses estudos, conforme foi apresentado no tópico anterior, já foi possível determinar alguns itens que deveriam estar presentes na ontologia, cuja forma de modelagem será abordada no próximo capítulo.

Capítulo 4

Ontologia do Processo DEA

4.1 Introdução

Com a conclusão do estudo sobre representação do conhecimento através das ontologias, da linguagem OWL, da ferramenta de desenvolvimento adotada para modelagem e do funcionamento do processo petroquímico adotado como estudo de caso, o próximo passo para a evolução desta pesquisa passa a ser o desenvolvimento da ontologia que irá descrever a planta DEA.

Neste capítulo será apresentada a forma como foi modelada a planta DEA utilizando o conceito de ontologias e aplicando-se a ferramenta Protégé.

4.2 Representação da planta DEA através de ontologia

Para a modelagem da planta DEA, a ontologia desenvolvida foi classificada como uma ontologia de domínio, que segundo Heijst, Schreiber e Wielinga (1997), expressa conceituações que são específicas de um domínio particular de conhecimento, ou seja, o processo químico analisado.

O desenvolvimento da ontologia da planta DEA seguiu os oito passos indicados por Antoniou e Harmelen (2004), que se constituem em uma adaptação do método proposto por Noy e McGuinness (2001), incluindo um passo adicional de verificação de inconsistências, que passaremos a apresentar nos subitens a seguir.

4.2.1 Determinação do escopo

O primeiro passo para a modelagem da planta é determinar o escopo da ontologia, esclarecendo qual o domínio que ela deverá cobrir e a razão pela qual a ontologia será utilizada.

Foi estabelecido que a ontologia desenvolvida deveria modelar o domínio particular restrito à planta DEA, que é um processo petroquímico específico, representando todos os seus elementos constituintes (vasos, torres, bombas, dutos, válvulas, etc.) e suas respectivas conexões, inclusive sensores e alarmes associados com os instrumentos. Através do modelo, buscou-se sintetizar as informações disponibilizadas em seu fluxograma de engenharia (*piping & instrumentation diagram*) e nos documentos descritivos de seu modo de operação.

Quanto à finalidade, a ontologia foi desenvolvida com o objetivo de disponibilizar a descrição da planta tanto para especialistas como para outros sistemas que sejam

desenvolvidos no futuro, em face da vantagem do uso da linguagem OWL não apenas para apresentar informações, mas também para ser utilizada por aplicações que precisem processar o conteúdo da informação.

Outro objetivo para a criação da ontologia foi responder à pergunta: como ocorre o fluxo de produtos na planta DEA? Porque a partir da resposta a essa pergunta pode ser possível entender como se propagariam eventuais falhas registradas na planta.

Através da análise dos documentos descritivos do modo de operação da planta de processo foram identificados os principais fluxos apresentados na figura 4.1. Ressaltando que, embora estes sejam os principais fluxos de produtos, existem trechos em que alguns produtos fluem misturados e mesmo não estando destacados na referida figura, foram modelados na ontologia para auxiliar na compreensão do processo.

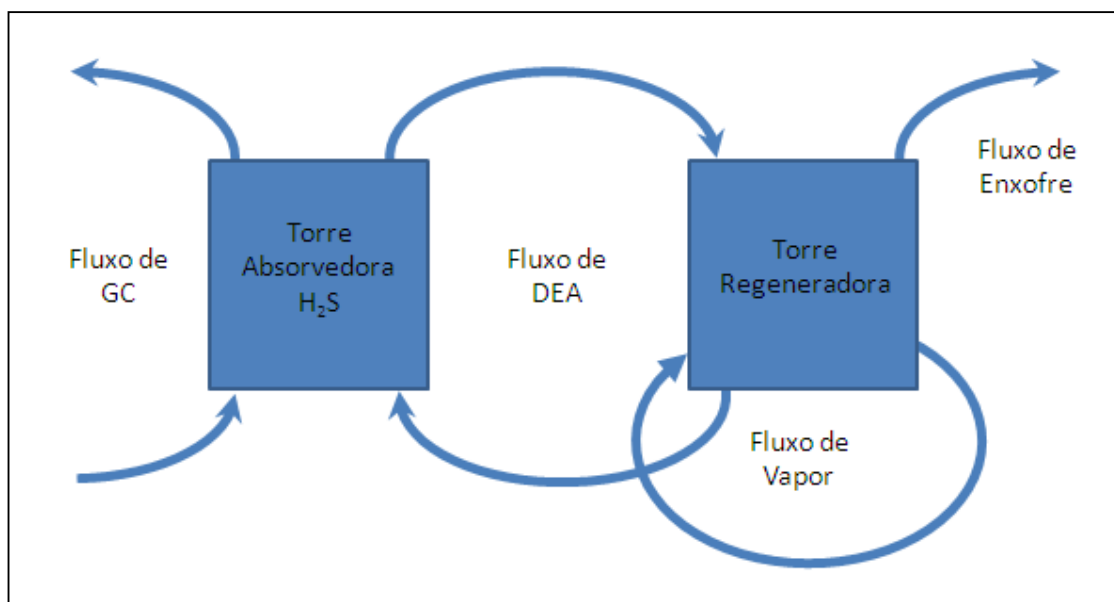


Figura 4.1. Simplificação da planta através da ilustração dos principais fluxos considerados.

Fonte: Elaborado pelo autor (2009).

Além dos documentos citados anteriormente, um dos especialistas no funcionamento do processo também forneceu informações ao pesquisador sobre a operação do processo e destacou no fluxograma de engenharia os principais caminhos percorridos pelos fluidos na planta, o que serviu de grande contribuição para as análises.

Com a conclusão da definição do escopo da representação e a maior compreensão acerca do funcionamento da planta, seguiu-se à próxima etapa, indicada a seguir.

4.2.2 Consideração de reuso

A possibilidade de reuso de ontologias é ancorada na disseminação da web semântica, que tende a aumentar o número de ontologias disponibilizadas para reuso e facilitar o início dos projetos (ANTONIOU; HARMELEN, 2004).

Cristani e Cuel (2005) sugerem que se procure outras ontologias que definam o domínio, consultando livrarias de ontologias reusáveis. Para facilitar essa consulta, o site do Protégé disponibiliza uma lista de ontologias compatíveis com o *software* na *home page*: http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library. Dentre as ontologias catalogadas, a que mais se aproximava do domínio estudado era a OntoCAPE, citada na página como uma ontologia de larga escala para o domínio da engenharia de processos químicos.

Após carregar a OntoCAPE na versão 3.4.4 do Protégé, verificamos que a ontologia é muito pesada, pois se constitui em um conjunto de várias ontologias. A tentativa de validação utilizando o *reasoner* Pellet, na versão que vem com o próprio Protégé, retornou uma série de inconsistências, que não foram possíveis de rastrear e sanar. Portanto, entendeu-se como mais simples a elaboração de uma nova ontologia, que ao final dos trabalhos mostrou-se mais fácil de compreender e utilizar.

4.2.3 Enumeração dos termos

Nessa etapa, Antoniou e Harmelen (2004) sugerem que seja construída uma lista não estruturada de todos os termos relevantes que se espera utilizar na ontologia, sendo provável que os substantivos se tornem nomes de classes e os verbos se tornem nomes de propriedades.

Portanto, examinando os fluxogramas de engenharia, é possível concluir que os elementos de tubulação, processos e instrumentos são os substantivos associados com a planta. Desta forma, esses elementos poderão ser agrupados nas seguintes categorias:

- Elementos de tubulação – são todos os tubos para condução (dutos) utilizados no processo, bem como seus acessórios (flanges, conexões, etc.), mas como apenas os dutos possuem identificação na planta, apenas eles estarão contidos nesta classe;
- Processos – são todos os elementos onde os produtos sofrem algum tipo de transformação física ou química. São exemplos de elementos que podem estar contidos: bombas, torres, vasos, filtros, trocadores de calor, etc.;

- Instrumentos – são todos os elementos de automação e instrumentação conectados à planta, tais como sensores, controladores e atuadores.

Além dos elementos listados anteriormente, que podem ser obtidos diretamente do fluxograma de engenharia, o escopo da ontologia também deve contemplar os alarmes que estão configurados para cada um dos instrumentos, ou seja, eles são substantivos que também fazem parte do levantamento inicial.

A partir do levantamento de termos e da análise do comportamento da planta, é possível tirar algumas conclusões que podem auxiliar na construção da ontologia:

- Na saída de cada processo, duto ou válvula, existe outro processo, duto ou válvula conectado – analisando os fluxogramas de processo, tubulação e instrumentação da planta DEA, é possível observar que a planta é composta por dutos, válvulas e processos conectados;
- Os dutos e processos possuem sensores para seu monitoramento – para que seja monitorado o comportamento da planta DEA, as grandezas físicas associadas aos dutos e processos são monitoradas por sensores instalados nesses elementos;
- Os atuadores da planta (válvulas) possuem controladores para seu acionamento – verificando os fluxogramas de tubulação e instrumentação é possível observar que as válvulas automáticas do sistema são acionadas pelos controladores instalados na planta;
- Sensores e controladores possuem alarmes – analisando a documentação do sistema, é possível constatar que os alarmes configurados para a planta estão associados com as grandezas monitoradas pelos sensores e controladores;
- Os controladores recebem sinal dos sensores para atuar – para que os controladores atuem sobre as grandezas físicas do sistema, é necessário que haja monitoramento constante dessas grandezas. Esse monitoramento acontece através dos sensores, que dentre outras funções, podem ser conectados à entrada dos controladores.

Os próximos passos levarão em consideração os termos enumerados e as propriedades listadas acima.

4.2.4 Definição das classes e taxonomia

Com os termos enumerados, a próxima etapa é a definição das classes e de sua hierarquia, que nesse processo foi realizada de cima para baixo (top-down), iniciando com a

definição dos conceitos mais gerais do domínio e a subsequente especialização (CRISTANI; CUEL, 2005).

As classes são elementos que se referem a entidades ou fenômenos que possuem os mesmos atributos em um domínio (CHENG E DU, 2008). Quando buscamos uma relação entre a etapa de definição das classes e as regras fundamentais estabelecidas por Noy e McGuinness (2001), podemos identificar as classes como os substantivos (objetos) nas sentenças que descrevem o domínio de interesse.

Portanto, partindo dos termos enumerados no subitem anterior, é possível concluir que os elementos de tubulação, processos e instrumentos são substantivos que podem ser representados por classes na ontologia. Desta forma, essas classes poderão ser agrupadas nas seguintes categorias:

- Elementos de tubulação – essa classe pode contemplar todos os dutos utilizados no processo, bem como seus acessórios (flanges, conexões, etc.), mas como apenas os dutos possuem identificação na planta, apenas eles estarão contidos nesta classe;
- Processos – essa classe pode contemplar todos os elementos onde os produtos sofrem algum tipo de transformação física ou química. São exemplos de elementos que podem estar contidos: bombas, torres, vasos, filtros, trocadores de calor, etc.;
- Instrumentos – essa classe pode contemplar todos os elementos de automação e instrumentação conectados à planta, tais como sensores, controladores e atuadores.

Além das categorias listadas anteriormente, que podem ser obtidas diretamente do fluxograma de engenharia, definiu-se no escopo que o modelo também deveria contemplar os alarmes que estão configurados para cada um dos instrumentos. Desta forma, percebe-se a necessidade de determinar uma classe adicional, que não é visível tomando apenas o diagrama P&I como referência, que é a classe de alarmes.

Esse agrupamento de objetos em categorias é uma parte muito importante da representação do conhecimento, mas será a especialização dessas categorias em subclasses que irá permitir a organização dessas classes em uma taxonomia (RUSSELL E NORVIG, 2004).

Quando a ontologia contém subclasses, elas herdam automaticamente as propriedades de suas respectivas superclasses (LICHTENSTEIN; SIGULEM, 2008).

Nas descrições apresentadas acima, a classe dos elementos de tubulação foi especializada apenas em uma subclasse de dutos, porque os demais elementos de tubulação constantes da ontologia (válvulas manuais, figuras “8”, etc.) não possuíam TAGs de identificação que possibilitassem a sua inclusão na ontologia.

Para a classe de processos, suas subclasses foram definidas como a lista de processos existentes na planta: bombas, condensador, filtro, permutador, torre e vaso.

Para o caso dos instrumentos, a classe foi especializada nas subclasses de sensores, controladores e atuadores, mas da mesma forma que a classe de alarmes, esses três conceitos ainda podem ser novamente especializados, em função das grandezas físicas envolvidas no processo, que são: temperatura, pressão, vazão e nível. Para o caso específico da pressão, os sensores podem medir pressão estática ou diferencial, sendo os alarmes também associados com esses tipos de pressão, razão pelo qual os conceitos de sensores e alarmes de pressão também foram novamente especializados.

A figura 4.2 apresenta a relação das classes definidas para descrever a planta, indicando a especialização de cada um dos conceitos.

Ainda na figura 4.2 é possível observar a existência de uma classe denominada *owl:Thing* acima das demais classes definidas para a ontologia. Isso acontece porque a referida classe é predefinida na linguagem OWL como a classe mais geral, fazendo com que todas as classes definidas na ontologia sejam obrigatoriamente um desdobramento deste conceito (GÓMEZ-PÉREZ; FERNÁNDEZ-LÓPEZ; CORCHO, 2004).

Por fim, entre as classes ou subclasses de mesmo nível hierárquico, todas foram definidas como disjuntas (*disjoint classes*), uma vez que conceitos como alarme, elementos de tubulação, processos, etc., inclusive suas subclasses, são elementos distintos. Essa definição foi necessária porque o OWL assume que classes podem estar sobrepostas (ter interseção), o que torna necessário delimitá-las explicitamente declarando-as disjuntas (BRILHANTE, 2005).

Após a definição das classes da ontologia e sua associação em uma hierarquia taxonômica, o próximo passo escolhido foi a definição das propriedades, que será abordada em detalhes a seguir.

4.2.5 Definição das propriedades

Em OWL, as propriedades representam relacionamentos entre dois indivíduos, podendo ser de três tipos (LICHTENSTEIN; SIGULEM, 2008):

- 1) propriedades de objetos, que servem para ligar indivíduos a indivíduos;
- 2) propriedades de tipos de dados (*Datatype*), que servem para ligar um indivíduo a um valor de tipo de dado em XML-Schema ou a um literal em RDF;

- 3) propriedades de anotação, que podem ser utilizadas para adicionar informação (metadados) a respeito de classes, indivíduos e propriedades.

