

Universidade Federal do Rio de Janeiro

INTEROPERABILIDADE ENTRE AMBIENTES DE SIMULAÇÃO E PROJETO
DE PROCESSOS DA ENGENHARIA QUÍMICA

Érica Conceição Fernandes Domingos

2010



COPPE/UFRJ

INTEROPERABILIDADE ENTRE AMBIENTES DE SIMULAÇÃO E PROJETO
DE PROCESSOS DA ENGENHARIA QUÍMICA

Érica Conceição Fernandes Domingos

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Química, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Química.

Orientador: Argimiro Resende Secchi

Rio de Janeiro
Setembro de 2010

INTEROPERABILIDADE ENTRE AMBIENTES DE SIMULAÇÃO E PROJETO
DE PROCESSOS DA ENGENHARIA QUÍMICA

Érica Conceição Fernandes Domingos

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA QUÍMICA.

Examinada por:

Prof. Argimiro Resende Secchi, D.Sc.

Prof. Enrique Luis Lima, D.Sc.

Profa. Jonice de Oliveira Sampaio, D.Sc.

Profa. Ofélia de Queiroz Fernandes Araujo, Ph.D.

RIO DE JANEIRO, RJ - BRASIL
SETEMBRO DE 2010

Domingos, Érica Conceição Fernandes

Interoperabilidade entre Ambientes de Simulação e Projeto de Processos da Engenharia Química/ Érica Conceição Fernandes Domingos. – Rio de Janeiro: UFRJ/COPPE, 2010.

XV, 104 p.: il.; 29,7 cm.

Orientador: Argimiro Resende Secchi

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia Química, 2010.

Referencias Bibliográficas: p. 102-104.

1. Software CAE. 2. ISO-15926. 3. Simulador de Processo. 4. Ciclo de Vida. 5. Plantas de Processo. 6. Interoperabilidade. I. Secchi, Argimiro Resende. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Química. III. Título.

Dedico esta dissertação a meus pais Tânia Lúcia F. Domingos e Manoel Domingos
Neto.

AGRADECIMENTOS

Inicialmente agradeço a todos que de alguma forma tornaram este trabalho possível e contribuíram para o alcance deste objetivo.

Agradeço à Chemtech Serviços de Engenharia e Software por reconhecer a importância de ter um funcionário especializado e viabilizar com todo o apoio necessário o desenvolvimento deste trabalho.

Agradeço aos colegas de trabalho da Chemtech que dedicaram um pouco do seu tempo para discutir idéias e orientar as dificuldades de programação encontradas ao longo deste desenvolvimento: Alessandro de Lima Pereira, Mau Siu Ling e Rafael Silva Netto.

Agradeço em especial a contribuição de Cristiane São Bento Gonzaga para o desenvolvimento das simulações de processo do estudo de caso aplicado nesta dissertação.

Reconheço também como fundamental para direção deste trabalho e decisão de escolha do tema todo o incentivo e tempo dedicado por Geraldo Rochocz para discussão de idéias sobre ontologias e pelo fornecimento da norma ISO-15926, base deste trabalho.

Agradeço ao Grupo de Tecnologia em Computação Gráfica – Tecgraf da PUC-RIO pela troca de idéias e informações sobre a ISO-15926.

Agradeço ao apoio da Innotec do Brasil no fornecimento das licenças do software COMOS®, sem as quais nosso estudo de caso não seria possível.

Agradeço ao Orientador desta Dissertação, o Professor Dr. Argimiro Resende Secchi, em primeiro lugar por ter escutado minha idéia de mente e coração abertos, por ter compreendido a importância deste tema para mim e para meu desenvolvimento profissional e ter tido a sensibilidade e flexibilidade para conciliar os interesses do programa e do aluno. Em segundo, por ter me direcionado e orientado tecnicamente

ao longo desta trajetória, com toda sua experiência e bom senso. E por fim, por ter mais que comprado este idéia comigo.

Agradeço em especial a minha família pelo suporte emocional ao longo do desenvolvimento deste trabalho e a todos que demonstraram seu amor por mim em forma de compreensão, palavras de apoio, cobrança para que eu finalizasse minha meta e até mesmo silêncio nos momentos em que precisei estar completamente entregue a este trabalho.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

INTEROPERABILIDADE ENTRE AMBIENTES DE SIMULAÇÃO E PROJETO DE PROCESSOS DA ENGENHARIA QUÍMICA

Érica Conceição Fernandes Domingos

Setembro/2010

Orientador: Argimiro Resende Secchi

Programa: Engenharia Química

As ferramentas de software disponíveis no mercado para suporte aos procedimentos de trabalho da indústria de processos são caracterizadas por pacotes separados e independentes entre si. A competitividade de uma empresa nesta área está altamente relacionada com a forma como é feito o gerenciamento das informações, de modo a reduzir os custos com a inadequada interoperabilidade entre as diferentes fontes de informações. Isso tem provocado discussões internacionais sobre interoperabilidade que resultaram na iniciativa de criação de um padrão específico para plantas de processo, a ISO-15926. Este trabalho desenvolve um aplicativo baseado nesta norma para comunicação entre ferramentas de simulação de processos e software CAE, de modo a transferir as informações de engenharia de forma automatizada, ágil e consistente entre as fases de simulação e projeto básico. O aplicativo foi testado através de um estudo de caso aplicado à área de exploração e produção de petróleo para executar a transferência entre as informações de um simulador de processos genérico e uma ferramenta CAE. Os testes práticos provaram que o modelo de dados apresentado na norma é adequado para transcrever as informações da simulação de processos para softwares CAE e que a estratégia de construção do aplicativo de interface é eficiente para a interpretação dos modelos de dados das ferramentas de simulação e de CAE.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

INTEROPERABILITY BETWEEN ENVIRONMENT FOR PROCESS SIMULATION AND DESIGN OF CHEMICAL ENGINEERING

Érica Conceição Fernandes Domingos

September/2010

Advisor: Argimiro Resende Secchi

Department: Chemical Engineering

The software tools available in the market to support the work processes in the process industry are characterized by separated and independent packages. The company competitiveness in this area is highly related to data management way, towards minimize costs of inadequate interoperability between different systems. International discussions on interoperability resulted in the creation of a specific standard for process plants, ISO-15926. This work develops an application based on this standard for communication between process simulation tools and CAE software, to transfer engineering information on an automated, flexible and consistent way between simulation and basic design phases. The application has been tested through a case study applied to exploration and production of oil to perform the transfer of information between a generic process simulator and a CAE tool. Practical tests proved that the data model presented in the standard is suitable for transcribing the information from the process simulation to the CAE software and that the strategy to build the application interface is efficient for interpreting the simulation data model and the CAE tools.

SUMÁRIO

1	INTRODUÇÃO	1
2	REVISÃO BIBLIOGRÁFICA	4
2.1	FASES DO PROJETO DE ENGENHARIA NA INDÚSTRIA QUÍMICA	4
2.2	A SIMULAÇÃO DE PROCESSOS E O SOFTWARE EMSO	7
2.3	FERRAMENTAS CAE – HISTÓRICO E APLICAÇÃO	9
2.4	PADRONIZAÇÃO DOS FORMATOS DE INFORMAÇÃO E O PAPEL DA LINGUAGEM XML	14
2.5	ONTOLOGIAS	17
2.6	PADRÕES PARA INTERCÂMBIO DE DADOS ELETRÔNICOS NA INDÚSTRIA E O PAPEL DA ISO-15926	24
2.7	LINGUAGENS DE ONTOLOGIA E SINTAXE OWL	37
2.8	CONTEXTO DO TRABALHO	40
3	METODOLOGIA	50
3.1	INTRODUÇÃO	50
3.2	PREMISSAS PARA O DESENVOLVIMENTO	55
3.3	CONSTRUÇÃO DO DIAGRAMA UML	57
3.4	LEITURA DOS DADOS DO SIMULADOR	63
3.5	MAPEAMENTO DOS MODELOS DO SIMULADOR PARA AS CLASSES DA ISO-15926-4	71
3.6	GERAÇÃO DO ARQUIVO DE TRANSFERÊNCIA DE DADOS NO PADRÃO ISO-15926	75
3.7	MAPEAMENTO DOS MODELOS DA FERRAMENTA CAE PARA AS CLASSES DA ISO-15926-4	79
3.8	LEITURA DOS DADOS NO PADRÃO ISO-15926 PARA A FERRAMENTA CAE	84
4	ESTUDO DE CASO	86
5	CONCLUSÕES E PERSPECTIVAS	100
5.1	CONCLUSÕES	100
5.2	PERSPECTIVAS	101

6	REFERÊNCIAS BIBLIOGRÁFICAS	102
---	----------------------------------	-----

LISTA DE FIGURAS

Figura 1 – Fases do Ciclo de Vida de Unidades de Processo.....	6
Figura 2 – Impacto do Custo de Mudança nas Diferentes Fases do Ciclo de Vida.	7
Figura 3 – Modelo de Atividades para o Ciclo de Vida de Plantas de Processo.	30
Figura 4 - Arquitetura em 3 esquemas	31
Figura 5 – Tipos de Classes / Fonte: ISO-15926-1, First Edition. 2004-07-15.....	32
Figura 6 – Parte de um P&ID / Fonte: LEAL (2005).	34
Figura 7 – Identificação e Classificação das Relações / Fonte: LEAL (2005).....	35
Figura 8 – Relações de Composição / Fonte: LEAL (2005).	35
Figura 9 – Relação de Conexão / Fonte: LEAL (2005).	36
Figura 10 – Automatismos desenvolvidos para integração entre ferramentas de <i>software</i> em projetos de engenharia.	44
Figura 11 – Utilização de um modelo de ontologia central, neutro e padronizado para intercâmbio de dados.	45
Figura 12 - Fluxo de trabalho em um projeto de engenharia de processos.	46
Figura 13 – Integração entre ambiente de simulação e projeto via aplicativo ORION.	49
Figura 14 – Atividades empregadas no desenvolvimento do trabalho.	52
Figura 15 – Etapas da Construção do ORION.	53
Figura 16 – Exemplo de Estrutura ISO-15926-2 e 4. Fonte (GULLA et al., 2006).	60
Figura 17 – Classe de Equipamentos.	61
Figura 18 – Diagrama UML – Classes ISO-15926-4 para Simulação de Processos....	63
Figura 19 – Componentes do Diagrama de Processo. Fonte (FERNANDES, 2009). ..	65
Figura 20 – Arquitetura de Objetos no EMSO. Fonte (FERNANDES, 2009).	66
Figura 21 – Parâmetros e Variáveis do Processo no EMSO. Fonte (Fernandes, 2009).	67
Figura 22 – Extração do Arquivo XML de saída do EMSO.	70
Figura 23 – Modelo Conceitual ISO-15926-2.	71
Figura 24 – Exemplo da aplicação dos conceitos da ISO-15926-2.	72
Figura 25 – Interface COMOS®.....	80

Figura 26 – Módulo de Processo do COMOS®.	81
Figura 27 – Equipamentos da Biblioteca de Objetos de Processo.	82
Figura 28 – <i>Link</i> do Componente do PFD com o modelo do Objeto na Biblioteca.	83
Figura 29 – Catálogo de atributos dos objetos de engenharia.	84
Figura 30 - Processo de Separação de Óleo – Compressão de Gás.....	87
Figura 31 - Definição dos Componentes no EMSO.....	89
Figura 32 - Definição dos modelos das fases líquida e vapor no EMSO.	90
Figura 33 – Conversão XML do EMSO para XML Padrão do ORION.	92
Figura 34 – Criação dos Equipamentos do Processo.	93
Figura 35 – Conectores das Correntes de Processo.....	94
Figura 36 – Conexão Automática no PFD.....	95
Figura 37 - Fluxograma de Processo no COMOS	96
Figura 38 – Importação Dados de Simulação do VF_01 para o software CAE.	97
Figura 39 – Importação dos Componentes das Correntes no Software CAE.	98
Figura 40 – Importação das Frações Molares dos Componentes.....	98

LISTA DE TABELAS

Tabela 1 – Tabela de conexões do arquivo de saída do EMSO.	68
Tabela 2 - Tabela de componentes das correntes do arquivo de saída do EMSO.	68
Tabela 3 – Tabela de Resultados por Device do arquivo de saída do EMSO.	69
Tabela 4 – RDLs ISO-15926-4 (ISO-15926-4, 2007)	73
Tabela 5 – Mapeamento entre os termos do EMSO e a Classe <i>Property</i>	74
Tabela 6 - Mapeamento entre os termos do EMSO e a Classe <i>UOM</i>	74
Tabela 7 - Mapeamento entre COMOS® e RDLs da ISO-15926-4	82
Tabela 8 - Prefixos definidos para representação do fluxograma da unidade de reforço de compressão de gás.	88
Tabela 9 – Resultado da Simulação para o Vaso VF_01.	97

LISTA DE SIGLAS E ABREVIATURAS

2-D	Bi-dimensional
3-D	Tri-dimensional
AEC	<i>Architecture, Engineering, and Construction</i>
AI	<i>Artificial Intelligence</i>
ALSOC	Ambiente Livre para Simulação, Otimização e Controle de Processos
AP	<i>Application Protocol</i>
API	<i>Application Programming Interface</i>
ASCII	<i>American Standard Code for Information Interchange</i>
CAD	<i>Computer-Aided Design</i>
CAE	<i>Computer-Aided Engineering</i>
CAM	<i>Computer-Aided Manufacturing</i>
CAPE	<i>Computer-Aided Process Engineering</i>
CIS/2	<i>CIMSteel Integration Standards/Version 2</i>
CLIP	<i>Conceptual Lifecycle Process</i>
DOLCE	<i>Descriptive Ontology for Linguistic and Cognitive Engineering</i>
DOM	<i>Document Object Model</i>
DTD	<i>Document Type Definition</i>
EML	<i>EMSO Model Library</i>
EMSO	<i>Environment for Modeling, Simulation and Optimization</i>
FD	Folha de Dados
FEED	<i>Front End Engineering Design</i>
HTML	<i>Hypertext Markup Language</i>
IAI	<i>International Alliance for Interoperability</i>
IFC	<i>Industry Foundation Classes</i>
IGES	<i>Initial Graphics Exchange Specification</i>
ISO	<i>International Organization for Standardization</i>
NIST	<i>National Institute of Standards and Technology</i>
NURBS	<i>Non-Uniform Rational B-Spline</i>
OntoCAPE	<i>Ontology Computer-Aided Process Engineering</i>
OWL	<i>Ontology Web Language</i>
P&ID	<i>Piping and Instrumentation diagram</i>
PFD	<i>Process Flow Diagram</i>
RDF	<i>Resource Description Framework</i>
RDL	<i>Reference Data Library</i>
SAX	<i>Simple API for XML</i>

SGML	<i>Standard Generalized Markup Language</i>
STEP	<i>STandard for the Exchange of Product Model Data</i>
SUMO	<i>Suggested Upper Merged Ontology</i>
SW	<i>Semantic Web</i>
UML	<i>Unified Modeling Language</i>
VB	<i>Visual Basic</i>
W3C	<i>World Wide Web Consortium</i>
WfMC	<i>Workflow Management Coalition</i>
XML	<i>eXtensible Markup Language</i>

1 INTRODUÇÃO

A engenharia química é hoje uma ciência considerada como estabelecida para área de pesquisa, que se desenvolveu e amadureceu ao longo do século passado. Nos últimos 50 anos, tem sido notória a complementação das bases fundamentais da engenharia química – química, física e mecânica – pelas operações matemáticas e ciências de computação na área de pesquisa (SCHNEIDER & MARQUARDT, 2002). Entretanto, manter-se competitivo nesta indústria frente à crescente globalização dos mercados é um dos principais desafios deste século e demanda um grande investimento para melhoria dos procedimentos de trabalho¹ atuais aplicados à indústria química.

As ferramentas de suporte atuais são caracterizadas principalmente por pacotes de *software* separados ou fracamente ligados. Dentro destas ferramentas, quantidades crescentes de dados, documentos, e outros tipos de informações são manipulados. Estes diferentes pacotes de informação têm de ser geridos, uma vez que são valiosos recursos de conhecimento. Enquanto as informações são criadas e tratadas em ferramentas diferentes, muitas vezes há dependências e sobreposições entre elas (BAYER & MARQUARDT, 2004).

O mercado atual de processos químicos demonstra forte preocupação em melhorar sua capacidade de competir através da forma pela qual seus dados são transformados em informação, utilizados, gerenciados, transmitidos e armazenados. A garantia deste objetivo está fortemente relacionada aos avanços nos sistemas de informação, que segundo o estudo *Technology Vision 2020* (1996) força a indústria a alcançar as seguintes metas:

- Incentivar os fornecedores de tecnologia a desenvolver sistemas abertos, com foco na capacidade de integrar conjuntos específicos de informações em sistemas maiores. Isso exigirá melhorias em segurança de dados, qualidade e confiabilidade, assim como em tecnologias de compressão de dados.

¹ Conjunto de ações conduzidas por executantes, internos ou externos, da organização para produzir resultados desejados.

- Incentivar o desenvolvimento de normas para transferência entre modelos de dados e parâmetros dos modelos para facilitar o emprego da tecnologia da informação na área de modelagem e simulação de processos.
- Capitalizar-se em tecnologia da informação, trabalhando com instituições acadêmicas e empresas de *software* para garantir a compatibilidade e integrar as ferramentas computacionais utilizadas pela indústria química. Desenvolver parcerias para o compartilhamento de informações sobre técnicas de automação e de modelagem avançada.
- Incentivar o desenvolvimento de sistemas especializados e ferramentas inteligentes de apoio à decisão que sejam flexíveis o suficiente para o modelo multinacional, empresas multi-produto para uso na tomada de decisão em negócios.

Dentro deste contexto e frente à importância do tema para a indústria de processos e desenvolvimentos futuros da engenharia química, o trabalho aqui apresentado busca desenvolver e avaliar uma metodologia para transferência de dados de ambientes de simulação para as fases seguintes de projeto de modo a transformá-los em informações consolidadas com o padrão ISO-15926 para reutilização automatizada e eficiente.

Os avanços tecnológicos de *hardware* e *software* impulsionaram recentes desenvolvimentos de ferramentas eficazes para fins específicos dentro dos procedimentos de trabalho dos projetos de plantas de processos químicos. Entretanto, a demanda atual é pela integração destes ambientes de *software*, que está totalmente a cargo dos usuários finais destas aplicações: proprietários de plantas e empresas de engenharia. Após uma série de investimentos e desperdício de tecnologias para alcançar a tão sonhada interoperabilidade, o mercado atual chegou à conclusão de que este objetivo somente poderá ser atingido através do desenvolvimento e aplicação de normas para transferência de informações de engenharia. O Capítulo 2 apresenta os procedimentos de trabalho aplicados à área de projetos de plantas de processo e o histórico de evolução das ferramentas de *software* disponíveis para suportá-los, bem como os esforços para o desenvolvimento de normas aplicáveis à padronização das informações de engenharia e sua transferência ao longo do ciclo de vida das plantas.

A possibilidade de desenvolver mecanismos eficazes, reutilizáveis e economicamente atrativos para transferência de informações de engenharia para indústria química leva este trabalho mais especificamente aos seguintes objetivos:

- Geração de um aplicativo de *software* para prover comunicação entre diferentes modelos de simulação de processos e ferramentas de desenvolvimento de projetos de plantas industriais;
- Avaliar o potencial do padrão ISO-15926 para representar e transferir as informações da fase de simulação de processos de forma consistente, coerente e livre de ambigüidades;
- Compreender e utilizar os conceitos da ISO-15926 para o desenvolvimento do aplicativo de comunicação neutra entre as diferentes ferramentas de *software*;
- Apresentar um estudo de caso prático para avaliação do aplicativo desenvolvido neste trabalho;
- Avaliar a capacidade dos modelos de *software* aplicados ao estudo de caso em disponibilizar seus dados no formato de informações úteis e reutilizáveis;
- Demonstrar a importância do desenvolvimento de padrões para transferência de informação entre os diferentes modelos de *software* na engenharia química;
- Demonstrar os benefícios da interoperabilidade aplicada aos procedimentos de trabalho da área de projetos de plantas químicas.

O Capítulo 3 apresenta a metodologia utilizada para o desenvolvimento do trabalho, o capítulo seguinte descreve um estudo de caso para geração dos resultados práticos desta aplicação e o Capítulo 5 conclui este estudo e apresenta as perspectivas de trabalhos futuros para continuidade do tema.

2 REVISÃO BIBLIOGRÁFICA

2.1 FASES DO PROJETO DE ENGENHARIA NA INDÚSTRIA QUÍMICA

O projeto de uma planta de processo industrial objetiva a produção de produtos químicos principalmente em larga escala. Um projeto pode ser executado para construção de uma nova unidade, modificação ou expansão de uma unidade de processo existente. Uma planta química cria um novo material via processos de transformação química ou biológica, ou simplesmente atua na separação de materiais. Dentre as plantas de processo destacam-se as farmacêuticas, alimentícias, geração de energia, produção de petróleo e gás, refino de petróleo e tratamento de água.

Tipicamente menos de 1% dos projetos de plantas químicas torna-se comercial, devido à quantidade de restrições que impactam esta área. A avaliação da viabilidade de uma planta de processo inclui fatores como custo de capital, espaço disponível, impactos ambientais, aspectos de segurança, produção de resíduos e custos de operação e manutenção. Em sua maioria, o estudo de reações químicas ou desenvolvimento de novas rotas industriais é feito em pequena escala nos laboratórios e os resultados obtidos são analisados por engenheiros para produção em maior escala. Ainda assim, antes da construção da unidade com capacidade industrial, são construídas unidades menores chamadas de planta piloto para obtenção do projeto e informações de operação (DOUGLAS & JAMES, 1998).

Um projeto de engenharia para plantas de processo envolve uma série de atividades multidisciplinares e uma equipe de especialistas distribuídos em diversas disciplinas como processo, mecânica, tubulação, arranjo e arquitetura, elétrica, instrumentação e automação, segurança, entre outros. A troca de informações entre os membros das equipes é intensa e um grande volume de informação é gerado em formato de documentos, que são considerados as entregas materializadas do projeto. Os processos do projeto envolvem todas as atividades relacionadas entre os membros das equipes durante o desenvolvimento dos trabalhos e até mesmo de outras companhias envolvidas no projeto. A multidisciplinaridade destas tarefas torna um projeto de construção de unidades químicas um empreendimento bastante complexo e

que necessita ser estruturado em fases, que compõem o chamado Ciclo de Vida do Projeto.

SCHNEIDER & MARQUARDT (2002) definem que o termo ciclo de vida tem origem na biologia e é definido como: Uma série de estágios através dos quais alguma coisa (indivíduo, cultura ou produto produzido) passa ao longo de sua vida. O autor destaca também que o conceito de ciclo de vida tem sido usado por alguns anos na engenharia de *software* e recentemente foi adotado por muitas empresas de engenharia e fornecedores de *software* para definir as várias fases de uma planta química. O ciclo de vida de uma planta pode ser dividido em 6 fases principais: projeto conceitual, engenharia básica, engenharia de detalhamento, construção e comissionamento, assim como gerenciamento de ativos, manutenção e *revamps*. O projeto conceitual e FEED (*Front End Engineering Design*), fase intermediária e opcional entre o projeto básico e o detalhamento, constituem a parte do ciclo de vida de impacto mais significativa no custo (MARQUARDT & NAGL, 2004). 80% do custo de capital são determinados nestas duas primeiras fases, onde as principais decisões que impactam fases futuras do empreendimento são tomadas (NAGL *et al.*, 2003). Nas fases iniciais, praticamente todas as decisões conceituais são definidas como seleção de processos, reações envolvidas, separação de materiais e transferência de energia, seleção dos tipos de unidades de processo (reatores químicos, unidades de destilação, troca de calor, etc.), suas interconexões, seus parâmetros de otimização e a obtenção dos objetivos do processo. Estes objetivos são geralmente a obtenção do produto com uso mínimo de taxa de material e energia, além de mínima geração de resíduos (TAYLOR, 2007). Os resultados destas fases formam a base para as fases subsequentes, engenharia básica e engenharia de detalhamento.

