

UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE
ÁREA: CIÊNCIAS DA COMPUTAÇÃO

PAULO JOSÉ MELO GOMES CORRÊA

**UMA ONTOLOGIA PARA REPRESENTAÇÃO DO CONHECIMENTO DO DOMÍNIO
DA QUÍMICA ANALÍTICA COM ADIÇÃO DE NOVOS AGENTES E
FUNCIONALIDADES PARA ANÁLISE E MONITORAMENTO DE COMBUSTÍVEIS.**

SÃO LUIS
2009

PAULO JOSÉ MELO GOMES CORRÊA

**UMA ONTOLOGIA PARA REPRESENTAÇÃO DO CONHECIMENTO DO DOMÍNIO
DA QUÍMICA ANALÍTICA COM ADIÇÃO DE NOVOS AGENTES E
FUNCIONALIDADES PARA ANÁLISE E MONITORAMENTO DE COMBUSTÍVEIS.**

Dissertação apresentada ao Programa de Pós-graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão, como requisito parcial à obtenção do título de Mestre em Engenharia de Eletricidade na área de Ciências da Computação.

Área de Concentração: Ciências da Computação

Orientador: Prof. Dr. Allan Kardec Barros.
Co-Orientador: Prof. Sofiane Labidi.

SÃO LUIS

2009

Corrêa, Paulo José Melo Gomes

Uma ontologia para representação do conhecimento do domínio da química analítica com adição de novos agentes e funcionalidades para análise e monitoramento de combustíveis / Paulo José Melo Gomes Corrêa. – São Luís, 2009.

128f.

Orientador: Prof. Dr. Allan Kardec Barros

Co-orientador: Prof. Dr. Sofiane Labidi

Dissertação (Mestrado em Engenharia de Eletricidade) – Programa de Pós-Graduação em Engenharia de Eletricidade, Universidade Federal do Maranhão, 2009.

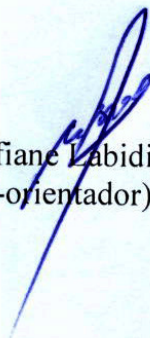
**UMA ONTOLOGIA PARA REPRESENTAÇÃO DO
CONHECIMENTO DO DOMÍNIO DA QUÍMICA ANALÍTICA
COM ADIÇÃO DE NOVOS AGENTES E FUNCIONALIDADES
PARA ANÁLISE E MONITORAMENTO DE COMBUSTÍVEIS**

Paulo José Melo Gomes Corrêa

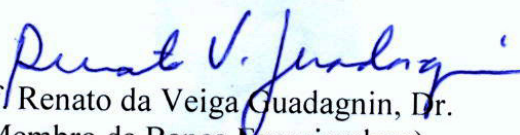
Dissertação aprovada em 14 de setembro de 2009.



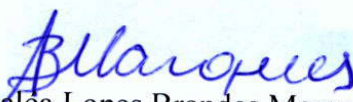
Prof. Allan Kardec Duailibe Barros Filho, Ph.D.
(Orientador)



Prof. Sofiane Labidi, Dr.
(Co-orientador)



Prof. Renato da Veiga Quadagnin, Dr.
(Membro da Banca Examinadora)



Profa. Aldaléa Lopes Brandes Marques, Dra.
(Membro da Banca Examinadora)

Aos meus pais.

“Sob o controle de um forte General,
não haverá jamais soldados fracos”.

(Sócrates)

AGRADECIMENTOS

A DEUS, por ter permitido que todos esses momentos pudessem ser realizados;

Aos meus pais, José Corrêa e Paula de Jesus pelo amor incondicional e primeiros ensinamentos que nortearam a minha caminhada em busca do conhecimento;

A minha irmã Pauliana, pelo amor e compreensão demonstrados mesmo quando tínhamos opiniões diferentes sobre o caminho a ser percorrido;

Aos meus sobrinhos queridos Luiza e André, por me proporcionarem a alegria com a energia de criança que possuem;

A minha avó Laurita Gomes, mais conhecida como Dona Mocinha, minha segunda mãe, pelo amor e carinho proporcionado durante todos esses anos;

A minha namorada Ivanice Coelho, pelo amor, carinho e compreensão pelas minhas ausências, nos momentos de alegria e de tristeza, sempre permanecendo firme e me apoiando em todas as decisões;

Ao Prof. Allan Kardec Barros, pela orientação e oportunidade de fazer parte deste projeto de pesquisa;

Ao Prof. Sofiane Labidi, pelas aulas e conhecimentos passados durante todos esses anos, pela amizade conquistada nos tempos de faculdade e por antes de entrar na faculdade, nos momentos de descontração do futebol de travinha na praia, sem que eu nem mesmo soubesse que ele seria meu professor e orientador, anos depois;

Aos professores do programa de pós-graduação, Alexandre Muniz, Aristófanos Corrêa, Francisco Silva, Mario Meireles, Braga, Vidal, Zair Abdelouahab e Rosário Girard, pelos ensinamentos durante as aulas;

A Prof^a. Aldaléa Marques, pela participação fundamental, sempre com palavras de incentivo e por proporcionar a oportunidade, motivação e determinação necessárias para a conclusão deste trabalho;

A coordenadoria do programa de pós-graduação em Engenharia de Eletricidade, em especial, Prof. Yuri Catunda pelo suporte geral oferecido;

Aos meus amigos, Delano Marques, Osevaldo Farias e Reinaldo Silva, que trabalharam diretamente comigo nesta pesquisa e me proporcionaram o conhecimento necessário para dar continuidade aos seus respectivos trabalhos;

Aos amigos queridos, Tiago Romão, Marina Lavareda, Fábio Carvalhal, Carlos Alberto, Caroline Benmuyal e Jardiane, que sempre me incentivaram a concluir o curso;

Aos amigos que ganhei no Laboratório de Sistemas Distribuídos (LSD), Gilberto Cunha, Eduardo Viana, Diego Souza, Pablo Durans, Estevão, Euziel, Alana, Renata Cantanhede e Vandecia Fernandes;

Aos meus amigos de curso, Geraldo Braz, Aline Lopes, Helaine Cristina, Pedriana Pavão, Rômulo Martins e Falkner Area, pelos momentos que passamos juntos, pelas trocas de idéias, conhecimentos e situações, das mais engraçadas as mais tristes;

Aos colegas do LAPQAP, Glene Henrique e Conceição, que foram de extrema importância para esta pesquisa;

Aos meus amigos de trabalho Carlos Rogério, Roberto Carlos, Diana Tavares e Agzael Viana, que foram compreensivos nos momentos que mesmo presente eu estava completamente ausente ao escrever essa dissertação;

Aos amigos do TRE-MA, Flavio Ricardo, Jocielma, Daniel, Valdeci, Lourencio, Wellison Fabio, Paulo Neves, Egidio Carvalho, Alisson e Ricardo Ferraz, que me incentivaram no decorrer desta pesquisa;

Aos amigos da Secretaria Municipal de Informação e Tecnologia (SEMIT), Sr. Ivanilson, Jucinei, Leonardo Cunha, Leonardo Nakahara, Elton, Alessandra Penha, Gleisse Moreira, Nogueira Jr, Andre Passaroni, Patrick Buna, Yona, Natalia Veiga, Das Neves, Edmilson, Marcos Pop, Claudio Sampaio e Joacy, pelo incentivo e compreensão por minhas faltas para assistir as aulas do curso;

Aos funcionários da coordenadoria do programa de pós-graduação em Engenharia de Eletricidade, em especial ao Alcides, pelo apoio e serviços oferecidos que foram de extrema importância para a realização deste trabalho;

E finalmente a todos que direta ou indiretamente foram importantes para a elaboração deste estudo.

RESUMO

Esta pesquisa apresenta estudos envolvendo as áreas de Engenharia de Eletricidade e Química Analítica de Petróleo e Bicomcombustíveis, tendo como objetivo o aperfeiçoamento das etapas de análise químicas do Programa de Monitoramento da Qualidade de Combustíveis (PMQC), instituído pela Agência Nacional de Petróleo e Biocombustíveis – ANP, executado no Estado do Maranhão pelo Laboratório de Análises e Pesquisa em Química Analítica de Petróleo e Biocombustíveis - LAPQAP. Para isto, são propostas melhorias no Sistema Inteligente de Monitoramento e Controle da Qualidade de Combustíveis – *SIMCQC* utilizado pelo laboratório no auxílio a tomada de decisões. Propõe-se a inclusão de novos agentes à sociedade multiagente, tendo como objetivo aumentar a quantidade de técnicas análises químicas no *SIMCQC*. Para o mecanismo de comunicação é mostrada a criação de uma ontologia de combustíveis que foi modelada utilizando-se da metodologia 101, cujo objetivo é a representação do conhecimento do domínio de análises químicas e o fornecimento de uma linguagem de conteúdo para o mecanismo de comunicação dos agentes da sociedade. Para o alcance dos objetivos foram utilizadas técnicas de Inteligência Artificial, motor de inferência *JESS (Java Expert System Shell)*, a tecnologia de Ontologia para representar o conhecimento do domínio e servir como vocabulário do processo de comunicação, o middleware *JADE (Java Agent DEvelopment Framework)* para execução do ambiente com suas melhorias e a metodologia de desenvolvimento de sistemas multiagente *PASSI* para a modelagem do sistema.

Palavras-chave: *Representação do Conhecimento, Sistemas Multiagentes, Ontologia, Metodologia 101, PASSI.*

ABSTRACT

This research presents studies involving Electricity Engineering and Oil and Biofuel Analytical Chemistry areas, whose objective is the perfectioning of chemical analysis steps for the Fuel Quality Monitoring Program, instituted by the National Agency of Petroleum and Biofuel - ANP, executed in the Maranhão State by the Laboratory of Analyses and Research in Analytical Chemistry of Petroleum and Biofuel - LAPQAP. For this, improvements were proposed for the Fuel Quality Control Multiagent System, and help for the decisions taking of the laboratory. The inclusion of new agents to the multiagent society is still considered, where the objective is to add a new technique automatized for chemical analyses, beyond additional functionalities. A fuel ontology was considered for the communication mechanism that will be shaped using 101 methodology, whose objective is the representation of domain knowledge of chemical analyses beyond supply of a communication language among the society agents. In order to reach the objectives we used Artificial Intelligence techniques, inference motor JESS (Java Expert System Shell), an ontology technology to represent the domain knowledge serving as vocabulary of the communication process, the middleware JADE (Java Agent DEvelopment framework) for environment execution with their improvements and the development methodology of multiagent systems named PASSI for the system modeling.

Keywords: *Knowledge Representation, Multiagent Systems, Ontology, 101 Metodology, PASSI*

LISTA DE SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
ACL	Agent Communication Language
ANP	Agencia Nacional de Petróleo Gas e Biocombustíveis
AROM	Aromaticos
ASTM	American Society for Testing and Materials
BEN	Benzenos
EC	Engenharia do Conhecimento
ER	Engenharia de Requisitos
ES	Engenharia de Software
ETA	Etanol
FIPA	Foundation for Intelligent Phisycal Agents
GUI	Graphical User Interface
IA	Inteligência Artificial
J2SE	Java 2 Standart Edition
JADE	Java Agent Development Environment
JESS	Java Expert System Shell
KDD	Knowledge Discovery Database
KQML	Knowledge Query and Manipulation Language
KR	Knowledge Representation
LAPQAP	Laboratório de Análises e Pesquisa em Quimica Analitica de Petróleo
MAS	Multiagente Systems
MON	Número de Octano Motor
NBR	Norma Tecnica Brasileira
OLEF	Olefinas
PASSI	Process for Agent Society Specification and Implementation
PAL	Protege Axiom Language
PCA	Analise dos Componentes Principais
PLS	Mínimos Quadrados Parciais

PMQC	Programa de Monitoramento da Qualidade de Combustível
RC	Representação do Conhecimento
RMSEP	Média da Raiz Quadrada do Erro de Predição
RON	Número de Octano Pesquisa
SAT	Saturadas
SE	Sistemas Especialistas
SIG	Sistema de Informação Geográfica
SL	Semantic Language
SMA	Sistema Multiagente
UFMA	Universidade Federal do Maranhão
UML	Unified Modeling Language
XML	eXtensible Markup Language
WWW	World Wide Web

LISTA DE FIGURAS

Figura 1.	Representantes legais da ANP no Brasil.....	17
Figura 2.	Metodologia de análise de combustível do LAPQAP.....	18
Figura 3.	Ensaio químicos realizados pelo LAPQAP.	22
Figura 4.	Combustíveis e suas propriedades.....	25
Figura 5.	Arquitetura de agente.	32
Figura 6.	Arquitetura do agente reativo.	33
Figura 7.	Modelo de percepção agente cognitivo.	35
Figura 8.	Containers e plataformas JADE.....	42
Figura 9.	Comportamentos (behaviours) em JADE.	43
Figura 10.	Etapas da metodologia PASSI.....	46
Figura 11.	Classificação de ontologias.	51
Figura 12.	Arquitetura da Web Semântica.....	53
Figura 13.	Passos da metodologia 101	55
Figura 14.	Tipos de hierarquia de classes.	58
Figura 15.	Slots da classe vinho.....	59
Figura 16.	Instancia da classe vinho.....	60
Figura 17.	Classes e hierarquia de classes.	72
Figura 18.	Slots da classe sample (amostras).....	74
Figura 19.	Plugin OntologyBeanGenerator para JADE.....	76
Figura 20.	Modelo físico da ontologia de combustíveis criado pelo plugin <i>OntologyBeanGenerator do Protege</i>	77
Figura 21.	Técnicas automatizadas pelo <i>SIMCQC</i>	78
Figura 22.	Diagrama de atividades do processo de captura e análise das amostras.	82
Figura 23.	Descrição do domínio da análise e monitoramento de combustíveis. ...	84
Figura 24.	Identificação de papéis <i>SIMCQC</i>	85
Figura 25.	Comportamento de monitoramento, captura e análise de amostras da sociedade do <i>SIMCQC</i>	86

Figura 26.	Diagrama da ontologia de domínio de combustíveis.	88
Figura 27.	Responsabilidade do agente Irox.	90
Figura 28.	Responsabilidades do agente Supervisor.	92
Figura 29.	Tela de inserção de dados complementares.	95
Figura 30.	Tela principal do agente Interface.....	96
Figura 31.	Tela de relatório de análises químicas do agente INTERFACE.	96
Figura 32.	Comportamento para criação de relatório do agente Interface.....	97
Figura 33.	Interface gráfica JADE.....	98
Figura 34.	Trecho da comunicação realizada pelo <i>SIMCQC</i>	99
Figura 35.	Agente Irox instanciando a ontologia.....	100
Figura 36.	Propriedades da técnica de composição (Gasolina).....	102
Figura 37.	Comportamento de envio do agente Irox.....	103
Figura 38.	Trecho da comunicação utilizando a <i>Fuel Ontology</i>	104
Figura 39.	Definição da ontologia no comportamento cognitivo.	107
Figura 40.	Exemplo de <i>defrule</i> baseado no <i>deftemplate</i> da classe <i>Property</i>	108
Figura 41.	Nova arquitetura do <i>SIMCQC</i>	110
Figura 42.	Diagrama de implantação/distribuição do <i>SIMCQC</i>	112

LISTA DE TABELAS

Tabela 1.	Ensaio regulares realizados e métodos utilizados.....	19
Tabela 2.	Comunicação da FIPA-ACL.....	40
Tabela 3.	Linguagens para descrição do conhecimento	54
Tabela 4.	Repositórios on-line de ontologias.....	68
Tabela 5.	Termos do domínio de combustíveis	69
Tabela 6.	Termos do domínio e suas propriedades	73
Tabela 7.	Agentes e responsabilidades do <i>SIMCQC</i>	80
Tabela 8.	Regras da base de conhecimento do agente cognitivo (Gestor).....	109
Tabela 9.	Comparativo de funcionalidades.....	114

LISTA DE QUADROS

Quadro 1.	Documento de descrição da ontologia.....	67
Quadro 2.	Funcionalidade do agente Irox.....	91
Quadro 3.	Funcionalidade do agente Supervisor	93
Quadro 4.	Funcionalidades do agente Interface.....	94

SUMÁRIO

LISTA DE FIGURAS	14
LISTA DE TABELAS	16
LISTA DE QUADROS	17
1. INTRODUÇÃO	16
1.1. Contexto	16
1.2. Definição do problema	20
1.3. Justificativa.....	23
1.4. Objetivos	27
1.5. Metodologia.....	29
1.6. Organização.....	29
2. FUNDAMENTAÇÃO TEORICA.....	31
2.1. Engenharia de software multiagente.....	31
2.1.1. Agentes	31
2.1.1.1. Agente Reativo ou Reflexivo	33
2.1.1.2. Agente Deliberativo ou Cognitivo	34
2.1.2. Arquitetura de Sistemas Multiagente.....	37
2.1.3. Comunicação na sociedade de agentes	39
2.1.4. O framework JADE.....	41
2.1.5. Metodologias para desenvolvimento de Sistemas Multiagente.....	43
2.2. Representação do conhecimento.....	47
2.3. Ontologia.....	49
2.3.1. Linguagens para desenvolvimento de ontologias	52
2.3.2. Metodologias para desenvolvimento de ontologias.....	55
2.4. Trabalhos relacionados	60
3. ONTOLOGIA DE COMBUSTIVEIS PARA O SISTEMA DE MONITORAMENTO DA QUALIDADE DE COMBUSTIVEL.....	64

3.1.	Definição da ontologia de combustíveis	64
3.2.	Domínio e escopo da ontologia	66
3.3.	Considerar a reutilização de ontologias existentes	68
3.4.	Enumeração dos termos da ontologia.....	69
3.5.	Definição das <i>classes</i> e suas hierarquias	71
3.6.	Definição das propriedades (slots) das classes	73
3.7.	Definição dos tipos de propriedades (<i>slots</i>)	74
3.8.	Criação do modelo físico das classes	75
4.	ESTUDO DE CASO: SISTEMA INTELIGENTE PARA MONITORAMENTO E CONTROLE DA QUALIDADE DE COMBUSTIVEL – <i>SIMCQC</i>	78
4.1.	O projeto <i>SIMCQC</i>	78
4.2.	Funcionamento do <i>SIMCQC</i>	79
4.3.	Estendendo o <i>SIMCQC</i>	81
4.3.1.	Descrição do Domínio	83
4.3.2.	Identificação dos papéis.....	85
4.3.3.	Modelo da sociedade de agentes	86
4.3.4.	Agente Irox.....	89
4.3.5.	Agente Supervisor.....	91
4.3.6.	Agente Interface.....	93
4.3.7.	Linguagem de comunicação do <i>SIMCQC</i>	98
4.3.8.	Melhorias no mecanismo de comunicação do <i>SIMCQC</i>	100
4.3.9.	O comportamento cognitivo do agente Gestor.....	105
4.3.10.	A nova arquitetura do sistema.....	110
4.4.	Resultados	113
5.	CONCLUSÃO E TRABALHOS FUTUROS	116
	REFERÊNCIAS.....	120
	ANEXO I - FORMULARIOS	126

1. INTRODUÇÃO

Neste capítulo é abordado o contexto em que se encaixa o presente trabalho, bem como a definição do problema, sua justificativa, os objetivos do trabalho, a metodologia utilizada para alcançar os objetivos da pesquisa e, por fim, é descrita a organização desta dissertação.

1.1. Contexto

As grandes revoluções, entre as quais a industrial, acontecida durante os séculos XVIII-XIX (OLIVEIRA, 2004), ou a do petróleo no Brasil, ocorrida a partir de 1997, com a lei nº 9478, mais conhecida como lei do petróleo, quebrando um monopólio da Petrobras (Petróleo Brasileiro S/A) nas atividades de exploração e petróleo, constitui-se em aspectos determinantes para mudanças de importantes paradigmas que têm resultado em fatores de desenvolvimento científico e tecnológico para o Brasil.

Na área de petróleo, o setor foi expandido para participação de empresas estrangeiras como reforço de arrecadação do setor público (BANCO NACIONAL DO DESENVOLVIMENTO ECONOMICO SOCIAL, 2009) e transformado, por completo, o panorama nacional da indústria, atuando diretamente na sociedade.

No Brasil, a revolução causada pela Lei do Petróleo, trouxe diversas mudanças, soluções e dúvidas sobre sua distribuição, controle e qualidade dos derivados de Petróleo.

Nesse sentido, foi instalada no país a Agência Nacional do Petróleo – ANP, ficando responsável por monitorar e regulamentar a distribuição e qualidade dos derivados do petróleo no território brasileiro. (AGÊNCIA NACIONAL DO PETRÓLEO GAS NATURAL E BIOCOMBUSTÍVEIS, 1997).

Entre os mecanismos de desenvolvimento estabelecidos pelo Governo Federal, na área de petróleo, foi implantado o CTPETRO (Ciência e Tecnologia na

área de Petróleo e Gás Natural), o primeiro de uma série de fundos setoriais, responsável por um grande aporte de recursos destinados a Instituições de Ensino e Pesquisa, para o desenvolvimento científico e tecnológico do país.

A UFMA, através da área de química, tem participado desta importante demanda científica nacional, desde o primeiro Edital CTPETTRO-FINEP, cujo primeiro projeto aprovado em 1999, teve como finalidade a implantação do Laboratório de Análises e Pesquisa em Química Analítica de Petróleo e Biocombustíveis – LAPQAP (LAPQAP, 2009).

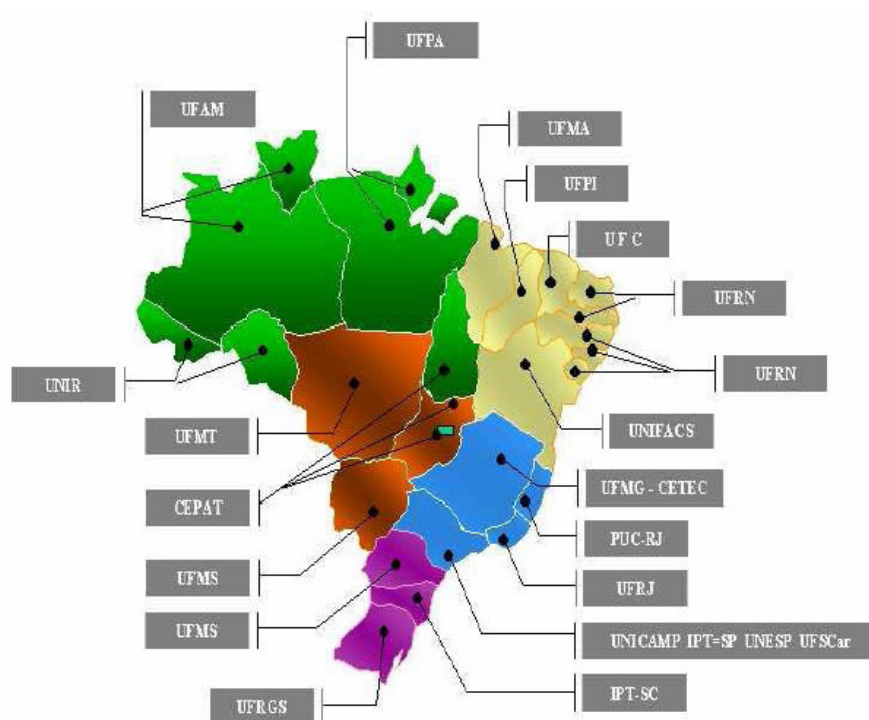


Figura 1. Representantes legais da ANP no Brasil.

Fonte: ANP, 2009.

Após a criação do laboratório LAPQAP, a UFMA passou a ser uma das 23 instituições contratadas pela ANP para o monitoramento da qualidade de combustíveis, parceria esta que já acontece há quase 10 anos.

O LAPQAP instituiu três fases para alcançar os objetivos definidos pela ANP, que são: Planejamento, Coleta e Análise. As fases de coleta e análises são definidas pelo Programa PMQC, do qual o LAPQAP faz parte, de acordo com as normas definidas em contrato firmado com a ANP.

A metodologia de análise dos combustíveis nos postos revendedores constitui-se basicamente de três etapas (MARQUES, 2003):

1. Planejamento: é responsável pela seleção aleatória dos postos revendedores a serem visitados e das amostras de combustíveis a serem coletadas;
2. Coleta: trata diretamente da coleta das amostras de combustíveis;
3. Análise: é responsável pela organização, cadastro, colocação das etiquetas, armazenamento criterioso das amostras e realização dos ensaios químicos propriamente ditos.

A seleção aleatória dos postos revendedores e amostras de combustíveis são realizadas uma vez por mês pelo Coordenador ou Supervisor do Laboratório de Combustível. Esse procedimento é realizado a partir da relação de postos revendedores fornecida pela Superintendência de Qualidade de Produtos da ANP e atualizado mensalmente pelo próprio laboratório. O roteiro a ser seguido é discutido e otimizado com o *amostrador*¹. Ressalta-se que os procedimentos adotados para a seleção dos postos e de coleta de amostras são informados mensalmente à ANP (MARQUES, 2003).

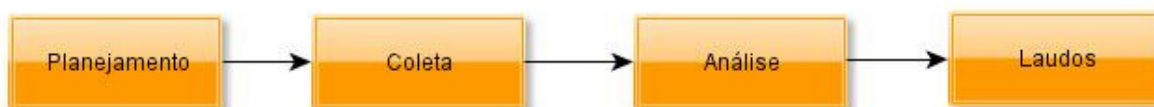


Figura 2. Metodologia de análise de combustível do LAPQAP.

Fonte: LAPQAP, 2009.

Após a fase de coleta, o *amostrador* realiza a entrega das amostras extraídas dos postos de combustíveis no laboratório, então, o técnico responsável realiza a etiquetagem manual das amostras, definindo um padrão de classificação

¹ Técnico responsável por efetuar a coleta das amostras.

de acordo com o tipo de combustível. Durante este procedimento manual de conferência e etiquetagem, nenhum mecanismo informatizado é utilizado como forma de auxílio, ressaltando os formulários que são preenchidos para cada amostra, contendo as informações do tipo de combustível a ser encaminhado para a fase de análise. (LAPQAP, 2009)

A principal fase, e a de objeto de estudo desse trabalho, trata-se da fase de análise, onde são realizadas as análises químicas das amostras coletadas pela fase de coleta. É nela, que o técnico atua diretamente com os equipamentos de análise, realizando os ensaios químicos para determinar a qualidade da amostra.

Durante esta fase, o técnico determina, de acordo com o tipo de combustível da amostra, quais os ensaios serão realizados em cada amostra, pois dependendo do tipo, serão utilizados equipamentos e técnicas específicas para aquela amostra. A seguir, podemos perceber na tabela de ensaios químicos, os tipos de ensaios realizados pelo LAPQAP.

Tabela 1. Ensaios regulares realizados e métodos utilizados.

Ensaio	Método	Gasolina	Diesel	Álcool	Biodiesel
Aspecto e cor	VISUAL	X	X	X	X
Cor ASTM	ASTM D1500		X		X
Composição	APG	X			
Teor de álcool	NBR13992	X			
Teor alcoólico	ASTM D4052			X	
Massa específica	ASTM D4052	X	X	X	X
Destilação	ASTM D86	X	X		
Condutividade	NBR10547			X	
Teor de hidrocarboneto	NBR13993			X	
pH	NBR10891			X	
Enxofre	ASTM D4294		X		X
Índice de Cetano	ASTM D4737		X		
Ponto de Fulgor	ASTM D93		X		X
Viscosidade cinemática	ASTM D445		X		X
Sódio e potássio	EM 14108				X
Corrosividade ao cobre	ASTM D130		X		X
Ponto de entupimento	ASTM D 6371		X		X
Índice de acidez	ASTM D664				X
Glicerina livre	ASTM D				X

	6584				
Glicerina TOTAL	ASTM D 6584				X
Monoglicérideos	ASTM D 6584				X
Diglicérideos	ASTM D 6584				X
Triglicérideos	ASTM D 6584				X
Metanol ou Etanol	EN14110				X
Estabilidade oxidação	a EM 14112				X

Fonte: LAPQAP, 2009.

Contudo, o laboratório instituiu uma 4ª etapa, a fase de laudos. Durante esta etapa o técnico de laboratório fica encarregado de efetuar a análise sobre os dados obtidos durante a fase de análise. Durante este procedimento, o técnico, primeiramente, verifica se algum ensaio realizado foi mal sucedido. Em caso de positivo, a amostra é novamente analisada, para obter uma contraprova. Até o limite de 3 tentativas, e assim, definindo os valores analisados para aquela amostra.

De posse dos resultados da fase de análise, e permanecendo os dados, o técnico então fará a análise sobre os dados e em seguida redigirá um relatório, baseado em valores de referencia informados pelo laboratório central da ANP, informando as possíveis adulterações que a amostra possui. Este documento ou laudo técnico, contem todas as informações sobre as amostras de combustíveis e seus diagnósticos de análise.

Os dados obtidos são digitados na ferramenta de Monitoramento da Qualidade de Combustível (MQC) para em seguida serem encaminhadas digitalmente para o laboratório central da ANP em Brasília, finalizando assim o processo de monitoramento e controle da qualidade de combustível.

1.2. Definição do problema

Nos últimos anos houve um crescimento da quantidade de petróleo produzido, como consequência existe uma queda natural no preço do produto. Com isso, há um aumento da compra e venda, ocasionando maior distribuição dos

derivados de petróleo pelo país. Esse crescimento de produção gera vários eventos, o primeiro, o de distribuição.

O crescente aumento de distribuidores de derivados do petróleo gera o segundo evento, para o monitoramento e tratamento da qualidade de combustível, considerando-se que dessa distribuição há um aumento da quantidade de revendas de derivados do petróleo como os postos de combustíveis, gerando assim um volume maior de combustíveis a ser coletado, monitorado e por sua vez analisado.

O volume de combustível analisado, por sua vez gera novamente outro evento, dessa vez, a quantidade de dados oriundos das análises dessas amostras, uma vez que para cada posto de combustível vistoriado, no mínimo uma amostra para cada tipo de combustível comercializado deve ser coletada para posterior análise.

Esse crescente, e conseqüente aumento, desde o barril de petróleo até os postos de combustíveis, geram dados que deverão ser monitorados para garantir a qualidade do combustível.

Dentro deste processo de crescimento, após a implantação do PMQC no ano de 1998, houve uma expansão deste programa no ano de 2005, tendo o mesmo alcançado todo o território brasileiro (ANP, 2009).

Nesse sentido, a engenharia de software e inteligência artificial, vem com a finalidade de utilizar técnicas de raciocínio lógico para realizar o tratamento dessas informações. Marques (2003) utilizou-se de técnicas de descoberta do conhecimento conhecidas como *KDD (Knowledge Discovery Database)* e *Data Warehouse* aplicadas nos dados recuperados após anos de análises realizadas pelo LAPQAP. Por outro lado, Silva (2008) propôs um sistema multiagente para auxiliar no tratamento das informações durante a fase de análise, utilizando-se para isso dados capturados por uma das técnicas usadas no monitoramento da qualidade de combustíveis, denominada de Destilação, cujos ensaios são realizados rotineiramente pelo LAPQAP.

Marques (2003) e Silva (2008) definiram procedimentos que foram de bastante utilidade no auxílio ao programa de monitoramento realizado pelo LAPQAP e trouxeram ao laboratório controle na fase de análise das amostras. Com seus

respectivos trabalhos o laboratório obteve mais autonomia sobre os dados analisados e durante o processo de análise de combustíveis.

Apesar do apoio gerado pelas propostas oriundas dos trabalhos de Marques (2003) e Silva (2008), estes solucionam apenas uma pequena parte do problema em face da grandeza do domínio do programa de monitoramento.