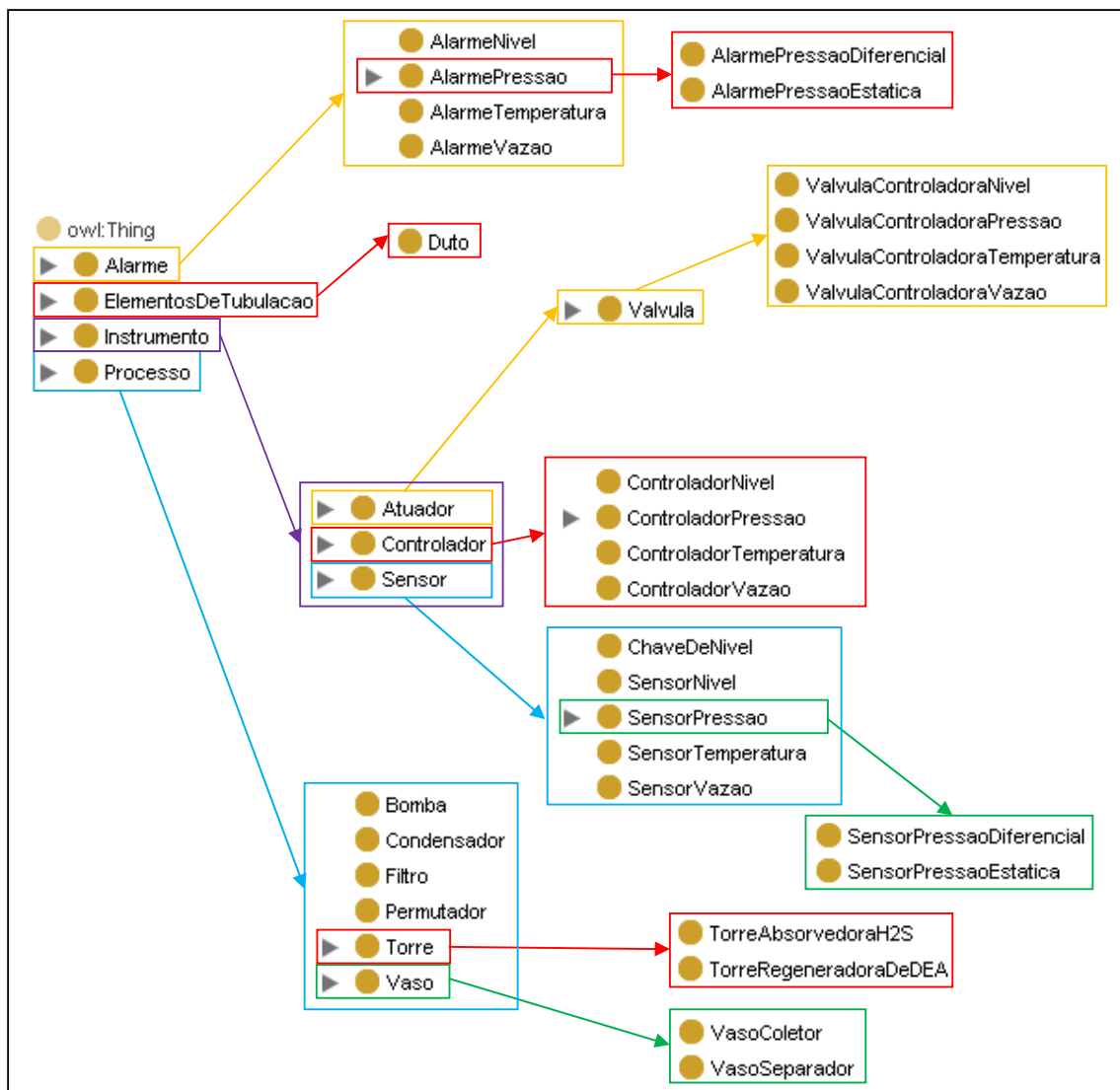


Figura 4.2. Classes e taxonomia definidas para a ontologia.

Fonte: Elaborado pelo autor (2010).

De maneira análoga às classes, que podem ser especializadas através da definição de subclasses, as propriedades de objetos também podem ser desdobradas em subpropriedades, que servem para especializar a ligação entre classes. Para exemplificar esse conceito, seria possível partir de uma propriedade que relaciona os familiares de um indivíduo “temFamiliar” e especializá-la através de subpropriedades tais como: “temIrmaos”, “temPrimos”, “temTios”, etc.

Assim, para a modelagem da planta DEA foram definidas apenas propriedades de objetos, consideradas pelo autor suficientes para representação da planta petroquímica. Além de definir esse rol de propriedades, também foi necessário especificar para cada uma delas um domínio (*domain*) e sua respectiva imagem (*range*), que era o mesmo que definir as classes que seriam conectadas pelas propriedades.

Embora nem sempre seja possível, buscou-se definir uma nomenclatura para as propriedades de acordo com a recomendação de Horridge *et al.* (2008, p.19):

Embora não existam regras únicas para nomear propriedades, recomenda-se que os nomes das propriedades comecem com letra minúscula, sem espaço, e que a primeira letra da próxima palavra seja em maiúscula. Recomenda-se também que as propriedades tenham como prefixo a palavra *has* (tem), ou a palavra *is* (é), por exemplo, *hasPart* (*temParte*), *isPartOf* (*éParteDe*), *hasManufacturer* (*temFabricante*), *isProducerOf* (*éProdutoDe*).

Para ajudar a seguir a recomendação de iniciar o nome das propriedades com os prefixos *has* e *is*, todas receberam denominações em língua Inglesa, o que não impede que no futuro tenham seus nomes modificados para o Português, de acordo com a necessidade dos usuários da ontologia.

Portanto, as propriedades básicas definidas para a ontologia estão listadas na tabela 4.1, onde também são apresentados o domínio (*domain*) e imagem (*range*) de cada uma delas, bem como uma descrição sucinta.

Nas propriedades que foram configuradas para possuírem múltiplas classes definidas como possíveis domínios e/ou imagens, o OWL interpreta essa definição de domínio e/ou imagem como a união das classes indicadas (BRILHANTE, 2005).

Conforme apresentado na tabela 4.1, o uso da propriedade *hasOutput* possibilita a representação do fluxo geral dos produtos na planta, uma vez que representa as conexões de todos os processos, válvulas e elementos de tubulação, mas não deixa claro o caminho percorrido por cada produto individualmente. A forma encontrada para solucionar essa lacuna foi especializar essa propriedade, conforme mencionado no início deste tópico, produzindo as subpropriedades listadas na tabela 4.2, que também herdaram as definições da super propriedade.

Na definição das subpropriedades também foi possível atender às recomendações de Horridge (2008), buscando estabelecer uma nomenclatura que evidenciasse os produtos que fluíam nos trechos da planta, conforme pode ser observado na descrição de cada uma das subpropriedades estabelecidas.

Tabela 4.1. Relação de propriedades básicas definidas para a ontologia.

| Propriedade | Domínio (domain) | Imagem (range) | Descrição |
|---------------------------|--|--|--|
| <i>hasAlarm</i> | Sensor Controlador | Alarme | Esta propriedade foi definida para indicar os alarmes da planta que estão associados a cada sensor ou controlador. |
| <i>hasController</i> | Atuador | Controlador | Esta propriedade foi definida para indicar quais os controladores que enviam sinais de acionamento para os atuadores instalados, que neste caso particular se restringiam a válvulas. |
| <i>hasOutput</i> | Processos ElementosDeTubulacao Válvula | Processos ElementosDeTubulacao Válvula | Esta propriedade foi definida para indicar o sentido dos fluxos na planta, representando as conexões entre os vários elementos. |
| <i>hasSensor</i> | Processo Duto | Sensor | Esta propriedade foi definida para indicar os sensores que estão instalados em dutos e processos, monitorando as grandezas físicas de controle (pressão, temperatura, vazão e nível). |
| <i>sendsExtraSignalTo</i> | Sensor | Controlador | Como os sensores possuem alarmes associados, mas também podem ser utilizados para enviar sinais para os controladores, esta propriedade foi utilizada para especificar o segundo caso. |

Fonte: Elaborado pelo autor (2010).

4.2.6 Definição das facetas

As facetas são características definidas para enriquecer as propriedades, como fruto da expressividade da linguagem OWL (ANTONIOU; HARMELEN, 2004). Conforme apresentado no capítulo 2, é possível definir a cardinalidade, os valores possíveis e as características relacionais das propriedades.

Portanto, algumas propriedades básicas puderam ser especializadas de acordo com esses conceitos, da mesma forma que foi possível definir as propriedades inversas, tanto para as propriedades básicas como para as subpropriedades, o que levou, finalmente, à definição do rol listado na tabela 4.3, ressaltando que as subpropriedades estão destacadas em cinza. A linguagem OWL ainda permitia especificar a cardinalidade e restrição de valores para as propriedades, mas as definições apresentadas foram consideradas suficientes para a modelagem da planta DEA.

Tabela 4.2. Relação de subpropriedades definidas para a propriedade *hasOutput*.

| Propriedade | Subpropriedade | Descrição |
|------------------|--------------------------|---|
| <i>hasOutput</i> | <i>hasDEAPGoingTo</i> | Foi definida para indicar os pontos do processo em que flui apenas a DEA pobre (DEAP), ou seja, a solução de dietanolamina pronta para ser utilizada na absorção de sulfeto de hidrogênio ou regenerada após uso. |
| | <i>hasDEARGoingTo</i> | Foi definida para indicar os pontos do processo em que flui apenas a DEA rica (DEAR), ou seja, a solução de dietanolamina contaminada com sulfeto de hidrogênio, após ter sido utilizada para absorver esse elemento químico. |
| | <i>hasGCGoingTo</i> | Foi definida para indicar os pontos do processo em que flui apenas o gás combustível (GC), que é o principal produto que se espera obter na saída da planta. |
| | <i>hasSGoingTo</i> | Foi definida para indicar os pontos do processo em que flui apenas o enxofre (S), que é o subproduto que se espera obter na saída da planta. |
| | <i>hasVaporGoingTo</i> | Foi definida para indicar os pontos do processo em que flui apenas o vapor, que é utilizado no interior da planta para fornecer calor a alguns processos. |
| | <i>hasDEAR_GCGoingTo</i> | Foi definida para indicar os pontos do processo em que flui uma solução composta por DEA rica e gás combustível. |
| | <i>hasGC_SGoingTo</i> | Foi definida para indicar os pontos do processo em que flui uma solução composta por gás combustível e sulfeto de hidrogênio. |
| | <i>hasS_DEARGoingTo</i> | Foi definida para indicar os pontos do processo em que flui uma solução composta por DEA rica e sulfeto de hidrogênio. |

Fonte: Elaborado pelo autor (2010).

A figura 4.3 apresenta a lista das propriedades definidas para a ontologia no *software* Protégé, onde também é possível observar que ao lado de cada propriedade está indicada a sua respectiva inversa.

No próximo tópico será abordado o último passo para a representação da planta DEA sob a forma de uma ontologia, que é a definição das instâncias.

4.2.7 Definição das instâncias

Com a conclusão da definição de classes, taxonomia, propriedades e facetas, para completar a ontologia restava apenas popular o conjunto de classes com as instâncias.

Tabela 4.3. Relação das propriedades definidas para a ontologia e suas respectivas características relacionais.

| (Sub)Propriedade | Propriedade Inversa | Característica Relacional | Descrição |
|---------------------------|-------------------------------|---------------------------|--|
| <i>hasAlarm</i> | <i>isAlarmOf</i> | Nenhuma foi definida | – |
| <i>hasController</i> | <i>isControllerOf</i> | Funcional | Um atuador possui apenas um controlador. |
| <i>hasOutput</i> | <i>isOutputOf</i> | Transitividade | Se um elemento <i>A</i> está conectado a um elemento <i>B</i> , que também está conectado com um elemento <i>C</i> , é possível concluir que <i>A</i> também está conectado a <i>C</i> . |
| <i>hasDEAPGoingTo</i> | <i>isReceivingDEAPFrom</i> | Transitividade | Pelo mesmo motivo da propriedade <i>hasOutput</i> . |
| <i>hasDEAR_GCGoingTo</i> | <i>isReceivingDEAR_GCFrom</i> | Transitividade | Pelo mesmo motivo da propriedade <i>hasOutput</i> . |
| <i>hasDEARGoingTo</i> | <i>isReceivingDEARFrom</i> | Transitividade | Pelo mesmo motivo da propriedade <i>hasOutput</i> . |
| <i>hasGCGoingTo</i> | <i>isReceivingGCFrom</i> | Transitividade | Pelo mesmo motivo da propriedade <i>hasOutput</i> . |
| <i>hasGC_SGoingTo</i> | <i>isReceivingGC_SFrom</i> | Transitividade | Pelo mesmo motivo da propriedade <i>hasOutput</i> . |
| <i>hasS_DEARGoingTo</i> | <i>isReceivingS_DEARFrom</i> | Transitividade | Pelo mesmo motivo da propriedade <i>hasOutput</i> . |
| <i>hasSGoingTo</i> | <i>isReceivingSFrom</i> | Transitividade | Pelo mesmo motivo da propriedade <i>hasOutput</i> . |
| <i>hasVaporGoingTo</i> | <i>isReceivingVaporFrom</i> | Transitividade | Pelo mesmo motivo da propriedade <i>hasOutput</i> . |
| <i>hasSensor</i> | <i>isInstalledOn</i> | Funcional | Um sensor está instalado em apenas um processo ou duto. |
| <i>sendsExtraSignalTo</i> | <i>receivesSignalFrom</i> | Nenhuma foi definida | – |

Fonte: Elaborado pelo autor (2010).

As classes em OWL dizem respeito a conjuntos que contêm indivíduos, sendo esses indivíduos conhecidos como instâncias dessas classes (HORRIDGE, 2008). Assim, para a classe “Duto”, por exemplo, deve ser definido um conjunto de várias instâncias, sendo cada uma delas correspondente a um duto da planta DEA, referenciado pela sua identificação (TAG) no processo, para o qual também devem ser indicadas todas as instâncias das demais classes com quem existe alguma relação através das propriedades.

Devido ao número de instâncias em uma ontologia ser tipicamente de uma ordem de magnitude muito maior que o número de classes, não é comum que a população das instâncias em uma ontologia seja realizada manualmente (ANTONIOU; HARMELEN, 2004). Para o

caso particular da ontologia da planta DEA foi possível povoar a ontologia manualmente, pois no final ela ficou apenas com um total de cento e quarenta e seis instâncias, sendo que a classe que possui o maior número delas é a classe “Duto”, com trinta e cinco indivíduos.