Uma unidade industrial tem um ciclo de vida de 50 a 60 anos, conforme ilustrado na **Figura 1**, sendo a duração da fase de projeto até a partida da planta de 2 a 5 anos. Isso se torna significativo porque normalmente 30 a 40% do total de custos do ciclo de vida de uma instalação ocorrem nestas duas primeiras fases e 60-70% na terceira fase (em dólares constantes, i.e., sem considerar inflações e deflações) (NIST, 2004).

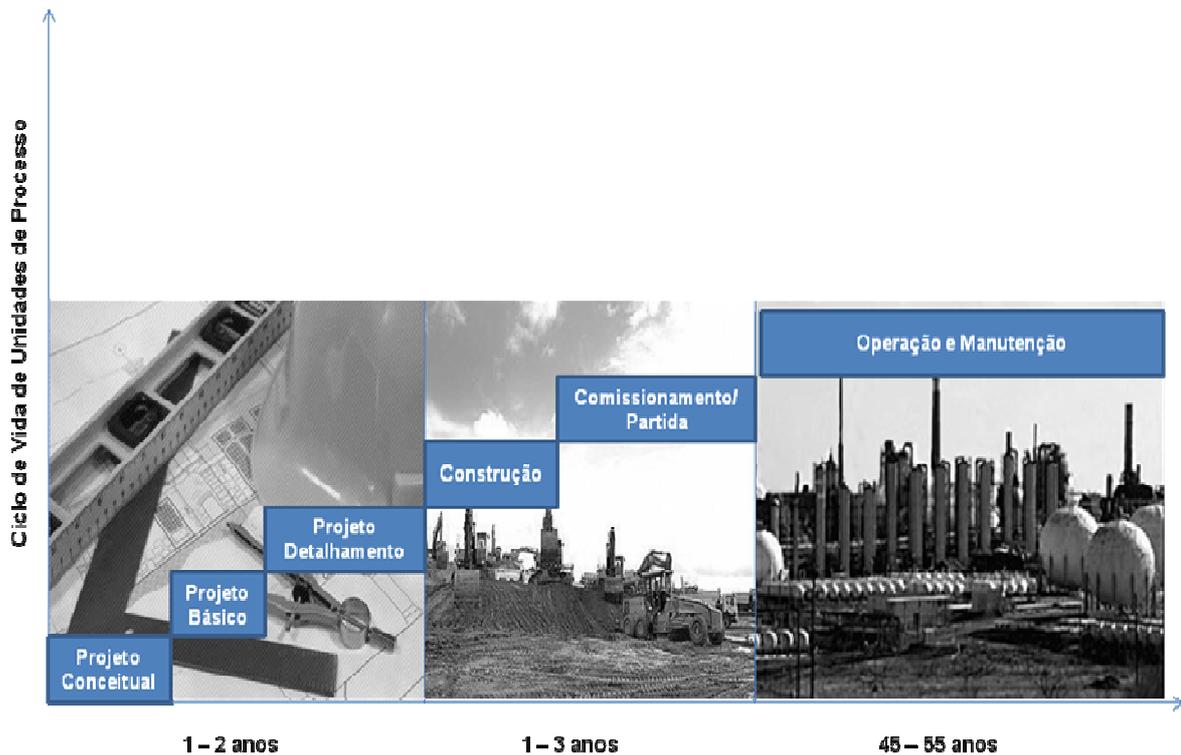


Figura 1 – Fases do Ciclo de Vida de Unidades de Processo

A **Figura 2** apresenta o impacto do custo da alteração de um projeto em várias fases do ciclo de vida do serviço. Um exemplo clássico desse impacto ocorre quando os aspectos de operação e manutenção de um tipo de aquecimento e ventilação não são considerados na fase de projeto. O projetista pode ter avaliado se o local onde o sistema de aquecimento e ventilação está projetado se encaixa fisicamente em termos de limitações de espaço e equipamentos existentes. Apesar disso, quando instalados os operadores das instalações podem identificar que, embora o equipamento esteja na sala, não seja possível a manutenção adequada devido às folgas limitadas entre o equipamento e as paredes da sala, o que exige um deslocamento físico de uma ou mais paredes. Este tipo de problema é muito caro para corrigir depois de a construção estar completa; a resolução deste conflito, antes da construção é mais eficiente e envolve menos interrupções (NIST, 2004).

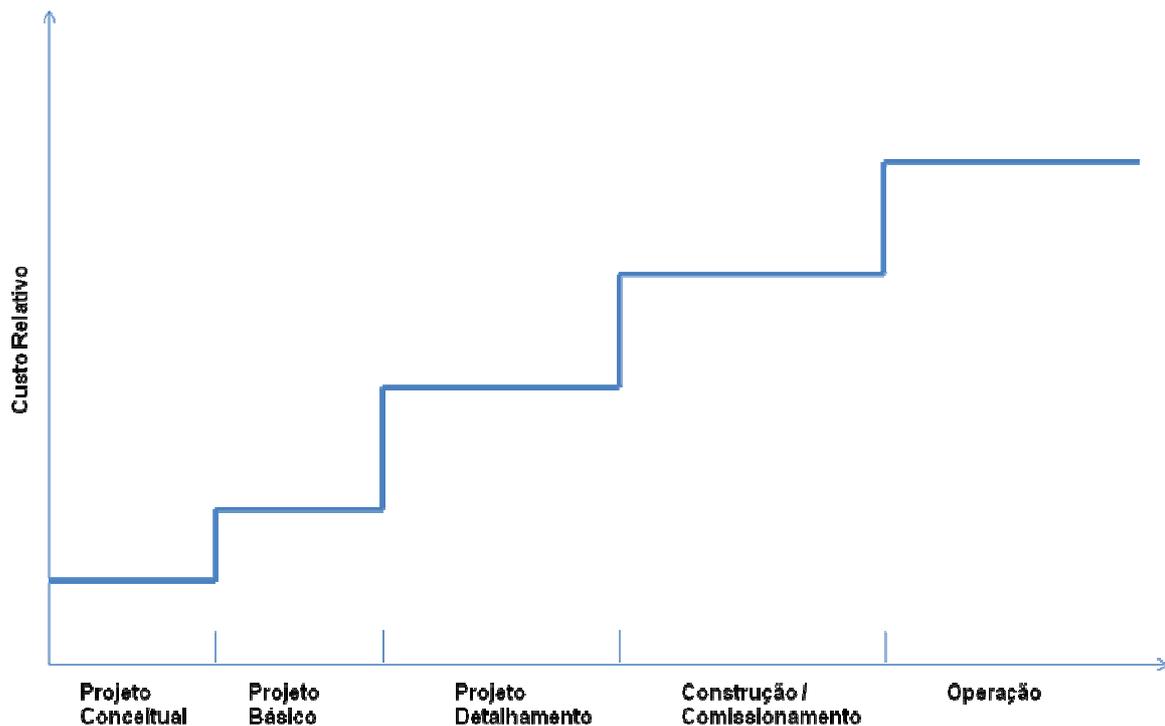


Figura 2 – Impacto do Custo de Mudança nas Diferentes Fases do Ciclo de Vida.

2.2 A SIMULAÇÃO DE PROCESSOS E O SOFTWARE EMSO

Existe uma grande variedade de razões para utilização de *softwares* de simulação de processos. Em muitos casos, a simulação auxilia na tomada de decisão, na redução dos riscos e na gestão estratégica, em níveis tático e operacional. Seis categorias de atuação dos *softwares* de simulação de processos estão listadas abaixo:

- Gestão estratégica;
- Planejamento;
- Controle e gestão operacional;
- Melhoria de processos e adoção de tecnologia;
- Compreensão e treinamento.

Quando se desenvolve um *software* de modelagem e simulação de processos, identificando os objetivos e as questões ou problemas de gerenciamento que devem

ser abordados, é fundamental a definição do escopo do modelo e dos dados que precisam ser coletados. Os simuladores de processo têm como proposta atender uma parte do ciclo de vida da planta (uma fase de projeto ou uma inspeção), o projeto de um novo produto ou a otimização de um processo existente. O resultado da simulação são informações necessárias para responder às questões fundamentais definidas na finalidade do modelo. Os resultados típicos de uma simulação de processos são listados a seguir (KELLNER *et al.*, 1999):

- Esforço/custo;
- Ciclo de tempo (duração);
- Nível de defeito;
- Taxa de utilização/desgaste;
- Requerimentos de desgaste ao longo do tempo;
- Custo/benefício, retorno do investimento ou outras medidas econômicas;
- Produtividade;
- Medição de atrasos.

Ao longo das últimas décadas, a indústria de *software* tem apresentado uma grande variedade de soluções para área de simulação de processos. Entretanto, a principal dificuldade é atender as exigências dos usuários deste mercado que buscam o alinhamento entre qualidade, velocidade de processamento e custo baixo.

O desenvolvimento do simulador de processos EMSO (*Environment for Modeling, Simulation and Optimization*) foi iniciado em 2001 com o objetivo de criar um simulador genérico para processos dinâmicos que atendesse às necessidades indisponíveis no mercado, principalmente no que se refere à complexidade das soluções.

O EMSO é um ambiente gráfico que permite a modelagem de processos complexos em regime transiente ou estacionário, utilizando a construção de blocos de modelos conectados entre si. A ferramenta também permite desenvolver novos modelos utilizando a linguagem de modelagem própria do simulador ou utilizar os modelos já existentes a partir do *EMSO Model Library* (EML) (SOARES, 2007).

A EML é uma biblioteca aberta de modelos escrita na linguagem de modelagem do EMSO. Esta linguagem é orientada a objeto e tem como chave o conceito de reutilização para o tratamento de complexidades. A utilização da EML para simulação de processos dinâmicos contempla simultaneamente dois níveis: baseada em equações e baseada em blocos. A Linguagem apresenta três entidades principais: *Models*, *Devices* e *Flowsheets*. Um *Flowsheet* é uma abstração do problema em estudo que é composto por um conjunto de componentes denominados *Devices*. Cada *Device* é baseado em uma descrição matemática, a qual rege o comportamento de suas variáveis. Esta descrição é chamada *Model* (SOARES, 2003).

Atualmente o *software* EMSO tem seu desenvolvimento continuado pelo projeto ALSOC (**A**mbiente **L**ivre para **S**imulação, **O**timização e **C**ontrol de **P**rocessos), uma iniciativa para aproximação entre universidade e indústria através da padronização e distribuição sem custo de especificações e ferramentas de *software*. O Projeto tem o objetivo de desenvolver, manter e distribuir especificações de uma linguagem de modelagem e uma biblioteca de modelos abertos para síntese, simulação, otimização e controle de processos em geral, certificar a conformidade de soluções externas com os padrões desenvolvidos e adicionar ao Projeto contribuições externas.

As características citadas acima, aliadas ao fato do simulador EMSO estar sendo desenvolvido pelo mesmo grupo de pesquisa deste trabalho, nortearam a escolha deste simulador para a avaliação da metodologia proposta.

2.3 FERRAMENTAS CAE – HISTÓRICO E APLICAÇÃO

Segundo o estudo realizado pelo NIST – *National Institute of Standards and Technology* – (2004) sobre o custo da inadequada interoperabilidade entre as ferramentas aplicadas à indústria, o maior volume de ferramentas de *software* está concentrado na primeira parte do ciclo de vida de uma planta de processo correspondente ao planejamento, engenharia e projeto. Nas três últimas décadas, tem havido uma tendência para substituir a correspondência em papel pelo correio eletrônico e também para introduzir planilhas computacionais e *softwares* em aplicações na fase inicial de planejamento e apoio orçamental para controle de custos. Além disso, tecnologias têm sido adotadas para tornar as funções de apoio às empresas mais eficientes.

No início dos anos 70, os profissionais da área de projeto utilizavam programas computacionais que operavam em computadores *mainframe*. Com a introdução de pequenos computadores pessoais e a utilização de mídias ópticas e magnéticas para distribuição da informação, os programas foram adaptados ao *hardware* e *software* mais comumente utilizados pela indústria no desenvolvimento de projetos. A especificação do desenvolvimento destes programas foi projetada para manipular e ser compatível com informação de construção, base de dados de custos e bibliotecas de especificação de produto. Este foi um grande passo em direção à modernização da comunicação no setor da construção. Isso permitiu o suporte da computação aos esforços de projeto e especificar o que deve ser construído e como.

Até o início dos anos 1980, alguns profissionais da área de projetos e engenheiros projetavam e tomavam decisões sobre as unidades de produção utilizando ferramentas computacionais para auxílio aos projetos e desenhos. Quando aprovados pela primeira vez, os pacotes CAD (*Computer-Aided Design*) foram usados para substituir as tarefas que normalmente eram feitas à mão e em papel (geralmente um desenho). Mas estes sistemas não eram inicialmente um investimento rentável porque eles eram caros, difíceis de aprender e usar, e não melhoraram significativamente a produtividade. Além disso, os primeiros sistemas CAD rodavam em grandes *mainframes* e sistemas de *hardware* dedicado.

Ao longo do tempo, no entanto, dados os requisitos das partes interessadas e da inovação dos pacotes de *software* pôde-se observar uma convergência para tornar o uso do CAD e, posteriormente, engenharia auxiliada por computador (CAE) e engenharia auxiliada por manufatura (CAM) mais eficientes e economicamente viáveis. Estes sistemas de *software* tornaram-se mais baratas, mais fáceis de usar, e tinham aplicações mais sofisticadas. Por exemplo, em vez de apenas exibições em 2-D imaginadas 20 anos atrás, agora existem *softwares* que permitem visualizar gráficos 3-D dos projetos. Deve-se notar, contudo, que há uma diferença fundamental entre sistemas de desenho que realmente não são sistemas de projeto, mas apenas uma forma de capturar os resultados do projeto em termos de exibições 2-D, e sistemas de projeto dos quais exibições em 2-D e 3-D podem ser extraídas. Estes sistemas de desenho não podem nativamente detectar conflitos, falta de componentes, conexões incompatíveis, as incoerências entre desenhos, configurações fisicamente impossíveis, e muitos outros erros que assolam a área de projeto. Sistemas de

modelagem podem executar este tipo de detecção e é por isso que eles estão substituindo os sistemas de desenho em projetos de unidades complexas.

A revolução na computação nos anos de 1980 e 1990 permitiu aos usuários executar esses aplicativos a partir de suas estações de trabalho usando *desktops*. Embora ainda existam questões a serem abordadas, a relação entre a transferência de dados de vários sistemas e interoperabilidade para questões relacionadas à interação com o cliente, o *software* CAD é amplamente aceito.

Durante a década de 1990, a Internet tornou-se muito popular e apresentou um novo meio de comunicação para trocar e visualizar informações. Da mesma forma, Intranets se tornaram uma ferramenta largamente utilizada pelas empresas para trabalharem juntos em projetos de seus próprios computadores. A tecnologia baseada em internet é aplicada para facilitar o intercâmbio de informação e compartilhamento de recursos entre as equipes de projeto, pois partes internas e externas podem se comunicar e compartilhar dados de forma mais rápida e eficaz.

O desenvolvimento de esforços para padronizar o formato das informações começou décadas antes de a *Web* se tornar popular, iniciando em um nível inferior com a padronização do ASCII e continuando em médio e alto nível como IGES, STEP, IFC e CIS / 2, que foram expulsos pela disponibilidade de múltiplos sistemas de informação incompatível. Os benefícios do intercâmbio eletrônico de dados estão empurrando a demanda por formatos melhores de compartilhamento de dados. Até o momento, não há opção clara e direta para formatos de arquivos neutros¹², apesar de várias organizações e institutos de normatização estarem investindo nestes desenvolvimentos. Atualmente, é necessário para as equipes de projeto discutir no início do projeto quais formatos serão usados e testar as conversões antes do projeto estar totalmente em curso. Mesmo com formatos que proporcionam troca de informações úteis, testes preliminares são essenciais para garantir interoperabilidade adequada.

² Arquivos digitais formatados para leitura e interpretação por qualquer aplicativo de software independente da arquitetura do modelo de dados, linguagem de programação ou regras semânticas empregadas ao sistema.

Atualmente, existem sérias deficiências no que diz respeito ao uso de ferramentas de *software* para auxílio ao desenvolvimento de projetos de plantas de processo. Algumas considerações importantes estão listadas a seguir (MARQUARDT & NAGL, 2004):

- As ferramentas estão determinando de forma significativa as práticas de projeto, porque têm sido amplamente impulsionadas pela tecnologia e não por uma situação do mercado no passado. As funcionalidades das ferramentas têm sido limitadas pela tecnologia, muitas vezes impedindo o atendimento das exigências do projeto. Normalmente, são ferramentas que oferecem funcionalidades de apoio para apenas uma parte das atividades de projeto ou um conjunto de tarefas do projeto.
- Existe uma integração limitada entre as ferramentas amplamente concentrada sobre *softwares* de um único fornecedor ou de seus parceiros/colaboradores. A integração do legado de ferramentas em tal ambiente ou a integração da infraestrutura de *software* de uma empresa é cara.
- A heterogeneidade do ambiente de *software* impede a cooperação entre as organizações.
- Os dados de projeto são representados de forma diferente nos vários instrumentos. Não são apenas técnicas, mas também sintáticas e semânticas as falhas que impedem a integração.
- Há uma falta de gestão das relações entre os dados e os documentos produzidos por diferentes ferramentas em diferentes atividades do projeto.
- Os *softwares* de gerenciamento e administração de projetos não são todos integrados com *softwares* de suporte aos projetos de engenharia. Assim, é difícil obter um bom planejamento e controle do projeto de processos.
- A integração entre as ferramentas é amplamente realizada por transferência de dados ou pela integração de dados via armazenamento central, negligenciando os requisitos dos procedimentos de trabalho.
- A comunicação na equipe de projeto só é suportada por ferramentas genéricas como o e-mail, vídeo-conferência, etc., que não estão integradas com as ferramentas de projeto de engenharia.

- A gestão dos processos de concepção criativa não é suportada por meio de ferramentas de domínio específico.

A maioria das ferramentas de aplicação para engenharia química foram desenvolvidas para fins específicos, chegando a um elevado nível de maturidade. Portanto, uma melhoria reconhecível dos procedimentos de trabalho só pode ser alcançada por meio da integração destes aplicativos existentes em um ambiente combinado com serviços comuns, como gestão de documentos, acesso a bases de dados comuns, ou o apoio aos procedimentos de trabalho. Durante os últimos anos, vários ambientes de *software* proprietários foram desenvolvidos para projetos de engenharia química como *Aspen Zyqad* ou *COMOS PT* da *COMOS Industry Solutions*. Nestas abordagens comerciais, principalmente as ferramentas de um fornecedor estão estreitamente interligadas; extensões com novas ferramentas e adaptações às particularidades dos procedimentos de trabalho dentro de uma empresa específica raramente são suportadas. Um completo entendimento do domínio da aplicação é necessário para o desenvolvimento do projeto aberto e ambientes flexíveis que permitam a integração dos instrumentos existentes e prestação de serviços e funcionalidades de suporte central. As ferramentas, as informações tratadas no âmbito destas ferramentas, e os procedimentos de trabalho que utilizam essas informações precisam ser entendidos juntamente com suas interdependências (BAYER & MARQUARDT, 2004).

WASSERMAN (1990) define quatro dimensões para integração de *softwares* de engenharia:

- Integração de dados: refere-se ao compartilhamento de dados e o gerenciamento das relações entre eles;
- Integração de controle: notificação (avisos de alerta) de ferramentas sobre certos eventos e ativação de ferramentas para responder a alguma solicitação de serviço;
- Plataforma de integração: facilita a execução de um sistema integrado de um conjunto de ferramentas em um ambiente heterogêneo e distribuído em rede de computadores;
- Apresentação da integração: tem como objetivo promover a interface homem-máquina de um conjunto de ferramentas integradas com uma visão comum.

2.4 PADRONIZAÇÃO DOS FORMATOS DE INFORMAÇÃO E O PAPEL DA LINGUAGEM XML

A integração entre os sistemas existentes e novas aplicações de *software* representa o atual desafio para os desenvolvedores de sistemas, visto a complexidade do trabalho que envolve formatos de dados e interfaces que não atendem aos requisitos atuais e o uso de ferramentas de *software* escritas em diferentes linguagens, para diferentes sistemas operacionais, redes e *hardwares*.

A necessidade de criação de padrões que pudessem especificar uma forma de compartilhar informações entre diferentes ambientes de aplicações deu origem à linguagem XML (*eXtensible Markup Language*) como uma ferramenta para viabilizar este intercâmbio eletrônicos independente da plataforma.

O desenvolvimento da linguagem XML começou em 1996 e foi derivado do SGML (Desenvolvido no início de 1980) e HTML (desenvolvido em 1990). Originalmente concebido para responder aos desafios da publicação eletrônica em grande escala, agora desempenha um papel na troca de uma ampla variedade de dados. O XML é atualmente a linguagem aceita para comunicação de dados pela Internet. O padrão utiliza tags para se comunicar com um computador, assim como para criar e definir os elementos dentro de um conjunto de dados e interpretar o conteúdo dos documentos eletrônicos transferidos. Os desenvolvedores do XML criaram um conjunto de diretrizes e convenções para projetar formatos de texto para estrutura de dados XML de modo a tornar fácil para os computadores gerar a leitura de dados, e garantir que a estrutura de dados seja inequívoca. O XML pode ser usado para armazenar qualquer tipo de informação estruturada e envolver ou sintetizar informações para processá-las entre diferentes sistemas de computação que seriam incapazes de se comunicar, além de ser gratuito (NIST, 2004).

A XML é uma linguagem de marcação para descrever e estruturar informações, com base em dados e metadados. Os metadados são informações que atribuem significado aos dados. A criação da marcação pode ser determinada de acordo com necessidades específicas, uma vez que as marcações não são pré-definidas na linguagem, tornando o padrão bastante flexível. O XML é desde 1998 uma

recomendação do W3C (*World Wide Web Consortium*) tratada como uma especificação pública. Dentre os principais objetivos originais da linguagem XML pode-se citar (BRAY, 2008):

- Deve ser diretamente utilizável pela Internet;
- Deve suportar uma ampla variedade de aplicações;
- Deve ser compatível com SGML;
- Deve ser fácil escrever programas que processam documentos XML;
- O número de características opcionais em XML deve ser mantido ao mínimo absoluto, idealmente zero;
- Documentos XML devem ser humano-legíveis e razoavelmente claros;
- Um projeto XML deve ser preparado rapidamente;
- Um projeto de XML deve ser formal e conciso;
- Documentos XML devem ser fáceis de criar;
- Concisão na marcação em XML é de importância mínima.

Segundo SOBRAL (2001), um elemento XML localiza-se entre uma marcação de início e uma marcação de fim, incluindo a própria marcação. Os elementos possuem correspondência com outros elementos, podendo ter uma relação de pai ou filho, formando assim uma estrutura de árvore. Um elemento pode ter como conteúdo outro elemento, além de conteúdos misturados (contém texto e outros elementos), conteúdo simples (texto) ou ainda conteúdo vazio (não contém informação). Os documentos XML geralmente possuem correspondência com uma tabela em uma base de dados. Os elementos XML podem conter atributos. Um atributo armazena informação adicional sobre os elementos, informação que não faz parte dos dados do elemento.

O Autor ainda ressalta que um documento XML considerado válido é um documento bem formatado e que está em conformidade com as regras de uma DTD (*Document Type Definition*). O propósito de um DTD é definir a estrutura do documento com uma lista de elementos possíveis. Com o uso de uma definição de documento, cada

arquivo XML pode carregar uma descrição do seu próprio formato. Deste modo, grupos independentes podem concordar em usar um DTD comum para a troca de informação. Uma aplicação pode usar um DTD para verificar se os dados que recebeu são válidos.