A arquitetura proposta por Silva (2008) foi definida como um Sistema Inteligente de Monitoramento e Controle da Qualidade de Combustível, denominado de *SIMCQC*. Este sistema é composto por uma sociedade de 4 (quatro) agentes, sendo 3 (três) reativos e 1 (um) cognitivo. Esse último tem como objetivo principal efetuar a inferência dos dados capturados previamente pelo dispositivo HERZOG utilizado na técnica Destilação. Contemplando apenas uma técnica de análise de combustíveis.

Aliados a estes detalhes, a arquitetura definida por Silva (2008) para efetuar a comunicação entre os agentes do *SIMCQC* utiliza-se de troca de mensagens de através de texto simples. Tornando a comunicação limitada e complexa caso houvesse necessidade de extensão. O motor de inferência, devido à limitação da comunicação dos agentes, não esta preparado para analisar maiores quantidades de dados, ficando limitado também a retornar quantidades previamente definidas de respostas.

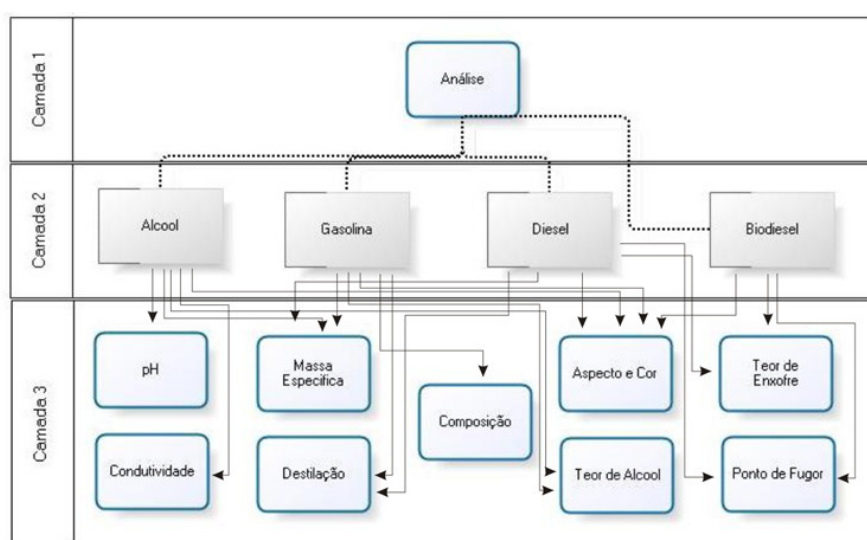


Figura 3. Ensaios químicos realizados pelo LAPQAP.

A Figura 3 mostra uma visão geral dos ensaios químicos realizados pelo LAPQAP. Podemos ver na camada 3 todas as técnicas de análise usadas no monitoramento dos combustíveis, realizadas pelo LAPQAP. Na camada 2 são apresentados os quatro diferentes combustíveis analisados. Silva (2008) propôs a automatização da técnica de Destilação, a qual é aplicada ao combustível Gasolina.

Além deste ensaio químico, o LAPQAP ainda realiza outras 8 (oito) etapas no processo de análise e controle rotineiros da qualidade dos combustíveis, as quais não são automatizadas. O processo realizado dessas etapas atualmente ocorre de forma manual, desprendendo um considerável tempo no preenchimento de formulários de análise, cujo tempo seria melhor aproveitado caso fosse realizado por um processo automático ou semi-automático. A percepção destes fatos motivou o presente estudo, o qual é considerado de alguma forma como a continuação dos estudos, primeiramente, realizados por Marques (2003) e Silva (2008).

Através de um diagnóstico dessa situação, foi constatada a importância de fornecer ao sistema proposto por Silva (2008) recursos que automatizassem uma quantidade maior de ensaios e técnicas utilizadas pelo LAPQAP, sendo este o principal foco e objetivo do presente estudo. As melhorias propostas têm a finalidade de cadastrar, recuperar os dados de análises e retornar resultados através de laudos ou relatórios técnicos, proporcionando um ganho de produtividade e permitindo ao técnico de laboratório ficar livre para realizar outras atividades de análise ao invés de preencher formulários que ainda assim, seriam novamente digitados para envio a ANP.

1.3. Justificativa

A química analítica é a parte da química que trabalha com a determinação da composição da matéria e a identificação dos componentes da mesma. Para a identificação, utilizam-se processos químicos nas etapas de análises, determinando assim além da identificação da propriedade o seu respectivo valor analítico. (ROSSI, 2009).

No processo de controle da qualidade de combustíveis, a química analítica tem como objetivo efetuar a identificação e determinar o quantitativo das propriedades de combustíveis. A partir da identificação da matéria, a propriedade e seu respectivo valor são verificados de acordo com os valores de referência, especificados pelo escritório central da ANP, atual responsável por assegurar o padrão de qualidade dos derivados de petróleo.

A equipe do LAPQAP, como responsável pelas análises químicas dos combustíveis no Estado do Maranhão, utiliza diferentes técnicas analíticas para realizar a identificação e a quantificação necessárias ao controle de qualidade das amostras. Para isso, são realizadas diligências aos postos e revendedores de derivados do petróleo para coleta de amostras representativas do ponto de vista estatístico, a serem posteriormente analisadas. (LAPQAP, 2009)

Atualmente o *SIMCQC* utilizado pelo LAPQAP não contempla todos os equipamentos e técnicas analíticas do processo de monitoramento da qualidade dos combustíveis. Com isso, não fornece o devido suporte para uma tomada de decisão rápida e segura.

Conforme Silva (2008) a operacionalização do processo de análise química apresentava inconsistências como:

- Falta de integração das bases de dados;
- Etapas de análises não automatizadas;
- Falta de padronização na comunicação do sistema com o MQC;

Com base nas inconsistências anteriores, Silva (2008) propôs o *SIMCQC* com o objetivo de sanar essas dificuldades. Porém, além desses problemas, tem havido ainda um aumento do número de análises químicas decorrente do crescente aumento no número de postos e na produção dos derivados do petróleo. A estes fatos está associado também, intrinsecamente um aumento do número de coletas.

Após a etapa de análises químicas, uma quantidade considerável de dados é gerada. Como os dados de números de postos revendedores, de coletas das amostras e do número de análises realizadas são crescentes, uma ação deve

ser tomada baseada no perfil peculiar a cada estado, baseado em sistemas que auxiliem nessa decisão.

Para cada combustível são analisadas as propriedades de acordo com a Figura 4.

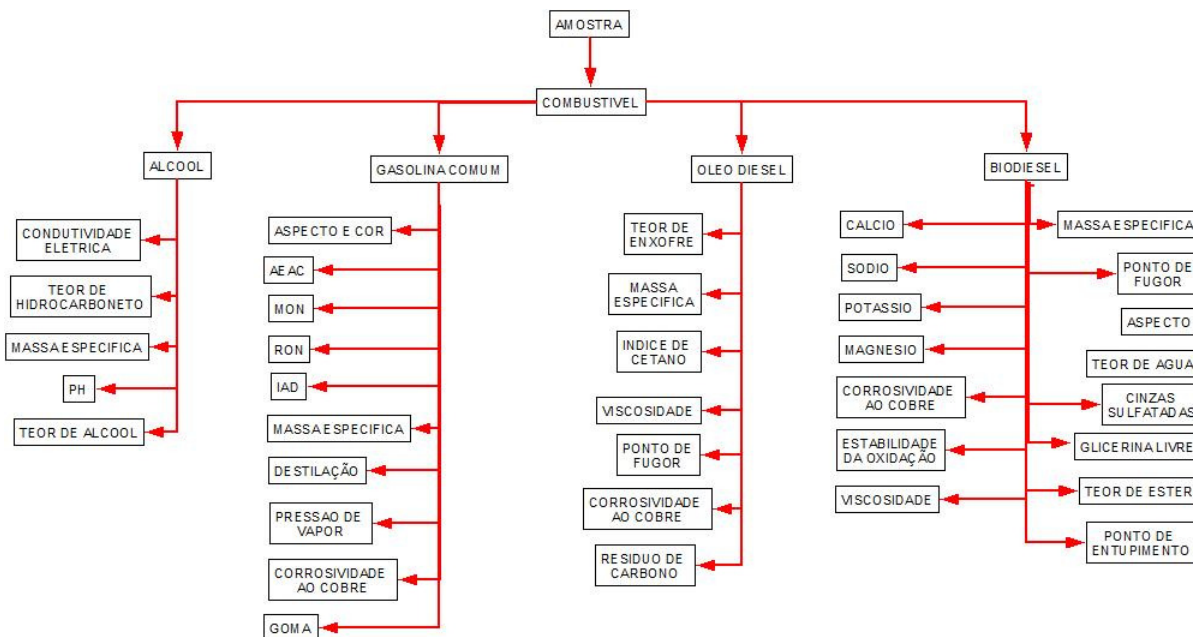


Figura 4. Combustíveis e suas propriedades.

Para realizar as análises químicas das amostras de combustíveis, técnicas e ensaios químicos são utilizados, somando um total de 9 (nove) técnicas ou ensaios. Durante esta etapa, equipamentos automáticos ou manuais são utilizados para verificar as propriedades de cada combustível. Muitas vezes, estes equipamentos são complexos e isolados, sem qualquer tipo de software de comunicação.

Nesse sentido, existe uma dificuldade para o *SIMCQC* recuperar os dados dos equipamentos e interpretar a totalidade das propriedades dos combustíveis, uma vez que além das propriedades existem as ações que o sistema irá oferecer.

Com o objetivo de aumentar o leque de opções do LAPQAP no processo de análises de combustíveis e também permitir a integração de maior número de técnicas automatizadas e/ou informatizadas o presente trabalho tem como objetivo

acrescentar melhorias ao sistema multiagente proposto por Silva (2008). O novo sistema proposto dá suporte no apoio de tomada de decisão do LAPQAP no processo do PMQC.

Primeiramente, para alcançar as metas propostas e dar suporte à etapa de análise, propôs-se a inclusão de uma técnica de análise não contemplada anteriormente. Trata-se do equipamento da técnica de infravermelho realizada pelo equipamento *Irox 2000 Gasoline*, da empresa *Grabner Solutions*, que realiza análise da composição de vários parâmetros de derivados de petróleo, do tipo gasolina e diesel.

Com a adição das novas funcionalidades propostas neste sistema, pretende-se proporcionar ao LAPQAP ainda mais agilidade no tratamento das informações. As técnicas a serem automatizadas e informatizadas pelas melhorias propostas, a serem incluídas, podem ser vistas na Figura 3, nas caixas vermelhas. São elas as seguintes: Destilação, Composição (Infravermelho), Massa Específica e Aspecto e cor.

A inclusão da nova técnica, proposta no presente trabalho, implica em necessárias modificações que deverão ser realizadas no *SIMCQC*, pois, a inclusão de um novo equipamento de análise na atual arquitetura do sistema exige um grande esforço de programação, tornando a tarefa complexa.

Para superar esse problema, uma segunda melhoria é proposta, a qual visa melhorar a escalabilidade² do *SIMCQC*, além de fornecer agilidade ao processo de captura de dados dos equipamentos durante as análises. Para isso, propõe-se a inclusão de uma *ontologia de combustíveis*.

Com a inclusão da *ontologia*, pretende-se, representar o conhecimento do domínio da aplicação, fornecido a partir da criação dos termos e conceitos da química analítica em classes.

Rosa (2008) afirma que a utilização de ontologias tem como vantagem a definição de um vocabulário único para a representação do conhecimento do

² Na engenharia de software, escalabilidade é uma característica desejável em todo o sistema, em uma rede ou em um processo, que indica sua habilidade de manipular uma porção crescente de trabalho de forma uniforme, ou estar preparado para crescer. (WIKIPEDIA, 2009)

domínio da aplicação. O objetivo principal deste vocabulário é fornecer uma conceitualização para que sejam prevenidas interpretações diferentes do mesmo.

Nesse sentido, através do uso da *ontologia*, será possível fornecer ao sistema uma linguagem de conteúdo para o mecanismo de comunicação mais concisa entre os agentes da sociedade, objetivando o envio do resultado das análises obtidas pelos equipamentos através do uso desse vocabulário. Com a inclusão da ontologia estão sendo adicionadas melhorias ao mecanismo de comunicação da sociedade multiagente.

Pretende-se ainda incluir no *SIMCQC* um novo agente, chamado *Interface*, tendo como objetivos: controlar os agentes, servir como meio de acesso às funcionalidades do sistema e fornecer um mecanismo de entrada de dados para inclusão de amostras, utilizando-se de técnicas ainda não automatizadas.

Ao final, com as inclusões propostas, o *SIMCQC* modificado poderá contemplar um total de quatro técnicas durante as etapas do processo de análise dos combustíveis, sendo que pelo menos duas são completamente automatizadas, a de destilação e a de infravermelho (determinação da composição). Outras duas são apenas informatizadas, porém, não automáticas, a de Massa Específica e a de Aspecto e cor. Contudo, as técnicas não automatizadas incluídas, também são analisadas pelo comportamento cognitivo do agente gestor, que deverá sofrer modificações para compreender os novos dados enviados em pacotes utilizando-se da ontologia de combustíveis, participando assim do processo deliberativo do mesmo.

Nesse sentido, após as inclusões das melhorias no *SIMCQC* objetiva-se fornecer ao LAPQAP melhor suporte no apoio de tomada de decisão no processo do Programa de Monitoramento da Qualidade de Combustíveis.

1.4. Objetivos

Este trabalho tem como objetivo acrescentar melhorias a um sistema multiagente para suporte e apoio na tomada de decisões do Programa de

Monitoramento e Controle da Qualidade de Combustível (MQC) realizado pelo Laboratório LAPQAP.

Pretende-se alcançar os seguintes objetivos específicos:

- i. Estudar os mecanismos oferecidos pelo Sistema de Monitoramento e Controle da Qualidade de Combustível (*SIMCQC*) anteriormente proposto por Silva (2008);
- ii. Pesquisar sobre formas de representação do conhecimento e utilização na área de engenharia de software e inteligência artificial;
- iii. Analisar o processo de monitoramento da qualidade de combustíveis realizado pelo LAPQAP, dando ênfase ao processo de ensaios químicos realizado para os combustíveis do tipo Álcool, Gasolina e Diesel;
- iv. Criar uma *ontologia de combustíveis* para representar o conhecimento do domínio da aplicação e, como consequência, servir de linguagem de conteúdo para o mecanismo de comunicação da sociedade de agentes;
- v. Acrescentar novas funcionalidades ao *SIMCQC*, visando contemplar mais uma técnica de análises químicas, através da inclusão de um novo equipamento de análise de combustível, ao sistema proposto;
- vi. Elaborar a especificação de Análise e Projeto das novas funcionalidades do *SIMCQC* utilizando para isso as técnicas de Engenharia de Software, de Conhecimento, de Agentes e de ontologias, a metodologia *PASSI* (COSSETINO, 2002) e a ferramenta *Protegé* (PROTEGE, 2009) respectivamente;
- vii. Modificar o comportamento cognitivo do agente gestor (SILVA, 2008) para fornecer suporte a novas funcionalidades a serem incluídas com os agentes e o novo mecanismo de comunicação com a utilização da *ontologia de combustíveis*.

1.5. Metodologia

A seguir, descreve-se a metodologia utilizada durante a presente pesquisa:

- Realização de pesquisas na literatura e consequente apresentação do estado da arte, referente às soluções utilizadas para auxílio do monitoramento e controle de qualidade de combustíveis, associando as áreas da Inteligência Artificial e Química Analítica;
- Realização de entrevistas com o técnico de laboratório, responsável pela realização dos procedimentos de análises químicas do LAPQAP;
- Análise técnica da comunicação e do comportamento cognitivo realizados atualmente pelo sistema utilizado no laboratório, o *SIMCQC* (SILVA, 2008);
- Análise técnica de documentos oficiais de análises químicas (ANEXO I) utilizados pelo LAPQAP, e definidos previamente pela ANP;
- Estudo dos laudos de análises de amostras previamente tratados, tendo como objetivo a determinação da inferência dos dados no papel do técnico de laboratório;
- Utilização de metodologias para o desenvolvimento das soluções propostas, como a metodologia 101 para desenvolvimento de ontologias e a metodologia de desenvolvimento de sistemas multiagente PASSI (COSSETINO, 2002).

1.6. Organização

Esta dissertação está organizada em cinco capítulos.

Este capítulo, o primeiro, teve como objetivo demonstrar o contexto da proposta destacando a ANP, o programa de monitoramento da qualidade dos

combustíveis, e o LAPQAP, em seguida, relatou-se o problema a ser solucionado, a justificativa do presente trabalho, bem como os objetivos a serem alcançados, a metodologia utilizada na pesquisa, e por fim, a organização do referido trabalho de pesquisa.

No segundo capítulo, apresenta-se o referencial teórico necessário para embasamento técnico acerca das tecnologias mencionadas neste trabalho. Neste sentido, descreve-se a Engenharia de Software Multiagentes e a tecnologia de ontologias, destacando seus conceitos, termos, metodologias e ferramentas envolvidas.

No terceiro capítulo, a criação da *ontologia* do domínio de combustíveis para representar o conhecimento do domínio da aplicação. Para isso, é utilizada a ferramenta Protegé (PROTEGE, 2009) durante a modelagem da mesma.

No quarto capítulo, todos os esforços são concentrados para a modelagem e implementação das melhorias a serem adicionadas ao *SIMCQC*, utilizando-se para isso da metodologia *PASSI* (COSSETINO, 2002) para modelar os novos agentes da solução proposta, incluindo-se a ontologia de combustíveis.

E finalmente, no quinto e último capítulo, são relatadas as considerações finais, bem como as dificuldades e sugestões para trabalhos futuros.

2. FUNDAMENTAÇÃO TEORICA

2.1. Engenharia de software multiagente

A Engenharia de Software (ES) tem como objetivo estudar melhorias nos procedimentos de desenvolvimento de sistemas como metodologias, ferramentas e ambiente de desenvolvimento de software (PRESSMAN, 2009).

Já Engenharia de Software Multiagentes (Orientada a Agentes), situada como uma das vertentes da Engenharia de Software, tem se destacado ao representar o mundo real (SILVA e LABIDI, 2007), oferecendo suporte para representar estruturas organizacionais utilizando-se para isso de técnicas que permitem alcançar maior comunicabilidade no que se refere às necessidades dos usuários, dos produtos gerados, das etapas de projeto e codificação.

2.1.1. Agentes

Diversas são as definições existentes na literatura sobre agentes computacionais em inteligência artificial, entre elas podemos citar:

Para Guizzardi (2007) um agente é tudo aquilo que pode ser visto percebendo um ambiente e interagindo sobre ele em busca de um conjunto de objetivos.

Já Russel (2004), define um agente como sendo uma entidade autônoma capaz de perceber o seu ambiente através de sensores, agindo sobre o mesmo ambiente através dos seus executores.

Girardi (2004), afirma que um agente é uma entidade autônoma que percebe seu ambiente através de sensores e age sobre o mesmo utilizando-se dos executores.

Wooldrige (2002) afirma que um agente é na verdade um programa de computador que está situado em um ambiente onde é capaz de interagir automaticamente para alcançar seus objetivos.

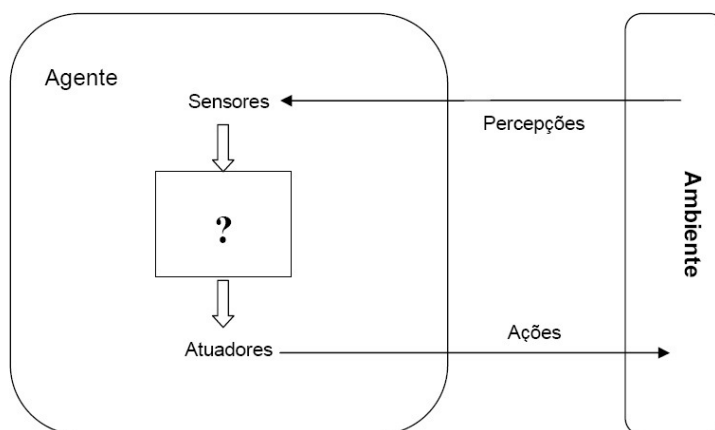


Figura 5. Arquitetura de agente.

Fonte: GUIZZARDI, 2007.

Nesse sentido, podemos afirmar que em todas as definições acima existe um senso comum, onde todos os autores afirmam que os agentes podem ou não ser autônomos, percebem o ambiente e agem sobre ele, porém, sempre em busca de um objetivo.

Um agente computacional é composto em sua estrutura básica por uma memória interna para reagir de acordo com o ambiente, onde esta se atualiza sempre que há modificações no ambiente, tal memória é utilizada para o processo de tomada de decisões, sendo responsável por acionar as ações que serão tomadas pelo agente.

Para um programa de computador, um software ou aplicação se encaixar na definição de um agente, ele deve possuir as seguintes características:

- Deve possuir AUTONOMIA e capacidade para agir sem a necessidade de intervenção de outros agentes.
- Possuir a capacidade de ter REAÇÕES de acordo com os estímulos do ambiente;

- Deve ter iniciativa própria para alcançar os objetivos, possuindo assim sua PROATIVIDADE;
- Deve ser capaz de se comunicar com outros agentes do ambiente, definindo assim sua SOCIABILIDADE.

Entre as classificações sobre os tipos de agentes computacionais existentes na literatura, podemos destacar: os que possuem a capacidade de reagir, chamados de agentes Reativos ou Reflexivos, os que possuem a capacidade de reagir e deliberar sobre uma percepção para assim tomar uma decisão, esses são chamados de agentes Deliberativos ou Cognitivos, e os que possuem capacidade de se locomover, os agentes moveis. A seguir destacamos os agentes que fazem parte do presente trabalho.

2.1.1.1. Agente Reativo ou Reflexivo

O agente reativo é aquele que possui apenas características de reagir ha uma mudança no ambiente. Para um programa de computador, um software ou aplicação se encaixar na definição de um agente reativo, ele deve possuir a seguinte arquitetura.

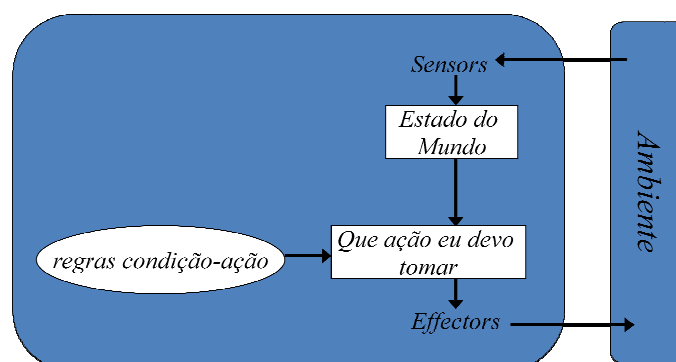


Figura 6. Arquitetura do agente reativo.

Fonte: GUIZZARDI, 2007.

Sua característica deve possuir sensores, regras de condição para efetuar ações e monitorar o ambiente desejado. Com isso, após o agente perceber as modificações no ambiente ele deve agir de acordo com suas regras de condição. Possuem autonomia apenas para executar tarefas. Esses agentes não possuem conhecimento de tarefas realizadas anteriormente a aquela percepção.

2.1.1.2. Agente Deliberativo ou Cognitivo

O agente cognitivo ou deliberativo possui a capacidade de perceber as mudanças no ambiente, analisar e decidir qual ação realizar. Para isso, o agente possui mecanismos de inferência utilizando-se de regras previamente inseridas no comportamento do agente para escolher a ação apropriada. O agente deliberativo possui ainda a capacidade de aprender com as experiências passadas, e assim tomar uma decisão baseada nessas experiências.

Para um agente se encaixar na definição de deliberativo ou cognitivo, ele deve possuir as seguintes características:

- Adaptatividade – deve possuir capacidade de adaptação, quando houver mudanças no ambiente determinando assim o aprendizado;
- Mobilidade – deve ser capaz de se mover entre os ambientes;
- Veracidade – deve ser capaz de fornecer conhecimento verdadeiro sobre suas ações;
- Benevolência – deve ser capaz de obedecer todas as requisições de serviços que são solicitadas.
- Racionalidade – deve ser capaz de agir sempre em busca dos próprios objetivos;

Vale ressaltar que para um agente cognitivo alcançar a característica de racionalidade o mesmo deve possuir uma base de conhecimento e assim efetuar a inferência. Somente assim o agente poderá decidir sobre qual decisão tomar.

O agente cognitivo deve possuir uma base de raciocínio para efetuar a inferência dos dados. Para isto ocorrer, existem algumas ferramentas que possibilitam o agente raciocinar, que são conhecidas como: motor de inferência.

O comportamento cognitivo de um agente cognitivo é baseado em regras, crenças e atitudes que são interpretadas pelo motor de inferência e acionadas a partir da percepção das mudanças ocorridas no ambiente.

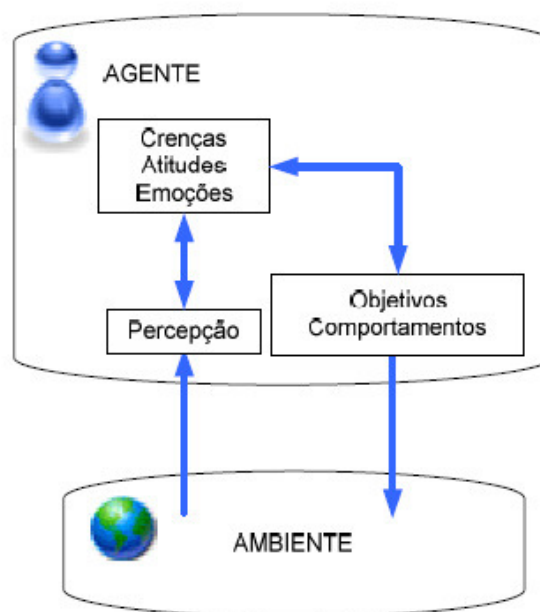


Figura 7. Modelo de percepção agente cognitivo.

Fonte: LINO, 2006.

O motor de inferência é responsável por controlar a atividade cognitiva do agente. Wikipedia (2009) define o motor de inferência como uma forma de estado finito que consiste de três ciclos: verificar, selecionar e executar as regras.

O primeiro ciclo consiste na verificação da regra que satisfaz a resolução do problema, sendo analisada entre todas as regras da sua base de conhecimento. O segundo ciclo é a seleção desta regra para solucionar o problema específico e o terceiro executa a ação que esta diretamente relacionada à regra escolhida, podendo inclusive ser uma nova premissa que poderá ser novamente analisada pela base de conhecimentos.

Existem na literatura algumas ferramentas consideradas motores de inferência, entre elas podemos citar: JESS - Java Expert System Shell (MENKEN, 2002), JIP - Java Internet Prolog (CHIRICO, 2009), EXPERT-RETE (SOUZA, 2005), entre outros.

Por ser uma das ferramentas objeto de estudo deste trabalho iremos detalhar o motor de inferência JESS a seguir.

A sigla JESS significa *Java Expert System Shell*, trata-se de um motor de regras e ambiente script desenvolvido completamente em Java (linguagem de programação desenvolvida pela *SUN Microsystems*). Foi originado do *CLIPS Expert System Shell*, porém, tomou formas completas e distintas. Com o JESS, os sistemas outrora "estáticos" ganham a capacidade de "raciocinar", conhecimento este adquirido através de "Regras Declarativas". Assim como o CLIPS, o JESS utiliza uma sintaxe de programação baseada no LISP (MENKEN, 2002).

O conhecimento no JESS é representado em 3 tipos:

- Rules (Regras) são inicialmente aplicadas para conhecimento heurístico baseado na experiência.
- Functions (Funções) foram criadas primeiramente para utilização no desenvolvimento *procedural*;
- Programação orientada a objeto, também criada primeiramente com intuito da utilização pelo desenvolvimento *procedural*. E com suporte para os objetos deste tipo de programação como: classes, abstração, message-handlers, encapsulamento, herança e polimorfismo.

Com isso, pode-se desenvolver um software inteligente utilizando-se apenas de regras, apenas de objetos ou uma combinação das duas opções anteriores.

O JESS fornece os elementos básicos para um sistema especialista (SE) que são (MENKEN, 2002):

- Lista de fatos e lista de instancias, que corresponde a memória global dos dados;
- Base de conhecimento, responsável por conter todas as regras, a base de regras.
- Motor de inferência, responsável pela execução das regras.

O motor de inferência é responsável pela decisão sobre quando e quais regras deverão ser executadas. Um sistema especialista baseado em regras escrito em JESS é um programa onde os fatos ou objetos, são os principais responsáveis por estimular a execução do motor de inferência.

O JESS ainda utiliza-se de outros mecanismos internos para realizar o tratamento dos dados, eles são: *defrules* e *deftemplates*.

Os *defrules* são as definições das regras propriamente ditas, já os *deftemplates* são as definições das estruturas dos objetos que servirão como entrada de dados, onde cada atributo de um *deftemplate* é chamado de *Slot*. Exemplo: uma classe *carro* definida com os atributos *nome* e *cor* seria um tipo de *deftemplate* *carro* contendo os *slots* *nome* e *cor*. Quando um objeto do tipo *carro* for encaminhado para o comportamento cognitivo, o JESS deverá possuir uma *defrule* para tratar este formato específico de *deftemplate*.

Com isso, podemos concluir que a base de conhecimento do JESS é um conjunto de *defrules* previamente incluídas no comportamento do agente, ou ainda armazenadas em banco de dados.

2.1.2. Arquitetura de Sistemas Multiagente

A arquitetura de um SMA fornece mecanismos para visualizar como o sistema funciona em sua essência. Percebe-se assim, como estão divididas as tarefas entre os agentes e como funciona a cooperação e coordenação dentro da sociedade.

De acordo com (RUSSEL e NORVIG, 2004) para um sistema ser considerado multiagente, o mesmo deve possuir as características de cooperação, coordenação e negociação entre os agentes e o usuário.

- Cooperar para alcançar em conjunto um objetivo em comum;
- Coordenar, para gerenciar as limitações e dependências das atividades dos agentes de forma a sincronizar as ações dos agentes para alcançar o objetivo em comum;
- Negociar, entre os agentes da sociedade para que seja viável a todos da sociedade.
- Precisa ainda possuir dois ou mais agentes, ou uma sociedade de agentes atuando juntos para solucionar um problema em comum, que não seriam capazes de resolver atuando de forma autônoma.

Na arquitetura, cada agente não possui todas as informações ou capacidade para resolver o problema, portanto:

- Um único agente possui um ponto de vista limitado;
- Não há controle global do sistema;
- Os dados estão descentralizados;
- A computação é assíncrona.

É importante ressaltar que as habilidades sozinhas não fornecem características necessárias para se formar uma sociedade multiagente. Farias (2009) complementa:

“[...] Adicionalmente, é preciso que estes sejam coordenados trabalhem de modo cooperativo e competitivo. Isso da o equilíbrio entre agentes coordenáveis e seus relacionamentos, associado às demais características descritas [...]” (FARIAS, 2009)

2.1.3. Comunicação na sociedade de agentes

A comunicação entre os agentes da sociedade é um fator determinante para o sucesso de um sistema multiagente. Para alcançar a cooperação e coordenação das tarefas entre os próprios agentes, é necessário que os mesmos interajam, conseqüentemente, estes devem realizar algum tipo comunicação entre si.