Figura 4.3. Relação de todas as propriedades diretamente no Protégé.

Fonte: Elaborado pelo autor (2010).

Em termos práticos, para povoar manualmente a ontologia no Protégé é necessário escolher a classe em que o indivíduo se enquadra, informar a identificação da instância (para o caso da planta DEA foram adotados os nomes indicados no fluxograma de engenharia ou na lista de alarmes) e depois indicar as demais instâncias com quem aquela instância possui ligações através das propriedades.

Conforme mencionado anteriormente, o preenchimento de cada instância da ontologia obedeceu rigorosamente às informações constantes do diagrama P&I, para que o produto final fosse fiel à descrição da planta.

Para exemplificar a forma de cadastramento das instâncias, vamos tomar como exemplo o cadastramento do controlador de nível LIC-98851, que será detalhado a seguir.

Para facilitar a compreensão do exemplo, a figura 4.4 apresenta um recorte da área de absorção da planta em que está situado o controlador de nível LIC-98851. Esse instrumento monitora o nível da torre de absorção (mostrada, em parte, no topo da figura) através do sinal que recebe do sensor LT-98851 e envia sinais de controle para a válvula (atuador) LV-98851.

Para cadastrar essa instância, devemos localizar a classe a que pertence; criar a nova instância e identificá-la através de sua nomenclatura no processo; em seguida, informar qual o instrumento (sensor) que lhe envia sinal; informar qual o atuador (válvula) que ele controla e, por fim, cadastrar os eventuais alarmes que estejam associados com o referido controlador, que foram obtidos em uma lista separada. Este procedimento é ilustrado através da figura 4.5.

O cadastramento de todas as instâncias é o último passo na criação da ontologia em si, embora ainda haja a necessidade de se verificar se ela possui alguma inconsistência para promover eventuais ajustes, o que será abordado a seguir.

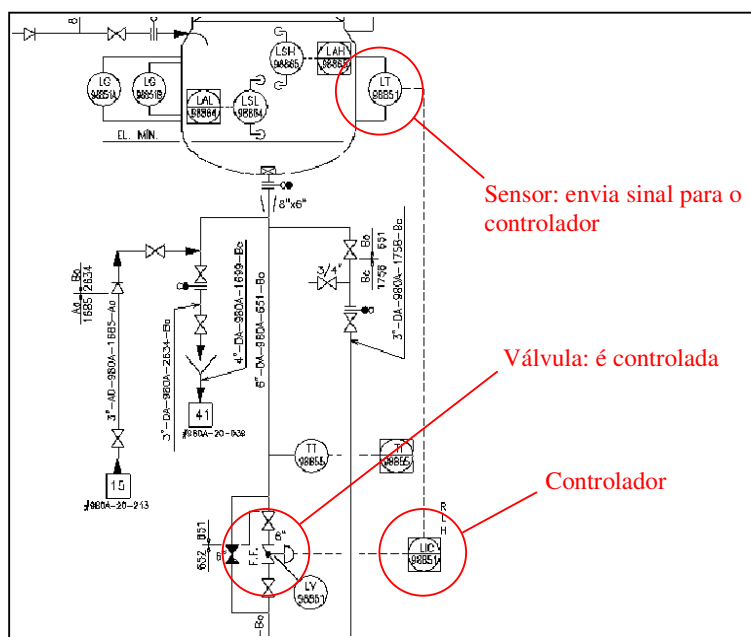


Figura 4.4. Trecho da planta DEA em que está situado o controlador LIC-98851.

Fonte: Adaptado de Petrobras (2001).

4.2.8 Checagem de inconsistências

Após a conclusão da ontologia, Antoniou e Harmelen (2004) recomendam que sejam checadas inconsistências.

A vantagem de utilizar o OWL como linguagem para o desenvolvimento das ontologias, combinada com a aplicação de ferramentas modernas, como é o caso do Protégé, é a

possibilidade de detectar inconsistências e erros, da mesma forma que um depurador em um ambiente de programação (SANTOS, 2006; ANTONIOU; HARMELEN, 2004).

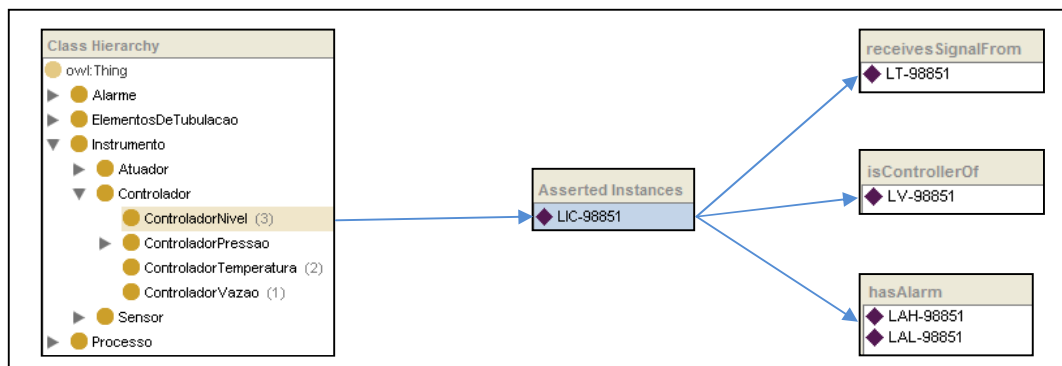


Figura 4.5. Procedimento de cadastramento de instância na classe “Controlador”.

Fonte: Elaborado pelo autor (2010).

No Protégé, a verificação de consistência (*consistency checking*) é um serviço padrão oferecido pelo mecanismo de inferência (*reasoner*). Esses mecanismos se baseiam na descrição das classes para verificar se as ontologias são consistentes. Eles também verificam se todas as classes podem ter instâncias, pois classes que não atendem a esse requisito são consideradas inconsistentes (HORRIDGE, 2008; SANTOS, 2006).

Para a verificação de consistência da ontologia foi utilizado o mecanismo de inferência que vem na versão 3.4.4 do Protégé, o Pellet 1.5.2. Através do uso desta ferramenta foi possível reparar algumas inconsistências verificadas durante o desenvolvimento da ontologia, embora em alguns momentos causas simples se propaguem por toda a ontologia, fazendo com que todas as classes apareçam marcadas em vermelho, como é o caso mostrado na figura 4.6a. Quando ocorria a propagação de inconsistências, a identificação da causa era obtida apenas após uma análise mais apurada do modelo.

Após o processo de verificação e ajustes, a ontologia fruto deste trabalho foi validada pelo Protégé. A figura 4.6b apresenta a tela que o Protégé fornece quando a ontologia é considerada consistente. Contudo, a tela de inconsistência irá variar de acordo com a causa.

4.3 Desenvolvimento da ontologia - conclusão

Através do processo descrito neste capítulo, foi possível modelar uma ontologia para descrever a planta DEA em seu fluxo normal de operação, condensando as informações contempladas em seus documentos descritivos e refinadas através de entrevista com o especialista na área.

Não foi objeto dessas análises a avaliação das mudanças nos fluxos em função do acionamento das válvulas manuais (*by-passes*) dos processos, o que pode ocorrer em situações de manutenção, por exemplo. A avaliação dessas situações pode ser realizada como um futuro aperfeiçoamento do modelo, uma vez que serviria apenas para enriquecer a representação e necessitaria de informações e avaliações adicionais. Na prática, o acionamento desses desvios poderia ser facilmente identificado, pois haveria redução na pressão e vazão dos trechos afetados.

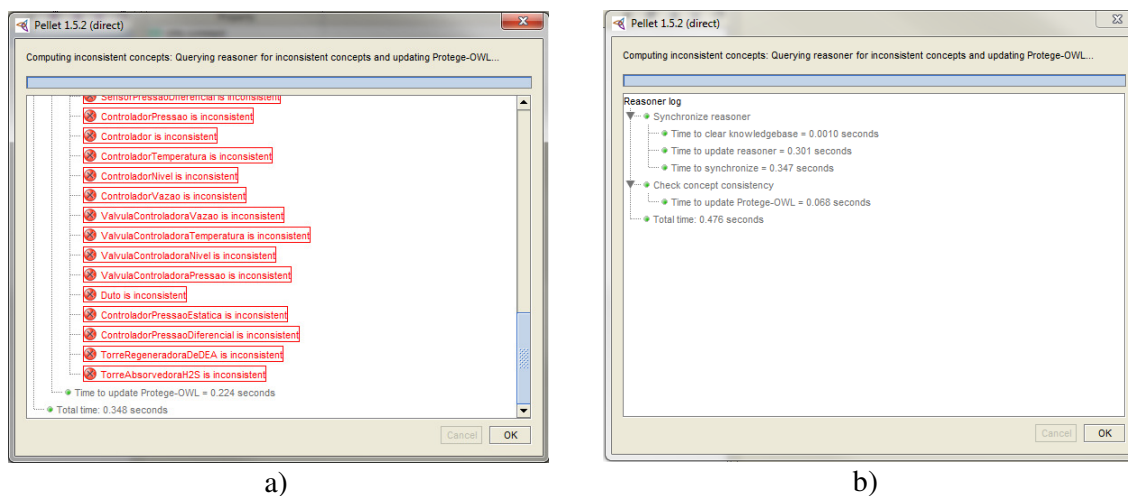


Figura 4.6. a) Tela do Protégé indicando a existência de inconsistências; b) Tela confirmando que a ontologia não possui inconsistências.

Fonte: Elaborado pelo autor (2010).

Com a conclusão da representação da planta, o próximo capítulo abordará a análise da representatividade do modelo e sua validação.

Capítulo 5

Análise de Resultados

5.1 Introdução

Após utilizar os passos mostrados no capítulo anterior, foi possível construir a ontologia da planta DEA seguindo seu diagrama P&D e a relação de alarmes, conforme inicialmente proposto.

Uma ontologia pode estar pronta e ser considerada consistente pelo mecanismo de inferência, mas ainda fica faltando uma validação quanto ao seu efetivo grau de representatividade do objeto de estudo. Essa validação fica difícil se a navegação entre as instâncias for realizada simplesmente abrindo cada uma delas e navegando pela ontologia na interface original do Protégé.

Este capítulo terá como objetivo a análise da ontologia obtida, com o intuito de provar que ela é de fato representativa da planta DEA. Para tanto será utilizado um *plugin* de visualização gráfica das ontologias desenvolvidas no Protégé, esperando-se que, ao final deste capítulo, a ontologia seja considerada válida através de uma análise simples e intuitiva.

5.2 O *plugin* Jambalaya

Conforme apresentado no item 2.3.2.2, o Protégé é um editor de ontologias desenvolvido em linguagem Java, com código aberto, que possibilita a seus usuários desenvolver *plugins* para expandir suas funcionalidades.

Há alguns anos o Chisel Group, localizado no Departamento de Ciência da Computação da Universidade de Victoria, vem trabalhando na área de visualização de ontologias e bases de conhecimento. O Jambalaya foi fruto do trabalho desse grupo, desenvolvido como uma extensão do conjunto de ferramentas de visualização gráfica ShriMP, reconfiguradas para serem utilizadas como um *plugin* para o Protégé, com o intuito de introduzir características para auxiliar na compreensão dos projetos desenvolvidos em RDF e OWL (LINTERN; STOREY, 2005).

O ShriMP inclui técnicas de visualização genérica e funcionalidades de navegação que facilitam o uso das habilidades cognitivas das pessoas quando estão utilizando telas de tamanhos limitados, que foram aplicadas no desenvolvimento do *plugin* (STOREY *et al.*, 2004).

No Jambalaya as classes e instâncias são representadas como nós em um grafo, que podem ser distinguidas através de diferentes tonalidades de cores. As relações entre os conceitos e instâncias são exibidas através das arestas (arcos), tais como relações “é-um” (*is-*

a) entre as classes no conceito de hierarquia; relações “é-instância-de” (*instance-of*) entre instâncias e classes (STOREY *et al.*, 2004).

Além dessas características, o Jambalaya também inclui modelos de visualização padrão do tipo aninhada (*nested view*), árvore de classes e indivíduos (*class & individual tree*) e domínio/imagem (*domain/range*), embora permita que o usuário também crie outras visualizações a partir destas.

O *plugin* utiliza alguns recursos comuns à maioria das ferramentas gráficas, tais como expandir e retrair vistas (*zoom*), ocultar e reexibir alguns trechos do grafo, reposicionar objetos do grafo com o *mouse*, etc. Outra funcionalidade importante é a possibilidade de rearranjar os objetos de um grafo na tela de acordo com alguns arranjos (*layouts*) predefinidos (*grid, radial, spring, tree - vertical, tree - horizontal*).

A aplicação deste *plugin* facilita muito a análise da ontologia, embora as visualizações geradas pelo Jambalaya geralmente precisem de ajustes manuais, porque quando o número de objetos utilizados na ontologia é grande, o *plugin* tem dificuldade de posioná-los na tela sem sobreposição, conforme pode ser observado na figura 5.1, que apresenta todos os elementos de tubulação e processos da planta DEA, incluindo a visualização dos sensores conectados aos elementos.