Um módulo de *software* capaz de ler documentos e fornecer acesso a seu conteúdo e estrutura é chamado de *XML Parser* e a interface de programação, incluindo os nomes dos métodos e atributos é uma API XML. Um desenvolvedor pode escrever uma API que pode ser executada em diferentes ambientes sem maiores modificações. Existem várias interfaces de programação (APIs) desenvolvidas ou em desenvolvimento para a manipulação de XML, porém, as duas principais especificações que estão se tornando padrões neste setor são: SAX e DOM. A XML DOM (*Document Object Model*) é uma API abstrata para documentos XML. Ela define uma maneira pela qual um documento XML pode ser acessado e manipulado. Como uma especificação da W3C, ela fornece uma API padrão para uma variedade de aplicações. DOM foi projetado para ser utilizado com qualquer linguagem de programação ou sistema operacional. SAX (*Simple API for XML*) não requer que um documento completo esteja na memória para processá-lo, pois o acesso ao conteúdo do documento é feito de forma seqüencial. Diferentemente do DOM, onde o arquivo é percorrido seguindo uma estrutura de árvore.

Um exemplo de XML aplicada à indústria é “aecXML” que é uma linguagem baseada em XML para representar Informações das áreas de Arquitetura, Engenharia e Construção (AEC). A iniciativa aecXML, originada pela *Bentley Systems*, é agora gerida pela Aliança Internacional para a Interoperabilidade (IAI). O aecXML visa estabelecer definições comuns do esquema, usando casos de negócios bem definidos, para os dados AEC através do padrão de formatação de linguagem XML. O aecXML destina-se a apoios específicos de transações pela Internet. Essas operações podem estar associadas com a transferência de recursos, como documentos de projeto, informações de materiais, peças e contato. O aecXML tem o potencial para permitir uma maior eficiência para as atividades, tais como propostas, projeto, estimativas, planejamento e construção (NIST, 2004).

2.5 ONTOLOGIAS

O termo ontologia, em grego *ontos* e *logoi* – “conhecimento do ser”, teve origem na filosofia quando muito tempo atrás Aristóteles e seus alunos o utilizaram em seus trabalhos como a ciência que estuda a natureza dos seres enquanto eles existem no universo. Posteriormente, este termo tornou-se um jargão bastante conhecido na área da ciência da computação para representação do conhecimento pela definição de conceitos dentro de domínios e relações entre estes conceitos.

No início dos anos 90, as ontologias tornaram-se um tópico de pesquisa bastante popular investigado por diversas comunidades no campo da Inteligência Artificial (AI), incluindo engenharia do conhecimento, processamento de linguagem natural e representação do conhecimento (STUDER *et al.*, 1998).

Dentre as muitas definições para ontologia apresentadas na literatura, destaca-se na área de engenharia de *software* a citada por GRUBER (1993): “Ontologia é uma especificação explícita de uma conceitualização”, onde o termo conceitualização se refere a um modelo abstrato de algum fenômeno no mundo pela identificação de conceitos relevantes deste fenômeno e que sejam compartilhados por todos.

GUARINO (1998) discutiu em seu artigo esta definição e apresentou uma idéia mais clara e ampla do termo, definindo uma ontologia como uma teoria lógica que corresponde ao significado intencional de um vocabulário formal, ou seja, um comprometimento ontológico com uma conceitualização específica do mundo. Em outras palavras, no contexto da engenharia do conhecimento uma ontologia é um entendimento compartilhado de um domínio, com um vocabulário comum de termos e relações.

Os componentes que constituem uma ontologia são:

- Indivíduos: instâncias ou objetos;
- Classes: conjuntos, coleções, conceitos, categorias de programação, tipos de objetos, ou tipos de coisas;

- Atributos: aspectos, propriedades, características ou parâmetros que os objetos (e classes) podem ter;
- Relações: formas em que as classes e os indivíduos podem estar relacionados um ao outro;
- Função: estruturas complexas formadas a partir de certas relações que podem ser usadas no lugar de um termo individual em uma sentença;
- Restrições: descrições formalmente declaradas que devem ser verdadeiras para que alguma afirmação seja aceita como entrada;
- Regras: declarações sob a forma de sentença IF-THEN (antecedente-conseqüente) que descreve as deduções lógicas que podem ser extraídas de uma afirmação em uma forma particular;
- Axiomas: declarações (incluindo as regras) de uma forma lógica que, juntas, compõem a teoria geral que uma ontologia descreve em seu domínio de aplicação. Esta definição difere das axiomas na gramática geral e da lógica formal. Nestas disciplinas, axiomas incluem apenas descrições afirmativas como um conhecimento a priori. Como usado aqui, axiomas também incluem a teoria derivada das declarações axiomática;
- Eventos: a mudança de atributos ou relações.

USCHOLD & GRUNINGER (1996) apresentaram o uso de ontologias para diminuir ou eliminar confusões conceituais e terminológicas e tornar o entendimento compartilhado, sendo utilizadas como base para:

- Comunicação entre pessoas com diferentes necessidades e pontos de vista sobre um determinado contexto;
- Interoperabilidade entre sistemas encontrada através de uma tradução entre diferentes modelagens, métodos, paradigmas, linguagens e ferramentas de *software*;
- Engenharia de sistemas para especificação, reutilização e confiabilidade.

O estudo realizado pelo NIST – *National Institute of Standards and Technology* – (2004) estima que \$158 bilhões são gastos por ano nos Estados Unidos devido à

inadequada interoperabilidade entre as ferramentas utilizadas em projetos e operações industriais. Muitas aplicações de ontologias são utilizadas para solucionar questões de interoperabilidade, promovendo o intercâmbio de dados entre diferentes usuários ou diferentes ferramentas de *software*. Por exemplo, diferentes ferramentas podem processar diferentes ontologias para definição de um mesmo domínio e, por alguma necessidade organizacional, precisam estar integradas. Neste caso, é necessário ter uma ontologia comum para que as diferentes ferramentas de *software* possam usar os dados. Este tem sido o maior desafio para a aplicação de ontologias, uma vez que não é possível impor um requerimento de integração entre as ferramentas que já estão em uso no mercado (USCHOLD & GRUNINGER, 1996).

Quando uma ontologia é projetada, alguns critérios de projeto precisam ser considerados. Para construção de uma ontologia que objetiva compartilhar conhecimento e promover interoperabilidade entre sistemas com base em uma conceitualização compartilhada, GRUBER (1993) considera os seguintes aspectos:

- Clareza: uma ontologia deve comunicar eficazmente o significado pretendido de termos definidos e ser objetiva;
- Coerência: Uma ontologia deve ser coerente. Pelo menos, os axiomas definidos devem ser logicamente consistentes. Se uma sentença que pode ser inferida a partir dos axiomas contradiz uma definição ou exemplo dado informalmente, então a ontologia é incoerente;
- Extensibilidade: Uma ontologia deve ser projetada para antecipar os usos do vocabulário comum. Deve oferecer uma base conceitual para uma série de tarefas previstas, e a representação deve ser trabalhada para que se possa ampliar e especializar as ontologias monotonicamente;
- Minimizar viés de codificação: A conceitualização deve ser especificada no nível de conhecimento sem depender de uma codificação de nível de símbolo particular. O viés de codificação deve ser minimizado, porque os agentes de compartilhamento de conhecimento podem ser aplicados em diferentes representações, sistemas e estilos de representação;
- Mínimo compromisso ontológico: uma ontologia deve requerer um compromisso ontológico mínimo e suficiente para suportar o compartilhamento

de conhecimentos destinados às atividades. Desde que o compromisso ontológico é baseado na utilização de um vocabulário consistente, o compromisso ontológico pode ser minimizado definindo apenas os termos que são essenciais para a comunicação do conhecimento compatível com a teoria.

Ontologias podem ser desenvolvidas usando abordagens *top-down* ou *bottom-up*. A abordagem *bottom-up* começa com os conceitos mais específicos em um domínio da aplicação. A abordagem *bottom-up* resulta em ontologias difíceis de modificar e integrar com outras ontologias desenvolvidas para outros domínios ou aplicações. A abordagem *top-down* começa com conceitos de alto nível que são assumidos para serem comuns a muitas áreas de aplicação. A abordagem *top-down* facilita a integração de aplicações com ontologias que são mais fáceis de manter. Infelizmente, os engenheiros utilizando a abordagem *top-down* são suscetíveis à imposição de categorias de alto nível arbitrárias que são prescritivas, não cumprindo os requerimentos dos usuários. Este problema pode ser evitado com uma ontologia superior. Uma ontologia superior define classes de nível superior, tais como objetos físicos, atividades, relações mereológica³ e topológicas das quais classes mais específicas e as relações podem ser definidas. Exemplos de ontologias superiores são SUMO, SOWA, DOLCE, CLIP e ISO 15926-2 (BATRES *et al.*, 2007).

BAYER & MARQUARDT (2004) apresentaram em seu artigo o modelo de dados CLIP (*Conceptual Lifecycle Process*) como uma ontologia superior desenvolvida no centro de pesquisa IMPROVE por Marquardt em 1998, cujo principal objetivo é a reutilização do conhecimento, das informações e estratégias de engenharia definidas em projetos anteriores para otimização dos procedimentos de trabalho. O modelo é uma solução integrada para Engenharia Química de modo a solucionar questões como: Dependência entre modelos de informação e modelos de procedimentos de trabalho; relação entre dados e documentos; integração de modelos de dados existentes.

Os autores explicam que o conhecimento pode ser definido como o fato ou a condição de saber alguma coisa com a familiaridade adquirida através da experiência ou da associação. Assim, o conhecimento está diretamente associado a uma pessoa que

³Teoria ou estudo lógico-matemático das relações entre as partes e o todo e das relações entre as partes no interior de um todo.

tem o conhecimento e que para ser transferido precisa ser transformado em informação. Uma pessoa pode explicar o seu conhecimento para torná-lo acessível para os outros; esta explicação corresponde à transformação do conhecimento em informação. A informação é caracterizada por seu conteúdo e contexto. O conteúdo da informação pode ser codificado como dados em ferramentas de *software* e armazenados em bases de dados. Exemplos de dados no domínio da engenharia química são o tamanho de uma planta ou equipamento (por exemplo, volume, diâmetro), as condições de funcionamento de uma etapa do processo (por exemplo, pressão, temperatura) ou as propriedades físicas de um composto químico (densidade, temperatura de ponto de ebulição). Os dados podem ser agregados aos documentos, como relatórios, arquivos ou fluxogramas. Os dados e documentos não podem ser analisados e descritos completamente sem considerar os procedimentos de trabalho onde são criados e usados. Para o desenvolvimento da estrutura de um *software* específico, não só o conhecimento sobre as informações é necessário, mas também o conhecimento sobre o fluxo de trabalho e as atividades. Visto tal complexidade o autor conclui que não é possível definir um modelo de dado detalhado que atenda potencialmente todas as propostas e sugere o uso de um *framework* que contenha conceitos básicos do domínio e a descrição com base em ontologia, servindo como vocabulário comum para um consenso sobre o entendimento. Os motivos que contribuem para este cenário são:

- Crescente complexidade e mudanças dinâmicas no domínio da Engenharia Química;
- Não é possível determinar todas as informações, procedimentos de trabalho e suas relações para todos os projetos e situações em um modelo de informação.

Tem sido crescente o número de ontologias desenvolvidas para o domínio da engenharia, tais como *PhysSys* para modelagem de sistemas físicos genéricos, *EngMath* que formula os conceitos matemáticos fundamentais para modelagem em engenharia e *YMIR* para representar conhecimentos de projetos de engenharia, além de trabalhos envolvendo o desenvolvimento de uma ontologia de nível superior para o domínio da engenharia química baseada no padrão ISO-15296 (ISO 2003), que pode ser usado para representar conhecimento sobre riscos e estudos de operacionalidade de fábricas de produtos químicos. O *OntoCAPE* é uma ontologia formal pensada para o domínio da engenharia de processos químicos. Neste domínio, o projeto, a

construção e a operação de plantas de produtos químicos são considerados como as principais atividades de engenharia. O campo de investigação para a promoção destas aplicações é conhecido como *Computer-Aided Process Engineering* (CAPE). OntoCAPE é o desenvolvimento de uma ontologia para CAPE, inicialmente desenvolvida no projeto *COGents*, que explora uma arquitetura baseada em agentes para simulação numérica de processos químicos. Seu desenvolvimento tem sido explorado pelo projeto IMPROVE, que tem o foco em novos conceitos e soluções de *software* de engenharia para suporte às atividades de projetos de engenharia (MORBACH & MARQUARDT, 2007).

BATRES et al. (2002) aplicaram os conceitos de ontologia para desenvolver a representação de um modelo de simulação que suporte ao intercâmbio de dados em diferentes níveis de detalhe. Utilizando a linguagem de modelagem *Modelica* eles representaram as equações dos modelos aplicados ao formalismo em que plantas, processos e produtos são representados como objetos relacionados em termos estruturais, comportamentais e operacionais. A ontologia trata os fenômenos físicos que se manifestam em mudanças nas propriedades do material, uma vez que o mesmo é processado na planta. Esta definição implica a existência de um modelo físico de comportamento que pode ser definido de forma independente de onde o fenômeno modelado ocorra.

Ontologias também podem desempenhar um papel na padronização de representações entre as ferramentas. As normas desempenham as mesmas funções para a interoperabilidade e de uma compreensão compartilhada. Alguns projetos estão sendo desenvolvidos para fornecer algum tipo de padrão em diferentes domínios de aplicação (USCHOLD & GRUNINGER, 1996):

- *Workflow Management Coalition* (WfMC)
- STEP e EXPRESS
- CORBA
- KIF

A fim de compreender a contribuição das normas para interoperabilidade é necessária uma definição inicial. A interoperabilidade é definida pelo IEEE como: "A habilidade de

dois ou mais sistemas trocarem informações e utilizarem estas informações trocadas". Duas questões distintas precisam ser abordadas quando se tenta aplicar esta definição para a noção de interoperabilidade de domínio. A primeira é a noção de sistemas ou componentes que trocam e a utilizam a informação. A segunda é a noção de uso da informação, porque as diferentes interpretações podem ter em diferentes contextos. Neste cenário, os padrões para interoperabilidade entre sistemas de computação podem ser divididos nas seguintes categorias (STEGWEE ET AL., 2003):

- Método: Uma maneira comum de pensar, trabalhar, e modelar durante o desenvolvimento ou utilização de um produto. Exemplo: Como faço para definir uma interface de comunicação entre dois sistemas.
- Meta-modelo: A descrição genérica do domínio, para ser utilizado em projetos que se baseiam em um método escolhido. Exemplo: Quais as funções genéricas podem ser distinguidas em uma arquitetura de comunicação por computador.
- Modelo Concreto: Uma descrição específica das interações e dados a serem trocados, tendo as relações entre cada um numa realidade específica. Exemplo: Quais são os dados que deve ser trocados para continuidade do fluxo de trabalho.
- Padrão Operacional: Especificação detalhada das interações e dados a serem trocados, que podem ser usados sem mais detalhes ou interpretação dos links de comunicação entre computadores. Exemplo: Como proceder para trocar dados entre diferentes partes e seus sistemas.

O *workflow* pode ser definido como a automação dos processos de negócio, como um todo ou parte, durante o qual os documentos, informações ou tarefas são transferidas de um participante para outro por ações em concordância com regras e procedimentos (QIU, *et al.*, 2008). O fluxo de trabalho dentro de uma organização, em geral inclui o objetivo do trabalho, as entradas e saídas de dados, restrições adicionais, decomposição do trabalho em unidades menores, chamadas de atividades (GERHARD *et al.*, 2003). Uma vez que os documentos, informações e tarefas são executadas em diversos pacotes de software específicos para cada área de conhecimento do projeto, não é possível obter *workflow* sem que haja interoperabilidade entre os sistemas. A especificação de fluxo de dados tem um papel fundamental para viabilizar este processo e deve expressar quais dados são criados e acessados através das atividades (GERHARD *et al.*, 2003).

2.6 PADRÕES PARA INTERCÂMBIO DE DADOS ELETRÔNICOS NA INDÚSTRIA E O PAPEL DA ISO-15926

O estudo realizado pelo NIST (2004) relata que diversos segmentos da indústria têm trabalhado para melhorar a capacidade de compartilhamento de dados durante os últimos 20 anos. Dentre estes esforços estão incluídos:

- *Interim Graphics Exchange Specification*: Em 1980, a organização *Interim Graphics Exchange Specification* (IGES) foi formada. Este foi o primeiro esforço que reconheceu a necessidade de intercâmbio de dados definido como um produto e não apenas um CAD. A IGES permite o intercâmbio de dados do produto de diferentes sistemas CAD / CAM.
- *Standard for the Exchange of Product Model Data*: Em meados dos anos 1980, o setor industrial criou a necessidade de uma norma para o Intercâmbio de Dados de Modelo de Produto (STEP - *Standard for the Exchange of Product Model Data*). Foi o primeiro esforço que reconheceu a necessidade de padronizar as representações dos dados do produto antes de expressá-las em um padrão de sintaxe de intercâmbio e formato através de protocolos de aplicação. O STEP, como parte do corpo de normas ISO, é um esforço mundial para desenvolver um mecanismo de intercâmbio e compartilhamento de dados de engenharia. O STEP trabalha para definições neutras de dados industriais, representação, e linguagem que suportem as funções do ciclo de vida de uma planta industrial. O uso de um formato de troca comum ajuda a reduzir os custos de tradução e a melhorar a qualidade de toda a utilização dos dados. O STEP permite compartilhamento de dados entre aplicações de *software* ao longo do ciclo de vida do produto, entre diferentes organizações envolvidas neste ciclo de vida, e fisicamente dispersas dentro de uma organização.
- *Industry Foundation Classes* (IFC): O IFC, em desenvolvimento pelo IAI desde 1998, foi projetado para fornecer uma solução completa, aprofundada, e precisa de construção de um modelo de dados para uma aplicação computacional utilizada entre participantes sem perda de informação. IFC são elementos que representam as partes ou elementos de um processo para uma determinada instalação e contém as informações relevantes sobre essas partes. Aplicações computacionais usam o IFC para montar um modelo de

computador de leitura óptica que constitui um banco de dados orientado a objetos. Este banco de dados pode ser compartilhado entre os participantes do projeto e continuar a crescer conforme o projeto continua através da concepção, construção e entrada em operação. O Conselho Europeu de Engenheiros Civis estima que o uso do IFC pode reduzir os fatores de risco para os operários da instalação em até 20% para edifícios novos e até 50% para outras estruturas.

- *CIMSteel Integration Standards/Version 2 (CIS/2)*: CIS/2 é um protocolo através do qual os programas *stand-alone*, como análise estrutural, CAD e sistemas de detalhamento, podem se comunicar. Ao fornecer um formato de dados neutro, o CIS/2 permite intercâmbio de dados entre uma grande variedade de tipos de programas. CIS/2 é um modelo lógico de produto e formato eletrônico de intercâmbio de dados para informações de projetos estruturais de aço. CIS/2 foi implementado em muitos projetos siderúrgicos, análise de engenharia, fabricação, construção e aplicações para criar um perfeito fluxo de informações integradas entre todas as partes da cadeia de suprimentos do aço e envolvidos na construção de estruturas de aço.

O STEP é descrito em diferentes partes da norma ISO-10303 e tem sido a tecnologia chave para o intercâmbio de dados dentro de muitas indústrias de grande porte por um longo período de tempo. A definição dos objetos, suas relações para outros objetos e suas restrições são definidas na linguagem EXPRESS (ISO-10303 parte 11). Isso significa que uma modelagem de dados muito poderosa foi desenvolvida em meados dos anos 80 anterior ao UML e XML. Foi destinada a ser flexível, prorrogável e em linguagem de modelagem escalável, fácil de ser lida por especialistas humanos. No entanto, o sucesso em muitas indústrias foi limitado a uma quantidade muito pequena de desenvolvedores que estão familiarizados com ele (BEETZ *et al.*, 2005).

GIELINGH (2008) cita em seu artigo que não existe arquivo neutro para intercâmbio de dados sem erros, e dá três exemplos de estudos realizados com o objetivo de evidenciar estas falhas e promover a melhoria dos padrões e normas disponíveis até o momento:

- Em 1994, um investimento europeu em P&D deu origem ao projeto PISA, que planejou uma demonstração do intercâmbio de dados geométricos entre dois

aplicativos CAE, usando tradutores comerciais para STEP AP203. O aplicativo de envio produziu um modelo de superfície com representação *Non-Uniform Rational B-Spline* (NURBS), que foi corretamente escrito em um arquivo físico STEP. O tradutor da aplicação recebida converteu-o em uma superfície de mosaico. A segunda aplicação receptora manteve a representação NURBS, mas mudou de pontos críticos de controle, provocando descontinuidades da superfície. Uma perda grave do projeto original ocorreu em ambos os casos. Surpreendentemente, verificou-se que todas as aplicações envolvidas aplicaram o padrão corretamente. AP203 permite a troca de várias representações geométricas em diferentes níveis de precisão. A norma não impede uma mudança de representação por cada pedido, nem uma mudança de precisão. Tais mudanças ocorrem como resultado do mapeamento entre o modelo neutro e o esquema interno do aplicativo. Na verdade, elas podem ocorrer duas vezes por troca unidirecional: uma vez da aplicação A para o modelo neutro, e uma vez a partir do modelo neutro para aplicação B. Esta experiência motivou a introdução de Classes de Conformidade no STEP.

- A equipe liderada por Robert Amor da Universidade de *Auckland* investigou recentemente o desempenho de tradutores IFC para três diferentes sistemas CAE. Neste teste relativamente simples, um arquivo IFC foi lido para o aplicativo e, posteriormente, reescrito como um arquivo IFC. Nenhuma mudança foi feita para o modelo. Os arquivos de entrada e saída foram então comparados. Nos três casos, diferenças significativas entre os arquivos foram encontradas: algumas entidades desapareceram, outras apareceram e, novamente, outros foram alterados. As três aplicações alteraram os dados do IFC de diferentes formas.
- Apesar de 13 anos de experiência prática com STEP AP203, o intercâmbio de dados ainda é preocupante. A empresa alemã *Prostep*, que atende o setor automotivo, com soluções em tecnologias de dados de produto, oferece a seus clientes a solução OpenDESC. Este *software* inclui conversão bi-direcional dos sistemas CATIA, IDEAS, e Unigraphics Pro/ENGINEER. Uma tubulação está disponível para cada combinação de sistemas (por exemplo, CATIA - Unigraphics). Embora os dados originais sejam convertidos usando o padrão interfaces (STEP AP214 e IGES), os erros inevitáveis da conversão são corrigidos por um conjunto de ferramentas de adaptação para cada tubulação.

Além disso, *Prostep* recomenda um conjunto de melhores práticas" para usuários de sistemas CAD minimizarem os erros.

A possibilidade de utilizar o STEP para indústria de processos foi reconhecida em 1990, e consórcios industriais foram formados na Europa, Estados Unidos e Japão para promover seu uso. Os Estados Unidos focaram em informações espaciais sobre planta de processo e o consórcio *US PlantSTEP* financiou o desenvolvimento da peça ISO-10303 227 "configuração espacial da planta", conhecida como AP (*Application Protocol*) 227. A Europa focou na informação funcional sobre as plantas de processo, e o consórcio europeu *EPISTLE* financiou o desenvolvimento da AP 221 "dados funcionais de planta de processo e sua representação esquemática", em paralelo com AP 227. Esta norma abrange os esquemas como P&IDs (*Piping and Instrumentation diagrams*) e PFDs (*Process Flow Diagrams*), e os comportamentos que os esquemas de engenharia representam. O trabalho sobre AP 221 encontrou dificuldades técnicas porque o consórcio *EPISTLE* exige um padrão que possa gravar as mudanças ocorridas em uma instalação de processo ao longo de sua vida. O objetivo do *EPISTLE* é definir um padrão para armazenar dados de instalações de um processo que contenha informações sobre:

- Os requisitos para uma planta de processo, e as alterações dos requisitos;
- O projeto para uma planta de processo e mudanças no projeto;
- Os objetos físicos que existem em uma planta de processo e alterações destes objetos físicos.