Sampaio (2007) afirma que existem diversas formas para os agentes alcançarem esta comunicação, e destaca entre eles os seguintes modelos:

- Comunicação direta – Neste modelo, um agente efetua a comunicação com outro agente diretamente, sem a necessidade de um intermédio;
- Federação – Para este modelo existe a presença de mediadores e facilitadores distribuídos em uma estrutura hierárquica de agentes para realizar a troca de mensagens;
- Broadcast – Neste modelo utiliza-se o envio de uma mensagem de broadcast para todos os agentes da sociedade. É utilizado quando o agente não conhece a entidade receptora ou quando há necessidade de todos os agentes da sociedade receber a mensagem;
- Blackboard ou quadro-negro – Neste modelo, bastante utilizado na Inteligência Artificial, atua como uma memória compartilhada, onde os agentes escrevem mensagens e os outros agentes da sociedade lêem esta mensagem diretamente da memória.

Para aplicar algum modelo mencionado anteriormente e assim realizar a comunicação entre agentes, existe linguagens específicas chamadas de ACL (*Agents Common Language*).

Atualmente algumas linguagens possibilitam esta comunicação, entre elas podemos citar:

- KQML – Knowledge Query and Manipulation Linguagem desenvolvida em 1990 pelo KSE (Knowledge Query and Manipulation Language), é uma linguagem e protocolo para comunicação de softwares que utilizam a tecnologia de agentes.
- FIPA-ACL – Foundation for Intelligent Phisycal Agents, é a linguagem para comunicação entre agentes criada pela FIPA (FIPA, 2009).

Contudo, as duas linguagens mencionadas possuem sintaxes semelhantes, porem não é objetivo deste trabalho a realização do aprofundamento no estudo da KQML, visto que o sistema em estudo utiliza-se da FIPA-ACL para realização da comunicação entre os agentes da sociedade.

A FIPA-ACL tem como objetivo principal padronizar a comunicação entre os agentes da sociedade através de performativas de comunicação, baseado na teoria de atos de fala ou comunicativos. (FIPA, 2009)

A FIPA-ACL esta dividia em três partes:

- Protocolos de Interação – IPS (Interactions Protocol);
- Atos de comunicação – CA (Communicatives Acts);
- Linguagem de conteúdo – CL (Content Language).

Como foi baseada na teoria de atos da fala, a FIPA-ACL possui as seguintes performativas de comunicação.

Tabela 2. Comunicação da FIPA-ACL

Performativas de comunicação		
ACCEPT_PROPOSAL	NOT-UNDERSTOOD	FAILURE
AGREE	PROPOSE	INFORM
CANCEL	QUERY-IF	INFORM-IF
CFP	QUERY-REF	INFORM-REF

CONFIRM	REFUSE	PROPAGATE
DISCONFIRM	REJECT-PROPOSAL	SUBSCRIBE
REQUEST	REQUEST-WHEN	PROXY
	REQUEST-WHENEVER	

Fonte: FIPA, 2009.

Inicialmente a FIPA-ACL não representava naturalmente o conteúdo da mensagem, com isso a FIPA desenvolveu a FIPA-SL (*Semantic Language*) (FIPA, 2009), no qual possibilitou que o mecanismo de comunicação pudesse representar a linguagem natural.

Através da FIPA-SL é possível utilizar-se de ontologias como forma de linguagem natural, fornecendo ao sistema o conhecimento necessário sobre o domínio da aplicação.

2.1.4. O framework JADE

Os SMAs são aplicações diferenciadas e necessitam de um ambiente propício para que seja realizada a cooperação, coordenação e comunicação entre os agentes da sociedade. Nesse sentido, o JADE fornece uma solução adequada para desenvolvimento de SMAs.

JADE é o acrônimo de *Java Agent Development Framework*, trata-se um middleware que proporciona a criação de aplicações multiagentes conforme as especificações da FIPA (FIPA, 2009). Foi desenvolvido na linguagem Java pelo CSELT (*Centro Studi Laboratori Telecomunicazioni Spa*) da Universidade de Parma na Italia.

O objetivo do middleware é facilitar o desenvolvimento de SMAs, para isso, o mesmo fornece a seguinte arquitetura:

- *Container* para a execução dos agentes;
- Biblioteca de classes para desenvolvimento dos agentes;
- Ferramentas gráficas para auxílio no gerenciamento, desenvolvimento e monitoramento dos agentes.

O *Container* é um dos principais recursos da arquitetura do JADE, ela possui dois agentes internos: O agente AMS (*Agent Management System*) e o agente DF (*Directory Facilitator*). O primeiro agente, é responsável por gerenciar os demais agentes da sociedade, garantindo que cada agente possua um identificador exclusivo denominado (Aid), além de possuir privilégios para criar ou excluir um agente. O segundo fornece um serviço de páginas amarelas (*Yellow-pages*) para cada agente localizar serviços de outros agentes dentro da sociedade.

O conjunto de *containers* é denominado de plataforma. Nesse sentido, uma plataforma possui vários containers, porém, existe somente um *Main-container* em cada plataforma.

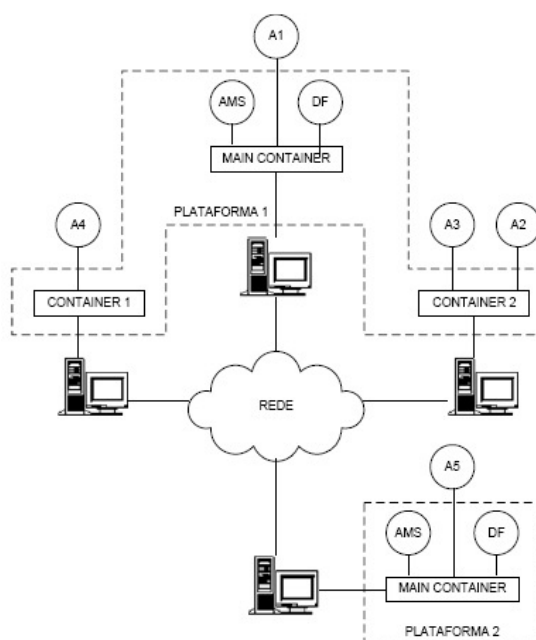


Figura 8. Containers e plataformas JADE.

FONTE: OLIVEIRA, 2006.

Dentro do JADE, tem-se um container principal ou *Main-Container*, e os demais containers, enumerados de acordo com sua criação (*container-1*, *container-2*, etc). Obrigatoriamente, os dois agentes mencionados anteriormente são iniciados automaticamente dentro do *Main-Container*.

As bibliotecas fornecidas pelo referido ambiente de execução fornecem mecanismos para criar a estrutura e comportamentos do agente. Um agente em JADE estende obrigatoriamente a classe *jade.core.Agent* fornecida, e seus

comportamentos devem implementar a classe `jade.core.behaviour.Behaviour`, conforme podemos ver na Figura 9.

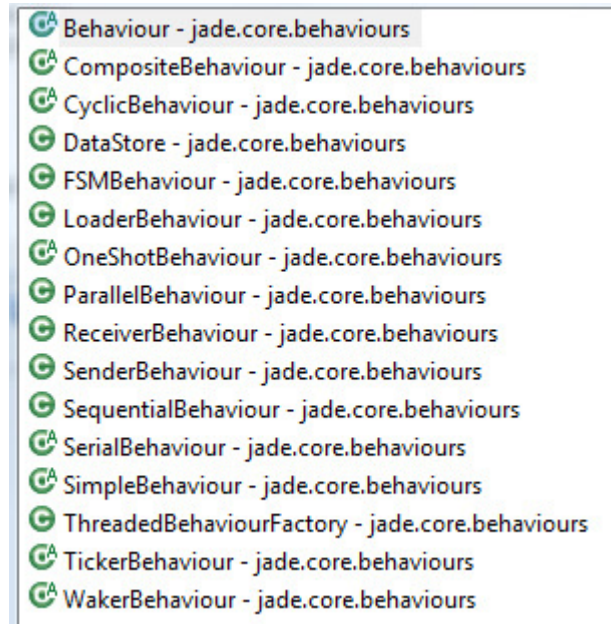


Figura 9. Comportamentos (behaviours) em JADE.

2.1.5. Metodologias para desenvolvimento de Sistemas Multiagente

Com a chegada da engenharia de software baseada em agente houve a necessidade de criar mecanismos para auxiliar a sua construção, com isso surgiram às metodologias de desenvolvimento SMAs.

O objetivo delas é fornecer técnicas de suporte para a modelagem desse tipo singular de aplicação. Contudo, ainda não foi possível reunir em uma única metodologia todas as qualidades fornecidas em cada uma delas. Com isso, podemos citar algumas metodologias para modelagem de sistemas SMA referenciadas na literatura como: MAS-CommonKADS (IGLESIAS et al., 1998), PASSI (COSSETINO, 2002), AGIL-PASSI (CHELLA, 2006) entre outras.

Abaixo será descrita com maiores detalhes a metodologia utilizada para modelar as responsabilidades dos novos agentes da sociedade.

PASSI (*Process for Specifying and Implementing Multi-Agent Systems Using UML*) é um processo para especificação e implementação de sistemas multiagente utilizando-se de notação UML (*Unified Modeling Language*).

Criada por Cossetino (2002), um dos objetivos da PASSI é fornecer ao desenvolvedor de software diferentes níveis de detalhes para modelar um sistema. A justificativa para o uso da notação UML de acordo com os autores, é que este tipo de notação é amplamente aceito no mundo acadêmico e industrial.

A PASSI esta dividida em 5 fases: a de modelo de requisitos do sistema, modelo da sociedade de agentes, modelo de implementação de agentes, modelo do código e modelo de implantação ou distribuição.

A primeira fase, do modelo de requisitos do sistema, descreve um modelo de requisitos do sistema contendo os termos dos agentes e suas finalidades, envolve 4 passos, que são:

- Criação de uma descrição do domínio (D.D). Utiliza-se de casos de uso UML para isto;
- Separação das responsabilidades dos agentes criando um diagrama de Identificação de agentes (A.Id). Utiliza-se de pacotes UML para representação;
- Identificação das responsabilidades dos agentes (R.Id). Criam-se diagramas de seqüencia mostrando o cenário da ação;
- Especificação de tarefas (T.Sp). Utiliza-se de diagramas de caso de uso e descrições das tarefas de cada agente.

A segunda fase, o modelo da sociedade de agentes, descreve o processo de interações e dependência da sociedade de agentes, este modelo envolve 4 passos:

- Identificação de regras (R.Id). Verificada na etapa anterior de requisitos do sistema;

- Descrição da Ontologia do Domínio (D.O.D.). Descreve-se através de diagrama de classes para representar o conhecimento dos agentes;
- Descrição das Regras (D.R). Utiliza-se de diagramas de classe para descrever os papéis desempenhados pelos agentes, as tarefas envolvem os papéis, comunicação, capacidades e dependências.
- Protocolo Descrição (P.D.). Representado por diagramas de seqüência para especificar a comunicação através de protocolos e termos das performativas de comunicação.

A terceira fase, o modelo de implementação dos agentes, cria-se um modelo da arquitetura proposta representado por classes e métodos, para isso utiliza-se de dois diagramas abaixo:

- Definição da Estrutura de Agente (A.S.D.). Utiliza-se de Diagramas de classe para demonstrar a estrutura da solução das classes do agente.
- Descrição do Comportamento dos Agentes (A.B.D.). Descreve-se através do uso de Diagramas de atividade para demonstrar o comportamento individual dos agentes.

A quarta fase, criação do código, descreve-se a criação e manutenção acerca do código fonte a ser criado ou utilizado pelo sistema, seguindo os passos a seguir:

- Biblioteca para Reuso de Código (C.R). Cria-se uma biblioteca com diagramas de classes e de atividades contendo o código de reuso associado;
- Base para complemento do código (C.C). Mantém o código fonte do sistema.

A quinta fase, o modelo de implantação ou distribuição, descreve um modelo de distribuição do sistema sobre o hardware, através do seguinte procedimento:

- Configuração da distribuição (D.C). usa-se diagramas de distribuição para descrever a alocação dos agentes pelas unidades de processamento disponíveis, descreve-se também quaisquer restrições sobre as migrações e mobilidade.

A Figura 10 contém as etapas da metodologia de desenvolvimento de sistemas multiagente PASSI.

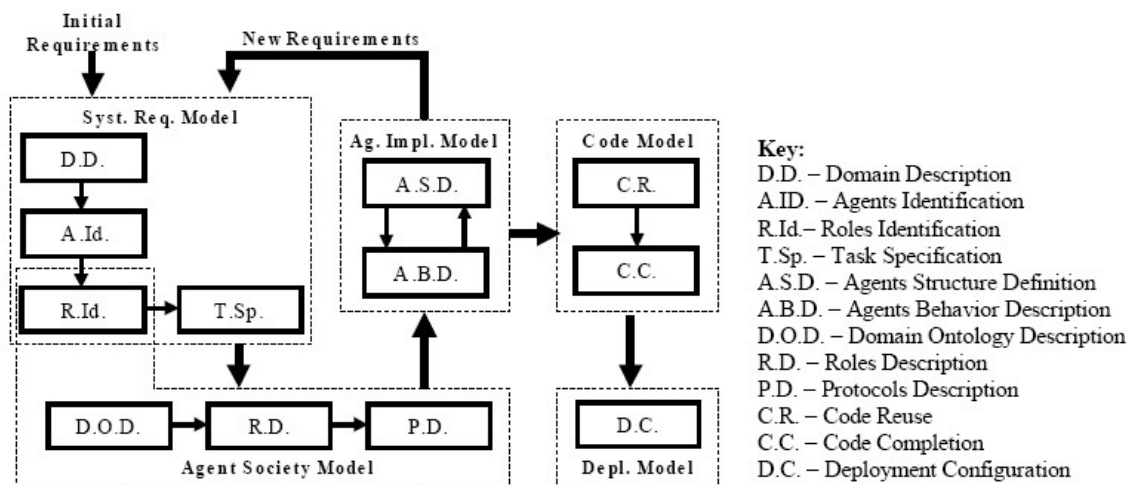


Figura 10. Etapas da metodologia PASSI.

Fonte: COSSETINO, 2002.

Conforme pôde-se perceber, a metodologia PASSI proporciona mecanismos para modelar com detalhes cada etapa do processo de desenvolvimento de um sistema multiagente.

Em seguida, são abordadas as técnicas existentes na literatura para representação do conhecimento e as metodologias existentes para auxílio na modelagem das mesmas.

2.2. Representação do conhecimento

Conhecer o domínio da aplicação é um fator determinante para o desenvolvimento de um software, pois, ao conhecer a área de atuação do trabalho, o desenvolvedor conseguirá uma boa especificação de requisitos e como conseqüência, será alcançado um software de qualidade.

Mypoulos (1981) afirma que a representação do conhecimento (RC) é um problema central da inteligência artificial. A importância da RC está relacionada ao fato de que os softwares inteligentes necessitam de conhecimento técnico sobre o sistema a ser desenvolvido. Com isso, o principal problema da RC é o desenvolvimento de uma notação que seja suficientemente precisa para representar o conhecimento.

Grigorova e Nikolov (2007) afirmam que o modelo ideal para representar o conhecimento deve ser flexível o suficiente a fim de apresentar o conhecimento com diferentes níveis de granularidade. Deve possuir habilidades de raciocínio, de utilização de contexto e trabalhar com informações incompletas e incertas;

Grigorova e Nikolov (2007) afirmam ainda que a representação do conhecimento é um elemento chave para os sistemas de inteligência artificial, e que são três os tipos de conhecimentos necessários pelos sistemas de interface natural, eles são: conhecimento sobre palavras, conhecimento sobre estrutura de frases e conhecimento sobre o domínio específico. Onde, o conhecimento sobre as palavras é relacionado a um dicionário, o conhecimento sobre a estrutura de frases está relacionado com a gramática e o conhecimento do domínio está relacionado aos objetos e relações do domínio.

Para Studer (1998) Representar o conhecimento é transferir conhecimento e adverte ainda que o simples desenvolvimento da representação do conhecimento possui conseqüências, tais como:

- O modelo construído é apenas uma representação daquela realidade, sendo que a mesma pode ser alterada e conseqüentemente o modelo sofrera mudanças;

- Devido às modificações, uma representação de um modelo é um processo cíclico, ou seja, esta em constante adaptação e por isso é um processo infinito;
- Todo processo de modelagem é apenas uma representação dependente do ponto de vista do desenvolvedor, portanto normalmente é um processo falho, com isso, uma avaliação de um processo de modelagem é necessária para o refinamento de um processo adequado.

Nesse sentido, a representação do conhecimento trata-se de uma subárea da Inteligência Artificial, o seu objetivo é representar o conhecimento do domínio de aplicação do software para tentar auxiliar no processo desenvolvimento e conhecimento do mesmo.

Com isso, pode-se afirmar que é um consenso entre os autores que não existe um modelo ideal para representação do conhecimento.

A literatura cita algumas técnicas para representação do conhecimento como: *redes semânticas*, *sistema de frames*, *thesaurus*, *dicionários*, *ontologias*, entre outras.

Abaixo, seguem os seus respectivos conceitos.

Redes Semânticas, esta técnica foi criada em 1956 por *Richard H. Richens* com o objetivo de representar o conhecimento, para isso, ela utiliza nós conectados por setas formando vértices entre si, esses nós representam os conceitos e os relacionamentos (setas) são identificados pelas relações semânticas entre eles (PINHO, 2007). Porém, devido a sua simplicidade, as redes semânticas são limitadas, não contemplando termos mais detalhados.

Sistemas de frames são considerados como a evolução das redes semânticas e trata-se de um conjunto de frames organizados em hierarquia. Frame descreve um objeto por atributos através de uma identificação por nome. (MINSKY, 1974, apud GRIGOROVA e NIKOLOV, 2007)

Os *thesauros* são outra forma de representação do conhecimento, de acordo com Jesus (2002), é definido como:

“[...] uma lista estruturada de termos associada empregada por analistas de informação e indexadores, para descrever um documento com a desejada especificidade, em nível de entrada, e para permitir aos pesquisadores a recuperação da informação que procura” (CAVALCANTI, 1978, apud, JESUS, 2002)

Já os *dicionários*, utilizam a forma de representação através de termos e conceitos, para cada termo utilizado, o dicionário conceitua utilizando-se de diversos sinônimos, muitas vezes fazendo a referencia para outros termos, conforme podemos perceber, são bastante utilizados na língua portuguesa, a exemplo temos: *Michaelis*, *Aurélio*, entre outros.

Por fim, a *ontologia*, que por fazer parte do objeto de estudo desse trabalho, será detalhada a seguir.

2.3. Ontologia

O termo *ontologia* teve origem na filosofia em meados do século XVII, com o estudo da filosofia primeira, sendo definido como:

“[...] o estudo ou o conhecimento da essência das coisas ou do Ser real e verdadeiro das coisas, daquilo que elas são em si mesmas, apesar das aparências que possam ter e das mudanças que possam sofrer.” (CHAUI, 2005)

Foi quando o filósofo alemão *Jacobus Thomasius* considerou que os estudos da filosofia primeira, realizados por *Aristóteles* seu antecessor, deveriam ser conhecidos através do termo *ontologia* (CHAUI, 2005). Com isso, o primeiro filósofo a definir os estudos de uma *ontologia* foi *Aristóteles*.

Na computação o termo *ontologia* é utilizado por áreas como inteligência artificial, web semântica, engenharia de software e arquitetura da informação, sempre com o objetivo de representação do conhecimento sobre o mundo ou uma parte dele. (WIKIPEDIA, 2009)

Para Gruber (1993) uma ontologia é uma especificação explícita de uma conceitualização, já Noy e McGuinness (2001) afirmam que uma ontologia tem como objetivos, compartilhar o entendimento comum da estrutura de informações com

pessoas ou agentes de software, permitir reutilizar o conhecimento do domínio, separar o conhecimento de domínio do conhecimento operacional e analisar o conhecimento do domínio.

Para Guarino (1995, 1998), uma ontologia é um artefato de engenharia, constituído por um vocabulário específico, utilizado para descrever uma determinada realidade, adicionado a um conjunto de intenções explícitas com o objetivo de dar significado a um vocabulário de palavras.

Existem várias tipos de classificações para as ontologias, sendo que uma das mais populares, citada nos trabalhos de (Kavouras, 2004), (Almeida, 2003), trata-se da proposta de Guarino (1995, 1998), que utiliza e as classifica como:

- Ontologias de alto nível: descrevem conceitos gerais como espaço, tempo, assunto, objeto, evento, ação, etc. São independentes de um problema particular ou domínio;
- Ontologias de domínio: descreve um vocabulário relacionado a um domínio genérico (como a medicina, ou automóveis);
- Ontologias de tarefas: descrevem uma tarefa ou atividade genérica (como diagnosticar ou venda), pela especialização dos termos introduzidos na ontologia de nível superior;
- Ontologias de Aplicação: descrevem conceitos sobre um determinado domínio ou tarefa, que muitas vezes são especialidades relacionadas ontologias.

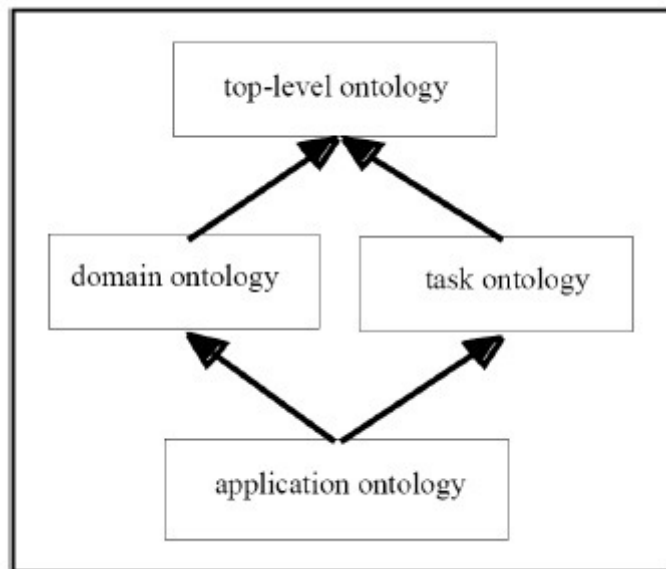


Figura 11. Classificação de ontologias.

Fonte: GUARINO, 1998.

Kavouras (2004) ainda reforça a classificação das ontologias de acordo com o nível de formalidade ou granularidade.

De formalidade podem ser de 3 (três) tipos:

- Informal, a mais simples das ontologias, sendo constituída por conjunto de conceitos e rótulos, organizados por hierarquia.
- Terminológica, composta por uma hierarquia, contendo conceitos definidos através da linguagem natural.
- Formal, que possui axiomas e definições declaradas em uma linguagem formal.

De granularidade são divididos em seis tipos:

- Ontologia de alto nível: responsável pela definição de conceitos gerais como: espaço, tempo, objeto, etc, que são independentes de um domínio particular;
- Ontologia geral: define conceitos fundamentais relativos ao conhecimento humano;

- Ontologia de domínio: define conceitos associados a um domínio específico;
- Ontologia de tarefas: define conceitos relacionados com a execução de uma determinada tarefa ou atividade;
- Ontologia de aplicação: define conceitos essenciais para o planejamento de uma aplicação específica;
- Meta-ontologia ou Genérica: definem conceitos fundamentais que são comuns em vários domínios, tais conceitos podem ser mais especializados do domínio;

De acordo com Gruber (1993) e Guarino (1995, 1998) a representação de conhecimento através da utilização da técnica de ontologias fornece vantagens como:

- Permitem que os termos da ontologia sejam compartilhados com sua utilização, fornecendo assim uma colaboração;
- Fornece uma descrição correta do conhecimento, evitando erros de interpretação dos termos do domínio;
- Permite que o vocabulário seja estendido, ao se usar uma ontologia genérica, pode-se estender a mesma para um domínio específico, facilitando assim o seu desenvolvimento;
- Fornecem uma visão do domínio da aplicação que se pretende modelar.

2.3.1. Linguagens para desenvolvimento de ontologias

Para se falar das linguagens para desenvolvimento de ontologias, faz-se necessário conhecer um pouco da Web Semântica. Este é o termo encontrado pelo W3C (W3C, 2009) para a tecnologia responsável por tentar explorar as funcionalidades e potencialidades da internet atual.

A internet foi projetada para humanos, utilizando-se de leitura ótica, e tornando esses dados incompreensíveis para máquinas (W3C, 2009), com isso, surgiram tecnologias para tentar “entender” esse material e assim fornecer mecanismos de leitura para computadores, utilizando-se para isso de padrões e ferramentas para dar significados às páginas, possibilitando assim que os usuários e programas trabalhem em cooperatividade.

Nesse sentido, a arquitetura da web semântica possui padrões para identificar os inúmeros recursos da web, bem como para a representação sintática, estrutural, semântica e lógica. Com base nessas informações foi criada a arquitetura da web semântica, como podemos ver na Figura 12.

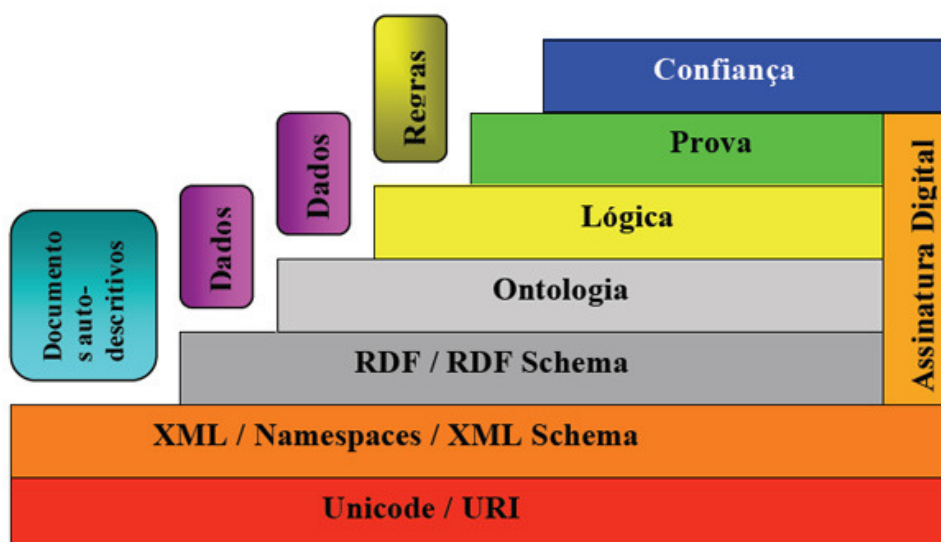


Figura 12. Arquitetura da Web Semântica.

Fonte: ROSA, 2002.

Existem linguagens para se definir recursos e ontologias, o uso das mesmas fornece auxílio no processo de representação do conhecimento do domínio. Tais linguagens envolvem as características necessárias como: classes, termos, conceitos, propriedades e seus relacionamentos.

Nesse sentido, o W3C (W3C, 2009), um consórcio internacional de empresas e organizações de diversas áreas espalhadas ao redor do mundo que tem como objetivo criação de padrões web desenvolveu algumas dessas linguagens

para descrição de ontologias e recursos. Entre elas podemos citar: OWL, RDF e RDF-SCHEMA. Porém, além das linguagens citadas acima, existem outras como: DAML+OIL, DAML+ONT e OIL.

Abaixo, segue uma breve descrição de algumas dessas linguagens.

Tabela 3. Linguagens para descrição do conhecimento

Linguagem	Descrição
RDF	Resource Description Framework (RDF) é uma linguagem para processar metadados, ou seja, descrever recursos, promovendo a interoperabilidade entre aplicações que trocam informações compreensíveis para máquinas na Internet. RDF enfatiza facilidades para permitir tratamento automatizado de recursos da web. (RDF, 2007)
DAML+OIL	DAML+OIL foi desenvolvida como uma extensão de XML e baseada na RDF, é uma linguagem para criação de ontologias, além de aumentar o poder de expressão e modelagem do conhecimento em documentos Web. Prove meios para modelar domínio de conhecimento. Incorpora aspectos da DAML e OIL. (DAML+OIL, 2009)
OIL	A OIL foi criada com objetivo de representar de forma semântica e de maneira que pudesse ser acessível por computadores, modelando para isso os domínios de conhecimento através de ontologias. Desenhada para fornecer suporte aos padrões do W3C, como o XML e RDF, a OIL utiliza a modelagem baseada em RDF-SCHEMA. (OIL, 2009)
OWL	OWL foi criada com o objetivo de descrever classes e as relações existentes entre elas, possibilitando a reutilização das mesmas, ou a possibilidade de ser herdada por aplicações. Para isso, utiliza-se de termos de vocabulários que podem ser representados explicitamente através de descrição de classes, propriedades e suas instâncias, bem como as relações entre entidades nestes vocabulários. OWL possui recursos que vão além das linguagens anteriores, permitindo maior compreensão do conteúdo representado. (OWL, 2009)

2.3.2. Metodologias para desenvolvimento de ontologias

De acordo com alguns autores [Gruber (1993), Noy e McGuinness (2001)], não existe uma forma correta para construção de ontologias. Com o objetivo de auxiliar nessa construção, a literatura menciona algumas metodologias, guias e técnicas. Entre elas podemos citar: a Methodology criada por Fernandez et al. (1997), a metodologia 101 criada por Noy e McGuinness (2001) e UML2ONTO criada por Victorette (2008).

Abaixo é descrita com detalhes a metodologia 101, por esta fornecer todos os recursos necessários para modelar uma ontologia no nível de detalhes desejado.

Desenvolvida por Noy e McGuinness (2001), é uma metodologia para criação de ontologias que fornece um processo iterativo baseado em 7 fases.

Para se construir uma ontologia os autores afirmam que algumas regras devem ser tomadas:

- Não existe um único modelo para modelagem de domínio, existem sempre alternativas. Ele afirma que a melhor proposta depende da aplicação que se deseja modelar;
- O processo de desenvolvimento de uma ontologia é iterativo;
- Os conceitos da ontologia devem ser os mais próximos da realidade e suas relações com o domínio.

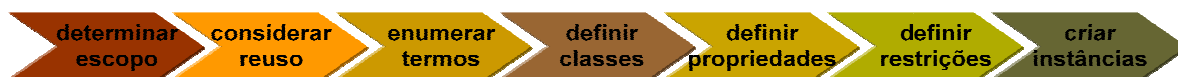


Figura 13. Passos da metodologia 101

Fonte: NOY e MCGUINNESS, 2001.

I. Definição do domínio e escopo da ontologia

O primeiro passo da metodologia 101 tem início com a definição do domínio e escopo da ontologia. Para isso, os autores definiram algumas perguntas que devem ser respondidas. Como:

- Qual é o domínio que a ontologia irá atuar?
- Qual o objetivo de se usar esta ontologia?
- Para quais perguntas as informações contidas na ontologia irão fornecer respostas?

É importante destacar que no decorrer da modelagem, as respostas a estas perguntas podem mudar, porém ajudam a limitar a aplicação da ontologia. Outra forma de se alcançar o escopo está na forma de “perguntas de competências”. Para os autores, o alcance da ontologia se dá através de uma lista de perguntas que devem ser respondidas com base no conhecimento da ontologia.

Ao se chegar às respostas das perguntas acima, um documento descritivo deverá ser gerado contendo o produto desta fase.

II. Considerar reutilização de possíveis ontologias existentes

A segunda fase descreve que deve ser considerada a existência de alguma ontologia no mesmo domínio e escopo para que possa ser refinada, com isso pode-se aperfeiçoar e ampliar a mesma para o nosso domínio e tarefa específicos. O autor afirma ainda que muitas ontologias estejam disponíveis em formato digital e que reutilização de ontologias existentes pode ser um requisito, caso o sistema tenha que interagir com outras aplicações. Nesse sentido, esta fase é dedicada à realização de pesquisas em sites especializados e em toda a literatura sobre ontologias de domínio para considerar a reutilização de possíveis ontologias compatíveis com o domínio.