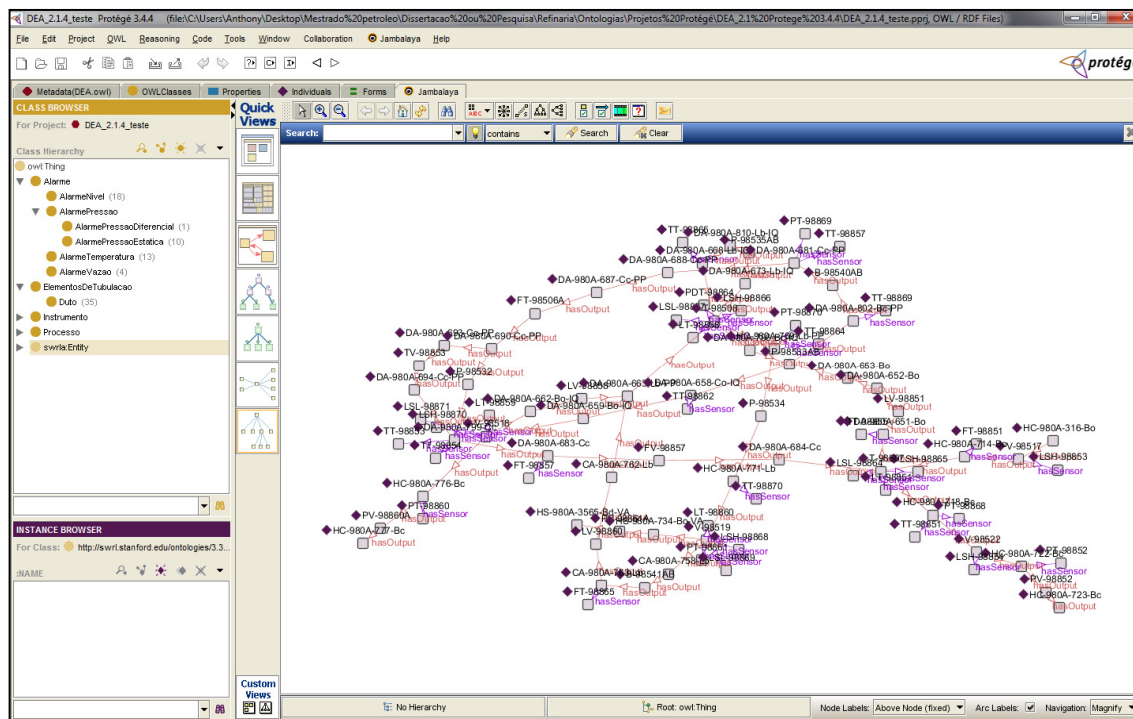


Figura 5.1. Visualização gerada no *plugin* Jambalaya de todos os elementos de tubulação e processos da planta DEA, inclusive sensores conectados.

Fonte: Elaborado pelo autor (2010).

Para criar o grafo de análise apresentado na figura 5.1 e os demais exibidos nos próximos itens, foi configurada uma visualização no Jambalaya do tipo *Spring*, que foi denominada *flow* (fluxo) e ajustada a cada etapa com a seleção da(s) propriedade(s) que deveria(m) ser exibida(s). A figura 5.2 apresenta um exemplo em que se está configurando a visualização *flow* para apresentar todos os elementos da planta conectados pela propriedade *hasOutput*, inclusive todos os sensores conectados aos elementos pela propriedade *hasSensor*. Para que fossem geradas outras visualizações, bastava localizar as propriedades que se desejava exibir e pressionar o botão “Run Quick View”.

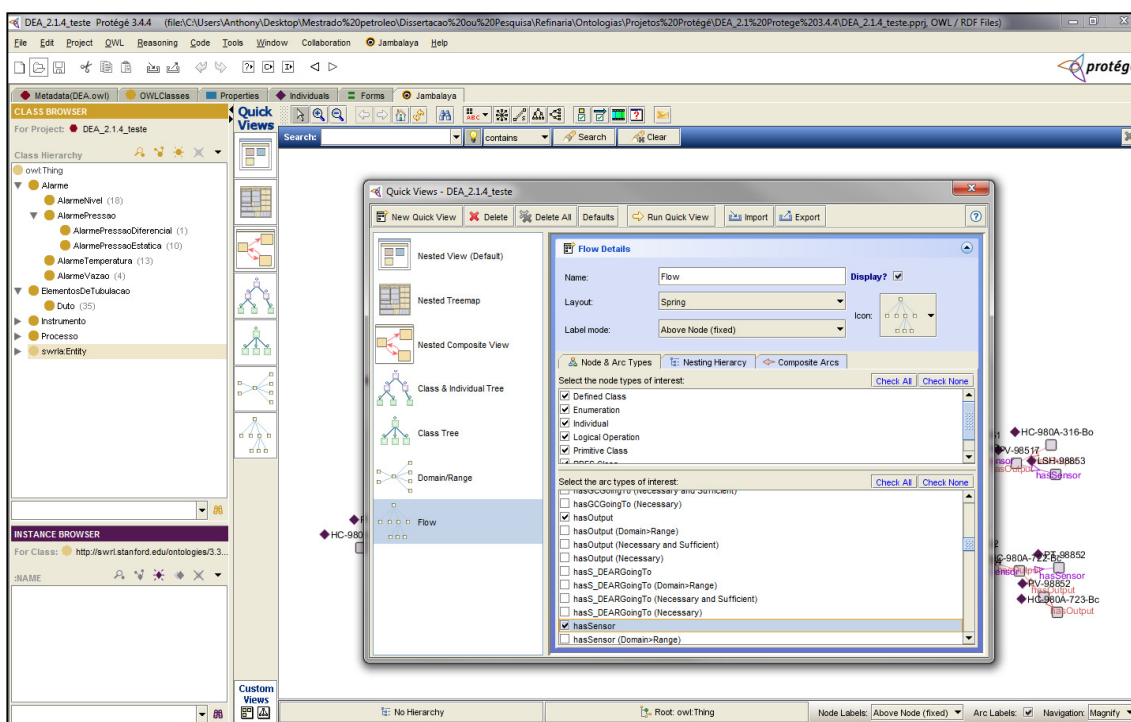


Figura 5.2. Menu de geração de visualizações no *plugin* Jambalaya.

Fonte: Elaborado pelo autor (2010).

As versões mais recentes do Protégé já passaram a disponibilizar esse *plugin* no próprio pacote do editor de ontologias, o que facilitou bastante a sua aplicação para as análises que serão mostradas nos tópicos seguintes.

5.3 Representação do fluxo dos produtos

Na etapa de modelagem, o objetivo da ontologia era descrever a planta DEA, os seus elementos constituintes e as conexões entre eles, inclusive indicando os instrumentos conectados aos processos e elementos de tubulação, bem como os alarmes configurados para

esses instrumentos. Dentro desse escopo também foi definido que a ontologia deveria indicar os fluxos percorridos pelos produtos na planta.

Em termos práticos, para validar parte do modelo, basta que sejam analisadas as representações dos fluxos, pois elas indicarão todos os elementos presentes em cada trecho da planta e suas respectivas conexões, que poderão ser comparados com a documentação descritiva da planta modelada. Entretanto, para validar a representação dos instrumentos e dos alarmes, faz-se necessário adicionar a indicação desses elementos à representação dos fluxos, que podem ser novamente comparadas com a documentação da planta para validação.

Com esse objetivo inicial, foi definida a propriedade *hasOutput*, que indica todos os pontos da planta em que acontecem conexões de dutos e processos, mas nesse primeiro momento ignora o tipo de produto que flui. Essa especialização do conceito será assunto dos tópicos seguintes.

Uma dificuldade verificada no uso do Jambalaya foi o tratamento das instâncias cujo nome é iniciado por um número, como é o caso dos dutos. A questão é que cada componente de uma ontologia deve ter um conjunto de *namespaces* XML, para prevenir colisões de nomes, isto é, para identificar unicamente os elementos da ontologia (LIMA; CARVALHO, 2005). Na aba inicial do Protégé existe uma área onde são sugeridos *namespaces*, que podem ser modificados pelo usuário e ficam como prefixo dos elementos da ontologia.

Quando são visualizadas as classes ou propriedades na tela, seja através das abas usuais do Protégé ou mesmo do *plugin* Jambalaya, normalmente os prefixos são suprimidos da exibição. Contudo, quando qualquer elemento da ontologia recebe um nome iniciado por um número, o *software* deixa de suprimir o prefixo quando o exibe na tela, conforme pode ser verificado na figura 5.3.

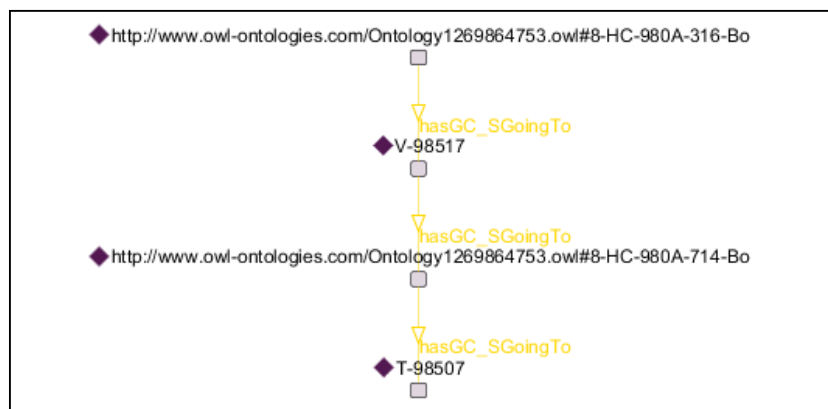


Figura 5.3. Visualização de instâncias cujo nome inicia com um número.

Fonte: Elaborado pelo autor (2010).

Durante a pesquisa não foi obtida explicação para esse fato, que dificultava a visualização das instâncias relacionadas aos dutos da planta, se fossem cadastradas pelas TAGs exatamente como mostradas nos fluxogramas de tubulação e instrumentação. Apenas para prevenir esse efeito e facilitar as análises, os nomes das instâncias dos dutos foram cadastrados sem o prefixo que indicava o diâmetro da tubulação.

Por fim, a ilustração do caminho percorrido por todos os produtos na planta é apresentada na figura 5.4, onde os nós entre os elementos correspondem à conexão das instâncias através da propriedade *hasOutput*, conforme foi destacado no canto inferior direito.

Para facilitar a visualização do grafo, o nome da propriedade entre os objetos foi propositalmente ocultado.

Por fim, através da comparação do grafo com os fluxogramas de tubulação e instrumentação é possível constatar que a ontologia consegue representar todas as conexões presentes no processo petroquímico em seu modo de operação normal.

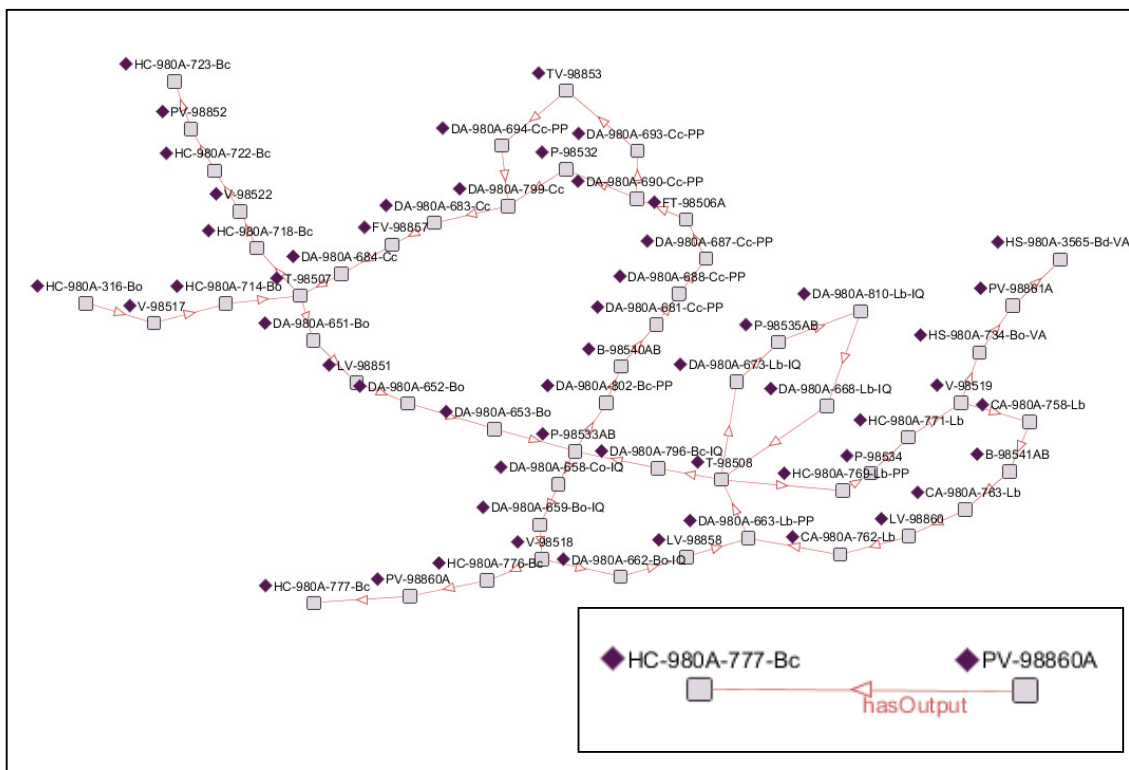


Figura 5.4. Representação da ligação entre todos os elementos da ontologia presentes na planta DEA, indicando o fluxo geral de todos os produtos.

Fonte: Elaborado pelo autor (2010).

5.3.1 Representação do fluxo de gás combustível com sulfeto de hidrogênio

Como o objetivo da planta é remover o sulfeto de hidrogênio do fluxo de gás combustível, o principal produto na entrada é justamente a mistura desses dois compostos. A mistura entra pelo duto “(8’’)HC-980a-316-Bo” e segue até a torre absorvedora de H₂S (T-98507), que é o ponto do processo em que vai entrar em contato com a solução aquosa de DEA pobre, para que o sulfeto de hidrogênio (H₂S) seja absorvido. Para representar esse fluxo inicial, a propriedade *hasOutput* foi especializada através da subpropriedade *hasGC_SGoingTo*, obtendo-se o resultado mostrado na figura 5.5, que corresponde ao fluxo descrito no respectivo fluxograma de tubulação e instrumentação.

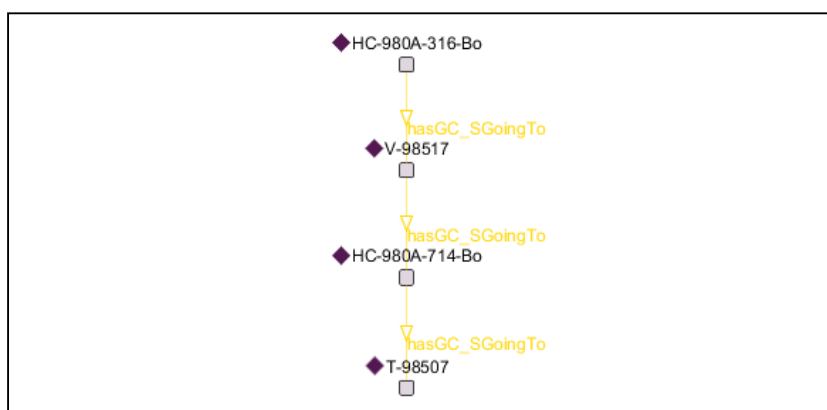


Figura 5.5. Representação do trecho em que fluem gás combustível e sulfeto de hidrogênio juntos.

Fonte: Elaborado pelo autor (2010).