Estes objetivos estão fora do âmbito do STEP. Como resultado, a ISO-15926 foi desenvolvida como um padrão complementar ao STEP. Ambos STEP e ISO-15926 são produtos do mesmo comitê ISO TC184/SC4. A AP 221 continua disponível, mas não considera a evolução de uma planta de processo através do tempo (LEAL, 2005).

A ISO-15926, *Norma de Automação Industrial e Integração de Sistemas no ciclo de vida de plantas de processo – incluindo as unidades de produção na área de Óleo e Gás*, é um padrão internacional para representação do ciclo de vida da informação de plantas de processo incluindo instalações de produção de óleo e gás, criado para facilitar a integração dos dados no suporte às atividades e processos de produção ao ciclo de vida da planta. O padrão, composto de 7 partes, especifica um modelo de

dados que define o significado do ciclo de vida da informação em um contexto único suportando todas as visões que engenheiros de processo, engenheiros de equipamentos, operadores, engenheiros de manutenção e outras especialidades podem ter a respeito da planta. Tradicionalmente, os dados associados a uma planta de processo têm-se concentrado em alguma visão individual da planta em um ponto no tempo. Esses dados são normalmente definidos e mantidos de forma independente de outros grupos de usuários, resultando em duplicação e dados conflitantes que não podem ser compartilhados dentro de uma empresa ou com parceiros de negócios de uma empresa (ISO 15926-1, 2004).

O desenvolvimento da norma foi iniciado em 1991 por um grupo de pesquisa chamado *ProcessBase* na Europa com o objetivo de criar um modelo de dados para o ciclo de vida da informação que atendesse às necessidades da indústria de processos. O grupo formado por um consórcio de empresas tomou como ponto de partida o padrão STEP (YOGUI, 2009).

A partir de 1997 o Consórcio POSC Caesar criou seus trabalhos com o padrão ISO-15926 e desde então preside o grupo responsável pela ISO TC184/SC4 e ISO-15926. A *POSC Caesar Association* é uma organização sem fins lucrativos, que promove o desenvolvimento de especificações abertas para serem utilizadas como padrões para permitir a interoperabilidade de dados, *software* e assuntos relacionados à engenharia. Sua responsabilidade principal é a manutenção e reforço do Padrão ISO-15926, uma vez que as partes 6 e 7 estão em progresso. A POSC Caesar funciona como uma organização global de normalização, em estreita colaboração com outras organizações de normalização na Europa e nos Estados Unidos. Entre seus membros, patrocinadores e colaboradores estão universidades e institutos de pesquisa, companhias produtoras de óleo e gás, empresas projetistas e consultoras em engenharia e fornecedores de soluções em *software* (POSC CAESAR, 2010).

A ISO-15926 está organizada em sete partes publicadas separadamente. A ISO-15926-1 apresenta uma visão geral e princípios fundamentais. A ISO-15926-2 contém o modelo de dados conceitual; Na ISO-15926-3 é definida a ontologia; A ISO-15926-4 apresenta a referência inicial de dados; A Parte 5 foi substituída pelo padrão “*ISO Maintenance Agency*” para manutenção dos dados de referência; Na ISO-15926-6 é discutida uma metodologia para o desenvolvimento e validação dos Dados de

Referência e a ISO-15926-7 apresenta Métodos de Implementação para integração de sistemas distribuídos (YOGUI, 2009).

A seguir estão apresentadas as 7 partes da ISO-15926:

Parte 1 – Visão Geral e Princípios Fundamentais: Especifica uma representação de informações relacionadas com a engenharia, construção e operação de plantas de processo. Esta representação suporta os requisitos de informação das indústrias de processos em todas as fases do ciclo de vida de uma planta de e no compartilhamento e integração de informações entre todas as partes envolvidas no ciclo de vida da planta. Descrevem os princípios fundamentais que são a base da ISO-15926, a relação da ISO-15926 com outras normas para dados industriais e as definições dos termos utilizados ao longo ISO-15926.

Parte 2 – Modelo de Dados: Especifica um modelo de dados conceitual genérico para representação técnica das informações de plantas de processo, projetado para ser usado em conjunto com dados de referência: instâncias padrão que representam informações comuns a um número de usuários, plantas de processo, ou ambos.

Parte 3 – Ontologia: Trata a informação gráfica (geométricas) como os modelos CAD 2D/3D (*Computer Aided Design*) e topológicas provenientes de sistemas de informação geográfica (GIS). Para isto o recurso utilizado é o *Reference Data Class* (Classe de Dados de Referência) (YOGUI, 2009).

Parte 4 – Dados de Referência Iniciais: Define o conjunto inicial de dados de referência para uso com os padrões de dados industriais ISO-15926 e ISO-10303-221– definição dos dados de referência são planilhas em Excel disponíveis on-line.

Parte 5 - ISO Maintenance Agency: ISO TC184/SC4 começou uma iniciativa para desenvolver dados de referência para manutenção que servirá como fonte para ISO-15926.

Parte 6 – Metodologia para o desenvolvimento e validação dos Dados de Referência: Em desenvolvimento.

Parte 7 – Métodos de Implementação para Integração de Sistemas Distribuídos:
Definir e testar metodologias de implementação. Em desenvolvimento.

Segundo as definições da ISO-15926-1, o modelo de atividades para o ciclo de vida de uma planta de processo está ilustrado na Figura 3.

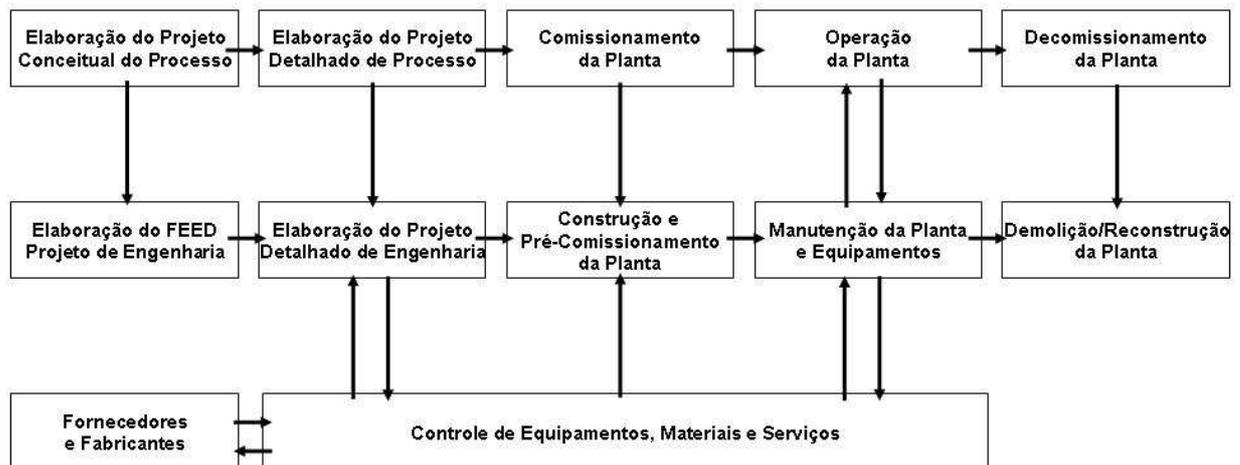


Figura 3 – Modelo de Atividades para o Ciclo de Vida de Plantas de Processo.

Fonte: ISO-15926-1, First Edition. 2004-07-15.

O suporte para uma atividade específica do ciclo de vida depende do uso adequado dos dados de referência em conjunto com o modelo de dados definido na norma ISO-15926-2.

O modelo conceitual de dados especificado na ISO-15926-2 é descrito conforme a arquitetura de três esquemas da ISO/TR 9007, exemplificada na Figura 4. O modelo exclui todas as regras de negócio referentes às aplicações específicas para permitir integração das informações e dar flexibilidade ao modelo em se adequar às mudanças de negócios. Esta arquitetura identifica três tipos de modelo de dados (ISO-15926-1, 2004):

- Modelo Externo: a estrutura de dados corresponde a uma visão para uma finalidade específica, que inclui regras sobre os dados que sejam adequados à finalidade particular.

- Modelo de dados conceitual: um modelo neutro, que é capaz de suportar qualquer ponto de vista válido que caia no seu âmbito. Este modelo só podem incluir regras para dados que são universalmente verdadeiras em todo o seu escopo.
- Modelo físico: a definição da forma como os dados são armazenados. Os tipos de dados refletem as coisas que são importantes para o armazenamento e acesso e não o significado dos dados para um negócio específico.

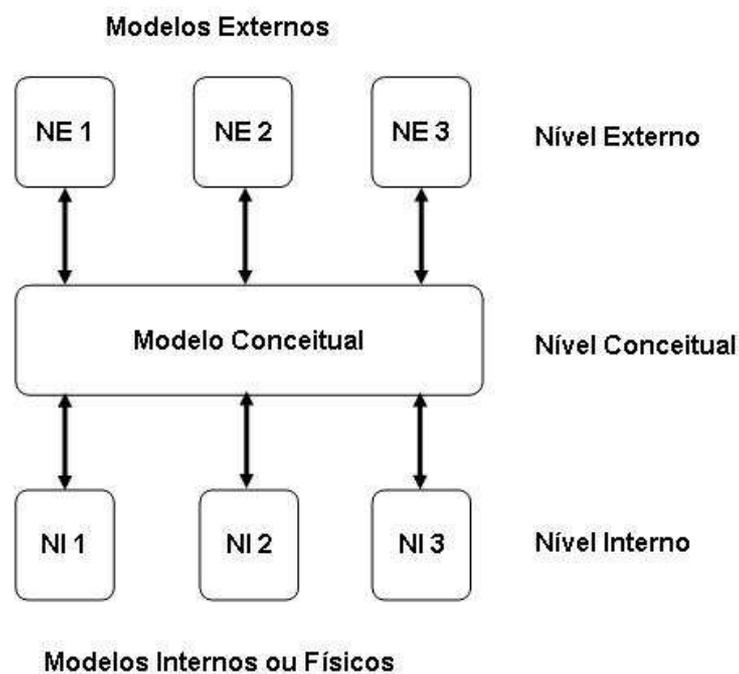


Figura 4 - Arquitetura em 3 esquemas

Fonte: ISO-15926-1, First Edition. 2004-07-15.

A Figura 5 ilustra a dependência desta norma com a utilização dos dados de referência compartilhados. Dados sobre uma planta de processo individual podem ser compartilhados e trocados somente se o remetente e o receptor utilizarem os mesmos dados de referência ou utilizarem referência de dados compartilhada. Estes dados de referência devem ser suficientes para garantir uma comunicação clara entre as partes. O modelo de dados especificado na ISO-15926-2 suporta o intercâmbio de dados, mas não fornece um significado específico de dados suficientes para permitir comunicação inequívoca.

Os Dados de Referência estão divididos em:

- Instâncias que representam referências dos indivíduos.
- Instâncias que representam referências das classes.

A última divisão dos Dados de Referência é subdividida nas seguintes categorias:

- Classes fundamentais;
- Classes proprietárias;
- Classes *defacto*;
- Classes padrões;
- Classes de produtos manufaturados;
- Classes de produtos commodity;
- Classes de produto proprietários.

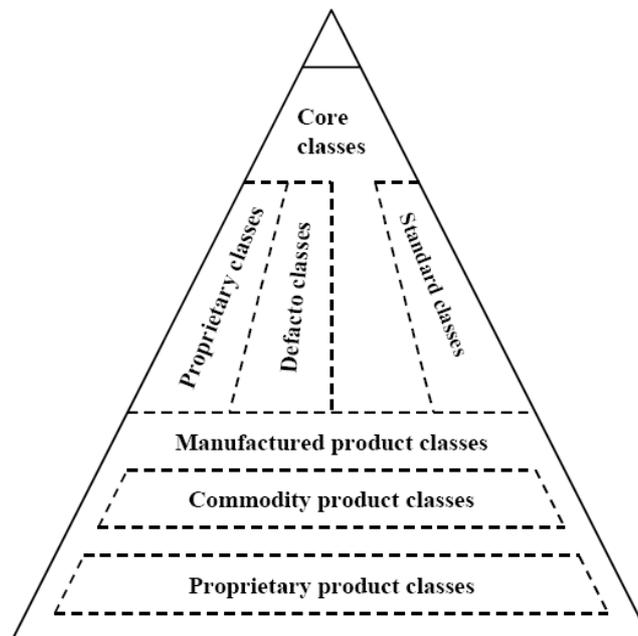


Figura 5 – Tipos de Classes / Fonte: ISO-15926-1, First Edition. 2004-07-15.

A posição de uma classe na direção do topo para a base do triângulo indica o grau de definição. As classes da parte superior são gerais e têm poucas restrições sobre a adesão, enquanto as da base são mais específicas. As classes da base do triângulo são especializações das que estão acima, e assim sucessivamente ao longo do triângulo.

LEAL (2005) define a ISO-15926 como um formato para a representação de informação sobre uma planta de processo. A base para a ISO-15926 é um registro de:

- Objetos físicos que existem dentro de uma planta de processo;
- Identificação dos objetos físicos;
- Propriedades dos objetos físicos;
- Classificações dos objetos físicos;
- Como os objetos físicos são montados;
- Como os objetos físicos estão ligados.

ISO-15926 não é apenas o registro da planta de processo, uma vez que existe num instante de tempo, mas também:

- Das mudanças de processo da planta, como resultado de manutenção e atividades de recuperação;
- Dos requisitos do modelo para a planta de processo, o que pode não corresponder diretamente a uma planta de processo tal como ele existe.

A classe ou tipo de um objeto físico é definido pela referência a um dicionário. Há centenas de milhares de classes de objetos físicos utilizados na indústria de processo. A ISO-15926 não tenta uniformizar todas essas classes, mas fornece um conjunto pequeno de classes de engenharia básica que pode ser especializado por referência a um dicionário.

A ISO-5926 Parte 4 padroniza um conjunto inicial de alguns milhares de classes genéricas. Espera-se que as empresas e associações da indústria criem dicionários para estender esse conjunto inicial.

O autor utiliza um exemplo prático de engenharia com parte de um P&ID, que define a identificação e conectividade de objetos em uma planta de processo e, para tornar mais claros os conceitos desta Ontologia, seus símbolos também fornecem uma base de classificação dos objetos.

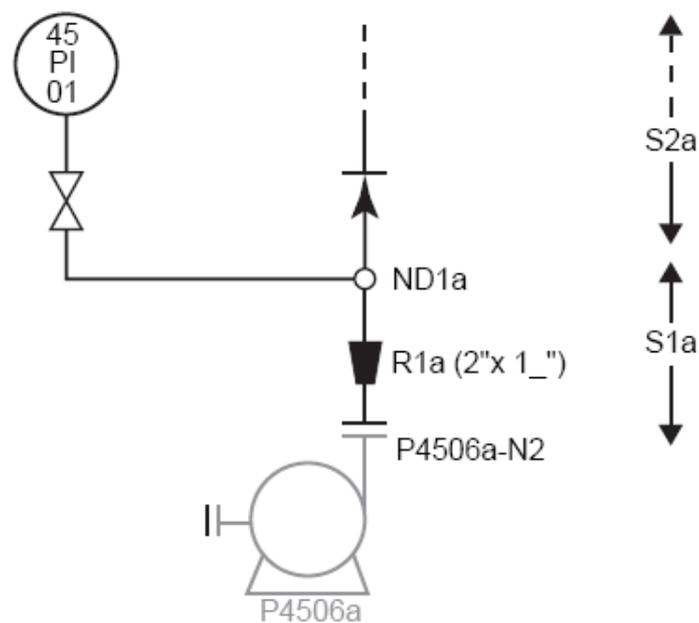


Figura 6 – Parte de um P&ID / Fonte: LEAL (2005).

No exemplo da Figura 6, há um objeto físico que:

- Tem identificador P4506a;
- É classificado como uma bomba centrífuga.

Esta informação é gravada pelas duas relações mostrada na Figura 7.

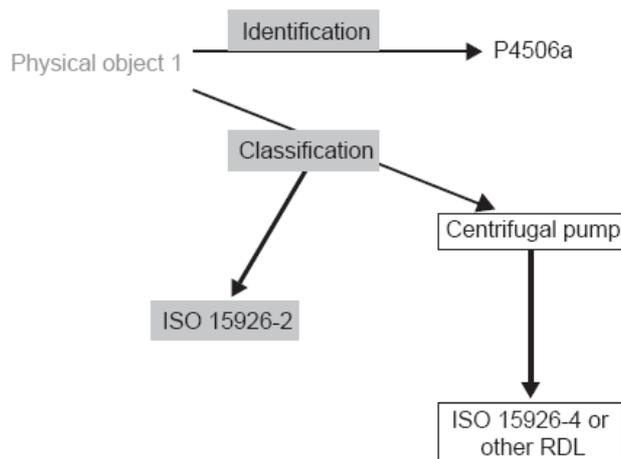


Figura 7 – Identificação e Classificação das Relações / Fonte: LEAL (2005).

A identificação e classificação são relações definidas na ISO-15926-2. A bomba centrífuga é uma classe definida no âmbito da ISO-15926-4. A um objeto físico podem ser dadas classificações adicionais mais precisas em relação a uma mercadoria da empresa ou a um dicionário de catálogo de fabricantes.

A Figura 6 também mostra uma redução R1a, um instrumento de pressão 45 PI 01, e dois segmentos de tubo S1A e S2A. Estas informações são gravadas pelas relações mostradas na Figura 8. O bocal P4506a-N2 está ligado ao segmento de tubo S1A.

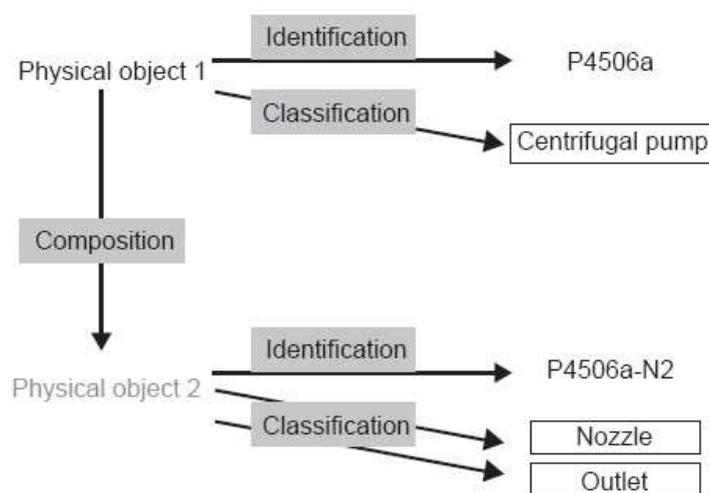


Figura 8 – Relações de Composição / Fonte: LEAL (2005).

O segmento de tubo é um objeto físico que:

- Tem uma relação de conexão com o bocal P4506a-N2;
- É classificado como um segmento de tubo.

Esta informação é gravada pelas relações mostradas na Figura 9.

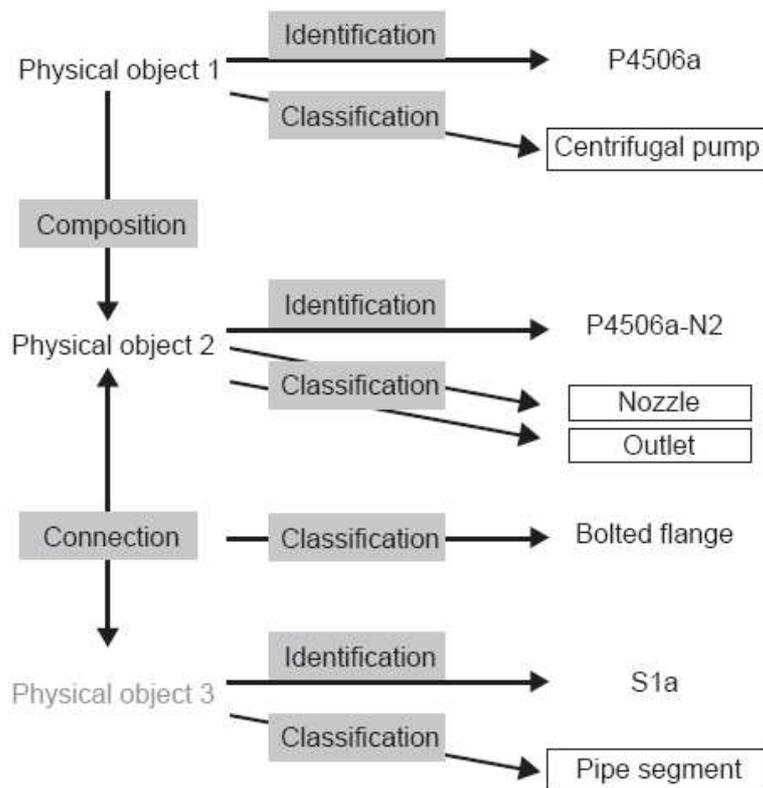


Figura 9 – Relação de Conexão / Fonte: LEAL (2005).

O esperanto pode ser utilizado como metáfora para explicar o papel da ISO-15926. A comunicação sempre foi um desafio para a humanidade. O esperanto é uma língua planejada que foi publicada em 1887 com a intenção de criar uma língua de fácil aprendizagem, que servisse como língua franca internacional, para toda a população mundial, mas sem qualquer intenção de substituir todas as línguas existentes. Para utilizar o esperanto cada membro de cada país precisa traduzir os termos de sua língua local para os termos do esperanto, o que pode ser considerado um mapeamento de termos. Nessa metáfora a ISO-15926 toma o lugar do Esperanto. A ISO-15926 é uma "linguagem" comum de intercâmbio de informação sobre plantas.

Independente da forma como cada um de nós armazena as nossas informações da planta, na interface, teríamos que "traduzir" este formato para o da ISO-15926 (POSC CAESAR, 2010).

BATRES et al. (2007) traduziram o código original EXPRESS da norma ISO-15926-2 para a linguagem OWL que pode ser usada diretamente em um número de pacotes de *software* de inferência. Apresentou os aspectos-chave da ontologia, descrevendo algumas das suas principais classes e propriedades e discutiu seus benefícios e aplicações no domínio da engenharia de processo, aplicando a um estudo de caso sobre os benefícios de uma ontologia superior para representar e interpretar informações produzidas durante um estudo de HAZOP (*Hazard and Operability*).

2.7 LINGUAGENS DE ONTOLOGIA E SINTAXE OWL

Embora a maior vantagem do XML seja o seu padrão, bem formatado com natureza de texto simples que permite aos desenvolvedores ler, compreender e trabalhar com ele em uma vasta coleção de ferramentas (disponível gratuitamente), a interoperabilidade entre elas exige uma quantidade significativa de engenharia, trabalho manual e coordenação. A razão para isto é que o XML não resolve o problema da inter-relação semântica de modelos de dados. Para automatizar ainda mais os processos de desenvolvimento e permitir o processamento de alto nível dos dados, uma iniciativa, chamada *Web Semântica* (SW), abriu a porta para uma nova era de "intercâmbio de dados inteligente", em que a informação tem um significado bem definido, permitindo que computadores, ou melhor, que as pessoas trabalhem em cooperação (BEETZ *et al.*, 2005).

No contexto da *Web Semântica*, o uso de Ontologias requer uma sintaxe e semântica bem definidas para suportar o raciocínio eficiente e a distinção entre suas classes, atributos, funções e relações.