III. Enumeração dos termos da ontologia

Na terceira fase chega-se ao momento de definição dos termos, com isso, o autor sugere que seja realizado um procedimento *Top-Down*. Iniciando com a listagem de todos os termos que serão utilizados para fazer declarações ou servir de informação para usuário. Para isso, outras “perguntas de competência” devem ser

respondidas como: Quais termos utilizar? Quais as propriedades de cada termo? E o que gostaríamos de dizer sobre esses termos?

Os autores afirmam ainda que, é importante tentar declarar todos os termos, mesmo que sendo sobrepostos, sem a preocupação de suas propriedades, relacionamentos ou propriedades, nem se são classes. Pois, nas fases subseqüentes serão realizados os desenvolvimentos das hierarquias, classes e definição das propriedades.

IV. Definição de classes e hierarquia de classes

Na quarta fase chega o momento de criar as classes e a hierarquia das mesmas, para isso o autor cita a classificação para desenvolvimento de hierarquia de classes definida por Ushold e Gruninger (1996):

Top-down – tem inicio com a definição dos conceitos mais gerais do domínio e em seguida passa-se para a especialização de cada conceito;

Botton-up – tem inicio com as classes mais especificas e sem seguida, passa-se ao agrupamento dessas classes em conceitos mais gerais.

Combinação – trata-se da união das duas técnicas anteriores.

Com isso, tem-se a primeira versão das classes e suas hierarquias, como podemos ver no exemplo a seguir sobre uma ontologia de vinhos:

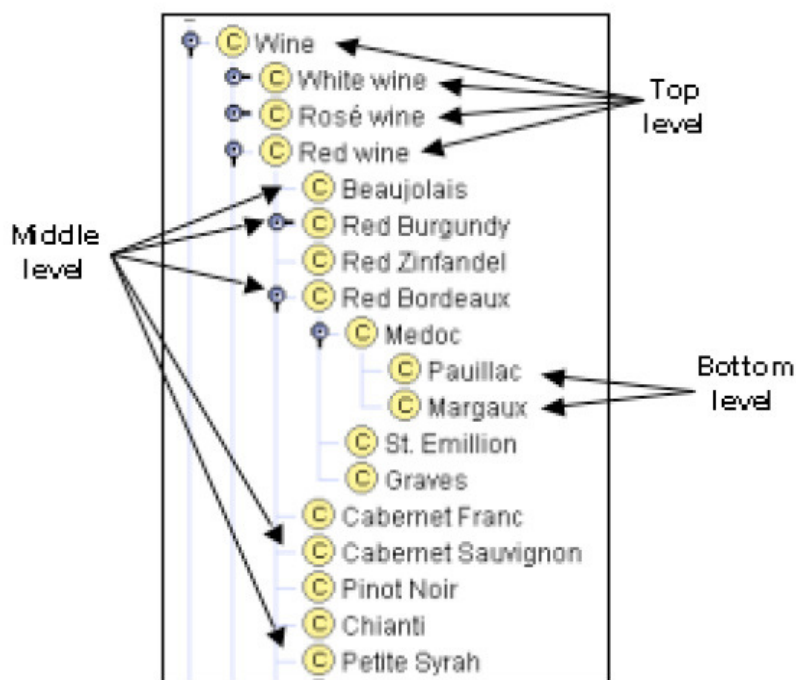


Figura 14. Tipos de hierarquia de classes.

Fonte: NOY e MCGUINNESS, 2001.

Na Figura 14 percebe-se o resultado da aplicação das técnicas mencionadas anteriormente. No lado direito, “top-level”, tem-se as classes de nível mais alto, contendo o vinho e seus tipos, ao lado esquerdo, “middle-level”, tem-se as classes intermediárias que são os tipos de vinho, no caso tipos de vinho tinto, e por fim, ao lado direito, em baixo, “bottom-level” tem-se as classes mais especializadas, contendo os tipos de vinhos tintos.

Noy e Mcguinness (2001) ainda afirmam que não existe uma técnica melhor das três mencionadas. Dependendo bastante do ponto de vista do desenvolvedor da aplicação.

V. Definição das propriedades das classes - Slots

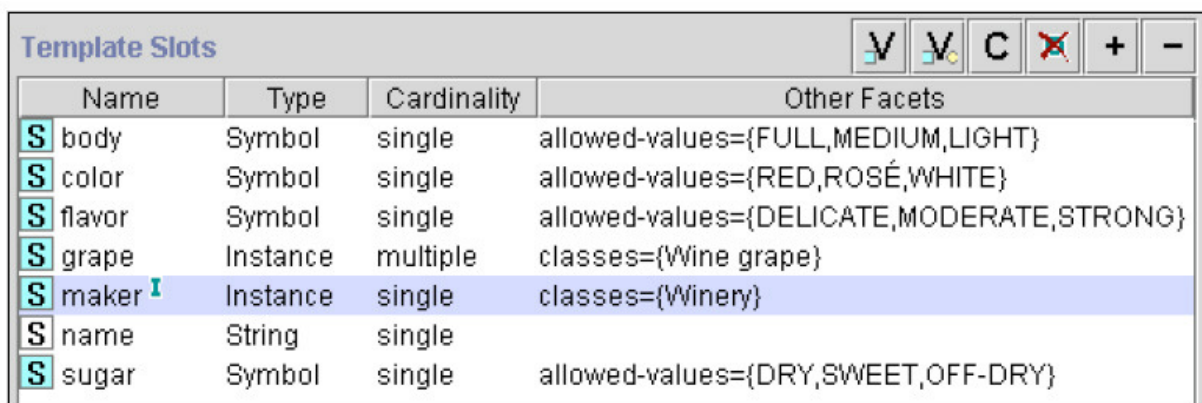
Na fase seguinte, passa-se para a descrição do conteúdo interno de cada classe. Com isso, uma vez definidas as classes, é preciso descrever a estrutura interna do restante dos conceitos, os *slots*, a partir das classes selecionadas da lista de termos criada na terceira etapa.

A maioria dos termos restantes é susceptível de serem propriedades dessas classes. Estas condições incluem, por exemplo, a cor do vinho, seria um *slot* da classe vinho.

Para cada propriedade na lista, é preciso determinar qual a classe que ela depende. O autor afirma que, provavelmente, o restante das propriedades podem se tornar *slots* das classes existentes.

VI. Definição dos tipos de *Slots*

Nesta fase, devem-se definir algumas propriedades dos *slots* criados na fase anterior, eles são: a cardinalidade que define a quantidade de valores um único *slot* pode ter, de um único valor a múltiplos valores, o tipo do valor do *slot* (*string*, *number*, *boolean*, *enumerated*, *instance-type*) e o domínio e alcance do *slot*.



Name	Type	Cardinality	Other Facets
S body	Symbol	single	allowed-values={FULL,MEDIUM,LIGHT}
S color	Symbol	single	allowed-values={RED,ROSÉ,WHITE}
S flavor	Symbol	single	allowed-values={DELICATE,MODERATE,STRONG}
S grape	Instance	multiple	classes={Wine grape}
S maker ^I	Instance	single	classes={Winery}
S name	String	single	
S sugar	Symbol	single	allowed-values={DRY,SWEET,OFF-DRY}

Figura 15. Slots da classe vinho.

Fonte: NOY e MCGUINNESS, 2001.

Na Figura 15 extraída da fase de definição dos tipos de *slots*, pode-se visualizar quais *slots* foram criados para a classe *wine* (Vinho) e seus respectivos tipos.

VII. Criação das instancias

Na ultima fase, deve-se criar individualmente as instancias de cada classe na hierarquia. Nesse momento será necessário realizar o seguinte procedimento: 1)

escolhe-se uma classe; 2) cria uma instância individual desta classe; 3) realiza-se o preenchimento de valores dos *slots*;

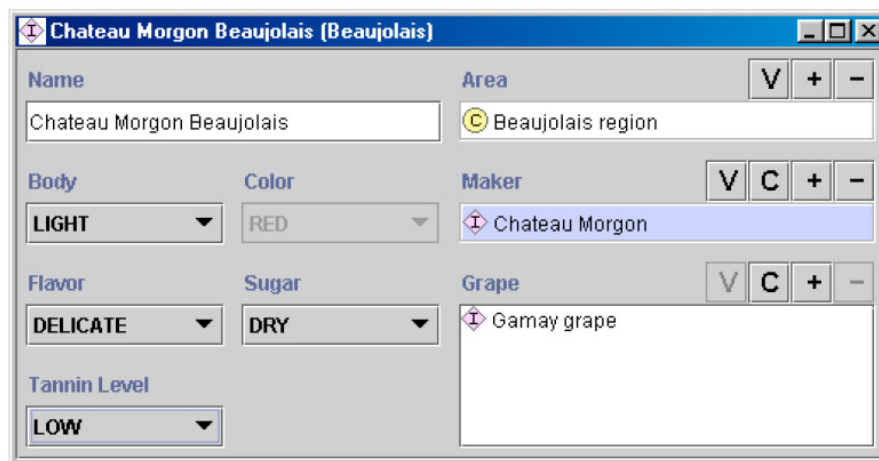


Figura 16. Instancia da classe vinho.

Fonte: NOY e MCGUINNESS, 2001.

Na Figura 16, podemos ver a criação de uma instancia da classe (Beaujolais) utilizando-se o nome "Chateau Morgon Beaujolais" e o preenchimento das suas propriedades, *body*, *color*, *flavor*, *suager* e *tammin level*. Nota-se nesse exemplo que as propriedades da classe *Beaujolais* são herdadas da classe principal, *Wine*.

Com isso, depois de realizadas todas as fases têm-se o modelo da ontologia finalizado, que pode ser exportado para diversas linguagens de criação de ontologias ou mesmo criado um modelo de classes para ser utilizado por sistemas multiagente.

2.4. Trabalhos relacionados

Vários trabalhos são encontrados na literatura apresentando diferentes propostas para a utilização de técnicas de representação do conhecimento, sistemas inteligentes e a combinação das duas técnicas.

Ying et al. (2009) apresenta um sistema inteligente baseado em agentes para a indústria com o objetivo de coordenar a execução da fabricação e dar suporte na tomada de decisão no processo químico industrial. O sistema multi-agente (MAS) foi desenvolvido para fornecer uma infra-estrutura flexível para a integração de processos químicos e do processo de informação modelos durante a fabricação.

O sistema proposto por Ying et al. (2009) possui uma base de conhecimento para que os agentes da sociedade possam se comunicar e cooperar entre si para trocar e partilhar informação, e assim, atingir decisões coordenadas para lidar com vários cenários no processo de fabricação e operações de gestão. Foi utilizada a técnica de inteligência artificial baseada em regras para suporte a tomada de decisão, utiliza-se ainda de técnicas de inteligência artificial tais como redes neurais artificiais. O middleware JADE (Java Agent Development Framework) é usado para execução do sistema inteligente. Ontologias foram utilizadas para especificar a infra-estrutura do agente com base no sistema, e para caracterizar as informações contidas nas mensagens trocadas entre agentes.

Sampaio (2007) propôs uma sociedade de agentes voltados para a detecção de fraudes em impostos municipais. Para isso, foi proposta uma modelagem a partir de dados históricos de ações e das funções de inferência de comportamento utilizando-se de dados da Prefeitura municipal de São Luís para aplicação do estudo de caso. O sistema foi modelado utilizando-se das metodologias MAS-CommonKADS e construída uma ontologia baseada na metodologia MADEM (Multi-Agente Domain Engineering Methodology),

A execução do referido sistema multiagente foi aplicado no middleware JADE (Java Agent Development Framework) e utilizado ainda o protege para desenvolvimento da ontologia. Já o motor de inferência foi criado utilizando a máquina de inferência em JESS, e sua base de conhecimento foi baseada em regras. Com isso, Sampaio (2007) provou que as fraudes municipais possuem padrões que podem ser detectados através da utilização das tecnologias mencionadas anteriormente.

Farias (2009) propôs uma aplicação multiagente para a seleção de falhas e tomada de decisão em viradores de vagões da VALE. Tendo como objetivo o apoio a tomada de decisão em células de descarga de minério. As particularidades das estruturas cognitivas de agentes foram criadas utilizando o motor de inferência

Jess, objetivando encadear decisões e recomendações de planos de manutenção feitas pelo sistema.

A representação do conhecimento sobre o domínio da aplicação foi expressa através de uma ontologia. Os artefatos produzidos durante as fases do ciclo de desenvolvimento têm a aplicação das técnicas da metodologia PASSI, Inteligência Artificial, do middleware JADE para execução do ambiente multiagente e do framework Protégé para criação da ontologia. Farias (2009) utilizou-se de ferramentas já consolidadas para criação do sistema.

Farias (2009) demonstrou que as técnicas de IA aplicadas a indústria fornecem soluções que auxiliam no processo de tomada de decisão quando há necessidade de escolhas em curto espaço de tempo.

Vianney et al. (2005) apresentou um estudo comparativo sobre análises químicas realizadas sobre o combustível diesel. Para isso, aplicou técnicas de redes neurais para determinar e validar as amostras do combustível. Para recuperação dos dados a serem analisados, utilizou os equipamentos e técnicas ASTM extraídas pelos métodos de referencia como: índice de cetano, densidade, viscosidade, temperaturas de destilação a 50% (T50) e 85% (T85) de valorização, bem como o conteúdo total enxofre (% w / w) modelados utilizando FTIR-ATR, FTNIR, e usando espectroscopia.

Ao final do experimento, Vianney et al. (2005) concluiu que os modelos PLS / FT-Raman apresentaram resultados razoáveis somente para a temperatura de destilação a 50% (T50). Contudo, nenhuma das técnicas foi capaz de gerar uma calibração adequada dos modelos PLS para a determinação do teor de enxofre.

Vianney et al. (2005) afirmou ainda que os modelos ANN / FT-Raman apresentaram os melhores desempenhos, em relação aos outros modelos R2, apresentando valores acima de 85% RMSEP, alguns deles com valores significativamente menores do que aqueles obtidos com FTIR-ATR e FTNIR. Por fim, os modelos ANN / FT-Raman e ANN/FTIR-ATR1 foram capazes de estimar o teor total com 0,01% (w / w) precisão.

Honorato et al (2008) realizou um estudo relacionado as amostras de gasolina. Onde foram coletadas 160 amostras do combustível comercial em cinco estados brasileiros e analisados por métodos ASTM para 13 propriedades. Análise de componentes principais (PCA) foi empregada para investigar o efeito de infravermelho na região espectral (ou próximo ao centro), calibração algoritmo

(principal componente regressão, por mínimos quadrados parciais ou múltiplos de regressão linear) e pré-processamento procedimento (derivado, nivelamento e variável seleção) resultando na média da raiz quadrada, erro de predição (RMSEP).

Com isso, Honorato et al (2008) apresentou um estudo sistemático da influência do espectro de gama, pré-tratamento e de procedimentos de calibração algoritmos sobre a determinação da gasolina com parâmetros espectroscópicos, com base em uma análise dos principais componentes valores relativos RMSEP.

Com base nos trabalhos acima descritos, podemos perceber que as técnicas de inteligência artificial, como agentes inteligentes e redes neurais, e representação do conhecimento (RC), através de ontologias, são comumente utilizadas para fornecer suporte no apoio de tomada de decisões nas mais diversas áreas de atuação.

3. ONTOLOGIA DE COMBUSTÍVEIS PARA O SISTEMA DE MONITORAMENTO DA QUALIDADE DE COMBUSTÍVEL

Este capítulo tem como objetivo, descrever o processo de modelagem da *ontologia de combustíveis* adicionada ao Sistema Inteligente para Monitoramento e Controle da Qualidade de Combustíveis – *SIMCQC* (SILVA, 2008).

O objetivo da ontologia é fornecer uma linguagem natural para a aplicação, contendo todos os termos utilizados no processo de análise de combustível do LAPQAP.

Com a ontologia, será possível representar o conhecimento do domínio da química, mais especificamente o domínio de análises químicas do Programa de Monitoramento e Controle da Qualidade de Combustível (PMQC).

Além da representação do conhecimento, outro objetivo a ser alcançado com a *ontologia*, é o de servir como linguagem de comunicação entre os agentes do Sistema Inteligente para Monitoramento e Controle da Qualidade de Combustível (*SIMCQC*). Após a inclusão da ontologia, os agentes da sociedade terão a capacidade de transmitir as amostras de combustíveis utilizando o vocabulário fornecido pela ontologia, e com isso, pretende-se fornecer uma linguagem completa do domínio da aplicação ao mecanismo de comunicação do *SIMCQC*.

Para modelar a ontologia foi utilizada a metodologia 101 e a ferramenta de auxílio na construção de ontologias, o framework Protégé (PROTEGE, 2009). Em seguida, são detalhadas as etapas para a criação da ontologia.

3.1. Definição da ontologia de combustíveis

Foi denominada como *ontologia de combustíveis* a ontologia responsável por representar o conhecimento do domínio dos combustíveis sobre os dados da etapa de análises químicas realizadas pelo LAPQAP. O domínio da aplicação faz parte da interpretação e análise de combustíveis para determinar o padrão de

qualidade no processo de análises químicas do programa de Monitoramento e controle da qualidade de combustível (PMQC).

De acordo com Kavouras (2004) a *ontologia de combustíveis* está classificada como uma *ontologia de aplicação* por fornecer uma solução para um software de um domínio específico.

Conforme visto anteriormente, é um consenso entre Gruber (1993), Guarino (1995, 1998), Kavouras (2004) que não existe uma forma e metodologia correta para criar uma ontologia. Em face dessas afirmações, para modelar a referida ontologia utilizou-se da Metodologia 101 (NOY e MCGUINNESS, 2001) quase em sua totalidade, por esta fornecer todos os requisitos que são utilizados nessa pesquisa.

Contudo, a última etapa do desenvolvimento da ontologia foi adaptada para utilizar a ferramenta para a construção de ontologias, o Framework Protégé (PROTEGE, 2009) e o *plugin OntologyBeanGenerator* (2008) com o objetivo de criar o modelo físico das classes *JavaBean* da ontologia no padrão J2SE e assim testar a aplicação no ambiente de execução, o middleware JADE (Java Agent Development Framework).

A modelagem foi dividida em 7 etapas: a primeira, responsável por determinar o domínio e escopo da aplicação; a segunda, realizam-se pesquisas na literatura sobre ontologias existentes; a terceira, baseia-se na construção das listas de termos importantes do domínio; a quarta, criam-se as classes; a quinta, criam-se os *slots* internos das classes; a sexta, são definidos os tipos dos *slots* e por último, realiza-se a criação do modelo físico da mesma.

A seguir, detalha-se o processo de modelagem da ontologia de combustíveis (*Fuel Ontology*).

3.2. Domínio e escopo da ontologia

A primeira etapa da metodologia 101 (NOY E MCGUINESS, 2001) fornece mecanismos para definição do domínio e escopo da ontologia. Para isto ocorrer, o objetivo é responder as perguntas criadas pelos autores da metodologia e ainda, criar novas perguntas para serem respondidas. Essas novas perguntas servirão para definir o escopo da ontologia.

Assim, após responder as perguntas definidas na primeira fase da metodologia podemos afirmar que a ontologia irá atuar no domínio das análises químicas do programa de Monitoramento e controle da qualidade de combustível realizada pelo LAPQAP. Conforme dito anteriormente, está enquadrada na classificação de Kavouras (2004) como uma ontologia de aplicação.

O objetivo principal da ontologia é representar os termos e conceitos do domínio em forma ontológica para serem tratados posteriormente por um sistema multiagente que irá efetuar a inferência sobre os dados representados.

As perguntas que a ontologia deve responder são:

- Qual o objeto principal de análise da ontologia?
- Quais técnicas o *SIMCQC* fornece como suporte no processo de análise da amostra de combustível do tipo Y?
- Quais são as propriedades para a amostra de combustível do tipo Y?
- Qual o resultado da análise da amostra de combustível do tipo Y para a técnica de destilação?
- Qual o resultado da análise da amostra de combustível do tipo Y para a técnica de Composição (Infravermelho)?
- Qual o resultado da análise da amostra de combustível do tipo Y para a técnica de aspecto e cor?

- Qual o resultado da análise da amostra de combustível do tipo Y para a técnica de Massa Específica?
- Qual o resultado final da análise de todas as amostras de combustível do tipo Y?
- Qual o responsável técnico pela análise das amostras de combustível do equipamento X?

Legenda: X = qualquer equipamento de análise de combustível, Y = qualquer tipo de combustível analisado.

Com o domínio definido, a ontologia de combustíveis deverá fornecer as respostas necessárias para as perguntas informadas anteriormente. Ao final desta etapa, obtém-se um documento descritivo contendo as informações da mesma.

Quadro 1. Documento de descrição da ontologia

Descrição da ontologia de combustíveis

Esta ontologia tem como objetivo apresentar uma visão do domínio de análises químicas do programa de Monitoramento da qualidade de combustível. Tendo como objetivo principal servir como vocabulário de termos e conceitos entre os agentes da sociedade multiagente do Sistema Inteligente para Monitoramento e Controle da Qualidade de Combustíveis (*SIMCQC*). Com isso, deverá fornecer suporte para a criação de pacotes contendo as amostras de combustíveis utilizando-se da ontologia para representar as informações durante o tráfego de mensagens dentro da sociedade. Desenvolvida por Paulo Jose Correa, Osevaldo Farias e Glene Henrique. Foram utilizadas as seguintes fontes de informações:

1. AGENCIA NACIONAL DE PETROLEO, 2008, [HTTP://WWW.ANP.GOV.BR](http://www.anp.gov.br)
2. PROGRAMA DE MONITORAMENTO E CONTROLE DA QUALIDADE DE COMBUSTIVEL, 2001, [HTTP://WWW.ANP.GOV.BR/PROGRAMA_MONITORAMENTO.ASP](http://www.anp.gov.br/PROGRAMA_MONITORAMENTO.ASP)
3. LABORATORIO DE ANÁLISE E PESQUISA EM QUIMICA ANALITICA DE PETROLEO, 2001, [HTTP://QUIMICA.UFMA.BR/LAPQAP](http://quimica.ufma.br/lapqap)
4. SISTEMA INTEGRADOR PARA MONITORAMENTO DO COMBUSTIVEL, 2005,

- | |
|--|
| <p>5. SISTEMA INTELIGENTE PARA CONTROLE DA QUALIDADE DE COMBUSTIVEL, 2008</p> <p>6. DOCUMENTOS TECNICOS PARA ANÁLISE DE COMBUSTIVEL, 2009, HTTP://QUIMICA.UFMA.BR/LAPQAP</p> |
|--|

3.3. Considerar a reutilização de ontologias existentes

Nesta etapa, foi realizada uma pesquisa sobre ontologias do domínio que pudessem ser reutilizadas. Nesse sentido, as pesquisas foram realizadas utilizando-se dos seguintes critérios: 1) compatibilidade com o domínio da química analítica, química ou bioquímica; 2) proximidade com o domínio de combustíveis; e 3) proximidade com programas de monitoramento de combustíveis;

Abaixo seguem alguns dos repositórios on-line de ontologias onde as pesquisas foram realizadas:

Tabela 4. Repositórios on-line de ontologias

Repositório	URL
KBS Ontology Projects Worldwide (2009)	http://www.cs.utexas.edu/users/mfkb/related.html
Ontology Registry and Repository (2009)	http://mmisw.org/or/
TONES Ontology repository (2009)	http://owl.cs.manchester.ac.uk/repository/browser
DAML Ontology Library (2009)	http://www.daml.org/ontologies/
OpenOntologyRepository (2009)	http://ontolog.cim3.net/cgi-bin/wiki.pl?OpenOntologyRepository

Após a realização da fase de pesquisas para localização de ontologias do mesmo domínio, foram encontradas duas ontologias do domínio da química e bioquímica. A primeira chamada *Biochemistry Ontology (primitive)* desenvolvida por

Michel Dumontier do *Dumontier Lab* em 01/10/2007 e a segunda, chamada de *Chemistry Ontology (primitive)*, também foi desenvolvida pelo mesmo autor, laboratório e data.

Após a análise das ontologias encontradas, percebeu-se que estas não forneceram suporte ao problema exposto no nível desejado, portanto, não foram aproveitadas.

3.4. Enumeração dos termos da ontologia

Nesta fase o objetivo é a criação de uma lista de termos que são utilizados no domínio da aplicação definido na primeira etapa. Para isso, respondemos as seguintes perguntas: Quais termos são utilizados? Quais as propriedades de cada termo? O que gostaríamos de fazer com esses termos?

Para responder estas perguntas foram utilizadas análises de documentos técnicos (ANEXO I), entrevistas com o técnico de laboratório (especialista do domínio), análise das técnicas utilizadas nos ensaios químicos e por fim análise das normas técnicas utilizadas pelo LAPQAP.

Ao final, com todas as perguntas respondidas, criou-se o produto desta fase, uma lista contendo todos os termos do domínio da aplicação.

Tabela 5. Termos do domínio de combustíveis

Termos do domínio	
TERMO	DESCRIÇÃO
Amostra	Nome da amostra em análise.
Combustível	Tipo de combustível da amostra.
Gasolina	Tipo de combustível/amostra
Álcool	Tipo de combustível/amostra
Diesel	Tipo de combustível/amostra
Biodiesel	Tipo de combustível/amostra
Relatório	Termo do sistema

Exportar	Funcionalidade do sistema
Converter	Funcionalidade do sistema
Propriedade	Propriedade de uma amostra
Técnica	Tipo de ensaio técnico realizado em uma amostra
Responsável	Nome do responsável pela análise química.
Nome da Amostra	Nome da amostra em análise.
Data de Análise	Data de análise da amostra
Etanol	Nome de propriedade de combustível
Benzeno	Nome de propriedade de combustível
Olefinas	Nome de propriedade de combustível
Saturadas	Nome de propriedade de combustível
Mon	Nome de propriedade de combustível
Ron	Nome de propriedade de combustível
Composição (infravermelho)	Nome de técnica realizada
Ponto de Fulgor	Nome de técnica realizada
Destilação	Nome de técnica realizada
Aspecto e Cor	Nome de técnica realizada
Massa Especifica	Nome de técnica Realizada
Teor de álcool	Nome de propriedade de combustível
Ponto de Fulgor	Nome de técnica Realizada
Condutividade	Nome de técnica Realizada
Teor de hidrocarboneto	Nome de propriedade de combustível
pH	Nome de técnica Realizada
Enxofre	Nome de propriedade de combustível
Índice de Cetano	Nome de propriedade de combustível
Viscosidade cinemática	Nome de propriedade de combustível
Sódio e potássio	Nome de propriedade de combustível
Corrosividade ao cobre	Nome de propriedade de combustível
Ponto de entupimento	Nome de propriedade de combustível
Índice de acidez	Nome de propriedade de combustível
Glicerina livre	Nome de propriedade de combustível
Glicerina TOTAL	Nome de propriedade de combustível
Monoglicérides	Nome de propriedade de combustível
Diglicérides	Nome de propriedade de combustível
Triglicérides	Nome de propriedade de combustível
Estabilidade a oxidação	Nome de propriedade de combustível

Na tabela acima podemos identificar todos os termos utilizados atualmente pelo processo de análises químicas do programa de monitoramento realizado pelo LAPQAP. Vale ressaltar que os termos citados representam os

conceitos, predicados, propriedades e ações que possam existir no domínio da aplicação.

Contudo, Noy e Guinness (2001) afirmam que a criação de um novo termo é passível de inclusão na lista de termos, pois o processo de criação sofre modificações no decorrer da modelagem.

3.5. Definição das *classes* e suas hierarquias

Em mãos da lista de termos da etapa anterior, passa-se para a criação do modelo das classes e suas hierarquias. Para isso foi utilizada a classificação *Top-down* mencionada por Ushold e Gruninger (1996), onde a criação das classes é realizada de forma que primeiramente os conceitos mais gerais da arquitetura são definidos para em seguida, serem detalhados e assim formar a hierarquia de classes.

Com base nessas premissas, tem-se o modelo a seguir das classes do domínio de combustíveis criado na ferramenta de modelagem de ontologias, o framework Protégé (PROTEGE, 2009).



Figura 17. Classes e hierarquia de classes.

No modelo da Figura 17 é possível perceber a utilização da hierarquia com base nos conceitos e predicados utilizados pelo middleware JADE (JADE, 2009). Cada termo da lista pode se encaixar na definição de predicado ou conceito, onde um predicado representa uma ação no qual se deseja realizar sobre os conceitos.

Ainda na Figura 17, temos quatro predicados: o primeiro, *Information*, é utilizado pelo sistema para encaminhar as amostras para análise; o segundo, o *Report*, é um termo utilizado para encaminhar os relatórios contendo o conceito de amostras; o terceiro, *Export*, é responsável por encaminhar as amostras para serem convertidas e em seguida exportadas para o MQC e o último, o *Recomentation*, é responsável por enviar as recomendações do sistema para o relatório, contendo informações como causa e solução sobre a amostra realizada.

Com isso, têm-se as classes e suas hierarquias definidas, restando definir a estrutura interna.

3.6. Definição das propriedades (slots) das classes

O objetivo desta fase é a conclusão das estruturas internas das classes definidas na fase anterior. Para isso, é necessário inserir as propriedades que deverão ser utilizadas pelas mesmas, conhecidas como *Slots*. Ressaltando que a participação do especialista do domínio é fundamental para a realização desta tarefa, pois é ele que irá detalhar a partir da lista de termos do domínio, quais dos termos que sobraram são candidatos a propriedades (*slots*) das classes principais.

No caso da análise de combustível especificamente, temos as classes Benzeno, Etanol, Saturadas, Olefinas, Aromáticos, RON e MON. Para cada um destes termos é necessário a propriedade (*slot*) valor. E assim sucessivamente para todos os termos das propriedades, independentemente do tipo de combustível.

Tabela 6. Termos do domínio e suas propriedades

Sample	Alcohol	Gasoline	Diesel
Name Date Technics	Name Date Techincs	Name Date Techincs	Name Date Techincs
Biodiesel	Property	Report	Information
Name Date Techincs	Name Value PropertyInstance	Sample Name	Sample Name
Export	Cause	Solution	Recomendation
Sample	Description Value	Description Value Cause Recomendation	Description Value
Techincs	Etanol	Bezeno	Olefinas
Description Value Property	Description Value	Description Value	Description Value
Saturades	Mon	Ron	Temperature
Description Value	Description Value	Description Value	Description Value
Aromáticos	Aspect	Composition	Density
Description Value	Description Value	Description Value	Description Value
Conduction	Destilation	FulgorPoint	pH
Description Value	Description Value	Description Value	Description Value

SulfurContent	CarbonContent	GasolineContent	AlcoholContent
Description Value	Description Value	Description Value	Description Value
Color			
Description Value			

Na tabela anterior podemos perceber todos os termos e suas respectivas propriedades utilizadas pelo *SIMCQC* no domínio das análises químicas.

3.7. Definição dos tipos de propriedades (*slots*)

Após termos todas as classes e suas devidas propriedades (*slots*) definidas, nesta fase definiremos o seu tipo de dados (*string*, *number*, *boolean*, *enumerated*, *instance-type*).