5.3.2 Representação do fluxo de gás combustível

Após chegar à torre absorvedora de H₂S (T-98507), a mistura gasosa composta por gás combustível e sulfeto de hidrogênio irá entrar em contato com a solução aquosa de dietanolamina (DEA), que recebe a nomenclatura de DEA pobre, em razão de ainda estar isenta da solução ácida.

Na torre, a solução de DEA pobre irá absorver o sulfeto de hidrogênio, passando a ser chamada de solução de DEA rica, e irá deixar a torre por uma saída no fundo. Por outro lado, o gás combustível irá sair pelo topo da torre, deixando o sulfeto de hidrogênio que foi absorvido pela dietanolamina. A partir desse ponto, o gás combustível será direcionado para os fornos, saindo do domínio da planta estudada, tendo como último ponto de passagem o duto “(8’’)HC-980a-723-Bc”.

Por outro lado, existe uma outra corrente de gás combustível em decorrência da DEA rica conseguir carregar um pouco do gás na solução, que irá fluir com essa mistura, mas conseguirá ser liberado no topo do vaso separador de hidrocarbonetos (V-98518), de onde seguirá para ser aproveitado em outro ponto da refinaria, tendo como último ponto de passagem o duto “(4”-)HC-980A-777-Bc”.

Para representar os fluxos de gás combustível, a propriedade *hasOutput* foi especializada através da subpropriedade *hasGCGoingTo*. Esses fluxos estão representados na figura 5.6.

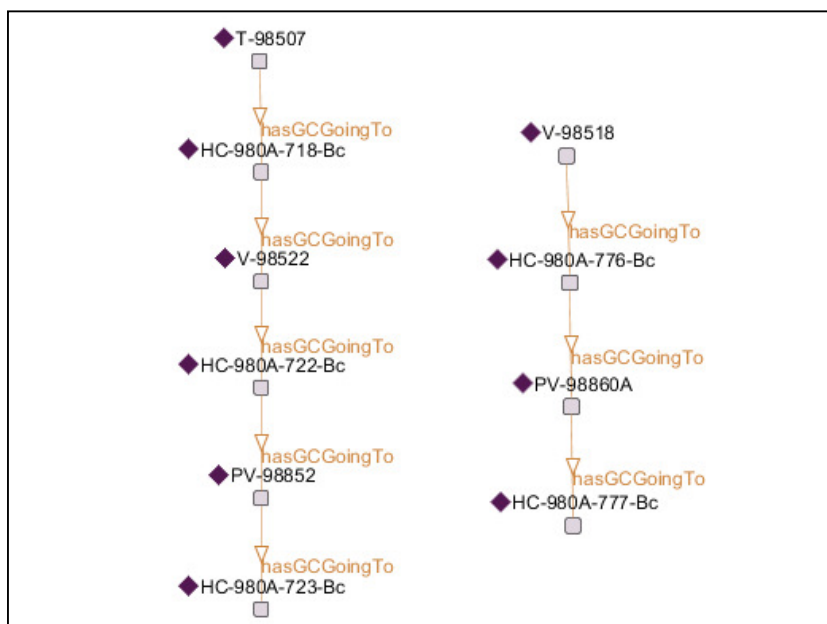


Figura 5.6. Representação dos trechos em que flui apenas gás combustível.

Fonte: Elaborado pelo autor (2010).

Comparando-se mais uma vez o grafo com os fluxogramas de tubulação e instrumentação, verifica-se que a ontologia consegue representar as referidas conexões da planta analisada.

5.3.3 Representação do fluxo de DEA rica com gás combustível

Conforme mencionado no item anterior, quando a solução aquosa de DEA pobre entra em contato com a corrente contrária de gás combustível e sulfeto de hidrogênio, além de absorver o sulfeto de hidrogênio, a dietanolamina também absorve um pouco do gás combustível.

O gás combustível sai da torre absorvedora de H₂S e segue o caminho mostrado no ramo esquerdo apresentado na figura 5.6. Por outro lado, a solução de DEA rica contaminada

com um pouco de gás combustível sai pela parte de baixo da torre absorvedora, para que a dietanolamina siga para regeneração. Nesse trajeto, o gás combustível passa em alguns dutos, em uma válvula, bem como no permutador de DEA pobre x DEA rica (P-98533A/B), até alcançar o vaso separador de hidrocarbonetos (V-98518), onde o gás combustível vai ser separado da mistura e seguir o trajeto mostrado no ramo direito da figura 5.6.

Esse fluxo percorrido pela mistura de DEA rica com gás combustível foi representado através da especialização da propriedade *hasOutput*, ou seja, de sua subpropriedade *hasDEAR_GCGoingTo*, conforme é mostrado na figura 5.7.

Quando são verificados os dois fluxogramas de tubulação e instrumentação que indicam o trecho por onde flui a DEA rica contaminada com gás combustível, é possível constatar que a ontologia representa corretamente o referido trecho.



Figura 5.7. Representação do trecho da planta em que flui DEA rica com gás combustível.

Fonte: Elaborado pelo autor (2010).

5.3.4 Representação do fluxo de DEA rica

Da mesma forma que aconteceu com o fluxo de gás combustível, há dois fluxos de DEA rica na planta.

O primeiro desses fluxos vai ter origem no vaso separador de hidrocarbonetos (V-98518), onde a DEA rica contaminada com gás combustível teve este segundo componente separado e liberado pela saída na parte superior do vaso. Da parte inferior do vaso será liberada a solução de DEA rica que seguirá para regeneração na torre regeneradora (T-98508), de onde sairão outros produtos.

O segundo fluxo ocorre porque do topo da torre regeneradora (T-98508) sairá o sulfeto de hidrogênio liberado da solução de DEA rica, mas esse fluxo consegue arrastar um pouco de DEA rica junto. Por esse motivo, ocorrerá uma nova separação no vaso de topo da regeneradora (V-98519), onde será liberada DEA rica, que será devolvida à torre T-98508.

Esses dois fluxos estão representados na figura 5.8, através da especialização da propriedade *hasOutput*, ou seja, de sua subpropriedade *hasDEARGoingTo*. O ramo esquerdo representa o primeiro deles e o ramo direito representa o segundo, ambos se encontrando no duto “(6’’)DA-980A-663-Lb-PP”, que fica na entrada da torre regeneradora T-98508.

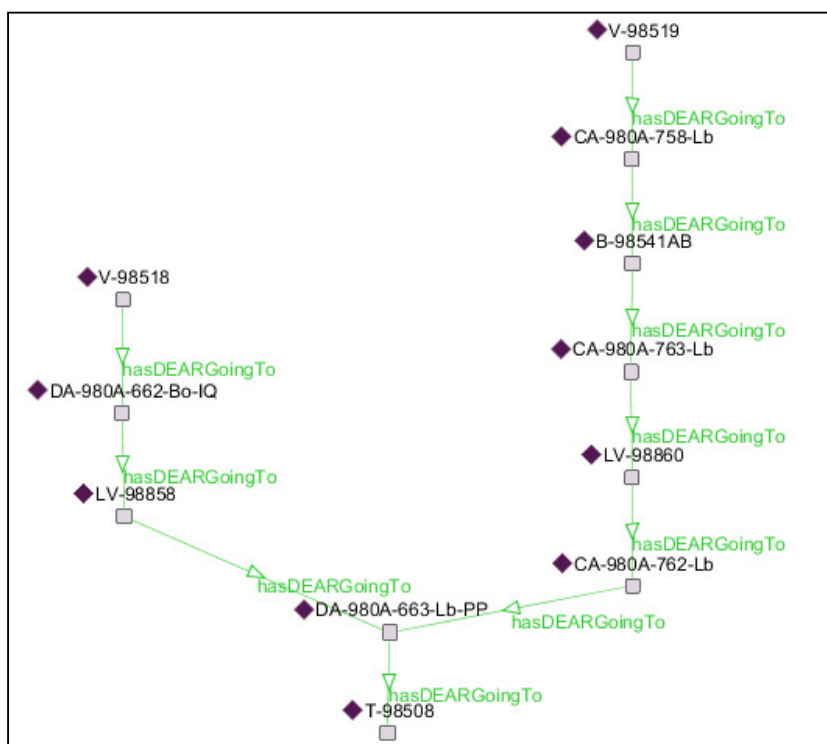


Figura 5.8. Representação do trecho da planta em que flui apenas DEA rica.

Fonte: Elaborado pelo autor (2010).

Portanto, a ontologia da planta DEA para os fluxos apresentados na figura 5.8, um deles com origem no vaso separador de hidrocarbonetos (V-98518) e o outro com origem no vaso de topo da regeneradora (V-98519), convergentes no trecho de duto de entrada da torre regeneradora (T-98508), pode ser considerada representativa dos caminhos indicados no fluxograma de tubulação e instrumentação da área de regeneração da planta DEA, estando em conformidade com as referidas informações.

5.3.5 Representação do fluxo de vapor

Para que ocorram tanto a reação de absorção do sulfeto de hidrogênio pela dietanolamina, como de regeneração da DEA rica, além das características da amina, também interferem nesse processo a pressão e a temperatura das torres onde ocorrem as reações.

O fluxo de vapor se insere nesse contexto, pois uma corrente de vapor supersaturado de baixa pressão circula na parte inferior da torre de regeneração da DEA (T-98508), com o objetivo de fornecer calor à solução DEA rica, favorecendo a liberação do sulfeto de hidrogênio da solução. Esse fluxo fica circulando dos refeedores da regeneradora (P-98535A/B) para a torre (T-98508), através de dutos que conectam esse sistema.

Esse fluxo percorrido pelo vapor foi representado através da especialização da propriedade *hasOutput*, ou seja, de sua subpropriedade *hasVaporGoingTo*, conforme é mostrado na figura 5.9.

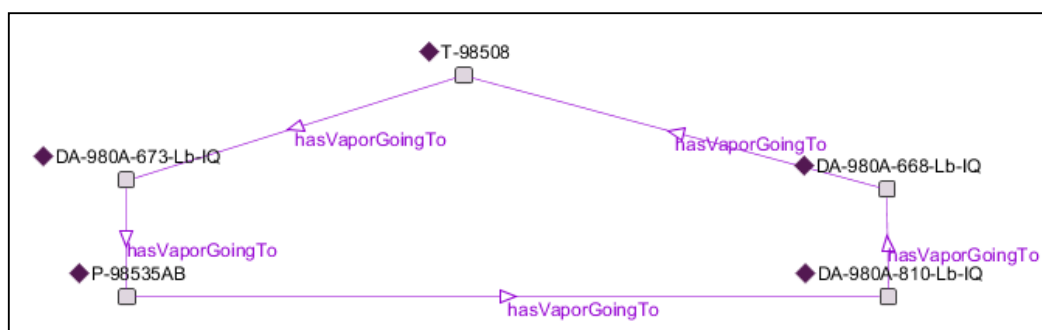


Figura 5.9. Representação do trecho da planta em que flui apenas vapor.

Fonte: Elaborado pelo autor (2010).

Da mesma forma que nos trechos anteriormente comentados, a partir da verificação do fluxograma de tubulação e instrumentação da área de tratamento e regeneração é possível constatar que a ontologia representa corretamente o referido trecho.

5.3.6 Representação do fluxo de sulfeto de hidrogênio com DEA rica

Embora o conceito de DEA rica possa levar a uma confusão entre esse fluxo com o descrito no item 5.3.4, é importante iniciar este tópico esclarecendo que são fluxos distintos.

Quando ocorre a regeneração da DEA rica na torre T-98508, o sulfeto de hidrogênio gasoso é liberado pelo topo da torre, mas carrega consigo um pouco da solução de DEA rica. Nessa situação, o produto principal que sai pelo topo da T-98508 é o sulfeto de hidrogênio, mas ele carrega consigo um pouco de DEA rica, que deverá ser separada e devolvida para a torre. No item 5.3.4, o produto principal é uma mistura da dietanolamina com o sulfeto de hidrogênio, que ainda passará por regeneração no futuro.

Essa solução gasosa vai para o condensador de topo da regeneradora (P-98534), onde parte dessa corrente será condensada, seguindo para o vaso de topo da regeneradora (V-98519), de onde sairão o gás ácido e a DEA rica separados, sendo que a partir desse vaso a DEA rica seguirá o ramo direito da figura 5.8, anteriormente descrito, com o objetivo de retornar à T-98508.

Esse fluxo percorrido pela mistura de sulfeto de hidrogênio com DEA rica também foi representado através da especialização da propriedade *hasOutput*, ou seja, de sua subpropriedade *hasS_DEARGoingTo*, conforme é mostrado na figura 5.9.

O trecho comentado também reflete o fluxo desses produtos conforme ilustrado no fluxograma de tubulação e instrumentação da área de tratamento e regeneração.

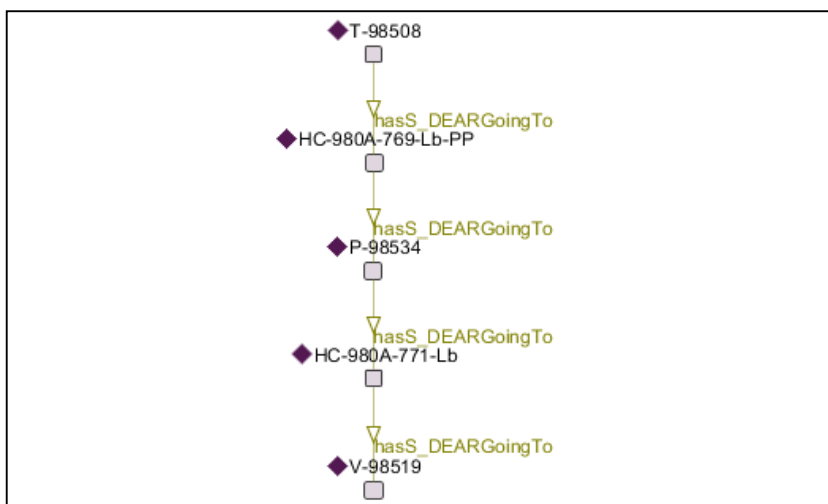


Figura 5.10. Representação do trecho da planta em que fluem sulfeto de hidrogênio e DEA rica.

Fonte: Elaborado pelo autor (2010).

5.3.7 Representação do fluxo de sulfeto de hidrogênio

Da mesma forma que o fluxo de gás combustível com sulfeto de hidrogênio, a descrição deste fluxo foi muito simples.