A idéia geral na *Web Semântica* é identificar cada pedaço de dados com uma máquina processável de descrições semânticas. Estas descrições devem ser especificadas de acordo com uma certa gramática e com referência ao domínio de um vocabulário padronizado. O vocabulário do domínio é referido como uma ontologia, e destina-se a representar uma concepção comum de algum domínio. A gramática é uma linguagem

semântica de marcação. Com estas anotações semânticas adequadas, aplicativos inteligentes podem recuperar e combinar documentos e serviços a nível semântico, eles podem compartilhar, compreender e raciocinar um sobre os dados do outro, e eles podem operar de forma mais independente e se adaptar a um ambiente em mudança consultando uma ontologia compartilhada. A interoperabilidade pode ser definida como um estado em que duas aplicações podem aceitar e compreender os dados do outro e realizar uma determinada tarefa de forma satisfatória sem intervenção humana. Muitas vezes é possível distinguir interoperabilidade sintática, estrutural e semântica como (GULLA *et al.*, 2006):

- Interoperabilidade sintática denota a capacidade de intercâmbio e compartilhamento de informações de dois ou mais sistemas por marcar dados de forma semelhante (por exemplo, usando XML).
- Interoperabilidade estrutural significa que os sistemas compartilham esquemas semânticos (modelos de dados) que lhes permitem trocar informações e estrutura (por exemplo, usando RDF).
- Interoperabilidade semântica é a capacidade dos sistemas de compartilhar e compreender informações a um nível formalmente definido e mutuamente aceito em conceitos do domínio, permitindo interpretação e raciocínio por máquinas de processamento.

Em 2001 o W3C formou um grupo de trabalho denominado *Web-Ontology Working Group* (WebOnt). O objetivo do grupo era fazer uma nova linguagem de marcação de ontologias para a *Web Semântica*, chamada OWL (*Ontology Web Language*). OWL é dividida em camadas: OWLite e OW (CORCHO *et al.*, 2003).

Segundo SMITH (2004), a OWL é uma linguagem para definir e instanciar ontologias *Web*. Uma ontologia OWL pode incluir descrições de classes, propriedades e suas instâncias. Dada uma ontologia, a semântica formal OWL especifica como derivar suas conseqüências lógicas, isto é, os fatos não literalmente presentes na ontologia, mas decorrentes da semântica. Esses vínculos podem ser baseados em um documento único ou vários documentos distribuídos que foram combinados usando mecanismos definidos em OWL.

Uma ontologia é diferente de um esquema XML, sendo uma representação do conhecimento e não um formato de mensagem. A maioria dos padrões da indústria baseados na *Web* consiste em uma combinação de formatos de mensagem e especificações do protocolo.

Uma vantagem das ontologias OWL é a disponibilidade de ferramentas que podem raciocinar através delas. As ferramentas dão um suporte genérico que não é específico para o domínio particular em questão, o que seria o caso se alguém construísse um sistema para raciocinar sobre um determinado esquema XML padrão da indústria.

A linguagem OWL fornece três sublinguagens:

- OWL Lite: suporta aqueles usuários que necessitam principalmente de uma hierarquia de classificação e restrição de recursos simples. Por exemplo, a OWL Lite suporta restrições de cardinalidade, com valores de cardinalidade 0 ou 1. Deve ser mais simples para fornecer suporte a ferramentas para OWL Lite que seus parentes mais expressivos, e fornece um caminho de migração rápida de diversas taxonomias.
- *OWL DL*: suporta aqueles usuários que querem a máxima expressividade, sem perder completude computacional (todos os vínculos estão garantidos para serem computados) e *decidibilidade* (todas as computações terminarão em tempo finito) de sistemas de raciocínio. OWL DL inclui todas as construções da linguagem OWL, com restrições, como a separação do tipo (a classe não pode ser um indivíduo ou propriedade, a propriedade não pode ser um indivíduo ou classe). OWL DL é assim chamada devido a sua correspondência com as lógicas de descrição (*Description Logic*), um campo de pesquisa que estudou um fragmento particular decidível da lógica de primeira ordem. OWL DL foi concebido para apoiar o segmento de negócio existente: Lógica de Descrição tem propriedades desejáveis para os sistemas computacionais o raciocínio.
- OWL Full: é destinado a usuários que querem máxima expressividade e a liberdade sintática do RDF sem garantias computacionais. Por exemplo, em OWL Full uma classe pode ser tratada simultaneamente como uma coleção de indivíduos e como um indivíduo em seu próprio direito. Outra diferença significativa da OWL DL é que uma *DatatypeProperty* pode ser marcada como uma *InverseFunctionalProperty*. OWL Full permite que uma ontologia aumente

o significado de um vocabulário pré-definido (RDF e OWL). É improvável que qualquer raciocínio de *software* seja capaz de suportar todas as características da OWL Full.

2.8 CONTEXTO DO TRABALHO

No início dos anos 90, uma série de estudos começou a ser publicada discutindo os próximos desafios para o desenvolvimento da engenharia química. O mais abrangente destes trabalhos foi resultado da participação de diversas representações acadêmicas e laboratórios nacionais dos Estados Unidos: *Technology Vision 2020: The U.S. Chemical Industry*, identificando as principais necessidades e demandas da engenharia química para as duas próximas décadas. A área de sistemas de informação recebeu destaque como uma das disciplinas de maior relevância para os desenvolvimentos futuros (SCHNEIDER & MARQUARDT, 2002).

Segundo o estudo feito pelo *American Chemical Society* (1996), diversas aplicações computacionais estão disponíveis no mercado para modelagem e simulação de processos, como Hysys, AspenPlus e PROII, controle e otimização de operações e fluidodinâmica computacional (CFD), como o CFX por exemplo. Essas tecnologias são utilizadas tanto para áreas de pesquisa e desenvolvimento quanto para projetos e plantas em operação. A realidade é que os simuladores de processos tornaram-se ferramentas indispensáveis para projetos de plantas de processo. A forma como os dados são transformados em informação e utilizados, gerenciados, transmitidos e armazenados está diretamente relacionada com a competitividade por toda a indústria química. Melhorias e sistemas de informação avançados são o cerne da visão de tecnologia do futuro, que vê a indústria química operando de forma altamente eficiente e econômica.

O estudo destaca ainda que para utilizar-se plenamente dos benefícios da tecnologia da informação e sistemas computacionais, algumas mudanças políticas são necessárias na indústria química de modo a alterar e melhorar a gestão dos dados:

- Os sistemas de informática e as redes devem estar disponíveis quando necessários.

- O uso de papel no local de trabalho deve ser significativamente reduzido.
- Os sistemas de controle de processos devem ser usados automaticamente como dados de entrada para outros sistemas. Dados de clientes e fornecedores podem ser transferidos entre redes de computadores em ambas as direções. Por conseguinte, os dados nunca devem ser digitados mais de uma vez em qualquer computador.
- O intercâmbio de dados deve ser transparente para o usuário e, para tornar isto possível, *software* e acessórios devem ser desenvolvidos.
- A proteção de informações proprietárias deve ser reforçada através de contínua melhoria dos sistemas de segurança computacionais.

GIELINGH (2008) afirma que não existem aplicativos CAE totalmente integrados e que ofereçam suporte ao ciclo de vida inteiro do produto. Como uma empresa moderna é uma articulação do esforço de empresas colaboradoras, modelos de produto só podem ser aplicados de forma eficaz se os dados puderem ser trocados e/ou compartilhados livremente além das fronteiras das organizações e das soluções de fornecedores específicos. Dado o fato de que há tantas aplicações CAE diferentes no mercado, é evidente que estes objetivos só podem ser alcançados através de normas. Entretanto, a adoção dos padrões lançados até o momento para obtenção da interoperabilidade dos sistemas ainda é baixa na indústria como um todo. Os desenvolvedores das normas assumem que os fornecedores de aplicações CAE devem suportar exportação e importação de dados de produto de forma normatizada, sendo os supostos responsáveis por investir nas traduções necessárias e até mesmo mudar seus produtos, a fim de se adequarem aos padrões. Isto significa que as empresas que têm de investir em interoperabilidade (fornecedores CAE) não são os que se beneficiarão dos resultados (operadores de plantas industriais, empresas projetistas). Para os fornecedores que já oferecem soluções integradas e têm um mercado consolidado não é interessante investir em soluções que tornam os produtos dos concorrentes mais atraentes. Mesmo se os fornecedores de *software* estiverem dispostos a se adequar ao padrão, eles vão investir apenas se houver uma demanda do mercado. Além disso, comunicação não será possível se os usuários não estiverem preparados. Todas as organizações em uma rede de abastecimento que desejam se comunicar via dados do produto devem estar prontas para produzir e utilizar os dados e todas as aplicações CAE relevantes que eles usam devem adotar o mesmo padrão da mesma versão e no mesmo nível. Novamente, como as condições são difíceis de

cumprir, são exigidos alto nível de gerenciamento estratégico, apoio, compromisso de longo prazo e normas estáveis.

A compreensão dos problemas de integração para suporte aos projetos na área de processos químicos tem sido a base para dar forma a grandes áreas de pesquisa consideradas como potenciais melhorias. Em particular, elas incluem (MARQUARDT & NAGL, 2004):

- A modelagem, análise e projeto de reengenharia de processos, que apresentam a integração ainda muito isolada em atividades de projeto ou através da definição de processos de projeto inovadores;
- O desenvolvimento de um modelo integrado de informação de um projeto de processo completo, no sentido de uma ontologia;
- O desenvolvimento de novos conceitos de ciência da computação e sua aplicação em um protótipo para a informação e colaboração no gerenciamento dos procedimentos de trabalho de engenharia em projetos de processos;
- A implementação de um exemplo de projeto de sistemas integrado para ilustrar a sinergia da integração e para provar o benefício adicional para o usuário final por meio de um cenário de projeto industrial relevante e realista;
- O desenvolvimento de tecnologias de *software* para integração das ferramentas existentes e suas funcionalidades com ênfase na geração automática para homogeneizar as interfaces.

Neste cenário, fica evidente que as normas e padrões internacionais estão disponíveis para possibilitar o intercâmbio e compartilhamento de informações de engenharia. Entretanto, as soluções de tecnologias não as suportam ou aplicam de forma ineficiente. Torna-se extremamente relevante o investimento em P&D para identificar as falhas dos padrões existentes e promover melhorias para torná-los eficazes e abrangentes na indústria química, assim como para identificar seus benefícios e melhor forma de utilizá-los. A aceitação destes padrões depende fortemente de estudos que comprovem sua eficiência na qualidade e otimização dos procedimentos de trabalho atuais.

Os procedimentos de trabalho atuais na área de projetos de plantas de processo envolvem uma série de ferramentas de *software* para modelagem de fluxogramas de engenharia (P&IDs), simulação do processo, cálculo e dimensionamento de linhas de tubulação, válvulas, bombas, trocadores de calor e cabos elétricos, que variam de projeto para projeto. Estes aplicativos são utilizados de forma independente, não sendo considerados os fluxos de dados e as relações entre os procedimentos de trabalho interdisciplinares. Diversas manobras são desenvolvidas muitas vezes pelos próprios engenheiros envolvidos tecnicamente no projeto com o intuito de minimizar a replicação manual das informações de engenharia. Porém, cada disciplina cria seu próprio método de recebimento, utilização e replicação das informações de forma independente das demais. Assim, a cada projeto novos métodos para executar o mesmo trabalho são desenvolvidos e o esforço anterior é desperdiçado, uma vez que os membros das equipes e as ferramentas de *software* são renovados. Isso acontece porque os processos desenvolvidos muitas vezes não são baseados em normas ou padrões e nem mesmo são considerados uma metodologia formal dentro da companhia.

A Figura 10 ilustra os mecanismos utilizados atualmente para gerar o intercâmbio de dados entre as ferramentas de *software*, onde as setas vermelhas indicam os automatismos desenvolvidos.

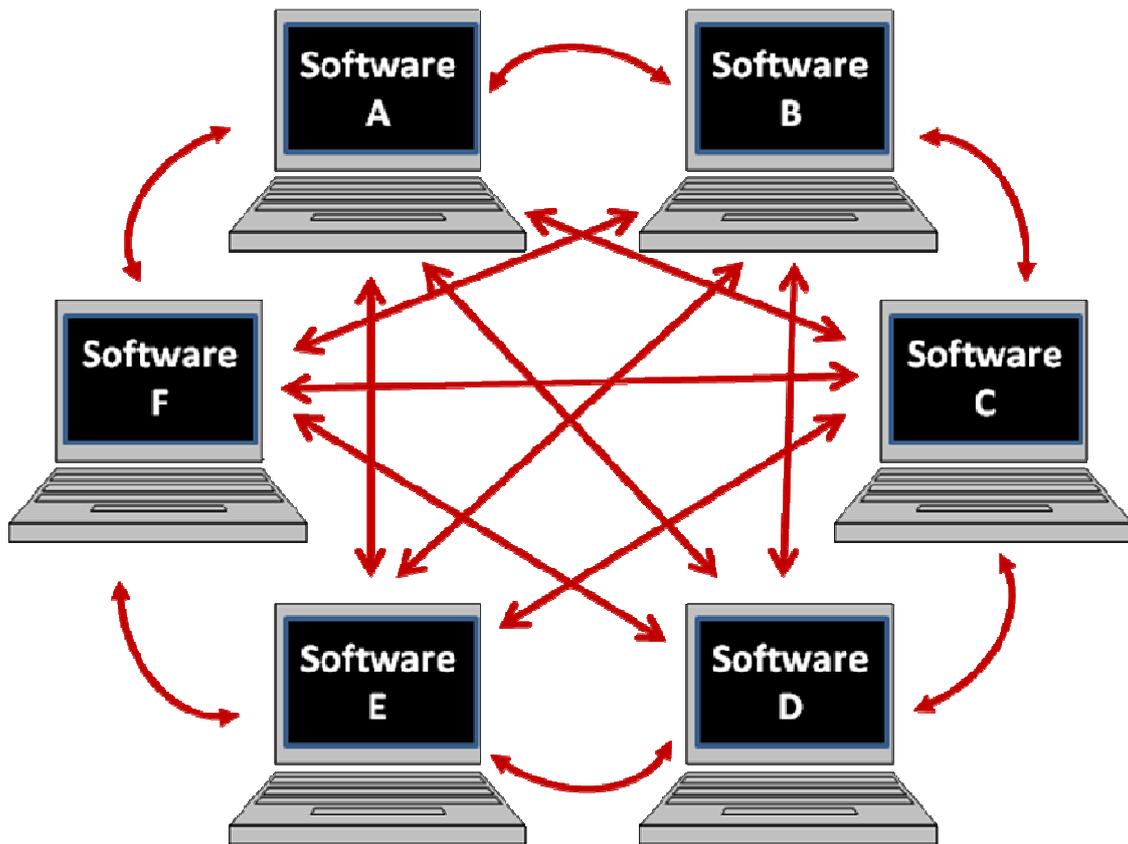


Figura 10 – Automatismos desenvolvidos para integração entre ferramentas de *software* em projetos de engenharia.

Todo este empenho para automatizar o fluxo de informações se deve ao fato de que falhas na fase de projeto podem causar sérios danos, retrabalho e atrasos na fase de construção e operação das unidades de processo, com impacto considerável no custo do empreendimento.

Este trabalho visa o desenvolvimento do aplicativo **ORION**, utilizando os conceitos de ontologia propostos na norma ISO-15926 e linguagem OWL para converter os dados de saída do simulador de processos EMSO em um arquivo XML neutro e padronizado, com o objetivo de avaliar o desempenho desta ontologia superior para representar os dados de processo de uma planta química e propagá-los ao longo das etapas do projeto em seus respectivos documentos elaborados na ferramenta CAE COMOS®.

A utilização de um modelo de ontologia minimizará a quantidade de automatismos ilustrados na Figura 10, tendo como base um meio central, neutro e padronizado para

comunicação entre todas as ferramentas de *software* envolvidas no projeto, conforme ilustrado na Figura 11.

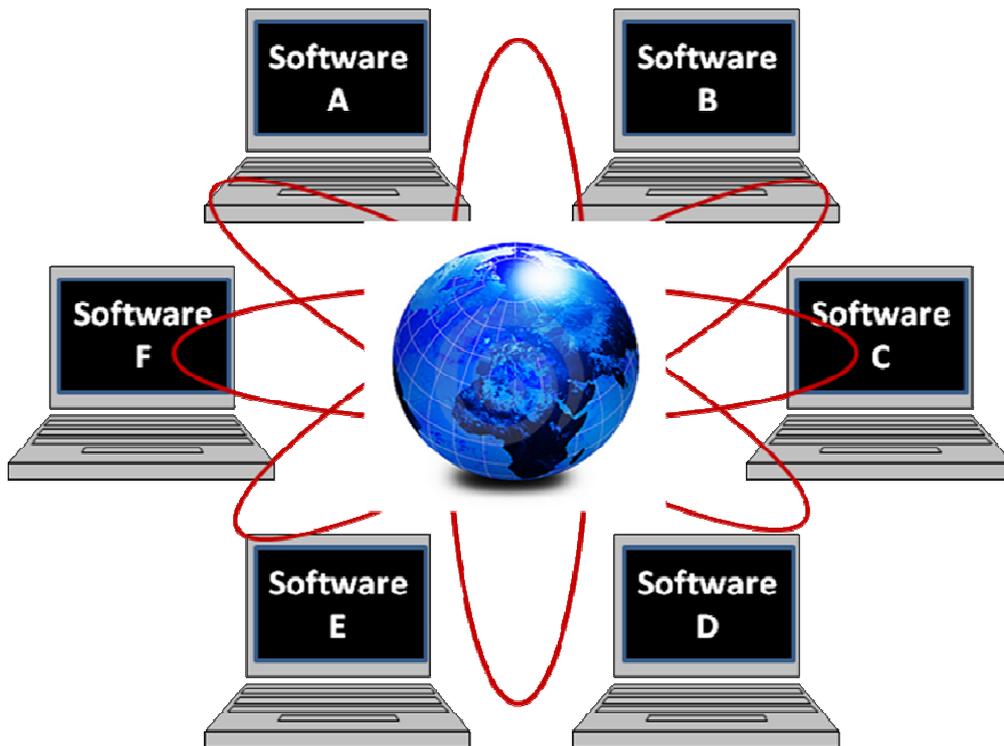


Figura 11 – Utilização de um modelo de ontologia central, neutro e padronizado para intercâmbio de dados.

A simulação de processos representa dentro do ciclo de vida do projeto de uma unidade de processo o início de uma série de atividades interdisciplinares, que geram um grande volume de informações e documentos, que demandam um elevado grau de controle para intercâmbio de dados de engenharia. A etapa de simulação alimenta diversos documentos da fase de projeto básico da planta, com informações operacionais que servirão como base para o dimensionamento e especificação de linhas de tubulação, instrumentos e equipamentos.

Diversos são os motivos que alteram a simulação do processo ao longo do projeto básico, sendo que a maioria deriva de mudanças nos requisitos do projeto por parte do cliente. Por exemplo, na área de exploração e produção de petróleo a curva de produção de petróleo pode ser alterada ao longo do projeto básico de acordo com a evolução dos estudos da equipe de pesquisas de campo. A verdade é que estas mudanças são inevitáveis e torna-se necessário garantir que todas as mudanças

realizadas na simulação do processo serão propagadas de forma consistente, abrangente e ágil para todos os documentos anteriormente alimentados com estas informações.

A Figura 12 exemplifica o fluxo de dados entre os documentos gerados na fase de projeto básico de uma unidade de produção de petróleo e a etapa de simulação de processo. Os requisitos do projeto, incluindo a curva de produção de petróleo, são recebidos na Base de Projeto e estas informações serão utilizadas como entrada para simulação do processo e geração de uma Lista de Equipamentos preliminar. A simulação do processo dá origem aos Balanços de Massa e Energia e fornece os dados de entrada para geração dos Fluxogramas de Processo (*PFD – Process Flow Diagram*). A partir destas informações, são geradas as memórias de cálculo do sistema e a Lista de Equipamentos é revisada. A memória de cálculo também servirá como entrada para geração dos Fluxogramas de Engenharia (P&ID) e Folhas de Dados de Processo de Equipamentos. Uma vez tendo o P&ID consolidado é feito – dimensionamento das linhas de tubulação e geração das Folhas de Dados (FD) de Processo dos Instrumentos.

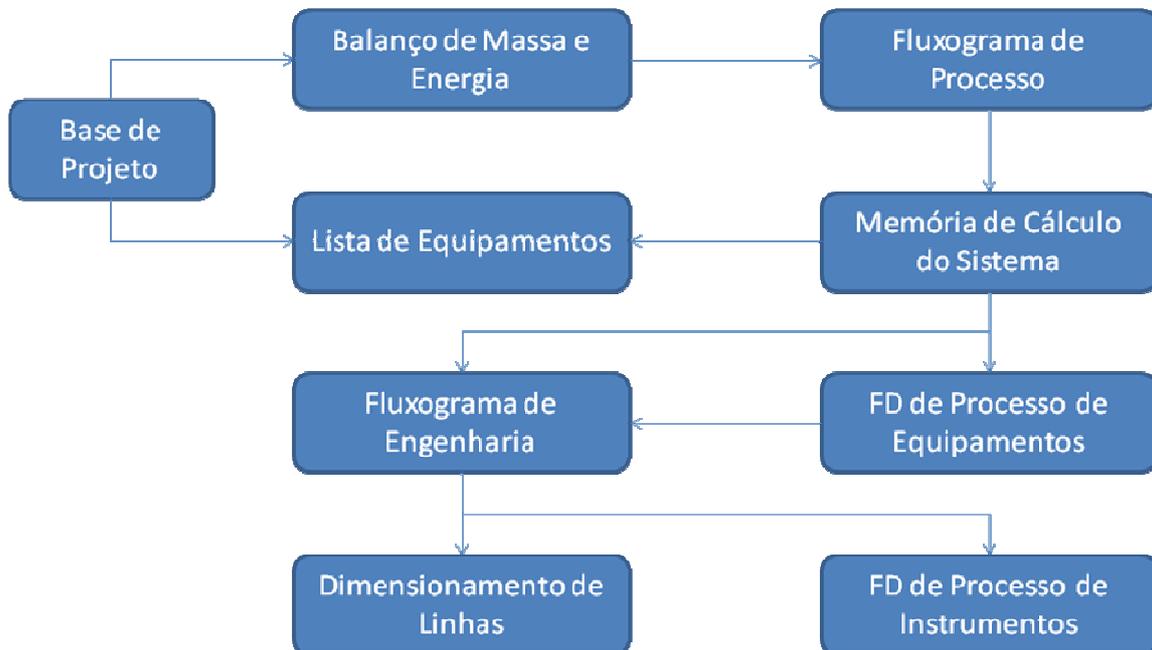


Figura 12 - Fluxo de trabalho em um projeto de engenharia de processos.

O ciclo de vida de um projeto é suportado por uma variedade de ferramentas de *software* desde aplicações como editores de texto e planilhas de cálculo (*Word, Excel, etc*), simuladores de processos para cálculos de engenharia, ferramentas CAD para representações gráficas, até aplicações CAE para modelagem suportada por banco de dados altamente complexos.

O conceito de dado centralizado e ferramentas de integração tem sido adotado na engenharia química e de processos desde o início dos anos 80, com uso de sistemas como PEDB da ICI, o VTPLAN da Bayer e base de dados de processos como o PRODABAS e DesignMASTER. Esse tipo de banco de dados foi arquitetado para armazenar informações de projeto para as fases conceitual e FEED de unidades de processo (SCHNEIDER & MARQUARDT, 2002).

A evolução da indústria de *hardware* e *software* trouxe ganhos consideráveis de processamento para a área de computação e suportou nos últimos anos o desenvolvimento de ferramentas mais direcionadas para ambiente de projetos na área de engenharia química. O conceito de banco de dados de engenharia está hoje disponível no mercado em ferramentas como *Aspen Basic Engineering (Aspen ZyqadTM)*, COMOS® da COMOS Industry Solutions e *SmartPlant P&ID* da Intergraph, com a proposta de atender o ciclo de vida do projeto desde a fase conceitual até a engenharia de detalhamento.