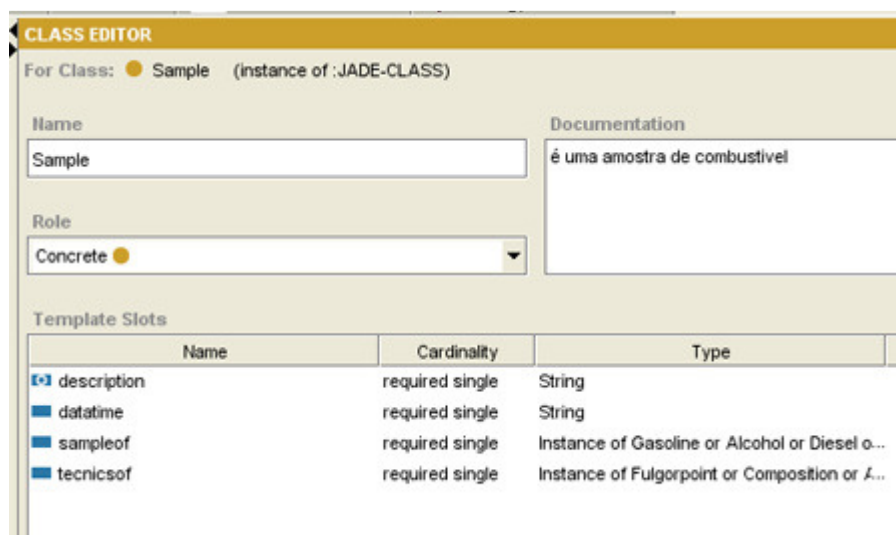


Figura 18. Slots da classe sample (amostras).

O exemplo da Figura 18 ilustra como foram definidos os *slots* das classes da ontologia. Nela podemos perceber o exemplo da classe *sample* (amostra), que possui os *slots* e seus respectivos tipos de dados definidos.

Primeiramente um campo descrição (*description*) contendo o nome da amostra de combustível em *String*, em seguida tempos um campo data para recuperar a *string* contendo a data da análise da amostra, logo após tem-se um campo *sampleof* que é uma instancia de um tipo de amostra de combustível, podendo ser (álcool, diesel, gasolina ou biodiesel) e somente estes. E por fim, um campo contendo uma instancia do tipo da técnica que foi utilizada para capturar esta amostra chamada de *technicsof*, salientando que somente técnicas de análise (Composição, Destilação, Aspecto, Ponto de Fulgor) são aceitas nesse campo.

O framework Protégé (PROTEGE, 2009) fornece ainda vantagens como a PAL (*Protege Axiom Language*), uma linguagem para criação de *facet*s.

As *facet*s representam as restrições que um *slot* possui. Por exemplo, um *slot* cor de uma classe carro, somente poderá ter as opções de cores (azul, branco ou verde). Através da PAL, o Protégé pode restringir o valor da propriedade a ser incluída em um *slot*, garantindo assim que apenas os valores desejados são utilizados. Com isso, a fase de definição dos tipos de *slot* está completa.

3.8. Criação do modelo físico das classes

A última da metodologia 101 foi modificada, pois, no processo original, devem-se criar as instâncias das classes para realização dos testes diretamente no framework Protégé. Porém, optou-se pela criação de um modelo físico para este ser incluído e testado no ambiente de execução do SMA, o middleware JADE.

Entretanto, para gerar o modelo físico da ontologia foi utilizado o *plugin OntologyBeanGenerator* (2008) para o framework Protégé (PROTEGE, 2009). O *plugin* tem a responsabilidade de gerar os arquivos para exportação da ontologia previamente criada na ferramenta. Ressalta-se que as classes da ontologia definidas a partir da lista de termos servem como entrada de dados para o referido plugin.

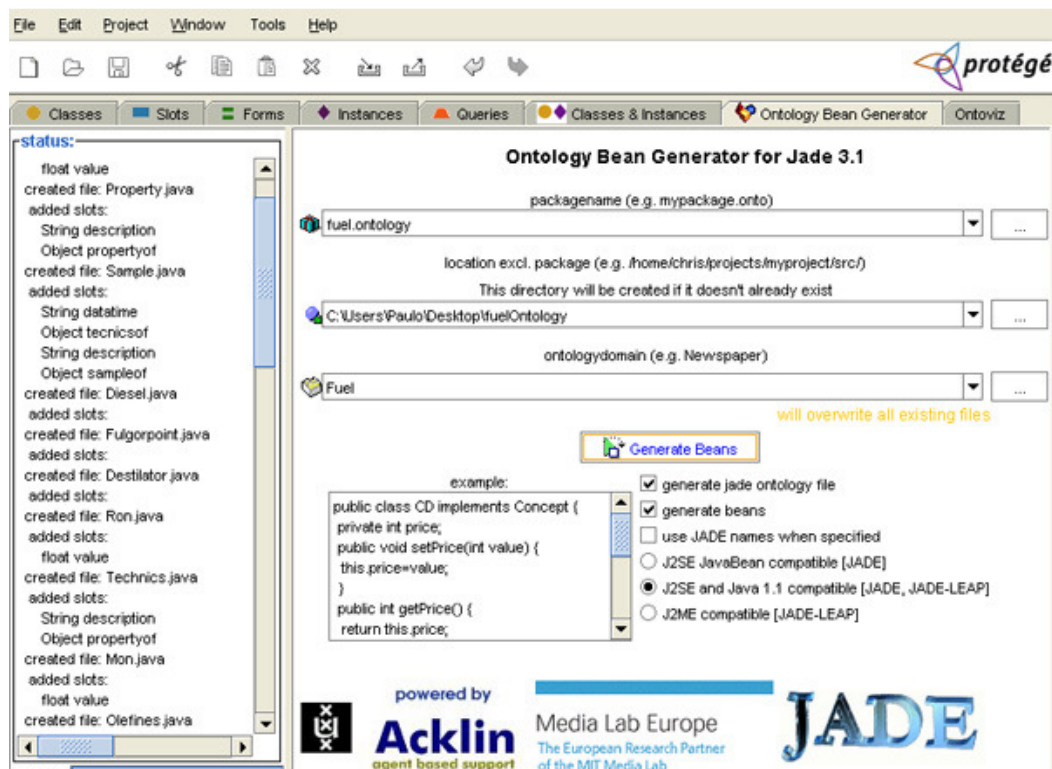


Figura 19. Plugin OntologyBeanGenerator para JADE.

A Figura 19 mostra a tela do plugin. Nela podemos perceber ao lado esquerdo, a criação das classes a partir da hierarquia de classes mostradas nos tópicos anteriores. Este plugin fornece a entrada de parâmetros para configurar a criação da ontologia, que são: nome do pacote que será criado para a ontologia, local para exportar as classes que serão geradas, e, por fim, o nome da ontologia do domínio, no caso acima está definido como “*Fuel*”.

Após sua execução, o referido plugin criou todo o vocabulário, contendo os conceitos e predicados no formato de classes *Java* utilizando-se do padrão *JavaBeans*.

As classes geradas pelo *plugin* possuem os mesmos nomes dos termos descritos pela ontologia. Em seguida, essas classes foram inseridas ao projeto do *SIMCQC* (SILVA, 2008), instanciadas e testadas pelo sistema.

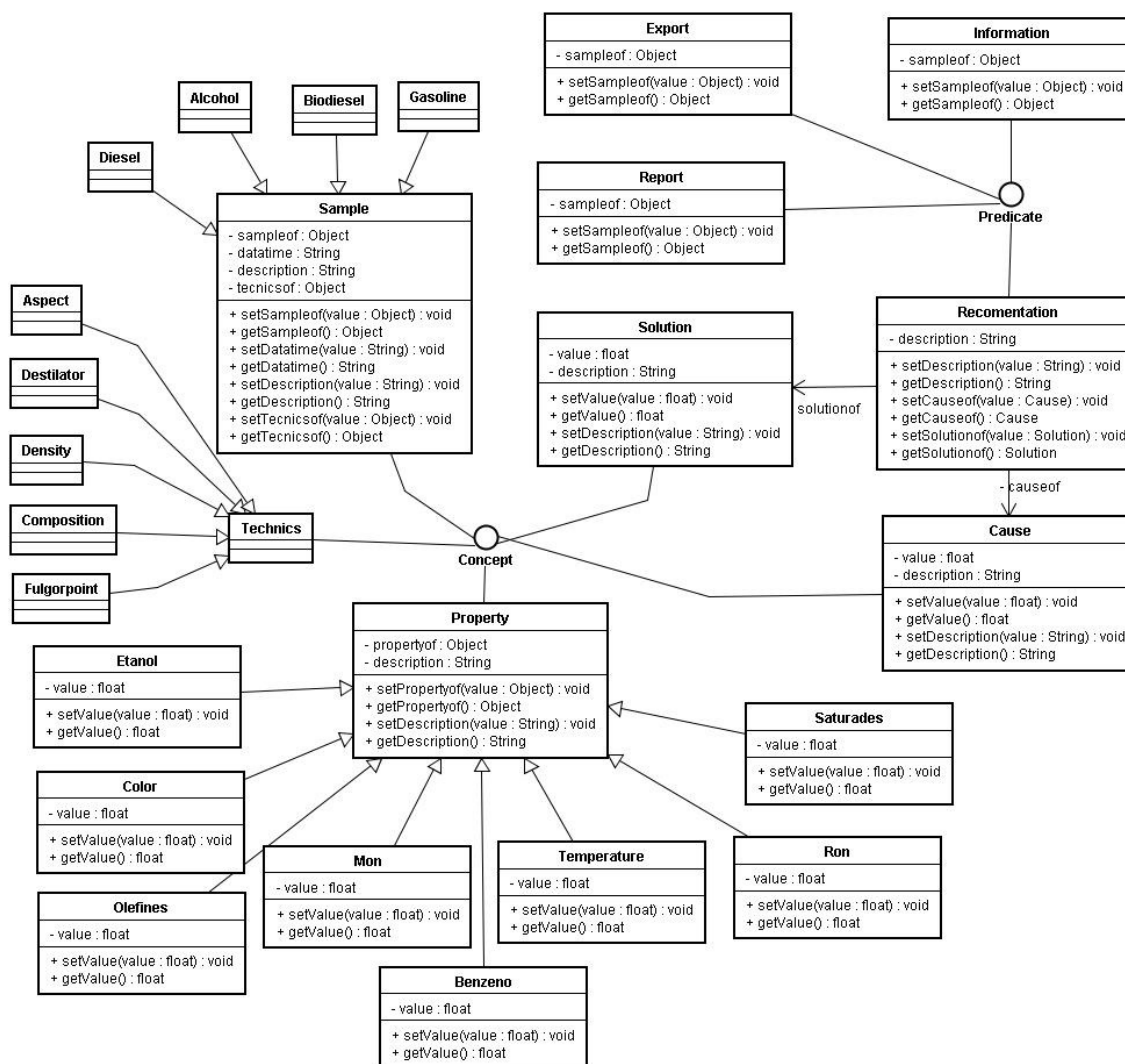


Figura 20. Modelo físico da ontologia de combustíveis criado pelo plugin *OntologyBeanGenerator* do *Protege*.

A Figura 20 representa o modelo físico da ontologia de combustíveis, gerado a partir do modelo de classes criado pelo *plugin OntologyBeanGenerator*(2008). O modelo possui a estrutura de classes UML contendo os termos necessários para a representação do conhecimento do domínio e para o sistema efetuar o tráfego das informações dentro da sociedade multiagente.

Com isso, finaliza-se o processo de criação da ontologia de combustíveis.

4. ESTUDO DE CASO: SISTEMA INTELIGENTE PARA MONITORAMENTO E CONTROLE DA QUALIDADE DE COMBUSTÍVEL – *SIMCQC*

Neste capítulo, é apresentado o Sistema Inteligente para Monitoramento e Controle da Qualidade de Combustível - *SIMCQC*, um software para auxílio na tomada de decisão do LAPQAP, logo após, são apresentadas as melhorias incluídas no sistema.

São descritos ainda os novos agentes da sociedade, em seguida, é destacada a inclusão da ontologia de combustíveis no mecanismo de comunicação da sociedade. Por fim, é mostrada a nova arquitetura do sistema e os resultados obtidos após a inclusão das melhorias no *SIMCQC*.

4.1. O projeto *SIMCQC*

Especificado por Silva (2008), o *SIMCQC* é um sistema multiagente que foi previamente concebido para atender às necessidades do LAPQAP em relação aos ensaios químicos, auxiliando o laboratório nas tomadas de decisões sobre os resultados das amostras analisadas pela técnica de destilação do programa de monitoramento da qualidade de combustível (MQC).

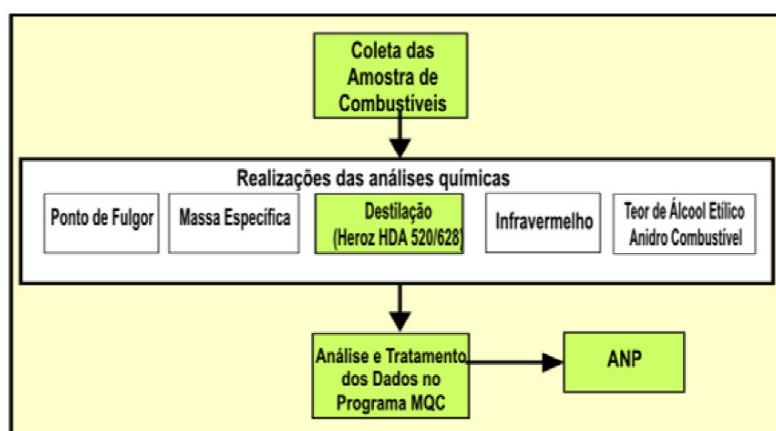


Figura 21. Técnicas automatizadas pelo *SIMCQC*.

Fonte: SILVA 2008.

Nesse sentido, o *SIMCQC* foi desenvolvido com objetivo de atuar na etapa de análises químicas, utilizando-se para isso da seguinte infra-estrutura: (SILVA, 2008)

Software Destilador – responsável por recuperar as amostras diretamente do equipamento de análise na técnica de destilação;

Sociedade de Agente Inteligente – tendo como objetivo efetuar a análise propriamente dita, através dos agentes do sistema (Gestor, Monitor, Conversor e DW);

MQC – O software de cadastro de resultados fornecido pela ANP;

Data Warehouse – criado para servir de repositório de informações analisadas pelo sistema, atuando apenas como um banco de dados.

Segundo Silva (2008), o modelo proposto automatiza diversas atividades (obtenção, organização e análise de dados) que são realizadas pelos pesquisadores.

O *SIMCQC* foi desenvolvido na linguagem *Java* (J2SE) e utiliza-se de tecnologias de agentes inteligentes e motor de inferência em JESS, sendo executado no middleware JADE para alcançar os seus objetivos.

4.2. Funcionamento do *SIMCQC*

A proposta do *SIMCQC* é baseada na captura e inferência de dados oriundos do processo de monitoramento do combustível. O fluxo tem início com a captura dos dados analisados pelo equipamento HERZOG (técnica de destilação) pelo agente reativo Monitor, em seguida, ele envia uma mensagem para o agente Gestor, informando que possui dados de amostras analisados, logo após, o agente Gestor solicita esses dados.

Tabela 7. Agentes e responsabilidades do SIMCQC

	ETAPAS	FUNCIONALIDADE DO AGENTE	TIPO DE AGENTE	INTERAÇÃO	OBJETIVO DO AGENTE
1	Consultar a base de dados.	Extrair informações de novo resultado e enviar para o agente Gestor.	Reativo	Base de Dados Textual	Consultar permanentemente a base de dados textual para extrair informações de novas amostras.
2	Analisar e tratar as informações.	Realizar a análise e tratamento das informações de novos resultados.	Cognitivo	Agentes Conversor e DW	Analisar e tratar as informações do resultado de amostra.
3	Converter o resultado da amostra tratadas em formato XML.	Aguardar mensagens do agente Gestor para disponibilizar as informações do resultado da amostra no formato XML.	Reativo	MQC	Esperar mensagens e convertê-las no formato XML e disponibilizar as informações para o Software MQC.
4	Manter um repositório do DW.	Montar um repositório do Data Warehouse com as informações do resultado da amostra advindas do agente Gestor, a fim de dar suporte para a equipe.	Reativo	Repositório Data Warehouse	Esperar mensagens contendo informações sobre as amostras do agente Gestor para serem armazenadas em um repositório do Data Warehouse.

Fonte: SILVA, 2008.

Após receber os dados, o agente Gestor realiza a inferência dos dados (amostras) de acordo com as regras de sua base de conhecimento em JESS. Na base de conhecimento, estão descritos os valores de referência através das regras de produção para efetuar a análise do combustível.

Por fim, o agente Gestor encaminha as amostras analisadas para dois agentes simultaneamente. O primeiro, o agente DW, é responsável por receber e armazenar os dados analisados para futuro tratamento utilizando-se de técnicas de mineração de dados. O segundo, o agente Conversor, recebe os dados e converte os mesmos para a linguagem XML, para ser exportado e utilizado no MQC.

Percebe-se que o funcionamento do sistema possui um fluxo de dados que se inicia com o agente Monitor, passa pelo agente Gestor e termina com os agentes, Conversor e DW.

4.3. Estendendo o *SIMCQC*

Para fornecer maior autonomia ao LAPQAP foram previstas inclusões de funcionalidades ao *SIMCQC* proposto por Silva (2008). Ao analisar os recursos básicos do sistema percebe-se que o mesmo atua apenas utilizando-se de uma técnica para efetuar a inferência dos dados.

Nesse sentido, o objetivo principal foi aumentar a quantidade de técnicas utilizadas para o sistema realizar a inferência dos dados. Para isso, foram realizados estudos no LAPQAP envolvendo o processo de análise de combustível, os equipamentos envolvidos durante a etapa de análise e entrevistas com o técnico de laboratório, responsável pela realização do processo de análises químicas do combustível.

Após a realização do estudo chegou-se a conclusão de incluir um novo equipamento no sistema *SIMCQC*, e como consequência, a adição de uma nova técnica para realizar o procedimento automatizado de inferência dos dados.

A técnica escolhida para inclusão no sistema foi a de *composição (infravermelho)*, realizada pelo equipamento Irox 2000 Gasoline, e sua escolha foi motivada devido aos seguintes fatores.

1. Trata-se da técnica responsável por analisar 7 (sete) propriedades químicas do combustível, sendo a técnica que fornece a maior quantidade de propriedades analisadas;
2. O equipamento *Irox 2000 Gasoline* da empresa *Grabner Solutions* fornece uma Interface de comunicação de rede, tornando possível a captura dos dados remotamente.

Podemos perceber que para efetuar a inclusão do equipamento foi necessária a adição de um novo agente na sociedade, denominado de *agente Irox*.

Continuando com o objetivo de adicionar novas funcionalidades ao *SIMCQC*, após a fase de entrevistas, foi relatado pelo técnico de laboratório sobre a dificuldade de interagir com o *SIMCQC* (SILVA, 2008), pois este, desprovido de conhecimentos da área de informática, tinha problemas para utilizar o sistema.

O objetivo dessa interface é tornar a interação homem – máquina menos problemática. Para isso, foi definido um agente Interface, responsável por controlar as ações fornecidas pelo sistema e de fácil manuseio.

Ao final, percebe-se a necessidade de aperfeiçoamento do mecanismo de comunicação da sociedade para que o sistema incorpore as melhorias explicitadas de maneira efetiva. Tal necessidade decorre do fato de que a atual arquitetura de comunicação exagera na quantidade de mensagens enviadas, sobrecarregando a comunicação entre os agentes do sistema.

Assim, para solucionar o problema da sobrecarga de mensagens dentro da sociedade propõe-se a inclusão da ontologia de combustíveis. A ontologia fornece uma linguagem do domínio de combustíveis para os agentes, diminuindo a quantidade de mensagens enviadas entre os agentes e aumentando a produtividade no processo de transmissão ao enviar pacotes contendo amostras completas e não somente texto.

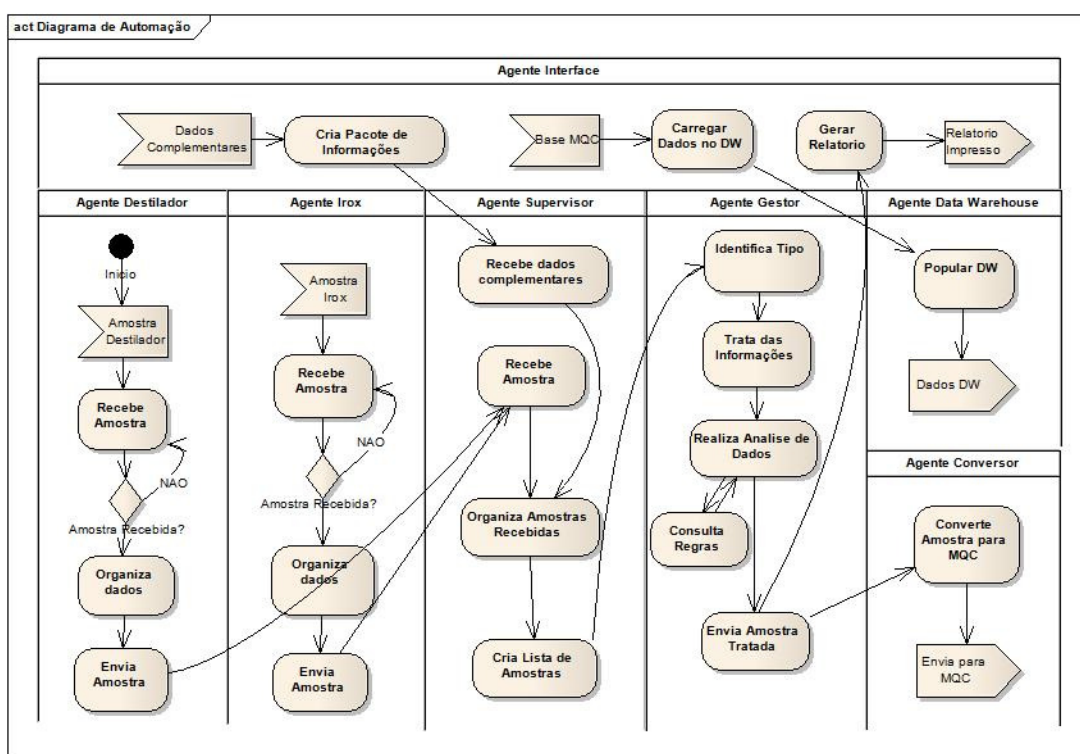


Figura 22. Diagrama de atividades do processo de captura e análise das amostras.

Na Figura 22, podemos perceber o fluxo da amostra no processo de análise de combustíveis, o qual deverá ser alcançado após a adição das melhorias no *SIMCQC*.

A fim de modelar as novas funcionalidades incluídas no sistema utilizou-se da metodologia PASSI (COSSETINO, 2002), a qual se aplica ao desenvolvimento de sistemas multiagentes.

A escolha desta metodologia foi motivada devido a sua capacidade de descrever as etapas de modelagem do projeto de forma detalhada e precisa através de seus diagramas. Assim, foi possível demonstrar com detalhes a concepção das melhorias a serem adicionadas ao sistema.

A seguir, são detalhadas as fases de modelagem, os novos agentes e funcionalidades a serem incluídas no *SIMCQC*.

4.3.1. Descrição do Domínio

A primeira fase da PASSI (COSSETINO, 2002) especifica procedimentos para a criação de modelos representativos do domínio da aplicação através de casos de uso UML. Os modelos servem para representar as ações que são realizadas pelos agentes da sociedade.

Na Figura 23, podemos ver todos os participantes da sociedade e suas respectivas ações. O referido modelo fornece uma visualização geral com a definição das ações de cada agente dentro da sociedade.

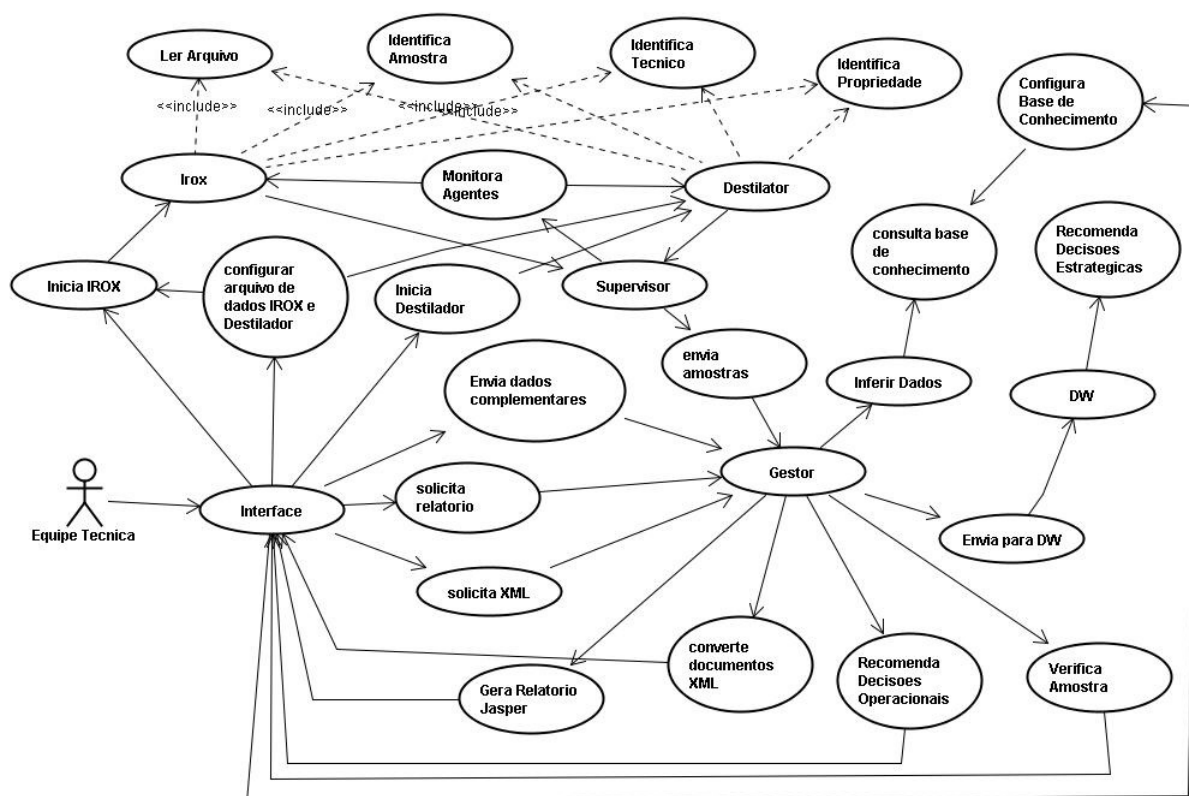


Figura 23. Descrição do domínio da análise e monitoramento de combustíveis.

Pode-se perceber que o ator externo atua de forma direta com o agente Interface e que todas as ações realizadas pela entidade externa *equipe técnica* são executadas pelo referido agente, pois o mesmo atua como uma ponte para executar as ações dentro do sistema.

No mesmo modelo, visualiza-se a inclusão das novas melhorias em relação ao modelo anterior proposto por Silva (2008). São elas:

- Os agentes – Interface, Supervisor e Irox;
- As ações – Configurar arquivos de dados dos equipamentos (Irox e Destilador), imprimir relatório jasper, exportar XML para MQC e configurar arquivo da base de conhecimento.

O próximo passo para inclusão das novas funcionalidades na solução está na identificação de papéis dos agentes da sociedade, que são descritas a seguir.

4.3.2. Identificação dos papéis

A Figura 24 ilustra os agentes da sociedade e suas respectivas funcionalidades, conforme é descrito na primeira etapa da metodologia PASSI (COSSETINO, 2002).

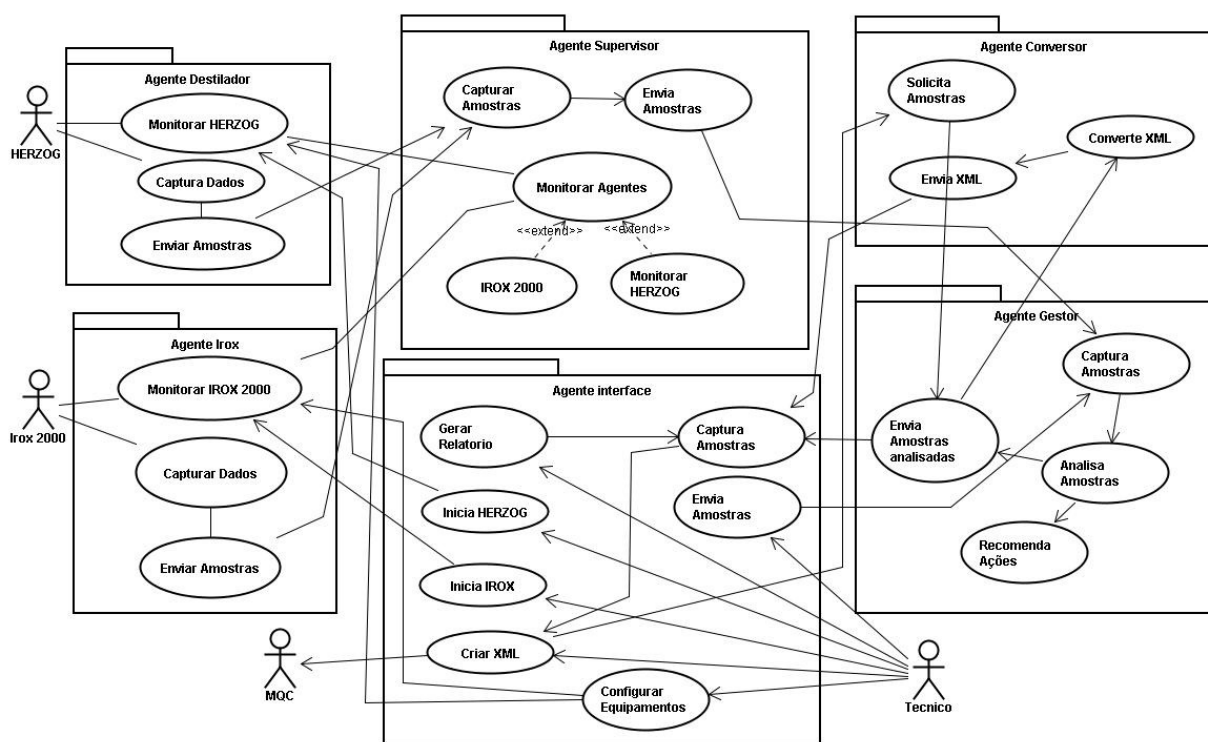


Figura 24. Identificação de papéis SIMCQC.

Na figura, percebe-se a sociedade de agentes após a inclusão dos novos agentes. Optou-se por estender a estrutura anterior proposta por Silva (2008), com isso, houve uma única modificação. Trata-se do agente Monitor (SILVA, 2008) que foi renomeado para Destilador, pois este realiza o papel de captura de dados do equipamento HERZOG (técnica de destilação).

Em seguida, foi inserido outro agente em seu lugar, o Supervisor, com a responsabilidade de monitorar e agrupar os pacotes recebidos dos agentes Irox e Destilador, e por fim encaminhar os pacotes recebidos para o agente Gestor.