Esse fluxo tem origem no vaso de topo da regeneradora (V-98519), após o processo descrito no item anterior. A fase gasosa deixa o vaso V-98519, ao comando do atuador PV-98861A, seguindo para a unidade de regeneração de enxofre (URE), embora o último ponto em que passe no processo seja o duto “(8’’)HS-980A-3565-Bd-VA”, que é o final do ramo descrito.

Esse fluxo percorrido pela corrente gasosa de sulfeto de hidrogênio, assim como os demais, foi representado através da especialização da propriedade *hasOutput*, ou seja, de sua subpropriedade *hasSGoingTo*, conforme é mostrado na figura 5.11.

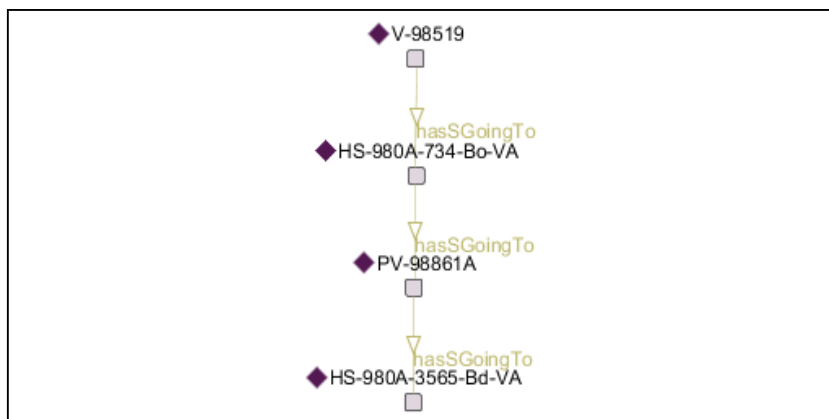


Figura 5.11. Representação do trecho da planta em que flui apenas sulfeto de hidrogênio.

Fonte: Elaborado pelo autor (2010).

O trecho mostrado na figura 5.11 reflete o fluxo da corrente gasosa de sulfeto de hidrogênio, conforme ilustrado no fluxograma de tubulação e instrumentação da área de tratamento e regeneração.

5.3.8 Representação do fluxo de DEA pobre

Dentre os fluxos analisados, o de DEA pobre foi o mais complexo.

Após a regeneração da DEA rica na torre T-98508, conforme explicado no item anterior, o fluxo gasoso de sulfeto de hidrogênio (e um pouco de DEA rica) sai pelo topo da

torre para ser encaminhado à URE, enquanto por baixo da torre T-98508 sai a DEA pobre para ser reaproveitada.

Essa DEA pobre segue para os permutadores de DEA rica x DEA pobre (P-98533A/B), onde perde um pouco de calor através da troca com o fluxo de DEA rica, sendo parcialmente resfriada. Dos permutadores, a DEA pobre segue para a sucção das bombas de circulação de solução DEA (B-98540A/B), para em seguida ir para o filtro para fluido de processo (FT-98506A), e em seguida para o resfriador de DEA pobre (P-98532).

Antes de retornar à torre absorvedora de H₂S (T-98507), a DEA é resfriada no resfriador de DEA pobre (P-98532), utilizado para controlar a temperatura da DEA em uma faixa que favoreça a absorção de H₂S na torre. A ontologia também representou um desvio controlado do permutador, através do atuador TV-98853, que passa pelo casco do equipamento, e é controlado pelo TIC-98853 (controlador de temperatura).

Pouco antes da DEA pobre entrar na torre T-98507, ela ainda passa pelo atuador FV-98857, que é controlado pelo FIC-98857 (controlador de vazão).

O fluxo percorrido pela DEA pobre foi representado através da especialização da propriedade *hasOutput*, ou seja, de sua subpropriedade *hasDEAPGoingTo*, conforme é mostrado na figura 5.12.

Com a chegada à torre T-98507, completa-se a descrição do fluxo de DEA pobre, que realmente corresponde ao caminho percorrido de acordo com os fluxogramas de tubulação e instrumentação das áreas de absorção e de regeneração.

Assim, conclui-se a verificação da ontologia da planta DEA, quanto aos fluxos de produtos, restando analisá-la quanto à representação dos instrumentos e dos alarmes, o que será realizado a seguir.

5.4 Representação dos pontos de conexão de instrumentos

Além da representação dos fluxos de produtos, que foi obtida através da modelagem das conexões entre os elementos de tubulação e processos, bem como dos atuadores (válvulas) presentes na planta, o escopo da ontologia também compreendia a indicação dos pontos de conexão dos instrumentos na planta.

Na ontologia, a classe de instrumentos continha os atuadores, controladores e sensores, embora os atuadores já tenham sido representados na etapa anterior, devido a estarem localizados nos trechos de fluxo da planta, servindo geralmente de conexão entre dutos.

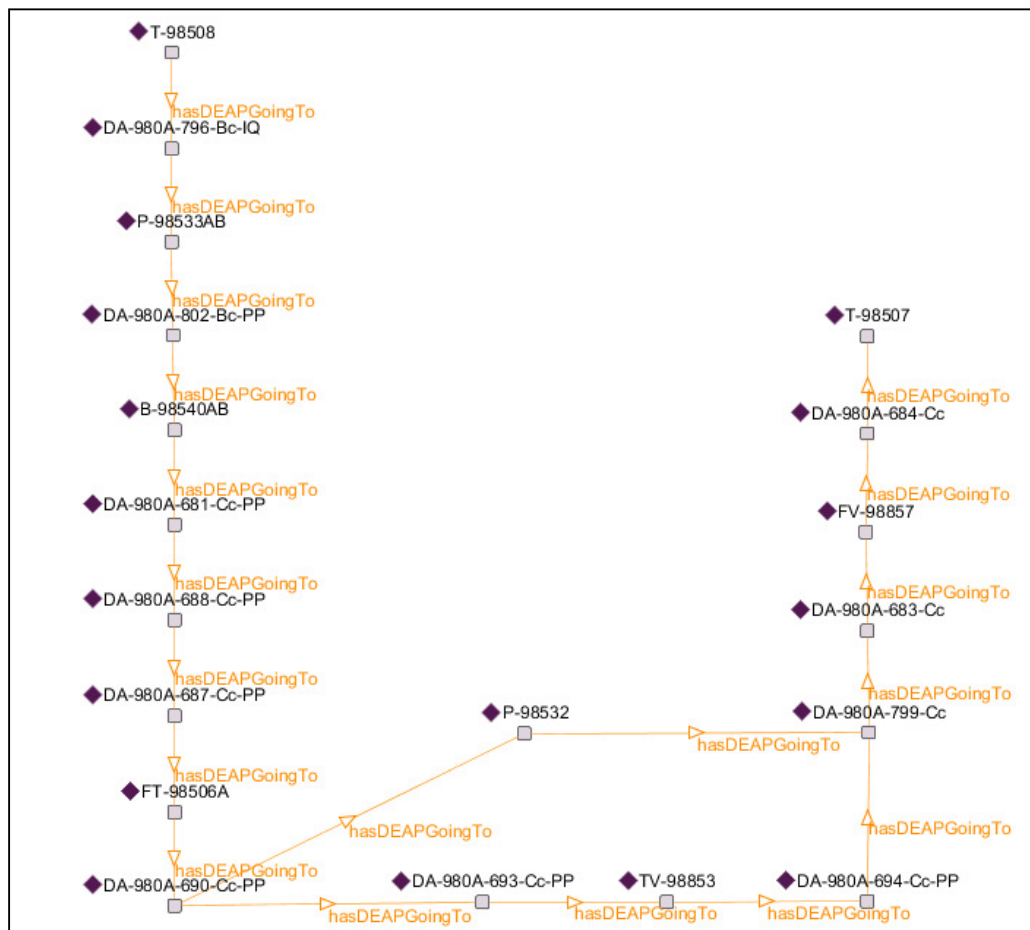


Figura 5.12. Representação dos trechos da planta em que flui apenas DEA pobre.

Fonte: Elaborado pelo autor (2010).

Portanto, nos itens seguintes será enfocada a representação dos sensores e elementos de controle na planta, completando a descrição dos pontos de conexão de instrumentos.

Para simplificar a visualização da ontologia será utilizado apenas o trecho de fluxo de DEA rica para análise de representatividade, uma vez que a aplicação das visualizações no modelo completo irá produzir um grafo de difícil compreensão, devido ao elevado número de elementos apresentados na tela.

5.4.1 Representação dos pontos de conexão dos sensores

Para garantir que uma planta industrial opere de acordo com as especificações, comumente os operadores acompanham a dinâmica de parâmetros químicos e/ou físicos preestabelecidos. Por essa razão, a planta DEA possui vários sensores conectados a elementos de tubulação e processos, com o intuito de monitorar pressão, vazão, temperatura e nível.

Com base nos valores indicados por esses sensores é que os controladores industriais enviarão sinais aos atuadores da planta ou haverá o disparo de alarmes indicadores de anormalidades operacionais.

Para representar os pontos de conexão de sensores, foi definida a propriedade *hasSensor* (inversa de *isInstalledOn*), utilizada para mapear os sensores de acordo com as indicações dos fluxogramas de tubulação e instrumentação.

As nomenclaturas utilizadas para os sensores na ontologia foram exatamente as mesmas indicadas nos fluxogramas de engenharia, que foram definidas de acordo com a norma ANSI/ISA S5.1.

O resultado da representação é apresentado na figura 5.13, lembrando que, embora estejam apresentados apenas os sensores do trecho correspondente ao fluxo de DEA rica, a ontologia contém a indicação de todos os sensores da planta.

Na figura 5.13, as conexões entre os elementos de tubulação e processos estão representadas pela propriedade *hasDEARGoingTo*, ilustrada graficamente pelas setas na cor verde clara, enquanto as conexões dos sensores aos demais elementos está indicada pela setas na cor roxa.

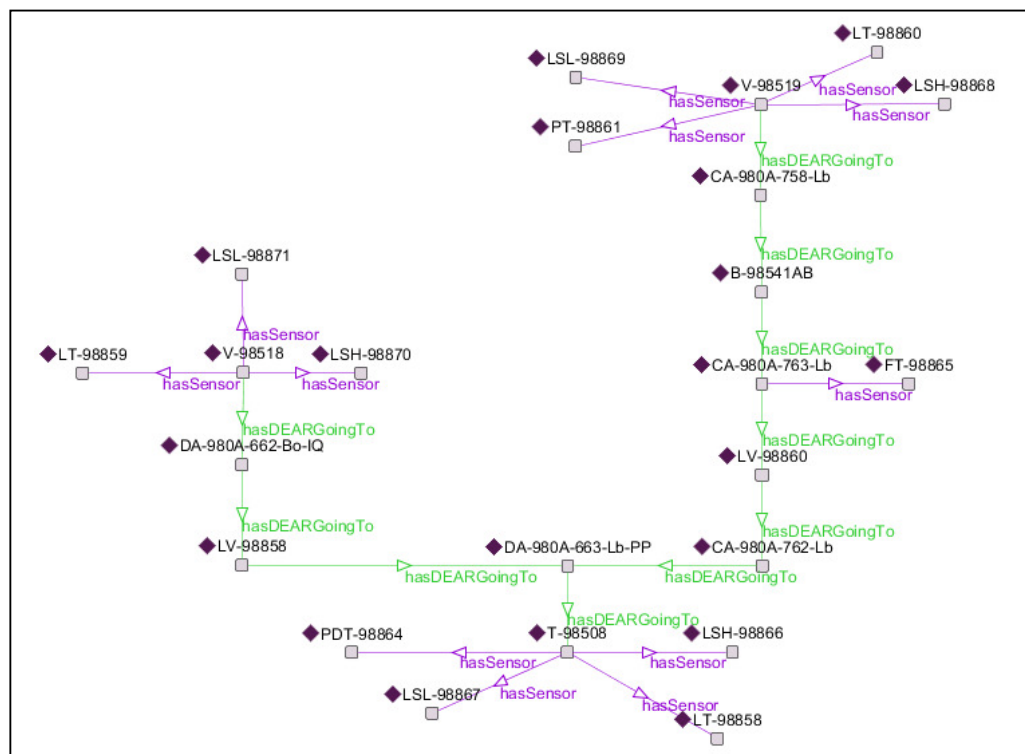


Figura 5.13. Representação dos sensores no trecho de fluxo da DEA rica.

Fonte: Elaborado pelo autor (2010).

Avaliando a conformidade da representação com os dados apresentados nos fluxogramas, é possível verificar que o modelo representa fielmente o posicionamento dos referidos sensores.

5.4.2 Representação dos elementos de controle

Modelada a parte de monitoramento, para concluir a representação dos elementos físicos na ontologia da planta restava apenas incluir os controladores.

Para que o processo sofra algum tipo de intervenção desses controladores é necessário que eles recebam indicações dos sensores (modelados no item anterior) e enviem sinais de controle para os atuadores do processo, que no caso da planta DEA são válvulas que já foram modeladas na etapa dos fluxos.

Para descrever as malhas de controle da planta, duas propriedades adicionais foram criadas: *receivesSignalFrom* e *isControllerOf*. A propriedade *receivesSignalFrom* (inversa de *sendsExtraSignalTo*) foi criada para indicar qual dos sensores da planta monitora a variável física sobre a qual o controlador atua no processo. Por outro lado, a propriedade *isControllerOf* (inversa de *hasController*) indica qual o atuador que recebe o sinal de controle enviado pelo controlador da variável de interesse.

Assim, a figura 5.14 apresenta novamente o trecho por onde flui apenas DEA rica, onde os elementos de tubulação e processos estão conectados pela propriedade *hasDEARGoingTo* e os sensores pela propriedade *hasSensor*, cujas cores das setas são as mesmas indicadas no item anterior. Adicionalmente, a figura apresenta os controladores, que recebem seus sinais de controle dos sensores a que estão conectados pela propriedade *receivesSignalFrom*, representada por setas na cor vermelha, e controlam os elementos a que estão conectados pela propriedade *isControllerOf*, representada por setas na cor verde escura.

Da mesma forma que ocorreu com as análises anteriores, o grafo exibido na figura 5.14 é representativo do trecho analisado de acordo com os fluxogramas de engenharia, tanto da área de absorção como de regeneração da DEA.

Com esse tópico encerram-se as análises da representação dos elementos físicos através da ontologia definida, que será finalizada com a inclusão dos alarmes associados com os instrumentos, conforme será mostrada a seguir.

dessa propriedade é apresentado na figura 5.15, que mostra todos os alarmes associados com cada instrumento do trecho onde flui apenas DEA rica na planta.