Neste conceito de banco de dados de engenharia, os diagramas de processo ganharam “inteligência” pelo uso de objetos que representam cada entidade da planta. A modelagem destes documentos tem relação direta com o fluxo de informações para folhas de dados e especificações de equipamentos. Cada documento gerado passa a ser apenas mais uma forma de representação do dado técnico na lógica das entidades que compõe a unidade de processo e o engenheiro projeta a planta virtualmente com a compreensão completa da unidade.

O presente trabalho fará uso do *software* COMOS® como banco de dados de engenharia para geração dos documentos da fase de projeto básico (PFD, P&ID, Lista de Linhas e Equipamentos e FDs de equipamentos e instrumentos). Esta escolha está fundamentada nos seguintes aspectos da ferramenta CAE: compreende em um único banco de dados o projeto das três disciplinas fundamentais para o projeto da planta – processo, instrumentação e elétrica; permite nativamente o fluxo de dados entre estas

disciplinas, permite manipulação de seus dados via programação externa; sua biblioteca pode ser acessada via Visual Basic .NET sem a necessidade de abertura da interface da aplicação.

O aplicativo **ORION** irá funcionar como um tradutor entre os ambientes de simulação e projeto para manter as informações atualizadas automaticamente e isentas de erros de digitação manual, frente às mudanças ocorridas na fase de simulação. A Figura 13 ilustra o processo de comunicação proposto entre o simulador de processos EMSO e o *software* de projeto de engenharia COMOS®. Para se obter este meio neutro de comunicação exposto anteriormente, é necessário realizar um mapeamento entre a nomenclatura e estrutura utilizada em cada *software* com as classes propostas e definidas pela ISO-15926. Uma vez realizado este mapeamento entre o simulador EMSO e as classes da ISO-15926, o simulador poderá se comunicar com qualquer *software* de engenharia que consiga ler um arquivo neutro padronizado de acordo com a mesma norma. Ou seja, o aplicativo **ORION** recebe o arquivo de saída do simulador como entrada, interpreta sua nomenclatura com o auxílio do mapeamento disponível e gera um novo arquivo de dados de simulação com as exatas informações de origem padronizadas para a nomenclatura da norma. O arquivo de saída do **ORION**, um XML considerado neutro e padronizado, é por sua vez alimentado no *software* de projeto de engenharia COMOS®. Mas, para que o *software* possa utilizar estas informações corretamente, um prévio mapeamento de seu modelo de dados precisa ser feito para obter a relações com as classes da ISO-15926.

É importante notar que o mapeamento entre as classes da ISO-15926 e as ferramentas de *software* é realizado uma única vez, entre o *software* e o aplicativo **ORION**, chamado aqui de meio neutro, possibilitando sua comunicação e transferência de dados com qualquer outro *software* mapeado com o aplicativo **ORION**.

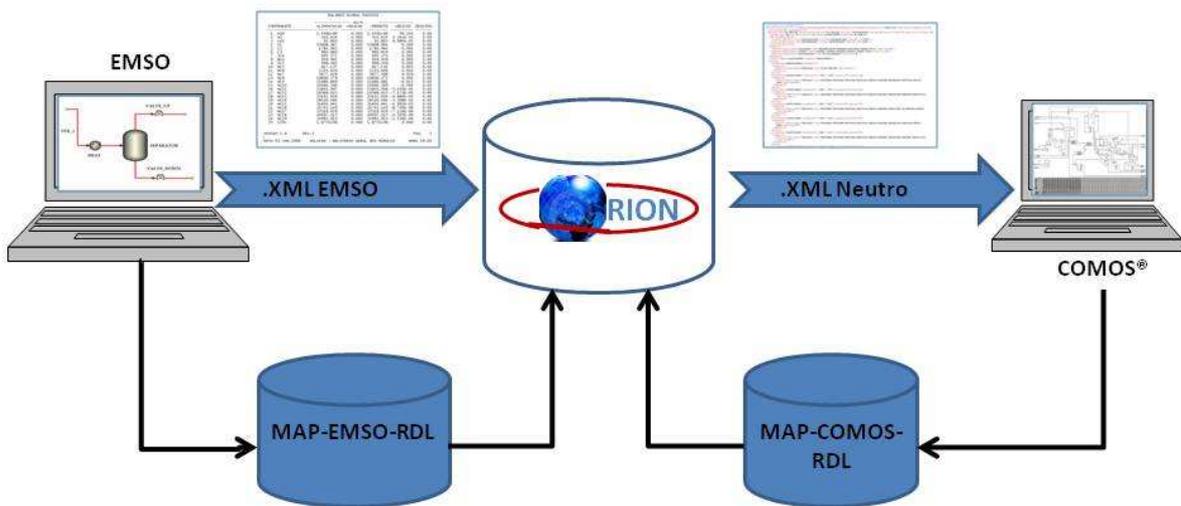


Figura 13 – Integração entre ambiente de simulação e projeto via aplicativo ORION.

O aplicativo **ORION** é o produto gerado por esta dissertação e atende uma transferência de dados unidirecional do EMSO para o COMOS, seguindo exatamente o fluxo de trabalho da fase de projeto em questão. Para que o ORION seja capaz de interpretar outros simuladores é necessário um estudo prévio das saídas geradas pelos *softwares* e programação do código de interpretação destes modelos.

Além disso, é importante ressaltar que este trabalho objetiva desenvolver uma programação para leitura, recriação e manipulação de arquivos em diferentes formatos e linguagens, sem a necessidade do desenvolvimento de um modelo de ontologia.

3 METODOLOGIA

3.1 INTRODUÇÃO

O conceito de modelo de dados envolve um conjunto de construções e regras (palavras e sintaxe), pelo qual uma parte de um mundo conceitual que aponta para o mundo real pode ser construída. A função de um modelo de dados é prover uma linguagem para as pessoas ou sistemas de comunicação (YANG & MCGREAVY, 1996). Neste trabalho são utilizados dois modelos de dados distintos para execução de duas etapas seqüenciadas do projeto de engenharia de plantas de processo: O EMSO, para simulação do processo e o COMOS® para desenvolvimento dos documentos da fase de projeto básico.

Os obstáculos para comunicação entre esses dois modelos de dados são devido às diferenças entre as metodologias empregadas para a modelagem dos dados e as regras e linguagem aplicadas à construção destes dois sistemas. Para solução deste problema é preciso empregar um modelo de normalização, que aqui será baseado no padrão ISO-15926, para construção de um meio tradutor entre os dois ambientes, simulação e projeto. O objetivo principal será a geração de um arquivo neutro para intercâmbio de dados entre os dois sistemas, atendendo aos seguintes requisitos de compartilhamento de dados:

- Coerência (ausência de ambigüidade);
- Consistência;
- Reutilização por outros;
- Atualização de mudanças, disponível e imediata.

Nos processos de trabalho dos projetos de plantas ou operações da planta, uma grande parte das atividades ainda envolve transferir as informações de um documento para outro manualmente. Por exemplo, após o engenheiro definir um instrumento, a única maneira prática para registrar as informações sobre o instrumento é ler os dados do fabricante, interpretá-los e decidir quais valores serão transcritos. Em seguida, identificar onde inserir os valores no sistema de projeto da planta. Algumas das operações são simples transcrição, como transferir um número de modelo a partir de

um ponto para outro. Mas algumas envolvem cálculo, tais como a mudança de uma unidade de medição para outra. Outras envolvem interpretação que vão desde ignorar o valor dos dados completamente às decisões que envolvem julgamento, tal como orientação. O trabalho desenvolvido em computadores, mas muitas vezes a única diferença real é que os próprios engenheiros fazem a digitação em vez de repassar esta tarefa a sua secretária (POSC CAESAR, 2010).

Em casos de atividades de transcrição de valores de dados a utilização de uma taxonomia seria suficiente. Entretanto, neste trabalho, para atender o *workflow* dos projetos de engenharia é necessário atribuir um contexto para cada informação.

Supondo que seja necessário transferir informações técnicas de uma folha de dados para outra, elaboradas em diferentes sistemas, e um sistema receba o valor 1034. Nenhuma ação pode ser executada sem que seja informado o significado para este valor. Por exemplo, “Pressão: 1034” possui um contexto dentro do sistema. Entretanto, esta informação ainda não pode ser utilizada, porque não informa a unidade de engenharia para medição da Pressão. Ainda que seja informada uma unidade de medição, no escopo de projetos de engenharia espera-se que esta medida esteja de acordo com o Sistema Internacional de Unidades (SI), como “Pressão: 1034 kPa”. Além disso, é necessária a informação sobre qual componente da planta opera sob esta condição de Pressão. Sendo assim, a informações a ser transferida deve ser: TAG Nº: P-001 / Pressão: 1034 kPa (POSC CAESAR, 2010). No momento da transferência da informação faz-se necessária uma interpretação (julgamento), porque a primeira folha de dados pode ter a nomenclatura “Pressão de Descarga” e “Pressão de Sucção”. Mas, a FD do outro sistema de projeto pode utilizar a nomenclatura “Pressão na Entrada” e “Pressão na Saída”.

Para viabilizar um *wokrflow* automatizado, baseado nos conceitos de interoperabilidade, faz-se necessária a utilização de uma ontologia que empregue as definições, classificações, atributos e restrições necessárias para que as informações possam ser utilizadas no contexto do fluxo de dados de projeto entre os diferentes sistemas sem a intervenção humana.

As atividades que constituem o desenvolvimento deste trabalho estão representadas de forma seqüenciada na Figura 14:

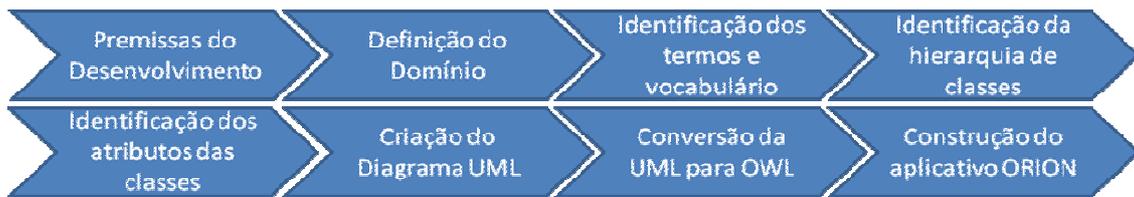


Figura 14 – Atividades empregadas no desenvolvimento do trabalho.

A definição de premissas é uma etapa fundamental do processo de desenvolvimento de qualquer aplicativo de *software*. Nesta etapa são listados todos os requisitos a serem considerados como base para o entendimento do ambiente e do sistema em desenvolvimento. As seis atividades seguintes constituem a análise do projeto, ou seja antes de iniciar a atividade de construção do aplicativo **ORION** é necessário definir o domínio a ser atendido e os componentes que constituem este domínio.

A ISO-15926 é composta por sete partes, porém apenas as partes 2 e 4 são relevantes para o desenvolvimento deste trabalho.

A Parte 2 especifica a linguagem de modelagem para a definição de terminologias específicas da aplicação, vindo na forma de um modelo de dados e inclui 201 entidades que são relacionadas em uma hierarquia de especialização de tipos e subtipos. Destina-se a fornecer os tipos básicos necessários para a definição de qualquer tipo de dados industriais. Ela tem sido especificada no EXPRESS, que tem uma definição formal, baseada na teoria de conjuntos e lógica de primeira ordem. A Parte 4 da ISO-15926 é composta por aplicação ou terminologias de disciplinas específicas, e é normalmente referenciada como *Reference Data Library* (RDL). Esses termos, descritos como classes RDL, são instâncias dos tipos de dados da Parte 2 e estão relacionados uns aos outros em uma hierarquia de especialização das classes e subclasses, bem como através de associações e relacionamentos. Se a Parte 2 define a linguagem para descrever as terminologias padronizadas, a Parte 4 descreve a semântica dessas terminologias. A Parte 4 contém hoje cerca de 50.000 conceitos gerais como motor, turbina, bomba, tubos e válvulas (GULLA *et al.*, 2006).

Definidas as premissas e tendo uma análise bem estruturada do projeto, a construção do aplicativo **ORION** pode ser iniciada. A metodologia utilizada para o desenvolvimento deste aplicativo está dividida em 5 etapas:

1. Leitura do arquivo de saída do EMSO, que é um XML formatado para abertura em Excel ou OpenCalc.
2. Mapeamento dos modelos do EMSO para as classes correspondentes no padrão ISO-15926-4.
3. Geração de um novo arquivo XML, utilizando linguagem padrão de ontologia OWL e o modelo ISO-15926-2.
4. Mapeamento dos modelos da ferramenta CAE COMOS® para as classes correspondentes no padrão ISO-15926-4.
5. Leitura dos dados do arquivo XML padronizado para ferramenta CAE COMOS®.

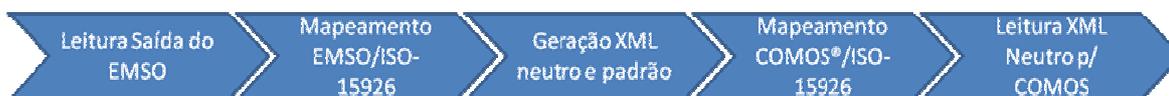


Figura 15 – Etapas da Construção do ORION.

A linguagem de ontologia OWL foi selecionada para esta implementação, porque diversos trabalhos recentes tiveram sucesso na utilização desta linguagem de marcação para descrever os termos semanticamente definidos na norma ISO-15926 Parte 2 (*Data Model*) e Parte 4 (*Reference Data Library*). Segundo BATRES *et al.* (2007), as razões para esta escolha são:

- Conhecimento representado em OWL pode ser processado por uma série de pacotes de *software* de inferência;
- Alguns conceitos da ISO-15926-2:2003 correspondem aos conceitos específicos da OWL (por exemplo, classe, relação);
- Apoio à criação de bibliotecas reutilizáveis;

- Uma variedade de ferramentas disponíveis publicamente para edição e verificação de sintaxe.

Para a construção deste aplicativo foi escolhida a plataforma de desenvolvimento *.NET Frameworks*, com o objetivo de produzir esta aplicação com a flexibilidade de execução tanto no ambiente *Web* quanto via *desktop*. O Visual Basic *.NET* foi utilizado neste desenvolvimento aplicado ao paradigma de programação orientada a objeto. As classes são o foco principal da programação orientada a objeto que é constituída por campos, propriedades e métodos (sub-rotinas e funções). As instâncias das classes são chamadas de objetos, que são criados para armazenar algum tipo de informação.

O desenvolvimento do aplicativo **ORION** foi estruturado em cinco classes para melhor organização e manutenção do sistema (códigos de programação). A ISO-15926 também é um modelo orientado a objeto e, portanto, é estruturada em classes e instâncias destas classes, porém as classes criadas no desenvolvimento VB *.NET* não possuem qualquer relação com a estrutura de classes da norma.

A classe inicial, *XMLExcelFileRead.vb*, tem como idéia principal receber o apontamento do arquivo de saída do simulador e armazenar os dados de interesse em memória no aplicativo **ORION**, por meio de um objeto do tipo *DataSet*. Este objeto é um banco de dados em memória que armazena as informações em objetos do tipo *DataTable*. Uma vez armazenadas as informações oriundas da simulação do processo, o aplicativo precisa encontrar o tipo de cada informação dentro das classes da ISO-15926. A classe *MapISO15926.vb* busca a relação em um arquivo que contém o mapeamento entre o nome da informação no EMSO, o nome da informação na RDL e a fonte onde esta informação pode ser encontrada e armazena no *DataSet*. As RDLs são definidas na Parte 4 da ISO-15926 que fornece uma biblioteca de dados de referência (RDL) onde as instâncias das classes principais estão definidas. Apenas com a definição da ontologia na Parte 2 não seria possível padronizar um arquivo de intercâmbio de dados, é necessário ter um vocabulário controlado. A RDL é fornecida pela ISO-15926 em formato de planilhas Excel, sendo uma para cada classe principal. Para construção do arquivo neutro de intercâmbio de dados é preciso buscar em cada uma das planilhas fonte de classes de referência da ISO-15926-4 os termos semânticos definidos na Parte 2. Esta consulta e armazenamento são executados pela classe *ISOReferenceData.vb*. Armazenada a terminologia para cada uma das informações de saída do simulador, a classe *XMLISOApplication.vb* executa a construção do arquivo neutro de intercâmbio de dados XML utilizando a linguagem de

marcação OWL. Assim como foi utilizado um mapeamento para relacionar as RDLs com as informações do simulador, a classe *MapISO15926.vb* faz também o armazenamento da relação das informações do modelo do COMOS® com as RDLs da ISO-15926 no objeto *DataSet*. A classe *ImportISOXML.vb* faz a leitura dos dados padronizados no arquivo XML neutro para o modelo de dados do *software* COMOS®, concluído o processo de transferência de dados entre os dois ambientes.

3.2 PREMISSAS PARA O DESENVOLVIMENTO

A literatura fornece informações relevantes para especificação da modelagem dos dados que descrevem as operações de plantas de processo. Estudos anteriores são utilizados como referência para as premissas utilizadas neste desenvolvimento.

Segundo YANG & MCGREAVY (1996), os dados de processo são informações necessárias para descrever, operar e permitir que o processo aconteça em uma planta de processo para geração de produtos. Os dados que são usados para descrever os processos suportados pelas plantas e seus aspectos dinâmicos são diferentes dos dados das plantas utilizados para descrever os aspectos estatísticos e as relações lógicas com os equipamentos da planta. Neste artigo, os autores identificam, para uso na planta de processo, o escopo dos dados precisa atender aos seguintes requisitos:

- O modelo de dados deve ser capaz de cobrir os processos (operações típicas) suportados pela planta e os materiais que estão sendo operados por esses processos;
- O modelo deve atender a todos os estágios do ciclo de vida:
 - Os requisitos do processo resultantes do projeto conceitual
 - O processo previsto resultante do projeto de detalhamento
 - O processo suportado pela planta durante a operação e manutenção
- O modelo deverá tratar os dois comportamentos em estado estacionário e dinâmico;
- O modelo deve abranger ambos os processos batelada e contínuo;

- O modelo deve ser capaz de manipular tanto o modelo a ser usado para descrever um processo quanto o processo em si.

Os dados de processo típicos identificados incluem informações sobre:

- Materiais e correntes
- Condições de operação
- Equipamentos de processo
- Lógica de controle de processo
- Segurança dos processos
- Potencial econômico de processos (não o custo da fábrica)
- Desenho de processos
- Disponibilidade de processos
- O controle dos processos
- Base de projeto de processos, tais como normas, códigos, suposições, cálculos e modelos de análise, projeto, programação, bases de dados técnicos, especificações dos fornecedores de materiais e equipamentos, caminho da reação, as condições do local, etc.

Os dados de processo podem ser classificados em dois grupos na prática. Um deles reúne os dados básicos necessários para descrever a lógica do processo e os procedimentos de operação. O outro são os dados úteis para ajudar na compreensão das suposições, métodos e modelo do processo em que o projeto foi baseado.

Neste trabalho, o aplicativo **ORION** deverá ser capaz de integrar os ambientes de simulação e projeto considerando os seguintes aspectos:

- Representação de todos os componentes da planta de processo (equipamentos, instrumentos e tubulações);
- Identificação das condições de operação dos componentes da planta;

- Identificação dos materiais das correntes de processo;
- Tratar os dados do comportamento em estado estacionário;
- Consistência dos dados de engenharia da planta;
- Representação dos dados de projeto de ambas as fases: simulação e projeto básico;
- Identificação de todas as conexões entre os componentes da planta para reutilização na construção dos Diagramas de Processo (PFD) e Diagramas de Engenharia (P&ID);
- Compartilhamento e intercâmbio de informações entre a ferramenta de simulação de processos e *software* CAE.

3.3 CONSTRUÇÃO DO DIAGRAMA UML

Apesar de o desenvolvimento de uma ontologia não ser parte deste trabalho, uma vez que será utilizada a ISO-15926 para este propósito, orientações disponíveis na literatura são utilizadas como base para uma melhor compreensão desta ontologia. A partir deste entendimento é gerado um Diagrama de Classes UML com a representação conceitual da Ontologia em um formato compreensível por humanos e sistemas computacionais.

METHONTOLOGY é uma metodologia de desenvolvimento de ontologias, com base em metodologias de Engenharia de *Software* e de Engenharia do Conhecimento, apresentada por GÓMEZ-PÉREZ *et al.* (2004). Estas instruções serviram como referência para a definição das atividades iniciais deste estudo que servem como suporte ao processo de desenvolvimento do aplicativo **ORION**.

Definição do Domínio da Ontologia:

O escopo da ISO-15926 é fornecer um modelo de dados conceitual para ser usado em conjunto com uma biblioteca de dados de referência (instâncias padrão que representam informações comuns) para suportar as atividades envolvidas em todo o

ciclo de vida de plantas de processo, incluindo projeto, engenharia, construção, operação, manutenção, desativação e demolição.

Deste escopo tão abrangente, este trabalho aplica os conceitos fornecidos pela norma somente em duas etapas do ciclo de vida da planta de processo referentes à fase inicial de projeto de engenharia: simulação e projeto básico. E a aplicação é avaliada em estudos de caso para área de exploração e produção de petróleo.

Neste cenário, somente uma parte do conjunto de instâncias das classes disponíveis na Parte 4 da ISO-15926, que seja relevante para este trabalho, será empregada para construção do Diagrama de Classes UML.

Identificação dos Termos e Vocabulário da Ontologia:

A identificação dos termos comumente utilizados pelos especialistas no discurso do domínio da ontologia é de fundamental importância para compreensão da definição das classes empregadas.

Os termos empregados em simulação de processos são utilizados para descrever o processo, os componentes da planta envolvidos neste processo e as condições em que ocorrem os processos. Uma vez que este trabalho visa transcrever os dados de simulação para o ambiente de projeto, torna-se relevante destacar alguns termos comumente utilizados nesta área:

- Planta de Processo: é formada por um conjunto de unidades de processo divididas em sistemas operacionais para produção do produto final;
- Diagramas de Processo: conjunto de equipamentos ou operações unitárias interconectadas para representação de um sistema de processo;
- Equipamentos: São objetos utilizados na transformação dos insumos em produtos (reatores, trocadores de calor, torres, bombas);
- Correntes de Processo: meio de conexão entre dois equipamentos para transporte dos fluidos envolvidos no processo;

- Modelo: representa o comportamento de um objeto componente do sistema através de uma descrição matemática da operação unitária ou equipamento;
- Variáveis: informações que determinam o comportamento do sistema;
- Parâmetros que são propriedades do processo.

Identificação da Classificação dos Termos e Hierarquia das Classes:

A relação entre o modelo de dados conceitual da Parte 2 da ISO-15926 e as instâncias das classes criadas na Parte 4 foi apresentada de forma bastante didática por GULLA *et al.* (2006). A estrutura de topo do modelo de dados da ISO-15926-2 é mostrada em notação EXPRESS na parte superior da Figura 16. O tipo de entidade “*Thing*” tem dois subtipos, *Abstract object* e *Possible individual*. O tipo de *class* tem um tipo superior *Abstract object*, e quatro subtipos (*Class of individual*, *Class of abstract object*, *Cardinality*, e *Role and domain*). As terminologias específicas da aplicação na norma ISO-15926-4 RDL são definidas como instâncias de tipos relevantes no modelo de dados. Por exemplo, a classe RDL *Pump* é uma instância da *Class of individual* a partir do Modelo de Dados. Ao mesmo tempo, as bombas são especializações da classe RDL mais geral *IndustrialArtifact*. Da mesma forma, a relação entre as classes da bomba e da tubulação é modelada como uma instância da classe *Class of relationship* do Modelo de Dados de relacionamento. Casos individuais, como uma bomba de uma instalação submarina particular (como mostrado na Figura 16 em *#myPump*), são representados como instâncias de uma classe RDL (*Pump*), bem como uma instância de um tipo de modelo de dados (*Possible individual*). Nota-se que isto também se aplica ao relacionamento entre instâncias individuais, como *#myConnection*.