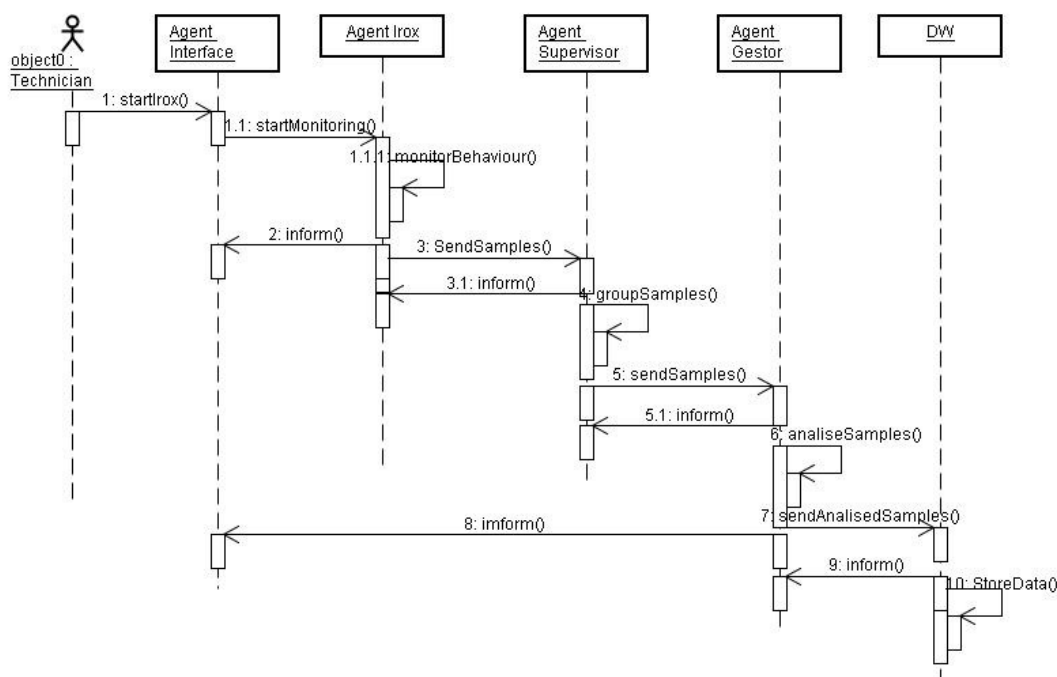


Figura 25. Comportamento de monitoramento, captura e análise de amostras da sociedade do *SIMCQC*.

Na Figura 25, podemos ver o principal comportamento a ser alcançado pela sociedade de agentes: o comportamento de monitoramento, captura e análise de amostras.

Podemos perceber que o processo tem início com o técnico acionando o agente Irox através do agente Interface. Em seguida, o agente Irox realiza a captura e encaminha para o agente Supervisor. Este por sua vez, agrupa as amostras e encaminha para o agente Gestor, que realiza a inferência dos dados e encaminha para o DW armazenar em sua base.

4.3.3. Modelo da sociedade de agentes

A segunda fase da PASSI (COSSETINO, 2002), descreve os processos e interações entre os agentes da sociedade. Não são necessários todos os modelos para alcançar os objetivos desta fase. Nesse sentido, é apresentado a seguir o modelo Descrição da Ontologia do Domínio (DOD).

O modelo DOD a seguir é representado através do diagrama de classes UML, no qual podemos visualizar todos os termos utilizados pelo domínio da aplicação. Porém, Noy e Mcguinness (2001) ressaltam que uma ontologia está em constante atualização, pois a qualquer instante um novo termo poderá surgir e modificar a mesma.

O atual modelo da ontologia possui 30 classes, sendo que 4 predicados responsáveis pelas ações do sistema e 26 conceitos e termos do domínio da aplicação

O diagrama de classes da Figura 26 representa a ontologia do domínio de combustíveis denominada como ontologia de combustíveis ou *Fuel Ontology*, cuja construção foi detalhada no capítulo 4 deste trabalho.

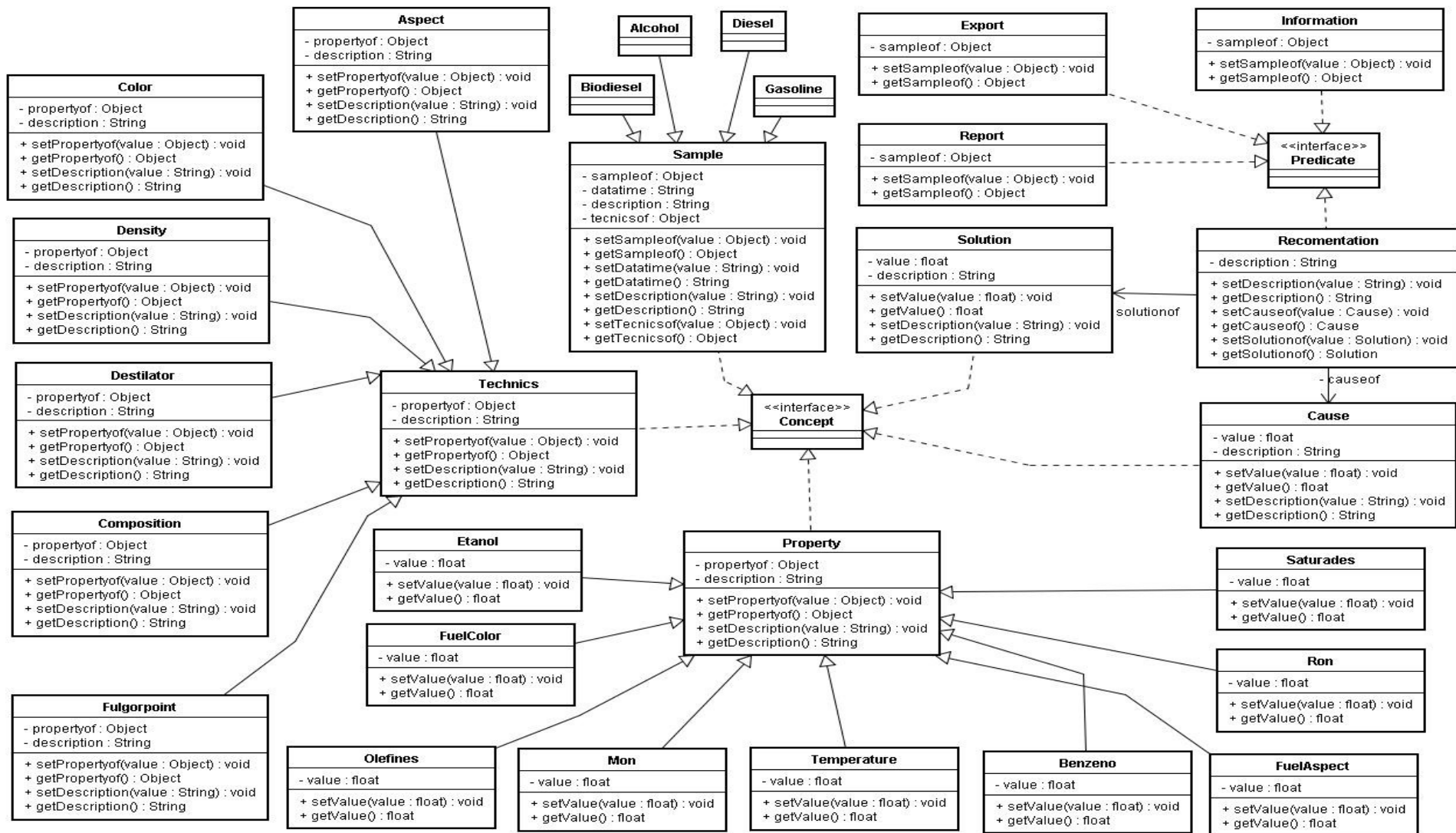


Figura 26. Diagrama da ontologia de domínio de combustíveis.

A seguir, são detalhados os novos agentes da sociedade, e com isso as suas respectivas responsabilidades também são detalhadas separadamente utilizando-se de diagramas de seqüência, de acordo com a metodologia PASSI (COSSETINO, 2002).

4.3.4. Agente Irox

O primeiro agente a ser detalhado, o agente Irox, é um agente reativo e possui a função de capturar a base de dados do equipamento *Irox 2000 Gasoline*, identificar as amostras capturadas pelo nome, criar o pacote contendo as amostras e suas propriedades e, em seguida, encaminhar o pacote de amostras para o agente Supervisor.

A captura da base de dados do equipamento *Irox 2000 Gasoline* é realizada através do software proprietário fornecido pela empresa *Grabner Solutions*, o *Irox – MiniWin*. Com ele os resultados analisados podem ser extraídos e obtidos através de uma base de dados textual.

O equipamento *Irox 2000 Gasoline* utilizado para capturar as amostras de gasolina possui as mesmas características de comunicação do *Irox Diesel* responsável por capturar os dados do combustível diesel. Ambos pertencem à mesma empresa, utilizando-se inclusive do mesmo software para realizar a captura dos dados.

Os testes e simulações não foram realizados com o equipamento Irox Diesel, pois o mesmo encontrava-se com sua base de dados vazia. Porém, a estrutura atualmente utilizada para realizar os procedimentos citados pode servir para capturar os dados deste equipamento também.

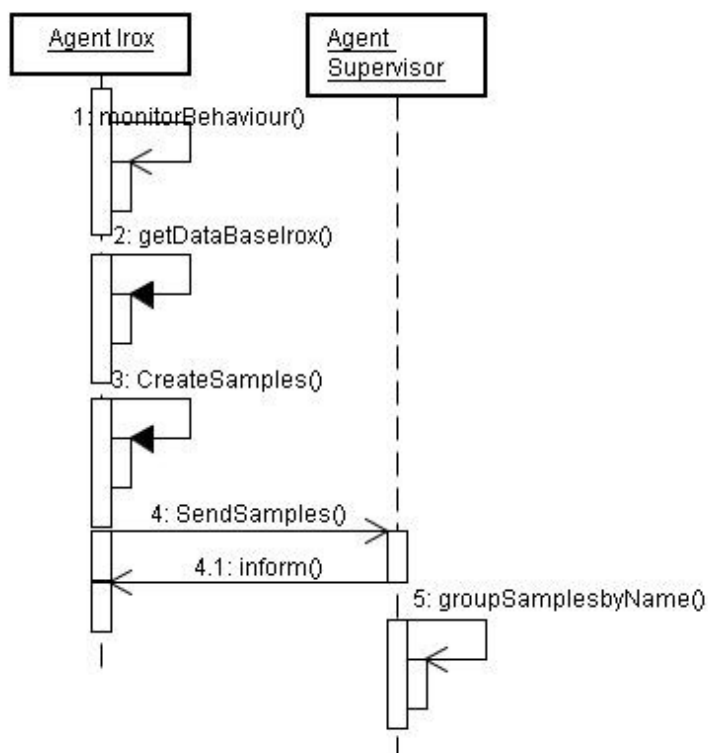


Figura 27. Responsabilidade do agente Irox.

O agente Irox possui comunicação direta com o agente Supervisor, enviando as amostras capturadas e sendo monitorado pelo mesmo. A Figura 27 se encaixa na primeira fase da metodologia PASSI (COSSETINO, 2002), definição dos requisitos, mais precisamente na identificação das responsabilidades dos agentes. Nela podemos perceber o modelo da comunicação entre os agentes da sociedade com o agente Irox na realização da tarefa de monitoramento e captura de amostras.

Agente Irox
<p>O agente Irox tem como objetivo capturar as amostras na base de dados textual extraídas diretamente do equipamento <i>Irox 2000 Gasoline</i>, de acordo com as instruções abaixo:</p> <ul style="list-style-type: none"> - Verifica o arquivo pela data do arquivo - Captura os dados e identifica as amostras pelo nome - Identifica as datas das amostras - Identifica o responsável técnico pelas análises. <p>Percepção: Consulta periodicamente a base de dados textual. Objetivo: Verificar e capturar as amostras analisadas pelo equipamento Irox. Ação: monitora, captura e identifica amostras Tipo de Agente: Reativo</p>

Quadro 2. Funcionalidade do agente Irox.

Ainda na Figura 27 percebe-se que o agente Irox realiza o monitoramento da base de dados Irox, em seguida, captura as amostras, cria as amostras a serem enviadas e por fim, envia para o agente Supervisor.

Com a adição do agente Irox o sistema pôde contemplar mais uma fase das análises químicas, a composição, utilizando-se da técnica de infravermelho. A referida técnica proporcionou ao sistema realizar a inferência dos dados sobre 40% do total das propriedades do combustível. Ao considerar 30% fornecidos pelos dados recuperados pelo agente Destilador (técnica de destilação), restam apenas 30% dos dados complementares para alcançar a totalidade de análise das propriedades sobre o combustível do tipo gasolina.

4.3.5. Agente Supervisor

O segundo agente a ser detalhado, o agente Supervisor, é um agente reativo que possui as seguintes funcionalidades: receber as amostras enviadas pelos agentes Irox e Destilador, monitorar os referidos agentes, agrupar as amostras pelo tipo de combustível e encaminha-las para o agente Gestor efetuar a inferência sobre elas.

Cada agente, o Irox ou Destilador, é responsável por criar pacotes contendo amostras capturadas por um equipamento que utiliza uma técnica específica para análise do combustível. Conforme dito anteriormente, um determinado tipo de combustível possui características que são analisadas por diferentes tipos de técnicas.

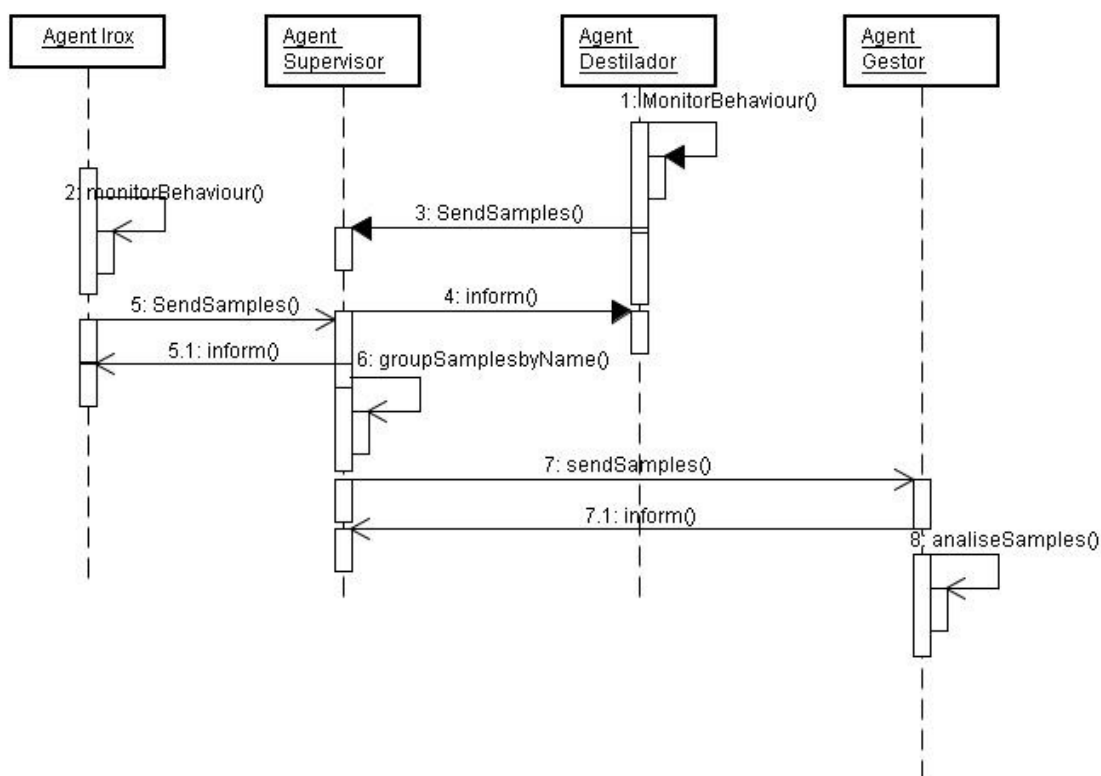


Figura 28. Responsabilidades do agente Supervisor.

Na Figura 28 pode-se perceber a responsabilidade do agente Supervisor. O agente recebe as amostras dos agentes Irox e Destilador, agrupa as amostras pelo nome da amostra, e as encaminha para o agente Gestor dar continuidade ao processo de análise.

Conforme foi possível ver na figura anterior, o agente Supervisor possui comunicação direta com 3 agentes da sociedade, primeiramente com os agentes Irox e Destilador, recuperando as amostras e efetuando o monitoramento dos mesmos, e logo depois com o agente Gestor, encaminhando as amostras recebidas dos agentes anteriores.

Agente Supervisor
<p>O agente Supervisor tem como objetivo monitorar os agentes responsáveis pela captura de dados dos equipamentos (agente Irox e agente Destilador), em seguida, recebe as amostras e encaminha para o agente Gestor, de acordo com as instruções abaixo:</p> <ul style="list-style-type: none"> - Monitora os agentes Irox e Destilador. - Recebe os pacotes de amostras e agrupa por nome. - Encaminha para o agente gestor. <p>Percepção: Consulta periodicamente os agentes para capturar as amostras e verificar se estão disponíveis.</p> <p>Objetivo: Monitorar agentes, agrupar amostras e encaminhar para gestor.</p> <p>Ação: Agrupar e encaminhar amostras para o gestor</p> <p>Tipo de Agente: Reativo</p>

Quadro 3. Funcionalidade do agente Supervisor

A partir da inclusão do agente Supervisor criou-se uma arquitetura na qual o agente Gestor não precisa estar ciente da comunicação dos agentes Irox e Destilador com o agente Supervisor, apenas da sua comunicação com o Supervisor para recebimento dos pacotes de amostras.

Nesse sentido, a inclusão de um novo agente na sociedade para encaminhar pacotes de uma nova técnica, poderia ser realizada sem maiores modificações, pois o agente Gestor não sofreria grandes modificações, apenas no comportamento cognitivo através da inclusão de regras para tratamento do novo pacote de amostras.

4.3.6. Agente Interface

O terceiro, o agente Interface, foi desenvolvido com o objetivo de solucionar o problema relacionado à utilização do sistema, pois antes da sua adição ao sistema o técnico de laboratório encontrava dificuldades para manusear o software.

O agente Interface é um agente reativo que fornece uma interface gráfica para o técnico efetuar o controle do sistema. Entre seus recursos, podemos destacar as funcionalidades de gerenciamento da aplicação, fornecendo ao técnico de laboratório opções como: configurar arquivo de dados dos agentes Irox e Destilador, inclusão manual de dados complementares das amostras de combustível, emissão de relatório, ativação de agentes de Monitoramento, impressão de relatórios e exportação de dados em XML para o MQC.

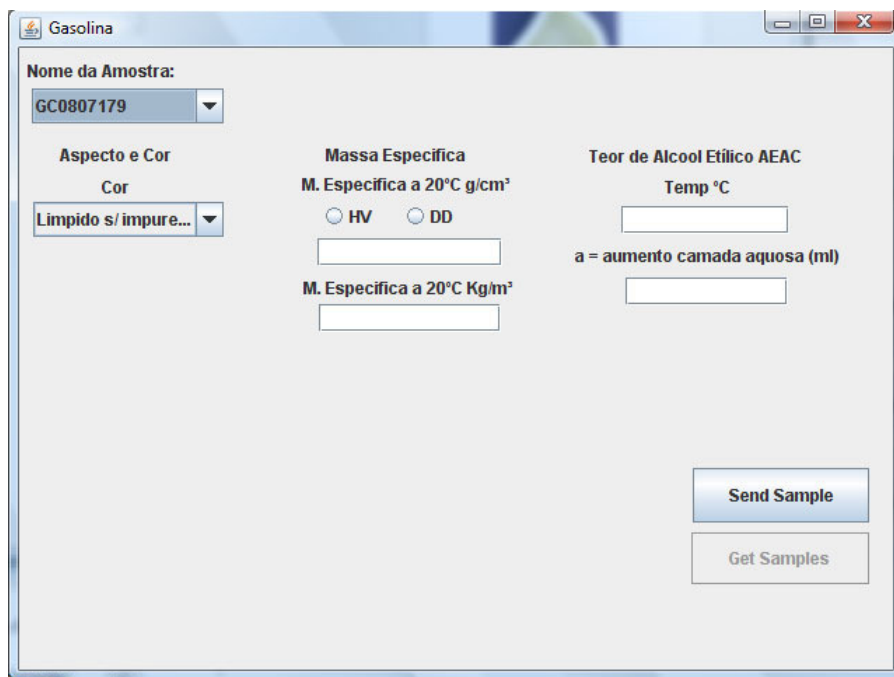
Agente Interface
<p>O agente Interface tem como objetivo fornecer uma Interface gráfica para o usuário acessar as funcionalidades do sistema, de acordo com as insruções abaixo:</p> <ul style="list-style-type: none"> - Monitora os agentes Irox e Destilador - Recebe os pacotes de amostras e agrupa por nome - Encaminha para o agente gestor <p>Percepção: Aciona os agentes da sociedade em busca de funcionalidades como emissão de relatórios e exportação de dados.</p> <p>Objetivo: Servir de Interface de comunicação com o técnico.</p> <p>Ação: Enviar dados complementares de amostras, imprimir relatórios, exportar dados para XML (MQC), configurar agentes (Irox e Destilador), configurar dados de análise técnica, configurar arquivo de regras JESS.</p> <p>Tipo de Agente: GuiAgent (Gráfico)</p>

Quadro 4. Funcionalidades do agente Interface

Sua inclusão forneceu mecanismos para o técnico de laboratório utilizar a ferramenta com facilidade. Através deste agente, o técnico pode inserir os dados complementares, necessários para completar as informações sobre o combustível gasolina (os agentes Irox e Destilador fornecem o restante dos dados).

Conforme informado anteriormente, os dados complementares equivalem aos outros 30% do restante das informações que não podem ser automatizadas pelo sistema, pois tais informações são realizadas por técnicas ou procedimentos manuais.

Os dados complementares compreendem as técnicas de *Aspecto e Cor*, *Massa Específica* e *Teor de Álcool Etílico*, por isso, o agente Interface fornece um formulário para inclusão manual destes dados (Figura 29).



The screenshot shows a software window titled "Gasolina" with a light gray background. At the top left, there is a label "Nome da Amostra:" followed by a dropdown menu containing the text "GC0807179". Below this, there is a section titled "Aspecto e Cor" with a sub-label "Cor" and a dropdown menu showing "Limpido s/ impure...". To the right of this section is the "Massa Específica" section, which includes two radio buttons labeled "HV" and "DD", and two empty text input fields. The first field is labeled "M. Específica a 20°C g/cm³" and the second is labeled "M. Específica a 20°C Kg/m³". To the right of the mass specific section is the "Teor de Alcool Etílico AEAC" section, which includes a label "Temp °C" and an empty text input field. Below this, there is a label "a = aumento camada aquosa (ml)" and another empty text input field. At the bottom right of the window, there are two buttons: "Send Sample" and "Get Samples".

Figura 29. Tela de inserção de dados complementares.

Nesse sentido, após a informatização da inclusão dos dados complementares restantes através do agente Interface, o sistema pôde alcançar a totalidade (100%) da análise das propriedades do combustível gasolina, finalizando assim o processo de inclusão de dados para este tipo de combustível.



Figura 30. Tela principal do agente Interface.

Na tela principal do agente Interface (Figura 30), podemos perceber as funcionalidades que estão disponíveis para o técnico. Duas merecem destaque, *iniciar o Irox* e *iniciar o Destilador*. Com elas, o técnico dá início ao processo de monitoramento de combustíveis nas técnicas automatizadas do sistema, que são a de composição e destilação. No cenário atual é necessária a interação humana para dar início ao processo de captura dos dados pelos agentes supracitados.

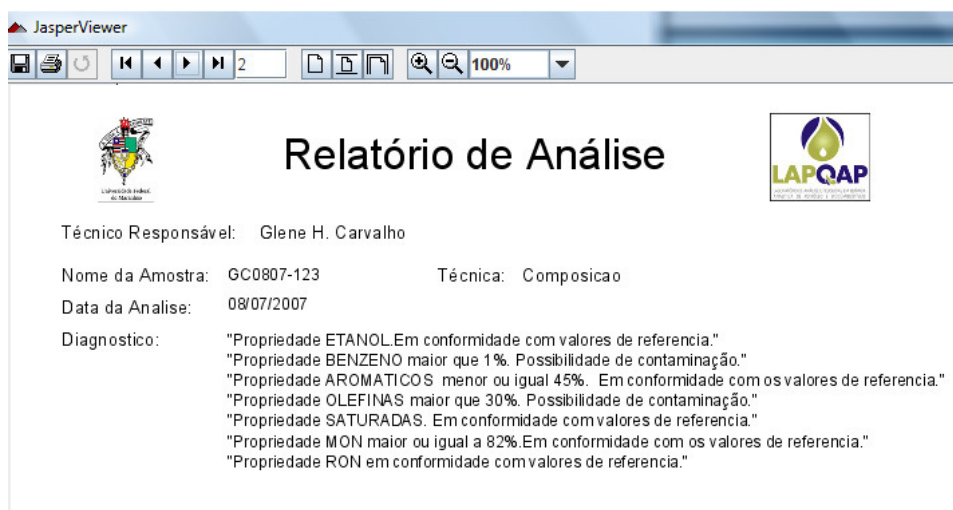


Figura 31. Tela de relatório de análises químicas do agente INTERFACE.

Na Figura 31, podemos ver o relatório extraído após análise dos dados realizado pelo motor de inferência do comportamento cognitivo do agente Gestor. O principal aprimoramento deste relatório está relacionado à quantidade de diagnósticos possíveis para uma única amostra, incluindo detalhes da mesma no nível de propriedade, o que na solução anterior não era possível.

O nível de detalhe no relatório foi alcançado devido ao uso da ontologia. Com a ontologia foi possível acoplar em uma única mensagem, uma amostra contendo todas as propriedades para serem analisadas, e, ainda retorno de todos os possíveis diagnósticos para cada propriedade analisada.

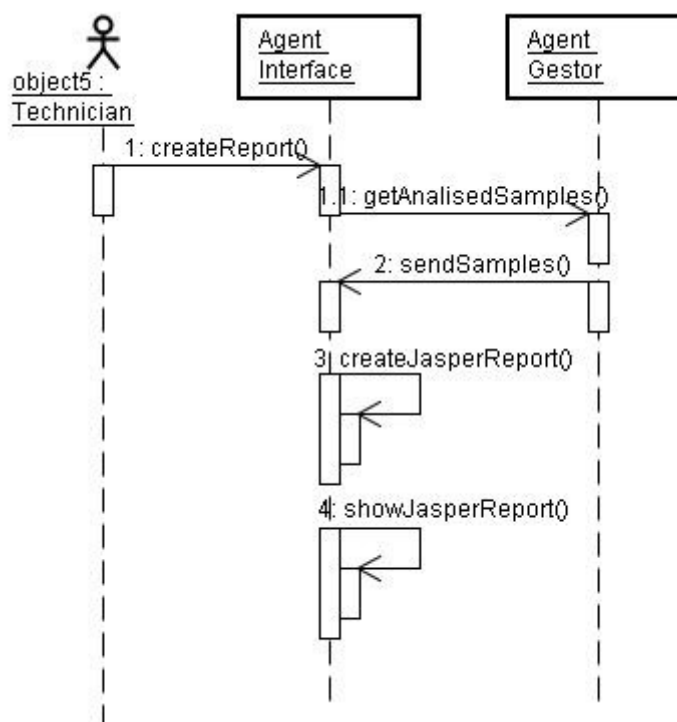


Figura 32. Comportamento para criação de relatório do agente Interface.

Na Figura 32, podemos ver o comportamento da agente Interface durante a funcionalidade de criação de relatório. Novamente a PASSI (COSSETINO, 2002) fornece mecanismos para visualizar a responsabilidade do agente.

Podemos perceber que, o técnico ao solicitar um relatório através do agente interface, envia uma mensagem para o agente Gestor, e este retorna um

pacote contendo as amostras analisadas. Em seguida, o agente Interface recebe as amostras e prepara o relatório utilizando-se para isso das informações contidas dentro das amostras. Com isso, foi alcançado mais um objetivo deste trabalho, o ganho de produtividade no processo de análises químicas.

Em seguida, é demonstrado com mais detalhes o processo de melhoria no mecanismo de comunicação da sociedade do *SIMCQC* (SILVA, 2008).

4.3.7. Linguagem de comunicação do SIMCQC

Em seu trabalho Silva (2008) mostrou que a utilização da linguagem de comunicação *FIPA-ACL* teve como benefícios: a padronização dos formatos de mensagens, uma arquitetura distribuída favorável para implantação do sistema e padronização dos protocolos de comunicação entre os agentes da solução.

Silva (2008) afirmou ainda que “a utilização da plataforma *JADE* para o desenvolvimento de agentes oferece um ambiente gráfico para acelerar a elaboração dos códigos de agentes”, conforme podemos ver na Figura 33.

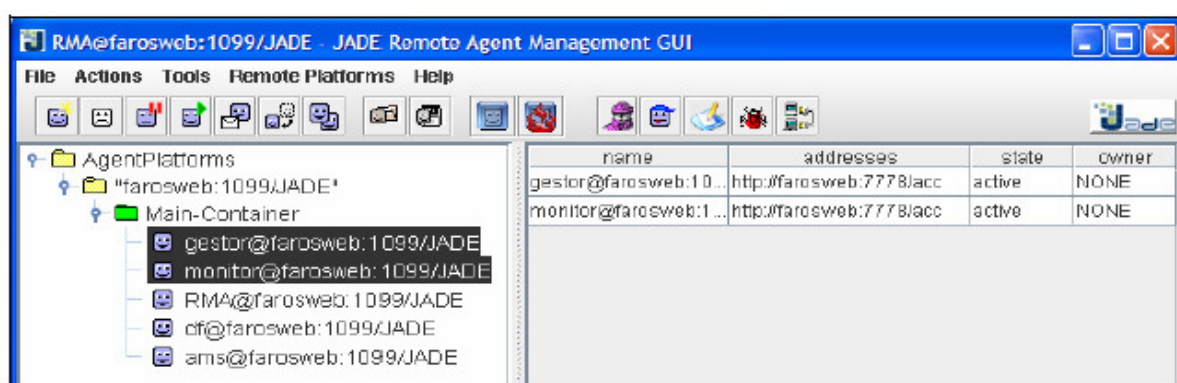


Figura 33. Interface gráfica JADE.

Fonte: SILVA, 2008.

A comunicação estabelecida por Silva (2008) para a sociedade multiagente utilizava-se da linguagem *FIPA-ACL* para comunicação entre agentes, contudo, fazendo o uso do padrão *ACL* (*Agent Communication Language*) apenas

para envio mensagens de texto simples de forma que somente uma única informação era trafegada por mensagem.

```

1. 15/04/08 13:52
2. (REQUEST :sender ( agent-identifier :name gestor@farosweb:1099/JADE :addresses
(sequence http://farosweb:7778/acc ))
3. :receiver (set ( agent-identifier :name monitor@farosweb:1099/JADE ))
4. :content "Amostra com indicio de contaminacao por material leve"
5. :protocol fipa-request)
6. 15/04/08 13:52
7. (REQUEST :sender ( agent-identifier :name monitor@farosweb:1099/JADE
:addresses (sequence http://farosweb:7778/acc ))
8. :receiver (set ( agent-identifier :name gestor@farosweb:1099/JADE ))
9. :content "90-145"
10. :protocol fipa-request)
11. 15/04/08 13:51
12. (REQUEST :sender ( agent-identifier :name gestor@farosweb:1099/JADE :addresses
(sequence http://farosweb:7778/acc ))
13. :receiver (set ( agent-identifier :name monitor@farosweb:1099/JADE ))
14. :content "Amostra com indicio de contaminacao por material pesado (ex. solvente
com ponto de ebulicao alto, oleo diesel ou teor de alcool alto)."
15. :protocol fipa-request)
16. 15/04/08 13:49
17. (REQUEST :sender ( agent-identifier :name monitor@farosweb:1099/JADE
:addresses (sequence http://farosweb:7778/acc ))
18. :receiver (set ( agent-identifier :name gestor@farosweb:1099/JADE ))
19. :content "10-65"
20. :protocol fipa-request)

```

Figura 34. Trecho da comunicação realizada pelo *SIMCQC*.