Na figura, as propriedades e cores mencionadas nos itens anteriores se repetem, com exceção da propriedade *hasAlarm*, que indica os alarmes associados a cada instrumento e é representada por setas na cor vermelha, da mesma forma que a representação da propriedade *hasSensor*.

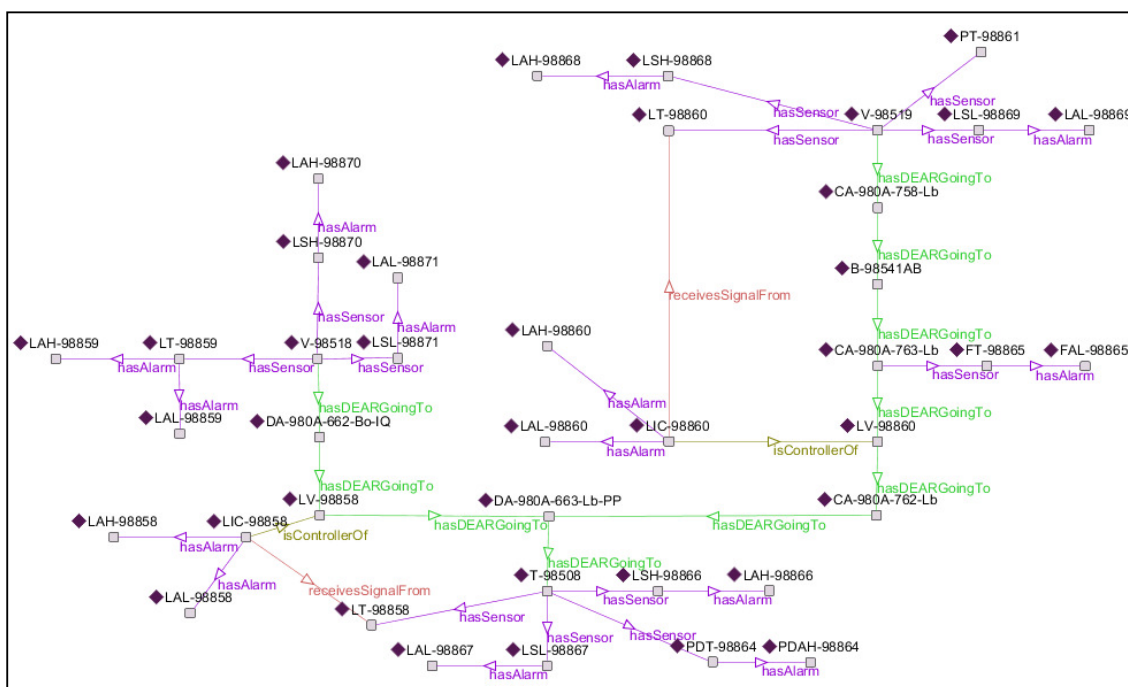


Figura 5.15. Representação dos elementos de controle no trecho de fluxo da DEA rica, inclusive dos alarmes associados aos sensores e controladores.

Fonte: Elaborado pelo autor (2010).

Os alarmes indicados na ontologia podem ser comparados com a lista originalmente mencionada, sendo de fato comprovada a representatividade da ontologia.

5.6 Análise da ontologia - conclusão

Neste capítulo, a adoção do *plugin* Jambalaya, distribuído junto com o *software* Protégé, possibilitou a apresentação de grafos para a visualização da ontologia desenvolvida para modelagem da planta estudo de caso.

Desta forma, foi possível verificar a representatividade da ontologia em relação ao escopo originalmente definido, pois os grafos tornaram possível a comparação dos modelos

com as informações disponibilizadas nos fluxogramas de engenharia do processo e a constatação de que ela representa os seus elementos, suas respectivas conexões e o fluxo dos produtos circulantes, que não puderam ser reproduzidos neste material, por se tratarem de informação de propriedade da Petrobras.

Para evitar a sobreposição de muitos arcos representativos das propriedades, como é ilustrado na figura 5.1, a análise relativa à representação dos instrumentos e alarmes apresentada neste texto contemplou apenas uma amostra do modelo, embora a validação do modelo tenha sido realizada com todos os trechos.

É importante lembrar que, embora as análises tenham sido realizadas com base em grafos, por trás das interfaces amigáveis do Protégé e do Jambalaya está a representação da ontologia em código OWL, da forma apresentada no item 2.3.2.1.

Aplicações importantes podem se desenvolver a partir de uma ontologia de domínio representativa do comportamento de uma planta de processo complexa. Dentre as possibilidades, uma ideia pode ser combinar essa informação com ontologias descritivas dos procedimentos operacionais, de controle e de segurança, para projetar sistemas inteligentes de diagnóstico de falhas, como sugerem Hangos, Németh e Lakner (2008).

Outra possibilidade é utilizar essa base de conhecimento como um dos elementos constituintes de um sistema de gerenciamento de riscos *offline*, onde se possam simular a propagação de falhas no sistema, que podem auxiliar na elaboração de documentos de segurança, como é o caso do HAZOP (*hazard and operability study*).

Laallam e Sellami (2007) desenvolveram uma ontologia para uma turbina a gás, com o objetivo de permitir aos agentes de manutenção o compartilhamento de uma compreensão comum sobre turbinas a gás e sua manutenção, bem como o reuso desse *know-how*. A ontologia da planta DEA poderia ter aplicação semelhante, ou até mesmo servir para o treinamento de novos operadores, porque condensa informações de vários documentos da planta em um forma mais simples.

Capítulo 6

Conclusões e Sugestões de Trabalhos Futuros

6.1 Conclusão

Ao final desta pesquisa foi possível representar uma planta real de adoçamento de gás natural, também conhecida como planta DEA, através de uma ontologia que indica os fluxos dos principais produtos da planta, em seu modo normal de operação, bem como os instrumentos de monitoramento e controle e ainda os alarmes associados aos mesmos.

Esta representação condensa informações do fluxograma de processo, dos fluxogramas de tubulação e instrumentação, da lista de alarmes disponível na planilha de racionalização, e dos descritivos de fluxo, que ainda foram refinadas através de uma entrevista não estruturada com um especialista do CENPES/Petrobras e de debates ocorridos junto ao grupo de pesquisa.

A adoção do editor de ontologia Protégé facilitou a criação da ontologia e o *plugin* Jambalaya possibilitou a visualização dos resultados, análise e comparação do produto final com os documentos utilizados como referência.

A ontologia desenvolvida pode ser interpretada como uma ferramenta de gestão do conhecimento, pois sintetiza todas as informações mencionadas anteriormente, e as mantém acessíveis para difusão junto a outros usuários, podendo inclusive facilitar o treinamento de operadores e a realização de aperfeiçoamentos na planta, por se tratar de um instrumento de simplificação de análises do sistema.

Do ponto de vista computacional, as ontologias se constituem em uma representação formal que pode ser utilizada por sistemas inteligentes para percorrer e selecionar informações relevantes, através de programas desenvolvidos em linguagem Java, que apliquem as *APIs* específicas Jena ou OWL. Outros conhecimentos também podem ser obtidos a partir de raciocínios realizados em cima da ontologia que se constitui em uma base de conhecimento.

A pesquisa aqui apresentada também pode ser aplicada para a descrição de outras plantas na refinaria, ou mesmo para outros processos industriais. Para tanto, basta que sejam seguidos os passos indicados a seguir, tomando este estudo de caso como referência:

- 1) Redefinir o escopo, coletando os documentos descritivos da(s) nova(s) planta(s), principalmente os fluxogramas de tubulação e instrumentação e a lista de alarmes. É desejável que haja uma conversa com um especialista na operação da nova planta, para que essas informações sejam complementadas e as dúvidas esclarecidas;
- 2) Na etapa de consideração de reuso, a ontologia desenvolvida nesta pesquisa pode servir como base para o novo modelo ou mesmo material orientativo;

- 3) Rever a lista de termos, buscando identificar as classes da ontologia atual que podem ser aproveitadas e as novas classes que precisarão ser cadastradas. Rever principalmente as classes que descrevem os instrumentos, uma vez que a planta analisada possuía apenas instrumentos associados com variáveis físicas (pressão, temperatura, vazão e nível), mas podem existir plantas que incluam dispositivos que controlem outras variáveis físicas não contempladas (peso, velocidade, etc.) ou mesmo variáveis químicas (concentração de soluções, pH, pOH, etc.). Também deve ser verificada a necessidade de cadastramento de novas relações entre as classes, que poderão se tornar novas propriedades;
- 4) Revisar a taxonomia, porque se houver introdução de novos termos no item anterior, a taxonomia poderá sofrer algum tipo de alteração;
- 5) Rever a lista de propriedades, podendo manter a que conecta os elementos (*hasOutput*), mas revendo as subpropriedades conforme revisão dos produtos que circulam. Quanto às propriedades que conectam instrumentos, alarmes, sensores, etc., poderiam ser mantidas;
- 6) Definir facetas para novas propriedades, mas apenas se forem cadastradas propriedades que descrevam realmente novas relações. Para o caso das subpropriedades geradas em função do fluxo de novos produtos na planta, não existe necessidade dessa definição, uma vez que elas irão herdar as facetas da propriedades *hasOutput* e sua inversa.
- 7) Repovoar as instâncias, cadastrando os elementos da ontologia de acordo com as *TAGs* e demais informações específicas do novo sistema. Nesta pesquisa, as instâncias foram cadastradas manualmente, mas pode ser estudado um cadastro automático, de acordo com a complexidade da nova planta;
- 8) Checar inconsistências ao final do processo;
- 9) Por fim, podem ser utilizados grafos montados a partir do *plugin* Jambalaya para validar a nova ontologia, sendo promovidos os devidos ajustes conforme forem sendo observadas as necessidades de melhoria.

Também é recomendável que o modelo final seja debatido junto aos especialistas que atuem na nova planta, lembrando que as ontologias são definidas como uma especificação formal de uma conceitualização compartilhada, o que indica a necessidade de haver consenso quanto à representação.

Em termos dos objetivos traçados no início da pesquisa, a ontologia criada atendeu ao escopo definido e está disponível como base de conhecimento para futuras aplicações.

6.2 Sugestão de trabalhos futuros

Sob a ótica da engenharia da computação, a ontologia desenvolvida pode ser considerada uma base de conhecimento, que pode ser utilizada no projeto de ferramentas de análise e diagnóstico da planta.

Essas ferramentas podem se beneficiar do uso de agentes inteligentes que combinem essa base de conhecimento com outras informações da planta, para auxiliar nas decisões de operadores ou no diagnóstico de falhas.

Os aplicativos que forem criados para utilizar a ontologia podem ser escritos em linguagem Java, pois existem *APIs* como a Jena e a OWL, que podem ser utilizadas para escrever programas que acessem e naveguem nas ontologias. Esse tipo de recurso pode facilitar o desenvolvimento de sistemas que simulem a propagação de falhas na planta, o que serviria para análise e diagnóstico desses sistemas, bem como para a elaboração de análises importantes de segurança, dentre as quais está o HAZOP, que é utilizado pela Petrobras.

Referências Bibliográficas

ALMEIDA, C. C. *O campo da ciência da informação: suas representações no discurso coletivo dos pesquisadores do campo no Brasil*. 2005. 395f. Dissertação (Mestrado em Ciência da Informação) – Centro de Ciências da Educação, Programa de Pós-Graduação em Ciência da Informação, Universidade Federal de Santa Catarina. Florianópolis.

ALMEIDA, M. B. *Um modelo baseado em ontologias para representação da memória organizacional*. 2006. 345f. Tese (Doutorado em Ciência da Informação), Programa de Pós Graduação da Escola de Ciência da Informação, Universidade Federal de Minas Gerais, Belo Horizonte. Disponível em: <http://www.enancib.ppgci.ufba.br/premio/UFMG_Almeida.pdf>. Acesso em: 05 junho 2010.

ALMEIDA, M. B.; BAX, M. P. Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção. *Ciência da Informação*, v.32, n.3, p.7-20, 2003.

ANTONIOU, G.; HARMELEN, F. V. *A semantic web primer*. Massachusetts: The MIT Press, 2004.

BARBOSA, D. P. *A influência do fator humano nos cenários acidentais de uma refinaria de petróleo*. 2009. 130 f. Dissertação (Mestrado em Sistemas de Gestão) – Centro Tecnológico, Mestrado Profissional de Sistemas de Gestão, Universidade Federal Fluminense, Niterói, 2009.

BORST, W. N. *Construction of engineering ontologies for knowledge sharing and reuse*. 1997. 243f. Thesis (Ph. D.) – Centre for Telematica and Information Technology, University of Twente. Enschede.

BRACHMAN, R. J.; LEVESQUE, H. J. *Knowledge representation and reasoning*. San Francisco: Elsevier, 2004.

BRILHANTE, V. *Linguagens para especificação de ontologias: OWL*. 2005. Disponível em: <<http://www.dcc.ufam.edu.br/~ontologias/slides/owl.pdf>>. Acesso em: 20 julho 2010.

BROWNE, W.; YAO, L.; POSTLETHWAITE, I.; LOWES, S.; MAR, M. Knowledge-elicitation and data-mining: fusing human and industrial plant information. *Engineering Applications of Artificial Intelligence* 19, p.345-359, 2005.

CARDOSO, J. Developing an OWL ontology for e-tourism. In: CARDOSO, J.; SHETH, A. P. **Semantic web services, processes and applications**. New York: Springer, 2006. P. 247-282.

CHENG, G.; DU, Q. The design and implementation of ontology and rules based knowledge base for transportation. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol XXXVII, Part B2, Beijing, 2008. Disponível em: <http://www.isprs.org/proceedings/XXXVII/congress/2_pdf/2_WG-II-2/30.pdf>. Acesso em: 31 maio 2010.

CRISTANI, M.; CUEL, R. A survey on ontology creation methodologies. *Int'l Journal on Semantic Web & Information Systems*, v.1, n.2, p.49-69, 2005. Disponível em: <http://www.infosci-journals.com/downloadPDF/pdf/ITJ2834_XhMFDg3SCU.pdf>. Acesso em: 16 junho 2010.

DAVIS, R.; SHROBE, H.; SZOLOVITS, P. What is a knowledge representation? *AI Magazine* 14(1), p.17-33, 1993.

DE FREITAS, F. L. G. Ontologias e a Web Semântica. In: JORNADA DE MINI-CURSOS EM INTELIGÊNCIA ARTIFICIAL, SBC, v.8, 2003. Disponível em: <http://www.inf.ufsc.br/~gauthier/EGC6006/material/Aula%203/Ontologia_Web_semantica%20Freitas.pdf>. Acesso em: 11 junho 2010.