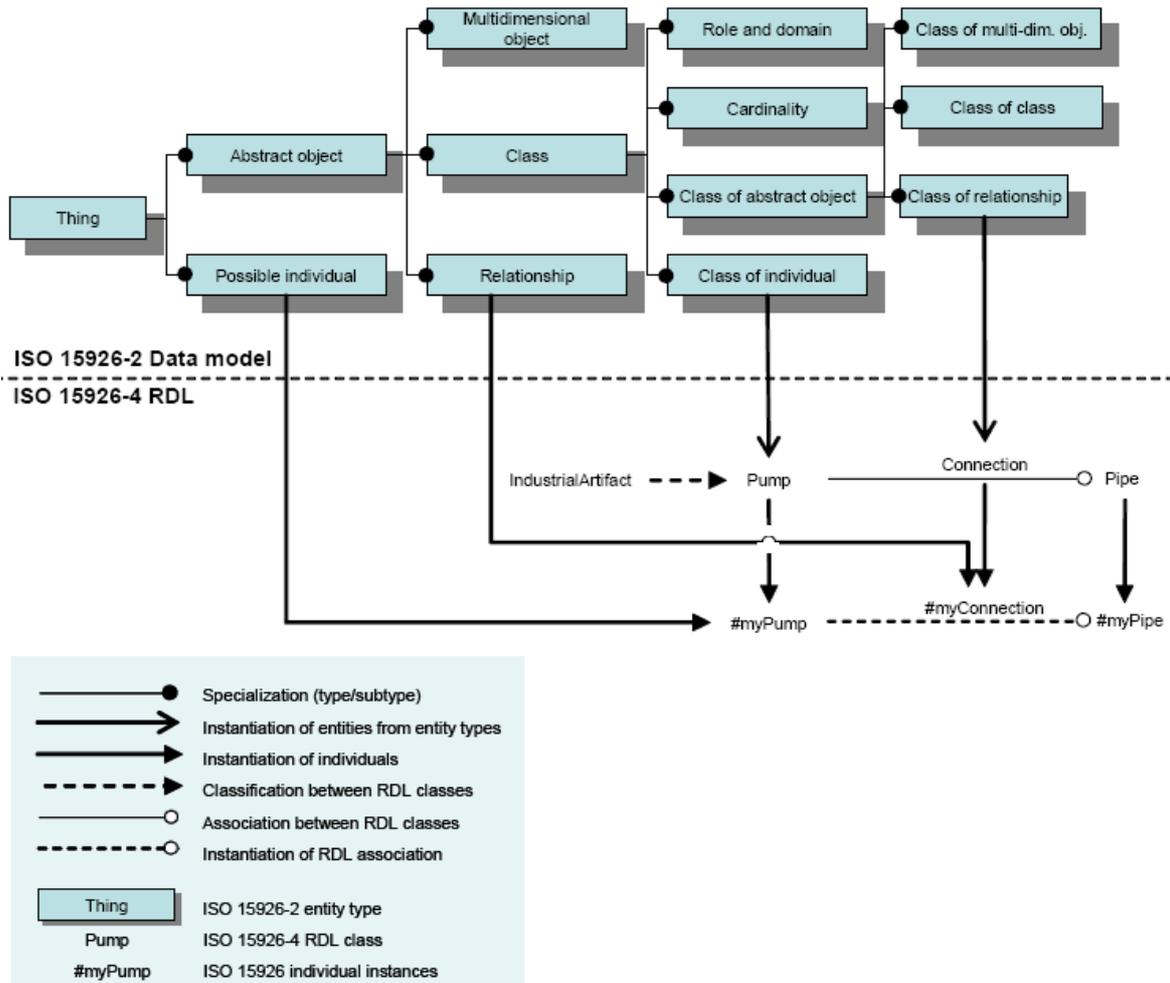


Figura 16 – Exemplo de Estrutura ISO-15926-2 e 4. Fonte (GULLA *et al.*, 2006).

Os modelos de equipamentos existentes no EMSO para indústria química e petroquímica foram classificados de acordo com as RDLs da classe *Equipments* da norma. A Figura 17 mostra um trecho do diagrama de classes UML construído com base neste levantamento e que posteriormente é utilizado como fonte para construção do mapeamento da nomenclatura do EMSO com a ISO-15926.

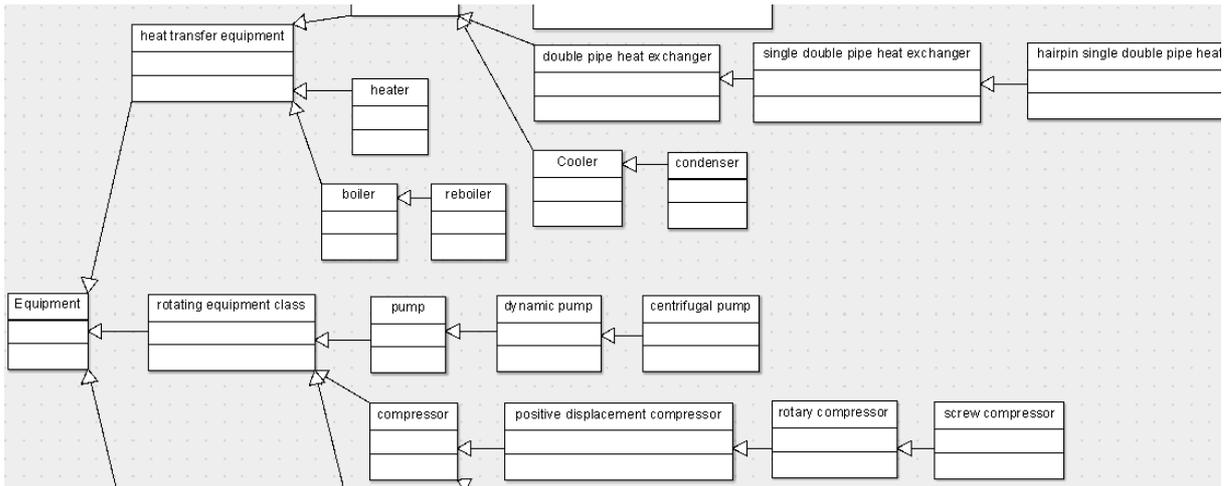


Figura 17 – Classe de Equipamentos.

Identificar os Atributos das Classes e as Relações entre as Classes:

Quando um modelo conceitual é definido, os atributos precisam ser definidos no nível das classes de objetos e a propagação destas informações feitas por herança entre subclasses e superclasses.

As propriedades contidas nos modelos de equipamentos do EMSO foram identificadas dentro da classe *Property* da norma. Este estudo serviu como base para o mapeamento entre os termos do EMSO e a ISO-15926.

Este trabalho identificou os seguintes atributos como pendentes na planilha fonte *Property* de RDLs da ISO-15926:

- *normal operating outlet pressure*
- *normal operating inlet temperature*
- *normal operating outlet temperature*

Para contornar este problema a propriedade “*normal operating outlet pressure*” foi inserida no mapeamento. No caso da Temperatura, foram utilizadas as terminologias

“inlet temperature” e *“outlet temperature”*. Após a revisão da norma e inclusão destes itens, a atualização das planilhas de mapeamento será suficiente para adequar o aplicativo **ORION** a nova versão da ISO-15926-4 sem qualquer necessidade de ajustes na codificação.

O diagrama UML, Figura 18, identifica todas as classes da ISO-15926-4 necessárias para interpretação do ambiente de simulação.

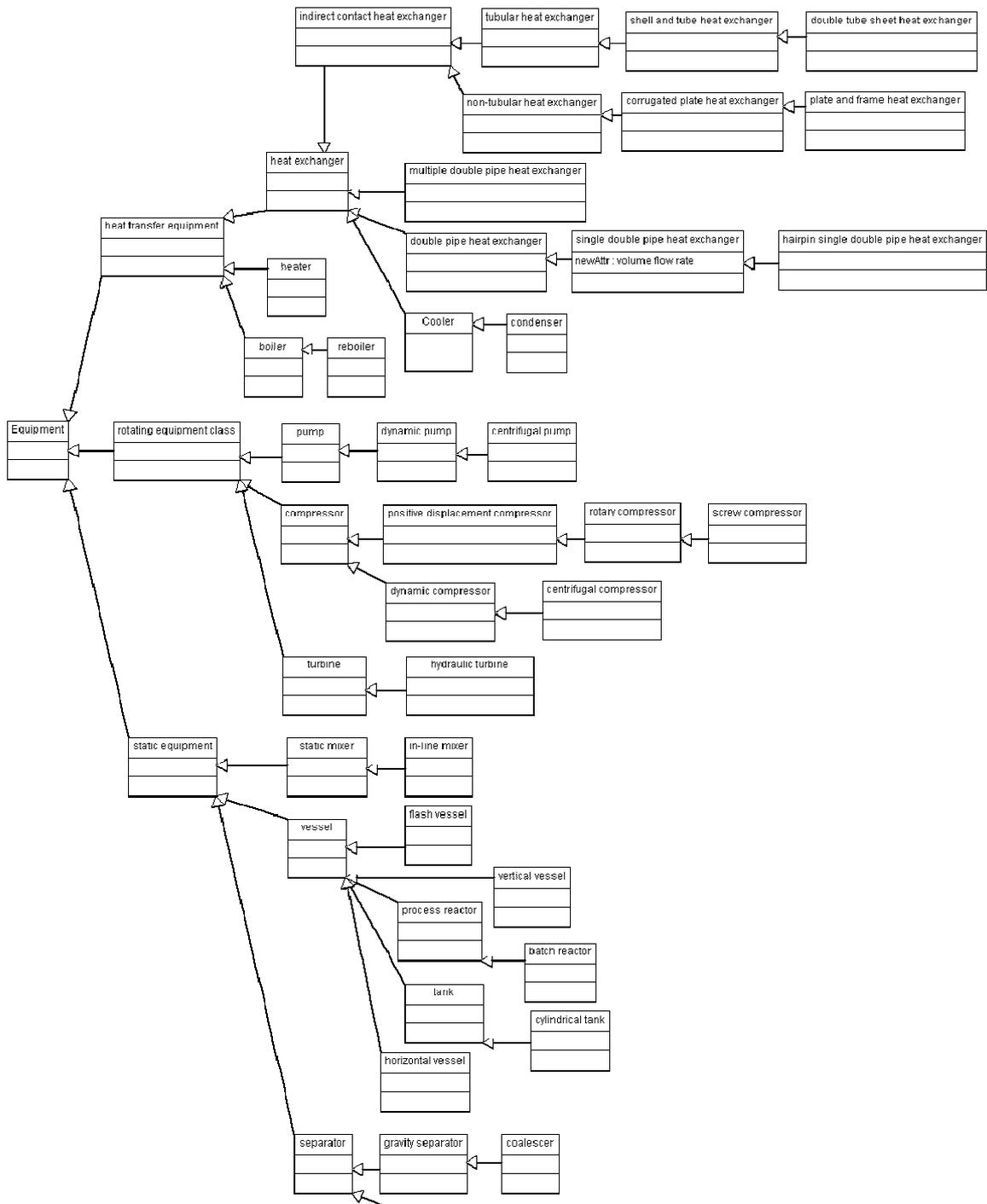


Figura 18 – Diagrama UML – Classes ISO-15926-4 para Simulação de Processos.

3.4 LEITURA DOS DADOS DO SIMULADOR

A leitura correta dos dados de saída do simulador está baseada no entendimento de sua modelagem, explorando a forma como os dados são representados. A partir deste

entendimento, devem ser selecionadas as informações de interesse para a fase de projeto básico.

Neste trabalho, o ambiente de simulação utilizado é o *software* EMSO. O EMSO está estruturado para estudar processos dinâmicos ou em estado estacionário em geral. Sua linguagem de modelagem é completamente orientada a objeto e formada por três componentes principais: *Flowsheets*, *Devices* e *Models*. O *Flowsheet* é o diagrama que representa o processo em análise. Neste diagrama, cada componente do processo é chamado de *Device*. Quando os *Devices* são conectados entre si dá-se origem ao *Flowsheet*. Cada *Device* possui uma descrição matemática associada, denominada *Model*. Um único *Model* pode estar associado a diversos *Devices*, que possuem a mesma estrutura, mas diferem em valores de parâmetros, conexões ou especificações (SOARES & SECCHI, 2003).

Neste contexto, a leitura dos dados de saída do simulador EMSO visa obter as especificações, parâmetros e conexões dos *Devices* pertencentes a cada *Flowsheet*, assim como as especificações das correntes que proporcionam as conexões entre os *Devices*.

No Diagrama de Processo da Figura 19, cada componente da planta destacado em vermelho é representado por um *Device* no *Flowsheet* modelado no EMSO.

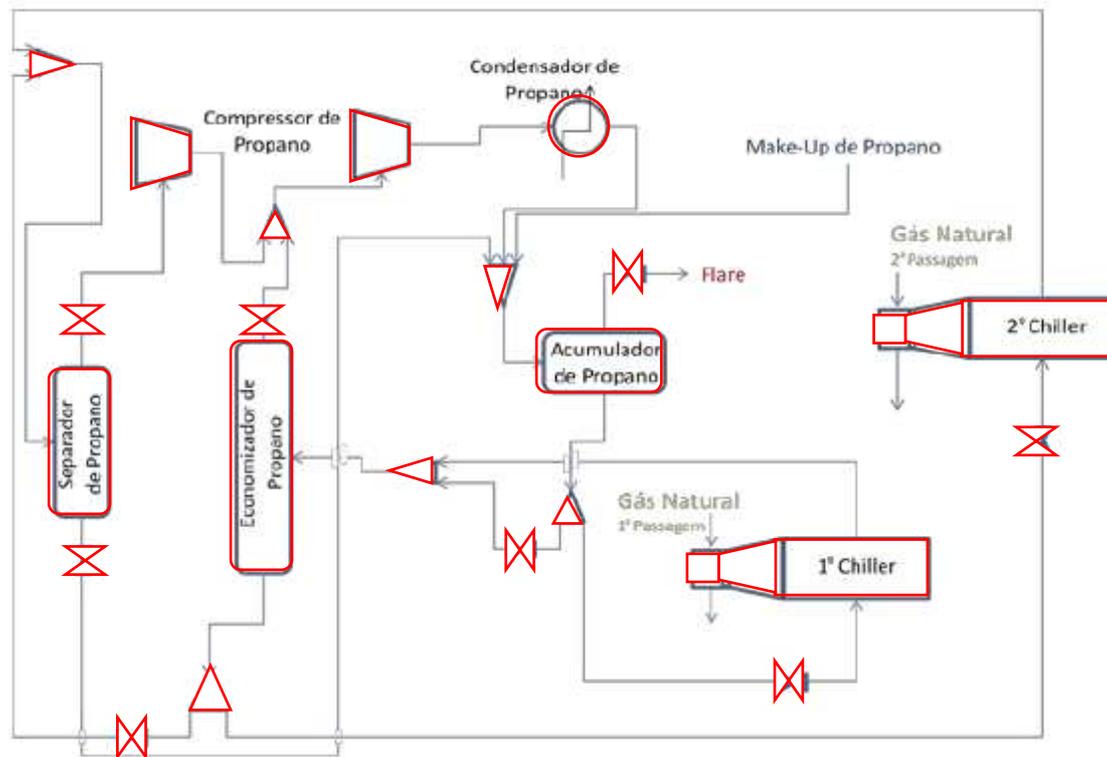


Figura 19 – Componentes do Diagrama de Processo. Fonte (FERNANDES, 2009).

Um *Flowsheet* é modelado segundo uma linguagem baseada em componentes, estruturada pela declaração de *Devices* e *Connections*, ilustrados na Figura 20. Estas informações são essenciais para fase de projeto, uma vez que fornecem a topologia do processo e dão origem ao documento Fluxograma de Processo. As informações de conexão entre os componentes também indicam a consistência e sentido de propagação das informações.

```

DEVICES
pidPdis1 as PIDIncr; #Pressure controller: First Discharge Tank
pidPdis2 as PIDIncr; #Pressure controller: Second Discharge Tank
pidL as PIDIncr; #Level controller: Propane Separator
pidP as PIDIncr; #Pressure controller: Propane Separator
pidL_eco as PIDIncr; #Level controller: Propane Economizer
pidP_eco as PIDIncr; #Pressure controller: Propane Economizer
pidP_ac as PIDIncr; #Pressure controller: Propane Accumulator
pidT_chiller_1 as PIDIncr; #Temperature controller: First Chiller
pidT_chiller_2 as PIDIncr; #Temperature controller: Second Chiller
f1 as flash;
C1 as Centrifugal_Compressor;
W1 as work_source;
C2 as Centrifugal_Compressor;
W2 as work_source;
Q as energy_source (Brief = "Heat supplied");
Discharge_1 as Tank_Discharge (Brief = "Discharge Tank for first stage compression");
Mixer_1 as Mixer (Brief = "Mixer for second stage compression");
Discharge_2 as Tank_Discharge (Brief = "Discharge Tank for second stage compression");
Condenser_1 as Heatex_LMTD (Brief = "After the second stage compression");
WaterCold as source;
Mixer_2 as Mixer_liq (Brief = "Mixer for second stage compression");
Propane_economizer as flash;
s3 as source;
Q_3 as energy_source;
Q_2 as energy_source;
Propane_accumulator as flash;
Splitter_1 as Splitter;
Mixer_3 as Mixer;
Valve_1 as Valve;
Chiller_1 as Heatex_LMTD_2;
s4 as source;
Valve_2 as Valve;
Chiller_2 as Heatex_LMTD_2;
Temperature_Control_1 as Temperature_Control;

CONNECTIONS
Chiller_2.OutletCold to f1.Inlet;
W1.OutletWork to C1.WorkIn;
f1.OutletV to C1.Inlet;
Q.OutletQ to f1.InletQ;
C1.Outlet to Discharge_1.Inlet;
Discharge_1.Outlet to Mixer_1.Inlet(1);
Propane_economizer.OutletV to Mixer_1.Inlet(2);
Mixer_1.Outlet to C2.Inlet;
W2.OutletWork to C2.WorkIn;
C2.Outlet to Discharge_2.Inlet;
Discharge_2.Outlet to Condenser_1.InletHot;
WaterCold.Outlet to Condenser_1.InletCold;
Condenser_1.OutletHot to Mixer_2.Inlet(1);
f1.OutletL to Mixer_2.Inlet(2);
s3.Outlet to Mixer_2.Inlet(3);
Mixer_3.Outlet to Propane_economizer.Inlet;
Q_3.OutletQ to Propane_economizer.InletQ;
Mixer_2.Outlet to Propane_accumulator.Inlet;
Q_2.OutletQ to Propane_accumulator.InletQ;
Propane_accumulator.OutletL to Splitter_1.Inlet;
Splitter_1.Outlet(1) to Mixer_3.Inlet(1);
Splitter_1.Outlet(2) to Valve_1.Inlet;
Valve_1.Outlet to Chiller_1.InletCold;
s4.Outlet to Chiller_1.InletHot;
Chiller_1.OutletCold to Mixer_3.Inlet(2);
Propane_economizer.OutletL to Valve_2.Inlet;
Chiller_1.OutletHot to Temperature_Control_1.Inlet;
Temperature_Control_1.Outlet to Chiller_2.InletHot;
Valve_2.Outlet to Chiller_2.InletCold;

```

Figura 20 – Arquitetura de Objetos no EMSO. Fonte (FERNANDES, 2009).

Um *Model* é descrito segundo uma linguagem baseada em equações e estruturada pela declaração de parâmetros, variáveis, equações e condições iniciais (Figura 21). Uma vez que os valores das variáveis descrevem o comportamento do processo que está sendo modelado, eles são considerados os resultados da simulação comparados aos parâmetros. Estas informações serão consideradas para fase seguinte de projeto de modo a compor documentos do tipo Folha de Dados.

```

FlowSheet Propane_Refrigeration_Cycle

PARAMETERS
PP as Plugin(Brief="Physical Properties",Type="PP",
Components = ["ethane", "propane", "water", "methane"],
LiquidModel = "PR",
VapourModel = "PR");
NComp as Integer;
pi as positive (Brief="Pi value", Default=3.141593, Symbol="\pi");
#Maximum and minimum values for Propane Separator
PMin as pressure;
PMax as pressure;
FVMin as flow_mol;
FVMax as flow_mol;
FLMax as flow_mol;
FLMin as flow_mol;
LMax as length;
LMin as length;

#Maximum and minimum values for Propane Accumulator
P_acMin as pressure;
P_acMax as pressure;
FV_acMin as flow_mol;
FV_acMax as flow_mol;

#Maximum and minimum values for Propane Economizer
P_ecoMin as pressure;
P_ecoMax as pressure;
L_ecoMax as length;
L_ecoMin as length;
FV_ecoMin as flow_mol;
FV_ecoMax as flow_mol;
FL_ecoMax as flow_mol;
FL_ecoMin as flow_mol;

VARIABLES
P_ad as Real (Brief="Dimensionless pressure Propane Separator");
Pdis1_ad as Real (Brief="Dimensionless pressure Discharge Tank First Stage");
L_ad as Real (Brief="Dimensionless level Propane Separator");
Pdis2_ad as Real (Brief="Dimensionless pressure Discharge Tank Second Stage");
P_eco_ad as Real (Brief="Dimensionless pressure Economizer");
L_eco_ad as Real (Brief="Dimensionless level Economizer");
P_ac_ad as Real (Brief="Dimensionless pressure Acumulator");
T_chiller_1_ad as Real (Brief="Dimensionless Temperature for First Chiller");
T_chiller_2_ad as Real (Brief="Dimensionless Temperature for Second Chiller");

#Set-Points
Psp as Real;
Lsp as Real;
Pdis1_sp as Real;
Pdis2_sp as Real;
P_eco_sp as Real;
L_eco_sp as Real;
P_ac_sp as Real;
T_chiller_1_sp as Real;
T_chiller_2_sp as Real;

#Controller parameters
KCP as Real;
KCL as Real;
intTL as Real;
intTP as Real;
KcPdis1 as Real;
KcPdis2 as Real;
intTPdis1 as Real;
intTPdis2 as Real;
KcPeco as Real;
KcLeco as Real;
intTLeco as Real;
intTPeco as Real;

```

Figura 21 – Parâmetros e Variáveis do Processo no EMSO. Fonte (Fernandes, 2009).

A topologia do processo é indicada na *worksheet* principal em uma tabela chamada “Connections”, conforme Tabela 1. Nesta mesma *worksheet*, estão indicados os componentes das correntes na aba de “Plugins”, Tabela 2.

Tabela 1 – Tabela de conexões do arquivo de saída do EMSO.

Connections	
From	To
Chiller_2.OutletCold	fl.Inlet
W1.OutletWork	C1.WorkIn
fl.OutletV	C1.Inlet
Q.OutletQ	fl.InletQ
C1.Outlet	Discharge_1.Inle-
Discharge_1.Outlet	Mixer_1.Inlet
Propane_economizer.OutletV	Mixer_1.Inlet
Mixer_1.Outlet	C2.Inlet
W2.OutletWork	C2.WorkIn
C2.Outlet	Discharge_2.Inlet
Discharge_2.Outlet	Condenser_1.InletHot

Tabela 2 - Tabela de componentes das correntes do arquivo de saída do EMSO.

Plugins	
Name	Value
Brief	Physical Properties
Type	PP
Components	ethane
	propane
	water
	methane
LiquidModel	PR
VapourModel	PR

Para cada *Device*, ou seja, equipamento, corrente ou instrumento do sistema, são criadas worksheets contendo o resultado da simulação. No caso de um equipamento, são informados os parâmetros dimensionais e os valores de entrada e saída das variáveis, ilustrado na Tabela 3.

Tabela 3 – Tabela de Resultados por Device do arquivo de saída do EMSO.