Fonte: SILVA, 2008.

Na Figura 34, do modelo de comunicação proposto por Silva (2008), podemos verificar os detalhes do trecho de código de uma comunicação retirado do *sniffer-agent* (JADE, 2009), realizada entre o agente Gestor e o agente Monitor.

No código acima, é encaminhada uma amostra do tipo gasolina analisada pela técnica de destilação. Na linha 9, o agente Monitor (SILVA, 2008) envia uma mensagem para o agente Gestor com o conteúdo “90-145”, em seguida, na linha 14, o agente Gestor responde a mensagem após efetuar a inferência, com o conteúdo “*Amostra com indicio de contaminacao por material pesado (ex. solvente com ponto de ebulicao alto, oleo diesel ou teor de alcool alto)*”.

Podemos perceber que no exemplo anterior os dados enviados de um agente para o outro, contém apenas um único valor por mensagem, o que demonstra assim a limitação da comunicação entre os agentes nesse modelo.

Como conseqüência, os agentes estavam limitados a transmitir poucos dados em uma única mensagem (Figura 34), pois, caso fossem enviadas muitas

informações, a interpretação dos dados exigiria um esforço de programação alto para tratar o volume da mensagem.

4.3.8. Melhorias no mecanismo de comunicação do SIMCQC

Com base nessas limitações a ontologia visa melhorar a comunicação entre os agentes, fornecendo para isso um vocabulário de termos e conceitos, e, assim, servir de linguagem de comunicação entre os agentes da sociedade. Isso possibilita aos agentes da sociedade enviar pacotes contendo informações completas sobre os dados trafegados, e, assim, superar a limitação do processo anterior.

Dessa forma, a adição de características ao modelo anterior (SILVA, 2008) fornece ao modelo um suporte para envio que contempla o pacote de amostras completo durante o tráfego de mensagens entre os agentes do SIMCQC e não apenas mensagens de texto simples.

```

1 public class AgenteIrox extends Agent {
2
3     private ContentManager manager = (ContentManager) getContentManager();
4     private Codec codec = new SLCodec();
5     private Ontology ontology = FuelOntology.getInstance();
6
7     [...]
8
9     protected void setup() {
10
11         manager.registerLanguage(codec);
12         manager.registerOntology(ontology);
13         addBehaviour(new HandleConfirmBehaviour(this));
14         ReadIroxDataBase();
15     }
16     [...]
17 }

```

Figura 35. Agente Irox instanciando a ontologia.

Na Figura 35, no trecho de código do agente Irox, podemos perceber a utilização da ontologia de combustíveis (*Fuel Ontology*). Primeiramente o agente

instancia um gerenciador (manager), responsável por controlar a configuração da comunicação do agente. Em seguida cria-se um objeto *codec* com uma instância de *SLCodec*. Este objeto indica a utilização da linguagem semântica SL (*Semantic Language*), o que significa que o agente utiliza esta linguagem para realizar a comunicação. Logo após, instancia-se um objeto que representará uma instância da *ontologia de combustíveis*, denominado aqui de *Ontology* apenas.

Realizada a instanciação dos objetos, encontra-se o método *setup()* do agente, o qual representa o coração do mesmo. Os objetos criados anteriormente são configurados dentro do método *setup()*, e em seguida, são incluídos no objeto principal de configuração do agente, o objeto manager. Assim, o agente Irox registra os objetos *codec* e *ontology* como podemos perceber nas linhas 11 e 12 do referido trecho de código. Com isso, o referido agente está apto para realizar uma comunicação com outro agente através da ontologia de combustíveis (*Fuel Ontology*). Vale ressaltar que para esta configuração ser realizada em sua plenitude, o outro agente da sociedade que fará parte desta comunicação também deve estar devidamente configurado com a referida ontologia.

Podemos perceber na linha 13 do trecho de código, dentro do método *setup()*, uma chamada para o método *ReadIROXDataBase()*. O referido método, localizado no coração do agente, é responsável pela função principal do agente Irox, ler a base de dados extraída do equipamento Irox e em seguida enviá-la para o agente Supervisor.

A seguir, detalha-se o processo de criação e envio das amostras de combustível do agente Irox.

```

1 public class AgenteIrox extends Agent {
2     [...]
3     protected void ReadIroxDataBase(){
4         Gasoline AmostraGasolina = new Gasoline();
5         List lista_propriedades = new ArrayList();
6         Property pETA = new Property();
7         Property pBEN = new Property();
8         Property pAROM = new Property();
9         Property pOLEF = new Property();
10        Property pSAT = new Property();
11        Property pMon = new Property();
12        Property pROM = new Property();
13        pETA.setName("ETA");
14        pETA.setValue((new Double(irox_Etha.replace(",",".")));
15        lista_propriedades.add(pETA);
16        pBEN.setName("BEN");
17        pBEN.setValue((new Double(irox_benz.replace(",",".")));
18        lista_propriedades.add(pBEN);
19        pAROM.setName("AROM");
20        pAROM.setValue((new Double(irox_arom.replace(",",".")));
21        lista_propriedades.add(pAROM);
22        pOLEF.setName("OLEF");
23        pOLEF.setValue((new Double(irox_olef.replace(",",".")));
24        lista_propriedades.add(pOLEF);
25        pSAT.setName("SAT");
26        pSAT.setValue((new Double(irox_satu.replace(",",".")));
27        lista_propriedades.add(pSAT);
28        pMon.setName("MON");
29        pMon.setValue((new Double(irox_mon_.replace(",",".")));
30        lista_propriedades.add(pMon);
31        pROM.setName("ROM");
32        pROM.setValue((new Double(irox_ron_.replace(",",".")));
33        lista_propriedades.add(pROM);
34        AmostraGasolina.setTypeSample("Composicao");
35        System.out.println("IROX: Nome da Amostra: "+irox_nome);
36        AmostraGasolina.setNameSample(irox_nome.replace(",",""));
37        AmostraGasolina.setDateSample(irox_data.replace(",",""));
38        AmostraGasolina.setProperties(lista_propriedades);
39        addBehaviour(new InformSupervisorBehaviour(this, AmostraGasolina));
40    }

```

Figura 36. Propriedades da técnica de composição (Gasolina).

No trecho de código da Figura 36 retirado do agente Irox, pode-se perceber a criação da amostra de gasolina e suas propriedades.

Primeiramente, na linha 4, o agente Irox cria um objeto *Gasoline*, que deverá conter os dados capturados do equipamento Irox. Em seguida, na linha 5, é criada uma lista de propriedades que irão conter todas as propriedades que a amostra vier a possuir.

Em seguida, da linha 6 a 12, através da classe *Property*, são criados os objetos de cada propriedade recuperada da amostra. Após a criação dos mesmos,

eles são inseridos dentro da lista de amostras, para novamente serem inseridos dentro da amostra *Gasoline*. Com isso, neste exemplo podemos perceber que uma amostra de combustível possui uma lista de propriedades na qual cada propriedade possui seu respectivo identificador e valor.

Por fim, na linha 39 é criado o comportamento (*InformSupervisorBehaviour*) para enviar a amostra de gasolina para o agente Supervisor. Este comportamento é uma classe interna do agente Irox responsável por efetuar o envio das amostras para o agente Supervisor.

```

1  class InformSupervisorBehaviour extends OneShotBehaviour {
2      [...]
3      public void action() {
4          ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
5          AID receiver = new AID("AgenteSupervisor",AID.ISLOCALNAME);
6          msg.setSender(getAID());
7          msg.addReceiver(receiver);
8          msg.setLanguage(codec.getName());
9          msg.setOntology(ontology.getName());
10         Information inform = new Information();
11         inform.setName(getAID());
12         inform.setStatus("Irox");
13         inform.setSample(it);
14
15         manager.fillContent(msg, inform);
16         myAgent.send(msg);
17         [...]
18     }

```

Figura 37. Comportamento de envio do agente Irox.

No trecho de código da Figura 37, podemos perceber o comportamento de envio do agente Irox. Na linha 4 é criado um objeto que representa a mensagem ACL a ser montada para enviar o pacote com as informações a serem enviadas. Na linha seguinte, cria-se um objeto para efetuar a comunicação direta. Neste caso, na linha 5, definimos o *Agente Supervisor* como o receptor da mensagem.

Nas linhas de 6 a 9, o objeto da mensagem é configurado com os devidos parâmetros de comunicação. Primeiramente com o nome do agente atual, em seguida com o nome do receptor da mensagem, logo após, a linguagem de comunicação SL e por último a ontologia que será utilizada para envio da

mensagem, no caso a *ontology* (instância da ontologia de combustíveis/*Fuel Ontology*).

Nas linhas 10 a 13, cria-se o objeto *information*. Este objeto é o predicado responsável por encapsular os dados de informação da amostra. Após o encapsulamento, o gerenciador *manager* insere o referido objeto (*information*) na mensagem ACL para posterior envio.

```

3 (INFORM
4 :sender ( agent-identifier :name AgenteIrox@pandora:1099/JADE :addresses (sequence http://par
5 :receiver (set ( agent-identifier :name AgenteSupervisor@pandora:1099/JADE ) )
6 :content "((INFORMATION (agent-identifier :name AgenteIrox@pandora:1099/JADE :addresses (seque
7 Irox (GASOLINE :serialID 0 :nameSample GC0807-171 :dateSample \"08/07/2007\"
8 :typeSample Composicao :properties
9 (sequence (PROPERTY :name ETA :value 0.0) (PROPERTY :name BEN :value 0.2)
10 (PROPERTY :name AROM :value 15.6) (PROPERTY :name OLEF :value 14.0)
11 (PROPERTY :name SAT :value 43.6) (PROPERTY :name MON :value 83.0)
12 (PROPERTY :name ROM :value 97.0)))))"
13 :language fipa-sl :ontology FuelOntology )
14
15 Gestor: Informação recebida do Supervisor. Mensagem :
16 Gestor Mensagem: (INFORM
17 :sender ( agent-identifier :name AgenteSupervisor@pandora:1099/JADE :addresses (sequence http
18 :receiver (set ( agent-identifier :name AgenteCognitivo@pandora:1099/JADE ) )
19 :content "((INFORMATION (agent-identifier :name AgenteSupervisor@pandora:1099/JADE :addresses
20 Supervisor (GASOLINE :serialID 0 :nameSample GC0807-183 :dateSample \"08/07/2007\"
21 :typeSample Composicao :properties
22 (sequence (PROPERTY :name ETA :value 0.0) (PROPERTY :name BEN :value 0.2)
23 (PROPERTY :name AROM :value 16.3) (PROPERTY :name OLEF :value 13.4)
24 (PROPERTY :name SAT :value 44.7) (PROPERTY :name MON :value 83.2)
25 (PROPERTY :name ROM :value 98.4)))))"
26 :language fipa-sl :ontology FuelOntology )
27

```

Figura 38. Trecho da comunicação utilizando a *Fuel Ontology*.

Na Figura 38 pode-se perceber a utilização da *ontology* proposta dentro do *SIMCQC*. O trecho de código da figura mostra a comunicação dos agentes da sociedade, onde no exemplo citado, o agente Irox encaminha um pacote de informação utilizando a *Fuel Ontology* (*ontologia de combustíveis*), contendo uma amostra do tipo gasolina e suas respectivas propriedades para o agente Supervisor. Logo em seguida, podemos perceber o agente Gestor, recebendo o pacote de informações (*INFORM*) enviado pelo Supervisor.

No primeiro bloco, da linha 7 até 13, pode-se perceber que o agente Irox encaminhou um pacote de dados contendo um objeto *information* que contém outros dois objetos, o agente que está enviando e uma amostra do tipo gasolina, com a

seguinte identificação nome (GC0807-171), data (08/07/2007) e ainda a técnica utilizada, no caso “composição”.

Ainda no primeiro bloco, porém nas linhas 9, 10, 11 e 12, seguem as propriedades da amostra que no caso são 7 (sete). Cada propriedade possui seu identificador e valor. Na amostra acima os identificadores de propriedades são: *ETA*, *BEN*, *AROM*, *OLEF*, *SAT*, *MON* e *ROM*, e seus respectivos valores são: *0.0*, *0.2*, *16.3*, *13.4*, *44.7*, *83.2* e *89.4*. Os identificadores de propriedades seguem os padrões de análises químicas de sua respectiva técnica. As sete propriedades citadas fazem parte da técnica de Composição (infravermelho).

Em vermelho, podemos perceber que os dois agentes, utilizam a linguagem de comunicação *FIPA-SL* e fazem o uso da *Fuel Ontology* (*ontologia de combustíveis*), a qual atua como um vocabulário e linguagem de comunicação entre os agentes.

Podemos perceber que, a utilização da ontologia de combustíveis (*Fuel Ontology*) proporcionou o encapsulamento dos dados das amostras capturadas pelos agentes, e também auxiliou no processo de transmissão de mensagens entre os agentes, através das classes fornecidas pela ontologia. Isso facilitou o processo de comunicação e proporcionou agilidade na transferência de dados entre os agentes da sociedade.

4.3.9. O comportamento cognitivo do agente Gestor

Após as inclusões dos agentes Irox, Destilador e Interface, o próximo objetivo foi a modificação do comportamento cognitivo do agente Gestor. Na arquitetura proposta por Silva (2008) o agente Gestor é responsável por inferir sobre os dados de amostras recebidas pela técnica de Destilação. Para isto, o agente analisa as mensagens enviadas pelo agente Monitor em formato de texto, em que, cada mensagem possui o valor de uma única propriedade da amostra.

Após o recebimento de cada nova mensagem, o agente Gestor dispara o comportamento cognitivo JESS e este realiza uma busca nas regras da sua base de

conhecimento. Em seguida, ele seleciona a regra que melhor satisfaz a situação, efetua a inferência por meio de comparação de variáveis de texto onde são comparados os dados recebidos na mensagem com os dados da base de conhecimento, e por fim, o resultado da análise é retornado para o agente Gestor.

No modelo de comunicação do *SIMCQC*, percebe-se que o envio de uma amostra, gera, no entanto, uma quantidade alta de mensagens dentro da sociedade, entre o agente Gestor e seu comportamento cognitivo JESS.

Estes problemas evidenciam que, o agente Gestor criado por Silva (2008) possui as seguintes limitações:

- Não está preparado para suportar a adição de um novo equipamento;
- O motor de inferência JESS não possui as regras necessárias para analisar as novas amostras do equipamento;
- O comportamento cognitivo utiliza o tratamento de texto, limitando a utilização dos objetos enviados pela ontologia.

Nesse sentido, as seguintes melhorias foram realizadas no comportamento cognitivo do agente Gestor:

- Modificação no mecanismo de comunicação do agente Gestor para corrigir o problema de excesso de mensagens, através da utilização da ontologia de combustíveis;
- Mudança do motor de inferência JESS para interpretar a programação orientada a objeto através das classes geradas pela ontologia de combustíveis (*Fuel Ontology*).
- Modificação do comportamento atual para tratamento dos novos pacotes de amostras oriundos do Destilador, desta vez recebido através da ontologia de combustíveis (*Fuel Ontology*).

- Criação novas regras na base de conhecimento JESS para tratamento das amostras oriundas do equipamento de análise da técnica de composição *Grabner Irox 2000 Gasoline*;

```
1 (import Ontologia.*)
2 (import examples.jess.*)
3 (defglobal ?*msg* = nil)
4 (defglobal ?*ok* = nil)
5 (deftemplate Information (declare (from-class Information)))
6 (deftemplate Sample (declare (from-class Sample)))
7 (deftemplate Gasoline (declare (from-class Gasoline)))
8 (deftemplate Diesel (declare (from-class Diesel)))
9 (deftemplate Biodiesel (declare (from-class Biodiesel)))
10 (deftemplate Alcohol (declare (from-class Alcohol)))
11 (deftemplate Property (declare (from-class Property)))
12
```

Figura 39. Definição da ontologia no comportamento cognitivo.

Conforme podemos perceber na Figura 39, a primeira alteração realizada no arquivo JESS do comportamento cognitivo, foi a definição das classes da ontologia, o que forneceu a orientação a objeto necessária para o comportamento cognitivo tratar os objetos encaminhados pelo agente Gestor.

Ainda na Figura 39, temos a declaração dos *deftemplates*, que são as estruturas das classes para serem interpretadas pelo JESS. Neste caso estamos definindo os *templates* de acordo com as classes que foram criadas pela ontologia.

Assim, quando um objeto for encaminhado para o comportamento cognitivo JESS e este possuir o formato de um dos *templates* definidos anteriormente, o mesmo saberá para onde encaminhar o pacote e assim realizar a inferência corretamente dos dados.

```

94 [...]
95 (defrule ETA
96   "regra para verificar as propriedades do tipo de combustivel"
97   ?p <- (Property {name == "ETA" && value <> nil}(name ?n) (value ?v) )
98   =>
99   (store ETAResultado "Propriedade ETANOL.Em conformidade com valores de referencia.")
100  (printout t "propriedade nome: " ?n " valor " ?v crlf)
101  ;(send ?*msg*)
102  )
103 [...]

```

Figura 40. Exemplo de *defrule* baseado no *deftemplate* da classe *Property*.

No exemplo da Figura 40 podemos perceber a utilização de uma regra. O *defrule* é o comando responsável por criar esta regra. A regra ETA acima é responsável por receber os pacotes enviados no formato do *deftemplate Property*, ou seja, todo pacote encaminhado pela ontologia do tipo *Property*, obrigatoriamente passará por esta regra e o comportamento cognitivo irá realizar as análises dos valores sobre os dados do objeto. Logo após, retornará para o agente através da variável *ETAResultado* o resultado da análise.

Nesse sentido, podemos afirmar que qualquer pacote encaminhado através ontologia para o comportamento cognitivo do agente Gestor poderá ser analisado pelo mesmo, contanto que, existam *defrules* com o formato do *template* da classe para analisar o pacote encaminhado.

Vale ressaltar que foram incluídas no comportamento cognitivo atual as *defrules* necessárias para analisar os pacotes das amostras enviadas pelos agentes Irox, Destilador e Interface do *SIMCQC*. No total de 30 regras para realização da inferência nos pacotes encaminhados pelos agentes descritos. Estima-se que para cada propriedade de combustível cria-se 2 regras de tratamento encadeadas, dessa forma, após a inclusão das regras para os demais combustíveis (Álcool, Diesel e Biodiesel) o sistema é provável que ultrapasse a quantidade de 100 regras.

Tabela 8. Regras da base de conhecimento do agente cognitivo (Gestor).

Linguagem Natural	Linguagem JESS
<p>SE Objeto <i>information</i> possui amostra <i>gasolina</i> ENTAO Verifica amostra <i>gasolina</i> FIM SE</p>	<pre>(defrule regra-information "regra para capturar a informacao e o combustivel" ?p <- (Information (Sample ?c)) => (assert (Gasoline ?p)))</pre>
<p>SE Objeto <i>gasolina</i> possui <i>propriedades</i> ENTAO Verifica <i>propriedades</i> de <i>gasolina</i> FIM SE</p>	<pre>(defrule regra-gasoline "regra para o combustivel gasolina" ?p <- (Gasoline (properties ?c)) => (assert (Property ?p)))</pre>
<p>SE objeto <i>propriedade</i> possui <i>ETA</i> ENTAO Verifica <i>propriedades</i> de <i>ETA</i> FIM SE</p>	<pre>(defrule regra-ETA1 "regra para verificar as propriedades ETA do combustivel gasolina" ?p <- (Property {name == "ETA" && value <> nil})(name ?n) (value ?v)) => (store ETAResultado "Propriedade ETANOL.Em conformidade com valores de referencia.") (printout t "propriedade nome: " ?n " valor " ?v crlf))</pre>

Na tabela acima podemos perceber algumas das regras que foram adicionadas ao comportamento cognitivo do agente Gestor, em linguagem natural e sua forma representativa em JESS.

No exemplo acima, podemos perceber o encadeamento das regras da base de conhecimento do agente Gestor. O fluxo se inicia no recebimento de um pacote contendo um objeto *information* que possui uma amostra de combustível gasolina e suas propriedades, que foram previamente analisadas e encaminhadas pelo equipamento *Irox 2000 Gasoline*.

O comportamento cognitivo recebe o pacote, identifica e aciona a regra que melhor se adapta a estrutura do pacote, no caso acima, a *regra-information*. Em seguida, a regra utilizada realiza uma nova chamada através do comando (*Assert (Gasoline ?p)*), dessa vez após a identificação do tipo de combustível. Novamente, o comportamento cognitivo realiza uma busca pela regra que melhor se adapta a nova estrutura do objeto, neste caso, foi acionada a *regra-gasoline*.

Após isso, a regra selecionada verifica as propriedades e encaminha o objeto contendo as propriedades da amostra gasolina através do comando (*Assert*

(*Property ?p*). Outra vez o comportamento cognitivo recebe o pacote contendo as propriedades e escolhe a regra que melhor se adapta para o tratamento. Finalmente, é acionada a última regra, a regra-ETA1, que irá verificar os critérios desta propriedade e retornar o resultado para o agente Gestor através do comando (*store ETAResultado "Propriedade ETANOL.Em conformidade com valores de referencia."*).

As modificações realizadas possibilitaram ao SIMCQC fornecer suporte para um total de 4 técnicas automatizadas e informatizadas para análise de combustível de gasolina em relação à arquitetura proposta por Silva (2008).

4.3.10. A nova arquitetura do sistema

A arquitetura de um SMA fornece mecanismos para visualizar como o sistema funciona em sua essência. Através dela, pode-se perceber a divisão de tarefas entre os agentes e como funciona a cooperação e coordenação dentro da sociedade.

A seguir é mostrada a arquitetura modificada do *SIMCQC*.

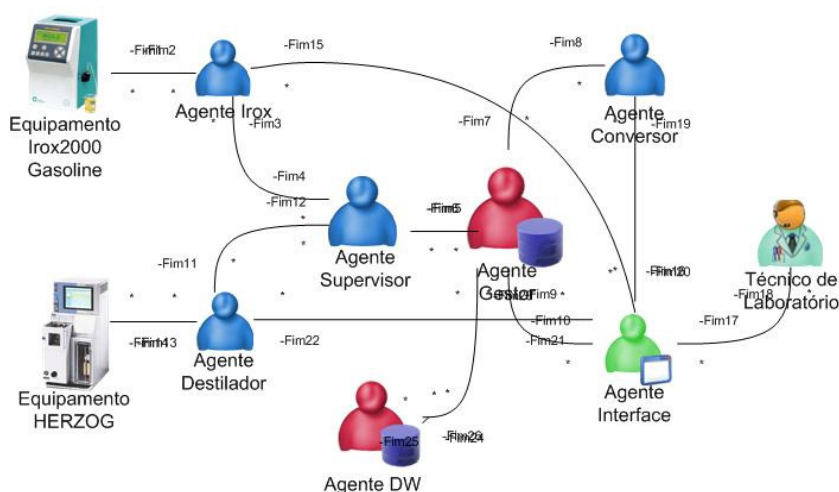


Figura 41. Nova arquitetura do *SIMCQC*.

Na Figura 41, vemos a inclusão dos novos agentes na sociedade do *SIMCQC*. Nela percebemos a distribuição dos agentes dentro da sociedade após a

inclusão dos agentes: Irox, Interface, Supervisor e Destilador. Este último, antigo agente Monitor do *SIMCQC* (SILVA, 2008), denominado nesta solução como agente Destilador.

Abaixo são detalhados os agentes da nova sociedade do sistema:

Agente Gestor – agente da solução anterior, responsável por fazer a inferência dos dados. Sofreu modificações no comportamento cognitivo para fornecer suporte a orientação a objetos e para utilizar a ontologia de combustíveis no mecanismo de comunicação. Foram adicionadas novas regras na base de conhecimento para interpretar as amostras recebidas pelo agente Irox e pelo agente Interface.

Agente Irox – novo agente incluído na sociedade, responsável por capturar as amostras da base de dados do equipamento *Irox 2000 Gasoline* e em seguida encaminhar para o agente Supervisor. Utiliza a ontologia de combustíveis no mecanismo de comunicação;

Agente Supervisor – novo agente incluído na sociedade, responsável por monitorar os agentes Irox e Destilador, agrupar as amostras pelo nome e em seguida encaminhar para o agente Gestor, utiliza a ontologia de combustíveis no mecanismo de comunicação;

Agente Interface – novo agente incluído na sociedade, responsável por fornecer uma interface gráfica para o usuário. Suas principais funcionalidades são: Iniciar captura de dados pelos equipamentos Irox e Destilador. Fornece ainda telas para o usuário inserir os dados complementares das amostras de combustíveis. Utiliza a ontologia de combustíveis no mecanismo de comunicação;

Agente Destilador – antigo agente monitor (SILVA, 2008) da solução anterior. Sofreu modificações em seu comportamento de captura de base de dados, pois o atual equipamento de análise da técnica de Destilação foi modificado. Sofreu modificações para utilizar a ontologia de combustíveis no mecanismo de comunicação

Agente Conversor – agente da solução anterior, é responsável por converter e padronizar os dados para exportação em XML. Sofreu modificações para utilizar a ontologia de combustives no mecanismo de comunicação;

Agente DW – agente da solução anterior, é responsável por armazenar os resultados das amostras analisadas pelo agente Gestor. Não sofreu maiores modificações, apenas a inclusão da ontologia de combustível no mecanismo de comunicação.

Após descrições dos agentes, a quinta e ultima fase da metodologia PASSI (COSSETINO, 2002) descreve a criação de um diagrama de distribuição do sistema, tendo como objetivo visualizar a implantação do mesmo. Nesse sentido, foi criado o diagrama a seguir para representar a implantação do sistema.

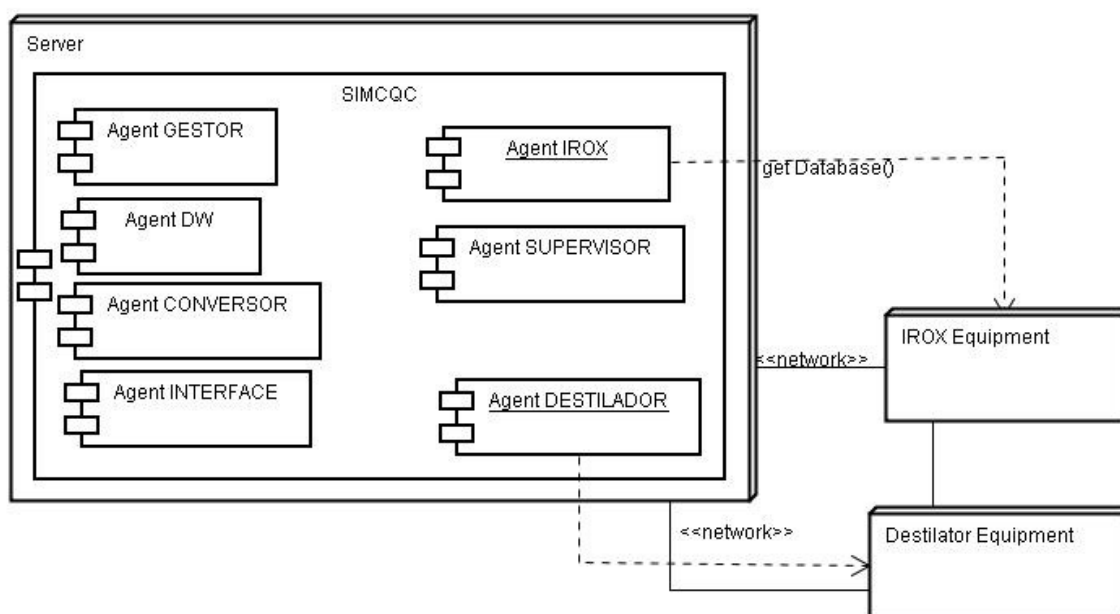


Figura 42. Diagrama de implantação/distribuição do SIMCQC.

No diagrama de implantação da Figura 42, podemos perceber como os agentes e equipamentos estão distribuídos, utilizando para isso os pacotes da notação UML.

O pacote *Server*, representa um servidor contendo uma instancia do *SIMCQC*, onde são destacadas as instancias dos agentes da sociedade (Irox, Destilador, Supervisor, Gestor, DW, Conversor e Interface).

Do lado direito, temos os pacotes referentes aos equipamentos externos *Grabner Irox 2000 Gasoline (IROX Equipment)* e *Destilator Opitdist (Destilator Equipment)*. A comunicação é representada pelo quadrado “*network*” e o acesso aos equipamentos é representado pelo método *getDatabase()* existente em ambos os agentes.

Dentro do pacote *SIMCQC*, temos os pacotes referentes as instancias dos agentes da solução (Irox, Monitor, Destilador, Supervisor, Gestor, Conversor e DW).

Vale ressaltar, que todos os agentes da solução foram modificados para utilizar a ontologia de combustíveis como linguagem de conteúdo do domínio da aplicação no mecanismo de comunicação da sociedade.

Após a finalização das etapas anteriores da PASSI, passa-se então para a realização dos testes do sistema.

A seguir, seguem os resultados e considerações sobre a utilização do sistema.

4.4. Resultados

O mecanismo de comunicação proposto por *Silva (2008)* é limitado, pois a possibilidade de extensão do sistema com a inclusão de novos equipamentos exige um esforço de programação alto ao tentar enviar varias amostras utilizando somente a Linguagem Semântica FIPA-SL (FIPA, 2009), tornando complexa a solução.

Nesse sentido, a inclusão de um vocabulário para representação do conhecimento do domínio através da *ontologia de combustíveis* forneceu uma linguagem de comunicação sem maiores esforços de programação para estender o sistema.

O motor de inferência (*JESS*) e o agente Cognitivo (Gestor) proposto por *Silva (2008)* tiveram que ser modificados para compreender a nova linguagem de comunicação (*ontologia*), depois de realizadas as modificações, o motor de inferência atuou de forma eficiente ao interpretar claramente os termos utilizados pela *ontologia*, justificando assim o uso da mesma.

Os testes com a ferramenta foram realizados nas dependências do LAPQAP na *UFMA*, com a utilização de arquivos de dados gerados a partir do equipamento *Irox 2000 Gasoline* responsável por analisar as amostras do tipo Gasolina pela técnica de Infravermelho (Composição) e do equipamento Destilador *OpitDist*³, responsável por analisar as amostras de diesel e gasolina pelo método de destilação.

Os arquivos gerados continham informações necessárias para o *SMA* efetuar a inferência nos dados e criar relatórios para o técnico efetuar a análise. O sistema analisou em 5 minutos um total de 108 amostras e informou ao técnico de laboratório através da ferramenta os resultados das análises.

O sistema logrou êxito ao final da análise, informando os resultados com base nos valores de referencias dos formulários de análise de combustíveis. Criou relatórios resumidos e possibilitou ganho de produtividade ao reduzir a jornada de trabalho do técnico de laboratório, pois este, após a utilização do sistema, despreendeu menos tempo para realizar as atividades de análises.

Tabela 9. Comparativo de funcionalidades

	Reuso	Ontologia	Interface Grafica	Relatórios	Exportação XML	Equipamentos		JESS	Técnicas
						Destilador	IROX		
SIMCQC (SILVA, 2008)					X	X		X	1
SIMCQC MODIFICADO	X	X	X	X	X	X	X	X	4

³ O equipamento *HERZOG 628* foi descontinuado, com isso o agente Destilador sofreu modificações no decorrer da pesquisa para obter os dados a partir do novo equipamento de destilação, o *OpitDist*.