DOKAS, I. M. Ontology to support knowledge representation and risk analysis for the development of early warning system in solid waste management operations. 2007. Disponível em: <[http://4c.ucc.ie/scewa/pdf/Ontology%20to%20support%20knowledge%](http://4c.ucc.ie/scewa/pdf/Ontology%20to%20support%20knowledge%20)

20representation%20and%20risk%20analysis%20for%20early%20warning%20system.pdf>.

Acesso em: 03 junho 2010.

ERIKSSON, H. Using JessTab to integrate Protégé and Jess. *Intelligent Information Processing*. p.43-50, march/april 2003.

FREIRE, J. L. F. *Engenharia de dutos*. Rio de Janeiro: ABCM, 2009.

FURGERI, S. *Representação de informação e conhecimento: estudo das diferentes abordagens entre a ciência da informação e a ciência da computação*. 2006. 160f. Dissertação (Mestrado) – Centro de Ciências Sociais Aplicadas, Programa de Pós-Graduação em Ciência da Informação, Pontifícia Universidade Católica de Campinas. Campinas.

GARCÊS, R.; FREITAS, P.; FERREIRA, N.; FREITAS, F. *Relatório: representação do conhecimento*. Funchal: Universidade da Madeira, 2006. Disponível em: <http://dme.uma.pt/edu/iia/acetatos%202005-2006/Apontamento%20rep_conhecimento.pdf>.

Acesso em: 20 maio 2010.

GARY, J. H.; HANDWERK, G. E. *Petroleum refining: technology and economics*. 4 ed. New York: Marcel Dekker, 2001.

GONZÁLEZ, R. G. *A semantic web approach to digital rights management*. 2005. 286f. Tese (Doutorado em Ciência da Computação e Comunicação Digital) – Departamento de Tecnologias, Universitat Pompeu Fabra, Barcelona. Disponível em: <<http://rhizomik.net/html/~roberto/thesis/Thesis.pdf>>. Acesso em: 31 maio 2010.

GÓMEZ-PÉREZ, A.; FERNÁNDEZ-LÓPEZ, M.; CORCHO, O. *Ontological engineering: with examples from the areas of knowledge management, e-commerce and the semantic web*. London: Springer, 2004.

GRIGOROVA, D.; NIKOLOV, N. Knowledge representation in systems with natural language interface. International Conference on Computer Systems and Technologies – CompSysTech'07, 2007, Rousse, Bulgária. Disponível em: <<http://ecet.ecs.ru.acad.bg/cst07/Docs/cp/SIII/IIIB.21.pdf>>. Acesso em: 27 maio 2010.

GRIMM, S.; HITZLER, P.; ABECKER, A. Semantic representation and ontologies: logics, ontologies and semantic web languages. In: STUDER, R.; GRIMM, S.; ABECKER, A. Semantic web services: concepts, technologies and applications. Berlin: Springer, 2007. P. 51-105.

GUARINO, N. Formal ontology and information systems. In: FOIS'98, 1998, Trento, Italy. **Proceedings...** Amsterdam: IOS Press, p.3-15. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.1776>>. Acesso em: 11 junho 2010.

HAAV, H. M.; LUBI, T. L. *A survey of concept-based information retrieval tools on the web*. 2001. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.109.6927>>. Acesso em: 09 junho 2010.

HANGOS, K.; NÉMETH, E.; LAKNER, R. A procedure ontology for advanced diagnosis of process systems. 2008. Disponível em: <http://daedalus.scl.sztaki.hu/PCRG/PCRG_publist_ISO.html>. Acesso em: 18 agosto 2010.

HEIJST, G. V.; SCHREIBER, A. T.; WIELINGA, B. J. Using explicit ontologies in KBS development. *International Journal of Human-Computer Studies*, v.46, n.2-3, p.183-292, February 1997.

HORRIDGE, M. *et al.* **Um guia prático para a construção de ontologias OWL, plugin Protégé-OWL 3.4**. Tradução de D. R. Soares e M. B. Almeida, 2008. Disponível em: <<http://www.das.ufsc.br/~gb/pg-ia/Protege08/OWL.pdf>>. Acesso em: 19 julho 2010.

HORROCKS. I.; PATEL-SCHNEIDER, P. F.; VAN HARMELEN, F. From SHIQ and RDF to OWL: the making of a Web ontology language. *Journal of Web Semantics*, 1, P. 7-26, 2003. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.7039>>. Acesso em: 11 junho 2010.

<http://paulinia.net/refinaria-de-paulinia-replan>, acesso em: 22 junho 2010.

http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library, acesso em: 15 setembro 2009.

KASSAHUN, A.; SCHOLTEN, H. A knowledge base system for multidisciplinary model-based water management. In: 3rd BIENNIAL MEETING OF THE INTERNATIONAL ENVIRONMENTAL MODELLING AND SOFTWARE SOCIETY, 2006, Vermont. **Proceedings...** Disponível em: <http://www.iemss.org/iemss2006/papers/s5/71_Kassahun_0.pdf>. Acesso em: 24 agosto 2010.

KIM, M.; KIM, M. Developing Protégé plug-in: OWL ontology visualization using social network. *Journal of Information Processing Systems*, v.4, n.2, p.61-66, 2008. Disponível em: <http://jips.kips.or.kr/dlibrary/JIPS_v04_no2_paper3.pdf>. Acesso em: 15 junho 2010.

LAALLAM, F. Z.; SELLAMI, M. Gas turbine ontology for the industrial processes. *Journal of Computer Science* 3, p.113-118, 2007.

LACLAVÍK, M. *Ontology and agent based approach for knowledge management*. 2005. 121f. Tese (philosophiae doctor) – Institute of Informatics, Slovak Academy of Sciences, Bratislava. Disponível em: <http://laclavik.net/publications/phd_2005-06-20.pdf>. Acesso em: 06 junho 2010.

LICHTENSTEIN, F.; SIGULEM, D. Criando uma ontologia em saúde com a ferramenta Protégé no padrão OWL. In: XI CONGRESSO BRASILEIRO DE INFORMÁTICA EM SAÚDE, 2008, Campos do Jordão. **Anais...** Disponível em: <<http://www.sbis.org.br/cbis11/arquivos/992.pdf>>. Acesso em: 14 junho 2010.

LIMA, J. C.; CARVALHO, C. L. *Ontologia – OWL (Web Ontology Language)*. 2005. Disponível em: <<http://ia.ucpel.tche.br/~lpalazzo/Aulas/IWS/m04/Recursos/OntoOWL.pdf>>. Acesso em: 15 agosto 2010.

LINTERN, R.; STOREY, M.-A. Jambalaya express: on demand knowledge visualization. In: 8th PROTÉGÉ CONFERENCE, 2005, Madrid. **Proceedings...** Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.129.3252>>. Acesso em: 22 julho 2010.

MANICA, H.; DANTAS, M. A. R.; TODESCO, J. L. Ontologia para compartilhamento e representação de conhecimento em saúde. *Diálogos & Saberes*, v.4, n.1, p.151-161, 2008.

MASTELLA, L. S.; ABEL, M.; LAMB, L. C.; DE ROS, L. F. Uma ontologia temporal para modelagem de conhecimento sobre ordenação de eventos. In: XXV CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 2005, São Leopoldo. **Anais...** Disponível em: <<http://www.sbc.org.br/bibliotecadigital/download.php?paper=321>>. Acesso em: 14 agosto 2010.

MEIRELES, N. R. Gestão do conhecimento/segurança. 2009. Disponível em: <<http://www.oguedes.com/site/seguranca-diversos/gest-o-do-conhecimento-na-seguranca-corporativa>>. Acesso em: 27 julho 2010.

MIZOGUCHI, R.; VANWELKENHUYSEN, J.; IKEDA, M. Task ontology for reuse of problem solving knowledge. In: ECAI'94 TOWARDS VERY LARGE KNOWLEDGE BASES, 1995, Amsterdam. **Proceedings...** Amsterdam: IOS Press, 1995. P.46-59.

MOKHATAB, S.; POE, W. A.; SPEIGHT, J. G. *Handbook of natural gas transmission and processing*. Burlington: Gulf Publishing, 2006.

NOY, N. F.; MCGUINNESS, D. L. *Ontology development 101: a guide to creating your first ontology*. 2001. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.136.5085>>. Acesso em: 16 junho 2010.

PARKASH, S. *Refining processes handbook*. Burlington: Gulf Publishing, 2003.

PASSOS, C. N.; ARAÚJO, M. A.; BRASIL, N. I.; CAMARGO, P. R. C. *Processos de refino*. Apostila do curso de Gestão em Engenharia de Petróleo e Gás Natural oferecido pelo IBEC, João Pessoa, 2006.

PEREIRA, H. J.; DONDONI, P. C. Gestão do conhecimento e prevenção a acidentes ambientais: estudo sobre as práticas de aprendizagem organizacional em empresa do setor de petróleo. *Produto & Produção*, v.9, n.1, p.14-40, fev. 2008.

PETROBRAS. *Unidade de tratamento DEA*. [s.d.]

PETROBRAS. *Área de tratamento DEA absorção*: fluxograma de tubulação e instrumentação. 2001a.

PETROBRAS. *Área de tratamento DEA regeneração*: fluxograma de tubulação e instrumentação. 2001b.

POLLOCK, J. T. *Semantic web for dummies*. Hoboken: Wiley Publishing Inc., 2009.

REZENDE, S. O. *Sistemas inteligentes: fundamentos e aplicações*. Barueri: Manole, 2005.

RODRIGUES, W. C. *Metodologia científica*. 2007. Disponível em: <http://www.ebras.bio.br/autor/aulas/metodologia_cientifica.pdf>. Acesso em: 28 julho 2010.

RUSSEL, S.; NORVIG, P. *Inteligência artificial*: tradução da segunda edição. Rio de Janeiro: Elsevier, 2004.

SANTOS, E. S. Uma proposta de integração de sistemas computacionais utilizando ontologias. 2006. 106 f. Dissertação (mestrado em informática) – Instituto de Ciências Exatas, Departamento de Ciência da Computação, Universidade de Brasília, Brasília.

SCHAUER, H.; SCHAUER, C. Modeling techniques for knowledge management. In: LYTRAS, M. D.; RUSS, M.; MAIER, R.; NAEVE, A. **Knowledge management strategies: a handbook of applied technologies**. New York: IGI Publishing, 2008, P. 91-115.

SCHREIBER, G. Knowledge engineering. In: VAN HARMELEN, F.; LIFSCHITZ, V.; PORTER, B. **Handbook of knowledge representation**. Amsterdam: Elsevier, 2008. P. 929-946.

SEMPREBOM, T.; CAMADA, M. Y.; MENDONÇA, I. *Ontologias e Protégé*. Artigo apresentado na disciplina de Inteligência Artificial Aplicada a Controle e Automação. Curso de Pós-Graduação em Engenharia de Automação e Sistemaas, Universidade Federal de Santa Catarina, 2007. Disponível em: <<http://www.das.ufsc.br/~gb/pg-ia/Protege07/ontologias.pdf>>. Acesso em: 14 junho 2010.

SOUZA, R. R.; ALVARENGA, L. A. A web semântica e suas contribuições para a ciência da informação. *Ciência da Informação*, v.33, n.1, p.132-141, jan/abr. 2004.

STANFORD - STANFORD CENTER FOR BIOMEDICAL INFORMATICS RESEARCH. <http://protege.stanford.edu/>. Acesso em: 14 junho 2010.

STOREY, M.-A.; LINTERN, R.; ERNST, N.; PERRIN, D. Visualization and Protégé. 2004. In: 7th PROTÉGÉ CONFERENCE, 2004, Bethesda. **Proceedings...** Disponível em: <<http://protege.stanford.edu/conference/2004/abstracts/Storey.pdf>>. Acesso em: 21 julho 2010.

SZKLO, A. S. *Fundamentos do refino do petróleo*. Rio de Janeiro: Interciência, 2005.

TANIGUCHI, T.; LEDERMAN, L. Knowledge management and networking for enhancing nuclear safety. In: INTERNATIONAL CONFERENCE ON NUCLEAR KNOWLEDGE MANAGEMENT, 2004, Saclay. Disponível em: <<http://www.iaea.org/inisnkm/nkm/CD-NKM/WS%20on%20NKM%20Trieste%202005/pdfs/25finalLarsUlfkjaer.pdf>>. Acesso em: 27 julho 2010.

THEOBALD, R.; LIMA, G. B. A. A excelência em gestão de SMS: uma abordagem orientada para os fatores humanos. In: XXV ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 2005, Porto Alegre. **Anais...** Disponível em: <http://www.abepro.org.br/biblioteca/ENEGEP2005_Enegep0405_0510.pdf>. Acesso em: 27 julho 2010.

USCHOLD, M.; GRUNINGER, M. Ontologies: principles, methods and applications. *Knowledge Engineering Review*, v.11, n.2, june 1996. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.5917>>. Acesso em: 08 junho 2010.

USCHOLD, M.; JASPER, R. A framework for understanding and classifying ontology applications. 1999. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.39.6456>>. Acesso em: 09 junho 2010.

VAZ, C. E. M.; MAIA, J. L. P.; SANTOS, W. G. *Tecnologia da indústria do gás natural*. São Paulo: Edgard Blucher, 2008.

VIINIKKALA, M. *Ontology in information systems*. 2004. Disponível em: <<http://www.cs.tut.fi/~kk/webstuff/Ontology.pdf>>. Acesso em: 11 junho 2010.

WAGNER FILHO, F.; LÓSCIO, B. F. Web semântica: conceitos e tecnologias. In: ERCEMAPI – ESCOLA REGIONAL DE COMPUTAÇÃO CEARÁ – MARANHÃO – PIAUÍ, 2009, Parnaíba. **Mini curso...** Disponível em: <<http://www.ufpi.br/ercemapi/arquivos/file/minicurso/mc9.pdf>>. Acesso em: 20 julho 2010.

World Wide Web Consortium – W3C. *OWL Web Ontology Language: overview*. 2004a. Disponível em: <<http://ia.ucpel.tche.br/~lpalazzo/Aulas/TEWS/arq/OWL-Overview.pdf>>. Acesso em: 12 junho 2010.

World Wide Web Consortium – W3C. *OWL Web Ontology Language: guide*. 2004b. Disponível em: <<http://www.w3.org/TR/owl-guide/>>. Acesso em: 14 junho 2010.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)