A tabela acima ilustra uma comparação dos recursos existentes na ferramenta *SIMCQC* proposta por Silva (2008) e da nova solução, denominada nesse exemplo de *SIMCQC-MODIFICADO*.

Os novos recursos incluídos na solução são:

- Ontologia para representação do conhecimento do domínio da química analítica de combustíveis, possibilitando assim o reuso da mesma;
- Interface gráfica com o usuário (GUI) – para facilitar a interatividade do técnico de laboratório com as funcionalidades do *SIMCQC*;
- Relatórios e laudos técnicos em formatos PDF;
- Inclusão da técnica de composição através do equipamento *Irox 2000 Gasoline*;
- Configuração de arquivos de dados dos equipamentos de análises químicas (*Irox* e Destilador) e da base de conhecimento JESS (base de regras);
- Total de quatro técnicas de análises – tornando assim o sistema capaz de inferir sobre 100% das propriedades da amostra de gasolina.

5. CONCLUSÃO E TRABALHOS FUTUROS

Nesta dissertação, foi mostrada a aplicação de tecnologias de Inteligência Artificial (IA) e Representação do Conhecimento (RC) para superar os desafios de diferentes áreas de atuação. Por um lado, a Engenharia de Software, com seus agentes inteligentes e ontologias, e por outro lado, a Química Analítica, com suas análises e ensaios químicos fornecendo suporte na análise e monitoramento da qualidade dos combustíveis.

Uma das contribuições deste trabalho foi a criação de uma ontologia de combustíveis, denominada *Fuel Ontology*, para representar o domínio das análises químicas de combustíveis do Programa de Monitoramento da Qualidade de Combustível (MQC) da Agência Nacional de Petróleo (ANP), realizado pelo Laboratório de Análise e Pesquisa em Química Analítica de Petróleo (LAPQAP).

A utilização da ontologia possibilitou confiabilidade no processo de tramitação das mensagens entre os agentes da sociedade, uma vez que agora, os dados estão descritos pelas classes da ontologia. Com isso, a ontologia atuou como linguagem entre os agentes no processo de comunicação da sociedade do *SIMCQC*.

O uso da ontologia possibilitou ainda, a representação das ações dentro do sistema, utilizando os termos do domínio da análise de combustíveis, proporcionando o encapsulamento dos dados na transmissão das mensagens exportadas para o MQC e posteriormente com a ANP, com ganhos de produtividade e sem atrasos operacionais.

Outra contribuição alcançada se deu através da inclusão dos novos agentes na sociedade de agentes do *SIMCQC*. Foram incluídos dois agentes. O primeiro, o agente Irox, foi o responsável direto pela inclusão de uma nova técnica ao *SIMCQC*, o que possibilitou que o sistema abrangesse mais uma técnica de análise de combustível, a técnica de composição (infravermelho).

Através da inclusão do agente Irox, o sistema tornou-se capaz de inferir sobre dados das amostras de combustível do tipo gasolina, recuperadas diretamente

do equipamento de análises químicas (*Grabner Irox 2000 Gasoline*), e retornar ao técnico de laboratório com precisão e rapidez os resultados obtidos através dos relatórios recuperados pelo sistema.

Com a inclusão do segundo agente, o agente gráfico, definido como Interface, foi proporcionado ao técnico de laboratório um mecanismo de comunicação direta com todas as funcionalidades do sistema, o que não era possível na arquitetura anterior do *SIMCQC* (*SILVA, 2008*).

Através desta Interface o técnico tornou-se capaz de interagir com o sistema, podendo utilizar-se de recursos como: escolher os arquivos de configuração dos equipamentos de análises, emitir relatório das atividades realizadas pela sociedade de agente, relatórios estes, que fornecem em sua totalidade os resultados contendo as propriedades de combustível analisadas e laudos técnicos, e por fim, exportar para o MQC através de arquivos *XML* os dados analisados.

A inclusão do agente Interface representa a terceira contribuição deste trabalho, pois tornou-se possível informatizar os procedimentos que eram realizados manualmente. A partir de sua inclusão o técnico de laboratório passou a dispor de uma ferramenta informatizada para efetuar a inferência sobre os dados capturados automaticamente dos equipamentos e sobre os dados complementares inseridos manualmente pela interface gráfica.

Após a criação dos agentes mencionados, todos os agentes da sociedade foram modificados para utilizar a *ontologia de combustíveis*. E após a modificação do mecanismo de comunicação, os agentes passaram a utilizar a ontologia como linguagem de termos contendo o vocabulário do domínio, deixando de se comunicar através de simples mensagens de texto.

As modificações realizadas no comportamento cognitivo do agente Gestor representam a última contribuição deste trabalho. O método de acesso às regras da base de conhecimento foi alterado. Anteriormente, o agente acessava a base de regras do JESS utilizando-se apenas de mensagens simples de texto. Após a inclusão da ontologia, o comportamento cognitivo foi modificado para utilizar a capacidade de orientação a objeto fornecido através da utilização das classes da

ontologia para o acesso a base de conhecimento JESS, e assim selecionar e acessar as regras.

Dessa forma, o comportamento cognitivo obteve maior desempenho no acesso e seleção de regras em relação ao modelo proposto por Silva (2008). Com isso, que o agente cognitivo tornou-se capaz de tratar maior quantidade de informações utilizando-se de menores quantidades de mensagens.

Ao final das inclusões percebe-se que o SIMCQC pode ser considerado estável, na medida em que se encontra preparado para receber futuras modificações estruturais, possibilitando o crescimento do mesmo de forma organizada.

Com este trabalho foi possível demonstrar que o uso das técnicas de Representação do Conhecimento (RC) aliadas às de Inteligência Artificial (IA) no processo de análises químicas proporcionou ganho na produtividade das atividades de análises químicas realizadas pelo LAPQAP, com redução de materiais e diminuição de custos do laboratório.

O estudo foi considerado satisfatório, facilitando a análise dos dados e o processo de tomada de decisão, podendo auxiliar também a interpretação de resultados para a melhoria do programa de Monitoramento da qualidade de combustíveis da ANP.

Durante a pesquisa de monitoramento de combustível foi possível demonstrar os resultados alcançados através da publicação abaixo:

Sistema multiagente de apoio à tomada decisão para controle da qualidade de combustíveis (CORREA et. al, 2009), artigo completo, aceito no 5º Congresso Brasileiro de Pesquisa e Desenvolvimento em Petróleo e Gás- 5º PDPETRO, Fortaleza-CE, 2009. Descreve a modelagem de melhorias no *SIMCQC* (SILVA, 2008) através da inclusão de novos agentes a sociedade multiagente do sistema.

A seguir são descritas algumas sugestões para trabalhos futuros:

- Acrescentar mecanismos de cadastro das informações dos postos de combustíveis em banco de dados local, pois, estas informações servirão de base para a inclusão da etapa de planejamento no *SIMCQC*.

- Com a inclusão das informações cadastrais, o mecanismo de comunicação deverá ser modificado, possibilitando assim a extensão da ontologia, desta vez, para contemplar a fase de Planejamento;
- Existe ainda a possibilidade de acrescentar novos equipamentos para análises químicas, estendendo a arquitetura atual, e assim contemplar outros equipamentos e fases do PMQC.
- A partir as informações dos postos será possível gerar mecanismos para auxiliar a fase de planejamento. Um mecanismo SIG (Sistema de Informação Geografica) poderia ser criado com a inclusão de um agente para acessar as informações de uma ferramenta (SIG), tendo como objetivo principal gerar as rotas para coleta de amostras.

REFERÊNCIAS

AGENCIA NACIONAL DE PETRÓLEO GAS E BIOCMBUSTIVEIS - ANP. **Os programas de Monitoramento da Qualidade**. Disponível em: <http://www.anp.gov.br/petro/programa_Monitoramento.asp/>, 2009. Acessado em: 13 de junho de 2009.

ALMEIDA, M. B.; **Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção**. 2003.

_____. **NBR 10520: informação e documentação: apresentação de citações em documentos**. Rio de Janeiro, 2002;

BANCO NACIONAL DO DESENVOLVIMENTO ECONÔMICO E SOCIAL – BNDES. **A revolução do petróleo**. Disponível em:< <http://50anos.bndes.gov.br/> >. Acessado em: 28/06/2009.

BELLIFEMINE, F.; CAIRE, G.; GREENWOOD, D. **JADE - Developing Multi-Agent System with JADE**. 2007.

CAIRE, G.; CABANILLAS, D. **Jade Tutorial. Application-Defined Content Languages and Ontologies**. JADE 3.3. <<http://jade.tilab.com/doc/CLOntoSupport.pdf/>>, (2004).

CHAUI, M.; **Convite a filosofia**. São Paulo, 13ª Ed (p. 183). Ática, 2005.

CHELLA, A., COSSETINO, M., SABATUCCI, L. and SEIDITA, V.; Agile PASSI: **An Agile Process for Designing Agents**. *International Journal of Computer Systems Science & Engineering*. Special issue on "Software Engineering for Multi-Agent Systems", 21(2). March 2006.

CHIRICO, U. **JIP – Java Internet Prolog**. Disponível em: < <http://www.ugosweb.com/jiprolog/doc/index.html> >. Acessado em: 20/05/2009.

CORREA, P. J. M. G., MARQUES, D. B., FARIAS O., CAVALCANTE, G. H., LABIDI, S., BARROS A. K, MARQUES, A. B.; **Sistema multiagente de apoio a tomada decisão para controle da qualidade de combustíveis**, 5º Congresso Brasileiro de Pesquisa e Desenvolvimento em Petróleo e Gás- 5º PDPETRO. 15 - 22 out., Fortaleza-CE. 2009.

COSENTINO, M, BURRAFATO, P; **Designing a multi-agent solution for a bookstore with the PASSI methodology** In: Fourth International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2002), Toronto (Ontario, Canada), May 27-28, 2002.

COSENTINO, M., POTTS C.; **PASSI: A Process for Specifying and Implementing Multi-Agent Systems Using UML**. 2002.

DAML+OIL Web Ontology Language (March 2001). Disponível em: <http://www.w3.org/Submission/2001/12/> . Acessado em: maio de 2009.

DAVIES, J., D. FENSEL and F. VAN HARMELEN, 2003, **Towards the Semantic Web: Ontology-driven Knowledge Management**. John Wiley & Sons, LTD, West Sussex, England.

FARIAS, O. S; **MODELAGEM E IMPLEMENTAÇÃO DE UM SISTEMA MULTIAGENTE PARA SELEÇÃO DE FALHAS E TOMADA DE DECISÃO EM VIRADORES DE VAGÕES**. Dissertação de Mestrado. 2009.

FIPA – **Foundation for Intelligent Physical Agents**, Disponível em: <http://www.fipa.org/specs/fipa00008/SC00008I.pdf> />, Acessado em: maio de 2009.

FRIEDMAN-HILL, E. J.; **Jess in Action: Rule-based systems in java**. USA: Manning Press, 2003.

FRIEDMAN-HILL, E. J., **JESS The Rule Engine for the Java Platform - Language Reference** – version 7.0b7 (11 may 2006) DRAFT, Sandia National Laboratories. Livermore, CA, USA.

GIRARDI, R.; **Engenharia de Software Baseada em Agentes**. Congresso Brasileiro de Ciência da Computação, Itajaí, 2004, 913-937. Itajaí, SC-Brasil, ISSN 1677-2822, 2004.

GRUBER, T. R., GUARINO, N., POLI, R; **Toward Principles for the Design of Ontologies Used for Knowledge Sharing (1993)**. Disponível em: citeseer.ist.psu.edu/gruber93toward.html/>.

GUARINO, N: **Formal ontology, conceptual analysis and knowledge representation**. Int. J. Hum.-Comput. Stud. 43(5-6): 625-640 (1995)

GUARINO, N.; **Formal Ontology and Information Systems**. In Formal Ontology in Information Systems, edited by N. Guarino (Amsterdam: IOS Press), pp.3-15. 1998.

GUIZZARDI, R S.S., **Inteligencia artificial**, 2007.

GUIZZARDI, Renata S. S.; PERINI, Anna. **Analyzing Requirements of Knowledge Management Systems with the Support of Agent Organizations**. In Journal of the Brazilian Computer Society (JCBS). Special Issue on Agents Organizations, vol. 11, n. 1, pág. 51-62. Julho 2005.

GUIZZARDI, Giancarlo ; FALBO, Ricardo; GUIZZARDI, Renata S. S . **A importância de Ontologias de Fundamentação para a Engenharia de Ontologias de**

Domínio: o caso do domínio de Processos de Software. Revista IEEE América Latina, v. 6, n.3, p. 244-251, 2008.

GRIGOROVA D., NIKOLOV, N.; **Knowledge Representation in Systems with Natural Language Interface.** International Conference on Computer Systems and Technologies - CompSysTech'07. 2007;

HONORATO, F. A., BARROS NETO, B., PIMENTEL M. F., STRAGEVITCH, L, GALVÃO R. K. H; **Using principal component analysis to find the best calibration settings for simultaneous spectroscopic determination of several gasoline properties.** 2008.

IGLESIAS C., GARIJO M., J., CENTENO-GONZALEZ J. VELASCO R.; **Analysis and Design of Multiagent Systems Using MAS-CommonKADS. Agent Theories, Architectures and Languages.** Lecture Notes in Artificial Intelligence. Vol. 1365, p. 313-326. Springer-Verlang. 1998.

JADE - **Java Agent Development Framework**, Disponível em: <<http://jade.cselt.it/>>, Acessado em: maio de 2009.

JENNINGS, N. R. **On Agent-based Software Engineering. Artificial Intelligence:** 117, p. 277-296. 2000.

JESUS, J B M. **tesauro: um instrumento de representação do conhecimento em sistemas de recuperação da informação.** 2002.

KAVOURAS, M.; **A unified Ontological Framework for Semantic Integration**, To appear in a special book "Next Generation Geospatial Information" by A.A. Balkema Publishers - Taylor & Francis, The Netherlands, P. Agouris (ed)., late 2004.

LABIDI, Sofiane; BASTOS FILHO, Othon; AXT, Margarete. **SISTEMA INTELIGENTE DE DESAFIOS ABERTOS IOCS: Uma proposta de adaptação dos padrões do Método Clínico Piagetiano em Plataforma Multiagentes.** RENOTE. Revista Novas Tecnologias na Educação, v. 4, p. 10, 2006.

LABIDI, Sofiane; SOUZA, C. M.; NASCIMENTO, E. **NetClass: Cooperative Learner Modeling in a Web-Based Environment** In: 6th Int. Conf. On Computer Based Learning in Science. Proceedings of the 6th Int. Conf. On Computer Based Learning in Science (CBLIS) Nicosia, Cyprus: University of Cyprus, 2003.

LABIDI, S. **CommonKADS Extension for Supporting Multi-Expertise.** In: The 17th International Conference of the British Computer Society on Expert Systems,(ES'97)., 1997, Cambridge. Proceedings of the 17th International Conference of the British Computer Society on Expert Systems, (ES'97)., 1997.

LABORATORIO DE ANÁLISE E PESQUISA EM QUÍMICA ANALÍTICA DE PETRÓLEO - LAPQAP; **A história do LAPQAP.** Disponível em: <<http://www.quimica.ufma.br/LAPQAP/>>, 2009. Acessado em: 14 de junho de 2009.

LINO N., TEDESCO P., SILVA D. R.; **Modelo de Percepção de Agentes Inteligentes baseados em Emoções**. DIGITAL PROCEEDINGS of the V Brazilian Symposium on Computer Games and Digital Entertainment, Recife, Brazil, 8-10 November 2006.

MARQUES, D; **Sistema Integrado de Monitoramento e Controle da Qualidade de Combustível**. Dissertação de Mestrado, UFMA, DEE, 2003. Acessado em: 29 jan. 2009. Disponível em: <<http://www.ppgee.ufma.br/aigaion/?page=publication&kind=single&ID=173/>>.

MARQUES, D. B., CORREA, P. J. M. G., CAVALCANTE, G. H., LABIDI, S., BARROS A. K; **Sistema de monitoramento e controle da qualidade de combustível**, 5º Congresso Brasileiro de Pesquisa e Desenvolvimento em Petróleo e Gás- 5º PDPETRO. 15 - 22 out, Fortaleza-CE. 2009.

MENKEN, M.; **Jess Tutorial**; Vrije Universiteit, Amsterdam, The Netherlands December 24, 2002.

MINSKY M.; **A Framework for Representing Knowledge**, MIT – AI Laboratory Memo 306, June, 1974. Disponível em: <<http://web.media.mit.edu/~minsky/papers/Frames/frames.html/>>

MYPOULOS, J.; **An overview of knowledge representation**. Proceedings of the workshop on Data abstraction, databases and conceptual modelling. Pages: 5 - 12, 1981.

NETO, M. G. de. M., **ANP Análise de Dados**. UFPE, ANP. Disponível em: <http://buscatextual.cnpq.br/buscatextual/visualizacv.jsp?id=K4781674U8&tipo=completo&idiomaExibicao=1#ProducaoCientifica/>>. Acessado em: 30 julho, 2009.

NOY, N F., MCGUINNESS, D L.; **Ontology Development 101: A Guide to Creating Your First Ontology**. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

OLIVEIRA, E M.; **Transformações no mundo do trabalho, da revolução Industrial aos nossos dias**, 2004.

OLIVEIRA, E. J. S.; **Comunicação segura e confiável para sistemas multiagentes adaptando especificações XML**. Dissertação de Mestrado, UFMA, DEE, 2006. Acessado em: 29 jan. 2009. Disponível em: <<http://www.ppgee.ufma.br/aigaion/?page=publication&kind=single&ID=173/>>.

ONTOLOGYBEANGENERATOR, 2008. Disponível em: <<http://protege.cim3.net/cgi-bin/wiki.pl?OntologyBeanGenerator> />. Acessado em 12/03/2008.

OWL. Ontology Web Language. Disponível em: <<http://www.w3.org/TR/owl-ref/>>. Acesso em: 20 jan. 2008.

PINHO, C. M. D.; KUX, H. H.; ALMEIDA, C. M.; **Elaboração de rede semântica para classificação de cobertura do solo de ambientes intra-urbanos: o Caso de São José dos Campos - SP**; Anais XIII Simpósio Brasileiro de Sensoriamento Remoto, Florianópolis, Brasil, 21-26 abril 2007, INPE, p. 637-644.

PIO, F. V.; CACIAN, M. H., FRANCA, R.; **JESS – the Rule Engine for the Java™ Platform**; Acessado em: 20 abr. 2009. Disponível em: <www.das.ufsc.br/~gb/pg-ia/Jess07/das6607-francesca-maiara-ricardo.pdf>.

PRESSMAN, Roger S. R.S. Pressman & Associates, Inc. Disponível em: <<http://www.rspa.com>>. Acesso em 07 abr. 2009.

PROTÉGÉ PROJECT. Disponível em: <<http://protege.stanford.edu/>>. Acessado em: 16 jul. 2009.

RDF. **Resource Description Framework**. Disponível em: <<http://www.w3.org/RDF/>>. Acessado em: 20 jun. 2007.

ROSA, J. H.; **O Uso de Ontologias para a Categorização de Produtos no Comércio Eletrônico**. 2008.

ROSSI, A. V., TORETTI G, A.; **PROQUIMICA: A Química Analítica**. Disponível em: <<http://proquimica.iqm.unicamp.br/>> Acessado em: 22/06/2009.

RUSSELL, S; NORVIG, P.; **Inteligência Artificial**. 2. ed. São Paulo: Campus, 2004. 1040 p.

SAMPAIO, C H.; LABIDI, S; FARIAS, O; **Using Agents for Detection of Fraud in Municipal Taxes in 7th International Conference on Intelligent Systems Design and Applications - ISDA'07 – Universidade Estadual do Rio de Janeiro - UERJ**. 2007.

SAMPAIO, C H; **uso de agentes na detecção de fraudes em imposto municipal - iss**. Dissertação de Mestrado. 2007.

SILVA, R, J; SOFIANE, L; MONTEIRO, M.S.; FARIAS, O; **Intelligent System for Fuels Quality Control and Monitoring Using PASSI Methodology.08**. Eighth International Conference on Volume, Issue , 10-12 Sept. 2008 Page(s):855 - 860.

SILVA, R, J; **SIMCQC: Sistema Inteligente para Monitoramento e Controle da Qualidade de Combustível**. Dissertação de Mestrado, UFMA, DEE, 2008. Acessado em: 12 dez. 2008. Disponível em: <<http://www.ppgee.ufma.br/aigaion/?page=publication&kind=single&ID=250> />.

SOUZA, R. R. A.; **EXPERT-RETE: Um Motor de Inferencia para Agentes cognitivos**. Dissertação de Mestrado. 2005.

SPEM - Software Process Engineering Metamodel Specification. Disponível em: <http://www.omg.org/technology/documents/formal/spem.htm>. Acessado em: 5 abr.2007.

STUDER, R; BENJAMINS, R V.; FENSEL, D.; **Knowledge Engineering: Principles and Methods**. In Data Knowl. Eng. 25(1-2):161-197, 1998.

USCHOLD, M., GRUNINGER, M.; **Ontologies: Principles, Methods and Applications**. Knowledge Engineering Review 11(2). 1996.

VIANNEY O. S. JR., F. C.C. OLIVEIRA, D. G. L., ANDREA C. P., EDGARDO GARCIA, P. A.Z. SUAREZ, J. C. R; **A comparative study of diesel analysis by FTIR, FTNIR and FT-Raman spectroscopy using PLS and artificial neural network analysis**. 2005.

YING A. G., SHANG Z. Kokossis A; **Agent-based intelligent system development for decision support in chemical process industry**. 2009.

W3C - The World Wide Web Consortium; Disponível em: <http://www.w3.org/>. Acessado em: 30 junho de 2009

WIKIPEDIA, a enciclopédia livre; **Ontologia em Ciências da Computação**. Disponível em: [http://pt.wikipedia.org/wiki/Ontologia_\(ci%C3%A2ncias_da_computa%C3%A7%C3%A3o\)](http://pt.wikipedia.org/wiki/Ontologia_(ci%C3%A2ncias_da_computa%C3%A7%C3%A3o))/>Acessado em: 30 junho de 2009

WOOLDRIDGE, M.; **An Introduction to Multi-Agent Systems**, John Wiley & Sons Ltda. Chichester, Inglaterra, 2002.

ANEXO I - FORMULARIOS

Formulário de Análise de Amostra de Combustível Gasolina

Identificação da amostra GC _____	REGIÃO	Data da Coleta:	Responsável pela Coleta:
	GASOLINA	Data de Entrada:	ANALISTAS: (01) -
		Data da Análise:	(02) -

Aspecto e Cor			ASTM D1500
Especificação	Aspecto (*) (01)	Cor Exceção: Azul	Analista
Análise 1	() 01 () 02 () 03 () 04		(01) (02) (03)
Análise 2	() 01 () 02 () 03 () 04		(01) (02) (03)
Análise 3	() 01 () 02 () 03 () 04		(01) (02) (03)

* (01): límpido e isento de impurezas; (02): límpido com impurezas; (03): turvo e isento de impurezas; (04): turvo com impurezas

Massa específica (m.e.)		Método: NBR 7148/ASTM D 1298 (HV) / NBR 14065/ASTM D 4052 (DD)		
Especificação	HV – ASTM D1298 DD – ASTM D 4052	Massa específica a 20°C (g/cm³)	Massa específica a 20°C (Kg/m³)	Analista
Análise 1	() HV () DD			(01) (02) (03)
Análise 2	() HV () DD			(01) (02) (03)
Análise 3	() HV () DD			(01) (02) (03)

* HV = Hidrômetro de Vidro; DD = Densímetro Digital (DMA 4500)

Destilação (Herzog HDA 620/628)						Método: NBR 9619/ASTM D 86	
Especificação	T PIE (°C)	T 10% (°C)	T 50% (°C)	T 90% (°C)	T PFE (°C)	Resíduo (%vol)	Destilador (01) (02) (03)
	–	máx. 65,0	máx. 80,0	min. 145,0 máx. 190,0	máx. 220,0	máx. 2,0	Analista
Análise 1							(01) (02) (03)
Análise 2							(01) (02) (03)
Análise 3							(01) (02) (03)

Infravermelho (IROX 2000)									
Especificação	Etanol (%vol)	Benzeno (%vol)	Aromáticos (%vol)	Olefinas (%vol)	Saturados (%vol)	MON	ROM	M+Ry2	Analista
	–	máx. 1,0	máx. 45	máx. 30	–	min. 82		min. 87,0	
Análise 1									(01) (02) (03)
Análise 2									(01) (02) (03)
Análise 3									(01) (02) (03)

Teor de Alcool Etílico Anidro Combustível (AEAC)				Método: NBR 13992
Especificação	T (°C)	a = Aumento da camada aquosa (mL)	Teor de AEAC (%vol) $V = 2 \times a + 1$ min. 24 máx. 26	Analista
	–	–		
Análise 1				(01) (02) (03)
Análise 2				(01) (02) (03)
Análise 3				(01) (02) (03)

Analista Responsável

Formulário de Análise de Amostra de Combustível Diesel

Produto: 01- Óleo Diesel B Comum 02- Óleo Diesel B Aditivada	Identificação da Amostra	Data da Coleta:	Responsável pela Coleta:
	OD___ - ___	Data de Entrada:	ANALISTAS: (01) -- (02) - (03) -
Região:	ÓLEO DIESEL	Data da Análise:	

Aspecto e Cor				NBR 14483 / ASTM D1500	
Especificação	Aspecto (*)			Cor	Analista
	Limpido e isento de impurezas				
Análise 1	() 01	() 02	() 03	() 04	(01) (02) (03) (04)
Análise 2	() 01	() 02	() 03	() 04	(01) (02) (03) (04)
Cor ASTM					(01) (02) (03) (04)

* (01): limpido e isento de impurezas; (02): limpido com impurezas; (03): turvo e isento de impurezas; (04): turvo com impurezas

Massa específica (m.e.)		Método: NBR 7148/ASTM D 1298 (HV) / NBR 14065/ASTM D 4052 (DD)			Analista
Especificação	Método (*)	Massa específica a 15° C (g/cm ³)	Massa específica a 20° C (g/cm ³)	Massa específica a 20° C (kg/m ³)	
		HV – ASTM D1298 DD – ASTM D4052	-	0,820 - 0,880 (Interior) 0,820 - 0,865 (Metropolitana)	820,0 - 880,0 (Interior) 820,0 - 865,0 (Metropolitana)
Análise 1	() HV () DD				(01) (02) (03) (04)
Análise 2	() HV () DD				(01) (02) (03) (04)
Análise 3	() HV () DD				(01) (02) (03) (04)

* HV = Hidrômetro de Vidro; DD = Densímetro Digital (DMA 4502)

Destilação (Herzog HDA 620/628)								Método: NBR 9619/ASTM D 86	
Especificação	T PIE	T 10%	T 50%	T 85 %	T 90%	T PFE	Resíduo	Destilador (01) (02) (03) (04)	
	-	-	min. 245 máx. 310,0	máx. 370 °C (Interior) máx. 360°C (Metropolitana)	-	-	% Vol	Analista	
Análise 1								(01) (02) (03) (04)	
Análise 2								(01) (02) (03) (04)	
Análise 3								(01) (02) (03) (04)	

Índice de Cetano		Método ASTM D 613 / ASTM D 4737		
Especificação	min. 42 ASTM D 613 Número de Cetano min. 45 ASTM D 4737 Índice de Cetano (Irox)	Metodologia		Analista
		() ASTM D 613	() ASTM D 4737	
Análise 1		() ASTM D 613	() ASTM D 4737	(01) (02) (03) (04)
Análise 2		() ASTM D 613	() ASTM D 4737	(01) (02) (03) (04)
Análise 3		() ASTM D 613	() ASTM D 4737	(01) (02) (03) (04)

Teor de Enxofre (RX 350 S)		Método: NBR 14533/ASTM D 4294/ ASTM 1552 / ASTM 2622		
Especificação	Teor (% massa)		Analista	
	máx. 0,18 (Interior)	máx. 0,05 (Metropolitano)		
Análise 1			(01) (02) (03) (04)	
Análise 2			(01) (02) (03) (04)	
Análise 3			(01) (02) (03) (04)	

Ponto de Fulgor (HFP 360)		Método: NBR 14598/NBR 7974/ASTM D 56 / ASTM D93 / ASTM D 3828		
Especificação	Método	Expectativa °C	C = ponto de fulgor observado (°C)	Analista
		-	min. 38	
Análise 1	() ASTM D53 () ASTM D93			(01) (02) (03) (04)
Análise 2	() ASTM D53 () ASTM D93			(01) (02) (03) (04)
Análise 3	() ASTM D53 () ASTM D93			(01) (02) (03) (04)

Analista Responsável

Formulário de Análise de Amostra de Combustível Álcool

Produto: Álcool Etilico Hidratado Combustível	Identificação da Amostra	Data da Coleta:	Responsável pela Coleta:
	AH ____ - ____	Data de Entrada:	ANALISTAS: (01) — (02) —
Cliente:	Região:	Data da Análise:	

Aspecto e Cor					
Especificação	Aspecto (*)			Cor	Analista
	Limpido e isento de impurezas				
Análise 1	() 01	() 02	() 03	() 04	(01) (02) (03) (04)
Análise 2	() 01	() 02	() 03	() 04	(01) (02) (03) (04)

* (01): limpido e isento de impurezas; (02): limpido com impurezas; (03): turvo e isento de impurezas; (04): turvo com impurezas.

Massa específica (m.e.)				Método: NBR 5992/ASTM D 4052
Especificação	Método (*)	Massa específica, da amostra (g/cm ³)	Massa específica, da amostra (Kg/m ³)	Analista
	HV – NBR 5992 DD – ASTM D 4052	min. 0,8076 g/cm ³ máx. 0,8110 g/cm ³	min. 807,6 Kg/m ³ máx. 811,0 Kg/m ³	
Análise 1	() HV () DD			(01) (02) (03) (04)
Análise 2	() HV () DD			(01) (02) (03) (04)
Análise 3	() HV () DD			(01) (02) (03) (04)

* HV = Hidrômetro de Vidro; DD = Densímetro Digital (DMA 4500)

Teor de Álcool				Método: NBR 5902
Especificação	T (°C)	TA - 4NPM	Metodologia	Analista
	—	min. 92,6 máx. 93,8	DMA 4500 / Tabela	
Análise 1			() DMA 4500 () Tabela	(01) (02) (03) (04)
Análise 2			() DMA 4500 () Tabela	(01) (02) (03) (04)
Análise 2			() DMA 4500 () Tabela	(01) (02) (03) (04)

Teor de gasolina			
Especificação	a= aumento da camada aquosa (mL)	Teor (mL/L) V=2x a + 1	Analista
		máx 30 mL/L	
Análise 1			(01) (02) (03) (04)
Análise 2			(01) (02) (03) (04)

Potencial Hidrogeniônico (pH)				Método: NBR 10891
Especificação	Equipamento	min. 6,0	máx. 8,0	Analista
	Análise 1	() pH 01 () pH 02		
Análise 2	() pH 01 () pH 02			(01) (02) (03) (04)
Análise 2	() pH 01 () pH 02			(01) (02) (03) (04)

Condutividade Elétrica				Método NBR 10547 / ASTM D 1125
Especificação	Equipamento	máx. 500 µS/m	Analista	
	Análise 1	() Condutiv. 01 () Condutiv. 02		
Análise 2	() Condutiv. 01 () Condutiv. 02		(01) (02) (03) (04)	
Análise 2	() Condutiv. 01 () Condutiv. 02		(01) (02) (03) (04)	

Analista Responsável