Name	UOM		
time	s	0,0000	18000,0000
fl			
NComp			
V	m³	7,0000	7,0000
Mw(1)	kG/kmol	30,0694	30,0694
Mw(2)	kg/kmol	44,0962	44,0962
Mw(3)	kg/kmol	18,0152	18,0152
Mw(4)	kg/kmol	16,0426	16,0426
diameter	m	0,5000	0,5000
inlet			
NComp			
F	kmol/h	735,8230	695,5480
T	K	251,0900	252,2430
P	atm	2,0156	2,0156
h	kJ/kmol	-3716,7500	-3639,0800
v		1,0000	1,0000
z(1)		0,0026	0,0027
z(2)		0,9974	0,9973
z(3)		0,0000	0,0000
z(4)		0,0000	0,0000
OutletL			
NComp			
F	kmol/h	3,2801	0,6948
T	K	248,2660	248,2660
P	atm	2,0156	2,0156
h	kJ/kmol	-	-
v		0,0000	0,0000
z(1)		0,0005	0,0005
z(2)		0,9995	0,9995
z(3)		0,0000	0,0000
z(4)		0,0000	0,0000

Neste trabalho foi implementada um rotina denominada *XMLExcelFileRead.vb* em linguagem VB .NET, cuja idéia principal é receber o apontamento do arquivo de saída do simulador e armazenar os dados de interesse em memória no aplicativo, por meio de um objeto do tipo *DataSet*. O *DataSet* é um banco de dados em memória que armazena as informações em objetos do tipo *DataTable*.

Um objeto *DataTable* é criado para armazenar todas as informações da etapa de simulação. O código varre o documento XML em busca de cada uma das tabelas citadas acima, lê e armazena as informações de cada uma delas, considerando um tag de *Device* (equipamento) por linha. Uma informação importante contida no XML é

a indicação do tipo de simulação, diferenciado entre estática ou dinâmica. Dependendo do tipo de simulação serão coletadas informações em posições diferentes do XML.

Durante o desenvolvimento desta etapa, foi identificada a ausência da informação sobre o tipo do equipamento no arquivo XML de saída do simulador, Figura 22. Apenas a identificação do tag dos equipamentos é fornecida. O tipo de equipamento é uma informação fundamental para o andamento do projeto e será solicitada sua inclusão do arquivo de saída do EMSO como melhoria do *software*.

```
        <Data ss:Type="String">Tank3</Data>
    </Cell>
</Row>
- <Row>
- <Cell ss:StyleID="BoldColumn">
    <Data ss:Type="String">k</Data>
</Cell>
- <Cell ss:StyleID="UnitCool">
    <Data ss:Type="String">m^2.5/h</Data>
</Cell>
- <Cell ss:StyleID="DadosCinza">
    <Data ss:Type="Number">4</Data>
</Cell>
```

Figura 22 – Extração do Arquivo XML de saída do EMSO.

3.5 MAPEAMENTO DOS MODELOS DO SIMULADOR PARA AS CLASSES DA ISO-15926-4

Ter o resultado da simulação lido automaticamente pelo aplicativo é apenas a primeira etapa deste desenvolvimento. Para geração do produto final, um arquivo de transferência de dados padronizado, é necessário que cada tipo de informação oriunda do simulador EMSO esteja identificado dentro da estrutura de classes do Padrão ISO-15926.

O mapeamento estabelece as conexões entre os conceitos equivalentes dos modelos de dados diferentes. A partir daí pode-se estabelecer uma relação de coerência entre diversos esquemas e obter exportação e importação de dados *interschema*.

A Parte 2 da ISO-15926 especifica o modelo de dados conceitual para representação computacional das informações técnicas de plantas de processo (ISO 15926-2, 2003). Ou seja, é uma ontologia onde é definido o que existe dentro de um domínio: neste caso uma planta de processo.

O modelo de dados consiste numa hierarquia universal de subtipo e supertipos. A Figura 23 mostra a raiz principal que é chamada “*Thing*” e subdividida em “*Possible individual*” e “*Abstract Object*”.

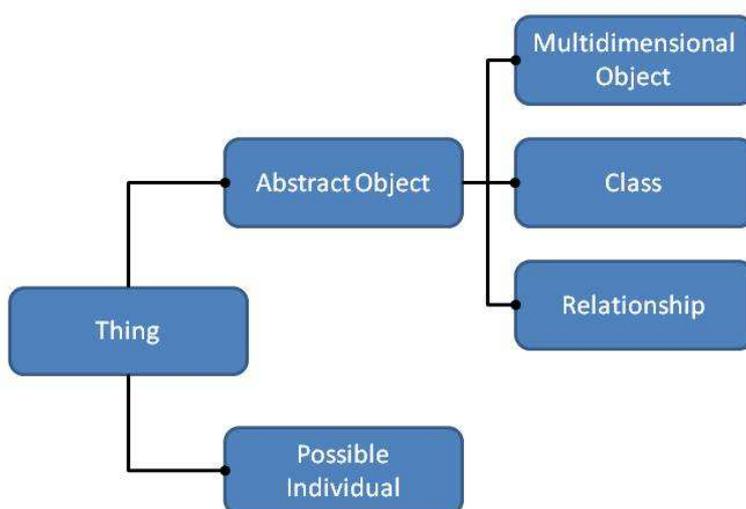


Figura 23 – Modelo Conceitual ISO-15926-2.

No exemplo de fluxograma citado no item 3.4, cada componente da planta identificado por um tag é considerado um *Device* no EMSO e dentro da ontologia da ISO-15926 é considerado um “*possible_individual*”, já que existem no tempo e no espaço do domínio. Já a classificação destes componentes da planta como equipamentos, correntes ou instrumentos é definida como um “*abstract_object*”. A Figura 24 ilustra os conceitos fundamentais desta ontologia aplicados ao exemplo. O Tag B-002 é considerado um “*possible_individual*” classificado como Bomba Centrífuga, que é subclasse de Bomba. As informações de entrada e saída da B-001 são consideradas “*connection_of_individual*” e suas propriedades físicas, como a temperatura de 25°C, “*property space*”.

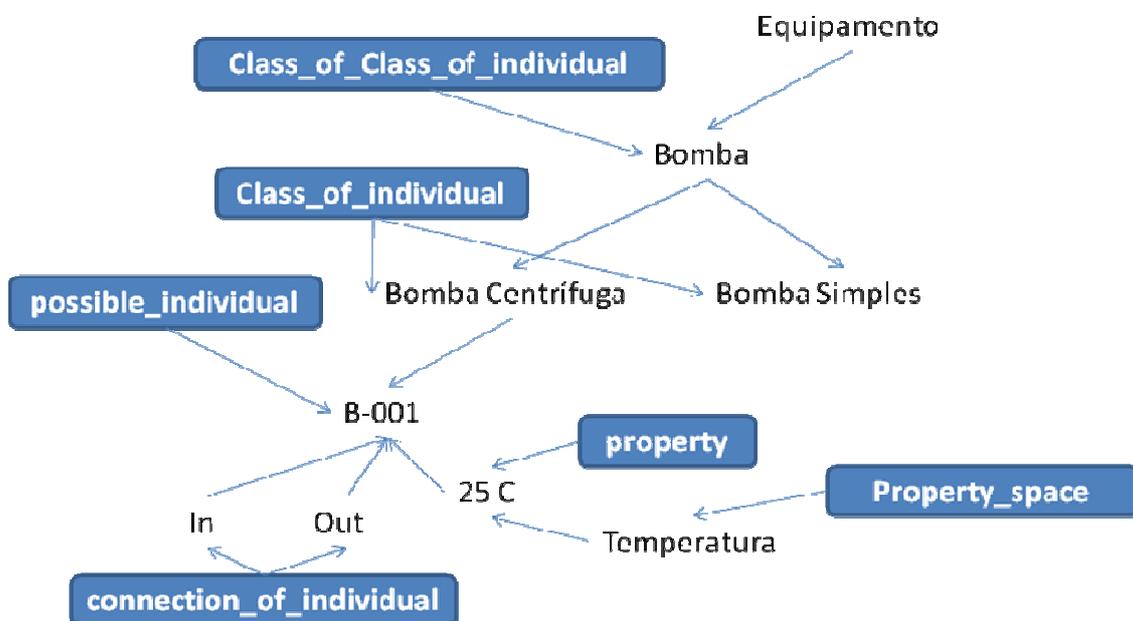


Figura 24 – Exemplo da aplicação dos conceitos da ISO-15926-2.

A Parte 4 da ISO-15926 fornece uma biblioteca de dados de referência (RDL) onde as classes principais estão definidas, conforme listado na Tabela 4. Apenas com a definição da ontologia na Parte 2 não seria possível padronizar um arquivo de intercâmbio de dados, é necessário ter um vocabulário controlado. A RDL é fornecida pela ISO-15926 em formato de planilhas Excel, sendo uma para cada classe principal.

Tabela 4 – RDLs ISO-15926-4 (ISO-15926-4, 2007)

name of set	description of set
activity	activities, including physical processes carried out within process plants and engineering activities carried out by people
basics	generic engineering classes which are referenced by other sets, but which are not specific to an engineering discipline
class of class	classifications of classes for information management purposes
connection material	equipment items and features of equipment items which are involved in the making of process connections
electrical	electrical equipment items including motors, generators, uninterruptible power supplies and transmission and distribution equipment
encoded information	languages and formats for information
control function	functions implemented by automatic control systems
heat transfer	heat transfer equipment
information	document types, including documents which specify process plant operations, and identifier types
instrumentation	equipment items involved in monitoring, communications, recoding and control
ISO 15926-2 super-classes	ISO 15926-2 entities which are superclasses of reference data items in this part of ISO 15926, or which have reference data items in this part of ISO 15926 as instances
piping	pipes and piping components
property	physical quantities and physical properties possessed by equipment items
protection	insulation (thermal and electrical) and safety systems for the protection of personnel and equipment
solid handling	handling of solid objects, including billets and particulate materials
static equipment	static process equipment, excluding heat exchangers, valves and piping. Within scope are tanks and vessels, reactors, separators, filters and static mixers.
transport	vehicles, and associated civil and marine structures and facilities
uom	units of measure and scales
valve	valves (for the control or prevention of fluid flow)

Deste modo, faz-se necessário relacionar os termos semânticos utilizados na construção do arquivo de saída do EMSO com as respectivas RDLs da Parte 4. No desenvolvimento do **ORION** o mapeamento consiste na construção manual de planilhas que armazenam este relacionamento no formato “De/Para”, baseado em um levantamento e estudo técnico das informações que especificam e caracterizam cada componente da planta de processo nos diferentes modelos de dados dos sistemas de software empregados no projeto. Uma planilha chamada “MAP-EMSO.xls” foi construída contendo o nome da informação no EMSO, o nome da informação na RDL

e a fonte onde esta informação pode ser encontrada. A Tabela 5 e 6 mostram respectivamente o mapeamento para as classes “*property*” e “*uom*”.

Tabela 5 – Mapeamento entre os termos do EMSO e a Classe *Property*.

EMSO	ISO15926-4	File
Fvol	<i>flow rate</i>	C:\ISO15926-4 - Reference Data\property.xls
T	<i>temperature</i>	C:\ISO15926-4 - Reference Data\property.xls
P	<i>pressure</i>	C:\ISO15926-4 - Reference Data\property.xls
Dh	<i>diameter</i>	C:\ISO15926-4 - Reference Data\property.xls
A	<i>area</i>	C:\ISO15926-4 - Reference Data\property.xls
V	<i>volume</i>	C:\ISO15926-4 - Reference Data\property.xls
h	<i>height</i>	C:\ISO15926-4 - Reference Data\property.xls

Tabela 6 - Mapeamento entre os termos do EMSO e a Classe *UOM*.

EMSO	ISO15926-4	File
m³/h	<i>cubic meter per hour</i>	C:\ISO15926-4 - Reference Data\uom.xls
K	<i>kelvin</i>	C:\ISO15926-4 - Reference Data\uom.xls
kPa	<i>kilopascal</i>	C:\ISO15926-4 - Reference Data\uom.xls
m²	<i>square meter</i>	C:\ISO15926-4 - Reference Data\uom.xls
m³	<i>cubic meter</i>	C:\ISO15926-4 - Reference Data\uom.xls
m	<i>meter</i>	C:\ISO15926-4 - Reference Data\uom.xls

Por exemplo, no arquivo de saída do simulador de processos EMSO está registrado o valor da pressão de entrada e de saída de um equipamento. Para que esta informação seja propagada corretamente para fase de Projeto Básico para compor a Folha de Especificação (FD) do equipamento, é necessário que a ambigüidade do termo seja removida. Na FD do equipamento existe a informação de pressão de operação, pressão de projeto e até mesmo os limites de pressão máxima e mínima para questões de alarmes ou controle de processos. Quando uma informação de pressão de entrada e saída de um equipamento da fase de simulação vai alimentar uma, é necessário informar que tipo de pressão este valor está representando. O mapeamento é utilizado com o objetivo de indicar que tipo de informação o modelo de origem faz referência com precisão, clareza e ausência de multiplicidade de interpretações.

A classe *MapISO15926.vb* do **ORION** acessa a planilha “MAP-EMSO.xls” e alimenta uma tabela do *Dataset* com estas informações. Em seguida, a classe *ISOReferenceData.vb* cria novas colunas na tabela que contém as informações da

simulação para armazenar o nome da RDL correspondente ao termo. No caso do tipo de equipamento, os valores desta coluna são acessados para cada linha (que representa um tag), o tipo é procurado na tabela MAP-EMSO para buscar a planilha fonte onde se encontra a classificação do termo na ISO-15926 Partes 2 e 4 e o valor é atualizado para cada linha. No caso das propriedades dos *Devices* (tags de equipamentos ou correntes), os valores numéricos dos atributos não são as informações de interesse, mas sim o nome do atributo em questão. O nome de cada coluna de atributo do tag é procurado na Tabela MAP-EMSO, a respectiva planilha fonte é acessada e a classificação do termo na ISO-15926 Partes 2 e 4 é atualizada com o mesmo valor para todas as linhas.

3.6 GERAÇÃO DO ARQUIVO DE TRANSFERÊNCIA DE DADOS NO PADRÃO ISO-15926

Nesta etapa do desenvolvimento é construído o arquivo neutro e padronizado para executar a transferência dos dados de simulação para a ferramenta de projeto. Uma vez que o aplicativo **ORION** faça a transcrição da saída do simulador de processos EMSO para a linguagem padronizada pela norma ISO-15926, esta poderá ser lida por qualquer ferramenta de *software* para projeto que seja capaz de interpretar este padrão ou que esteja mapeada para o aplicativo **ORION**. Este arquivo de saída do **ORION** é um XML construído com linguagem de marcação OWL para representação do modelo de ontologia da ISO-15926. Inicialmente uma breve descrição é apresentada, baseada em estudos da literatura, de como a linguagem OWL pode ser utilizada para representação do modelo conceitual da ISO-15926-2. Depois, os conceitos e superclasses deste modelo conceitual, que se aplicam às fases do ciclo de vida da planta estudada neste trabalho, são discutidos. E, finalmente, é apresentada a classe *XMLISOApplication.vb* que executa todo o conteúdo exposto neste capítulo.

A representação deste modelo de dados em OWL utiliza o conceito *RDFS:Class* com subclasse *OWL:Class*. Além disso, os meta-modelos *OWL:DatatypeProperty* e *OWL:ObjectProperty* são subclasses de *RDFS:Property* e instâncias de *OWL:Class*. A entidade do tipo *Class_of_individual* da ISO-15926-2 e seus subtipos são modelados como subclasses da *OWL:Class* e instâncias da subclasse *Class_of_abstract_object*. A entidade do tipo *Class_of_relationship* e seus subtipos são modelados como subclasses da *OWL:ObjectProperty* e instâncias da subclasse *Class_of_class_of_relationship*. A entidade do tipo *Possible_individual* e seus subtipos

são representados como subclasses de *OWL:Thing* e instâncias da subclasse *Class_of_individual*, o tipo de entidade *Relationship* e subtipos são modelados como propriedades em OWL e instâncias da subclasse pertinente de *Class_of_relationship*. As classes e instâncias na norma ISO-15926-4 são tratadas de maneira similar. Classes RDL são modeladas como subclasses de *OWL:Thing* e instâncias do tipo de entidade competente da ISO-15926-2. Qualquer especialização de uma classe RDL é modelada como uma subclasse da classe RDL. Relacionamentos em ISO-15926-4 são representados como instâncias da subclasse pertinente de *Class_of_relationship* (GULLA *et al.*, 2006).

Ao longo do ciclo de vida das plantas de processo, os equipamentos passam por diversas substituições para manutenção e garantia da operação da unidade. Esta é uma realidade freqüente em unidades de processo e as substituições ocorrem por questões de avanços tecnológicos do maquinário empregado ou por questões de desgaste das peças. Na fase de projeto um equipamento é identificado e projetado utilizando-se um Tag, que não pode ser considerado o objeto materializado, mas sim uma função. Na fase de construção um item material é instalado para executar a função B-001, considerado como exemplo de tag definido para uma bomba na fase de projeto, e terá um número de série fornecido pelo seu fabricante. Na fase de operação e manutenção da planta o desgaste das peças pode demandar a substituição deste equipamento, o que implicará no recebimento de um novo item identificado por outro número de série fornecido pelo seu fabricante. A equipe de engenharia permanece com o processo inalterado porque tem seus trabalhos baseados na função B-001, porém para a equipe de manutenção da unidade a alteração fica visível.

O modelo de dados da Parte 2 da ISO-15926 foi idealizado para suportar esta evolução dos dados ao longo do tempo. Um *physical_object* é um *possible_individual* considerado uma distribuição de matéria e/ou energia no tempo e no espaço. São considerados subtipos dos *physical_object* (ISO-15926-2, 2003):

- *Materialised_physical_object*: São objetos concretos como uma bomba mecânica com o número de série do fabricante;
- *Functional_physical_object*: É a função pretendida para o indivíduo, que é mantida mesmo que todos os componentes materiais do objeto tenham sido alterados. Exemplo: Tag da bomba B-001;

- *Stream*: Material ou energia que se move ao longo de um caminho, onde este caminho é a base para identificação e pode ser restringido. Exemplo: A Nafta que flui na tubulação de uma unidade de destilação de petróleo;
- *Spatial_location*: A identidade está baseada na continuidade de uma posição relativa. Exemplo: Licença para uma área *offshore*.

Para transposição dos componentes da simulação para a ferramenta de projeto de engenharia são utilizados somente os subtipos *Functional_physical_object* e *Stream* da classe *physical_object* para representação dos Tags dos equipamentos principais, informação de sua classificação segundo as RDLs da Parte 4 da norma e das correntes do processo. Neste caso, a serialização do XML em OWL é representada como:

```
< owl:PossibleIndividual ID = "B-001" >
  < rdf:type rdf:resource = "&iso15926_2;#FunctionalPhysicalObject" / >
  < rdf:type rdf:resource = "&iso15926_4;#CentrifugalPump" / >
< / owl:PossibleIndividual >
```

Uma informação fundamental para garantir a correta transformação do *Flowsheet* (diagrama do processo em análise) do EMSO em um Fluxograma de Processo na fase de projeto básico é a conectividade entre os componentes da planta, denominada Topologia do Processo. No modelo da ISO-15926-2, a *connection_of_individual* é um relacionamento que indica que a matéria, a energia ou ambas podem ser transferidas entre os membros do *possible_individual* que estão ligados, direta ou indiretamente.

As propriedades dos indivíduos do modelo ISO-15926-2 são traduzidas para propriedades do OWL como *OWL:ObjectProperty* e *OWL:DataTypeProperty* (SKJÆVELAND & KLÜWER, 2010). A *connection_of_individual* é considerada como uma propriedade no modelo de dados e é assim declarada como *OWL:ObjectProperty*. Considere que a bomba B-001 tem conexão com um bocal de entrada que possui um Tag N-001 e com um bocal de saída N-002. A serialização do XML está a seguir representada para este exemplo:

```
<owl:Class rdf:ID="ConnectionOfIndividual">
  <rdfs:subClassOf rdf:resource="#Relationship" />
</owl:Class>
```

```

< owl:PossibleIndividual >
  < rdf:type rdf:resource = "&iso15926_4;#CentrifugalPump" / >
  < iso15926_2:partOf rdf:resource = "#B-001" / >
  < iso15926_2:ConnectionOfIndividual>
    < iso15926_2:hasSide1 rdf:resource = "#N-001" / >
    < iso15926_2:hasSide2 rdf:resource = "#N-002" / >
  < / iso15926_2: ConnectionOfIndividual >
< / owl:PossibleIndividual >

```

Para completar a transferência dos dados de simulação é necessário transcrever as condições operacionais dos componentes da planta. Segundo BATRES *et al.* (2007), ontologia superior apóia a idéia de que os objetos físicos e atividades não devem ser autorizados a definir grandezas físicas (3 kg, 5 m, etc.), como atributos, pois uma quantidade física não é uma propriedade inerente de um objeto. O mapeamento entre um objeto físico e uma quantidade de temperatura, por exemplo, pode ser definido como uma instância da *class_of_indirect_property*. Esta classe é implementada como uma subclasse de *OWL:FunctionalProperty*, cujo domínio é dado por membros da *class_of_individual* e cuja escala é dada pelos membros da *property_space*. No que diz respeito às unidades de medida, a abordagem da ISO 15926-2 é classificar a quantificação da propriedade, em outras palavras, uma relação de classificação é utilizada para mapear uma instância de quantificação de propriedade para uma instância de escala. A abordagem utilizada aqui define a escala como uma *OWL:Property*.

SKJÆVELAND & KLÜWER (2010) fornecem como resultado de seu trabalho "A *mapping of ISO 15926-2 in EXPRESS to OWL*" a serialização de um XML em OWL para transcrição da nomenclatura do código EXPRESS, base do modelo ISO-15926-2, para OWL 1.0. Este arquivo XML foi utilizado como base para construção do arquivo de saída do aplicativo **ORION**.

3.7 MAPEAMENTO DOS MODELOS DA FERRAMENTA CAE PARA AS CLASSES DA ISO-15926-4

O COMOS®, da *COMOS Industry Solutions*, é uma solução de *software* orientado a objeto para representação holística e uniforme de uma máquina ou unidade industrial. A representação holística aplica-se ao chamado Gerenciamento de Informações do Ciclo de Vida de Ativos (*Life Cycle Asset Information Management*), permitindo implementação de projetos de Engenharia Básica, FEED e Detalhamento assim como acompanhar as demais fases do ciclo de vida da planta de processo. A ferramenta é utilizada de forma global e interdisciplinar nas áreas de engenharia de processos, tubulação/isométricos, instrumentação e controle industrial, elétrica, planejamento funcional, manutenção e gerenciamento de documentos do projeto.

O *software* utiliza e disponibiliza o conceito de biblioteca de objetos de engenharia, onde a idéia de objeto está baseada na realidade. A consideração principal está na idéia de uniformizar e generalizar a descrição da aplicação dos componentes que realmente existem na planta. Por exemplo, uma bomba tem muitos aspectos que quando vistos em conjunto constituem a imagem global da bomba. Cada disciplina técnica traz sua visão própria sobre cada componente visualizado em conjunto na unidade industrial e o modelo uniforme deste componente deve convergir para atender todas as áreas. Além disso, sua arquitetura aberta permite a customização desta biblioteca padrão para inclusão de normas e padrões de engenharia específicos de cada setor ou companhia (*COMOS Industry Solutions*, 2010).

Os diagramas de processo são construídos utilizando os modelos disponíveis na biblioteca de objetos de engenharia e a conexão bi-direcional dos gráficos com o banco de dados impedem inconsistências. A Figura 25 mostra a visualização da árvore de componentes da planta, diagrama de processo e uma de suas respectivas folhas de dados de processo de componente e lista de equipamentos na interface do COMOS®.

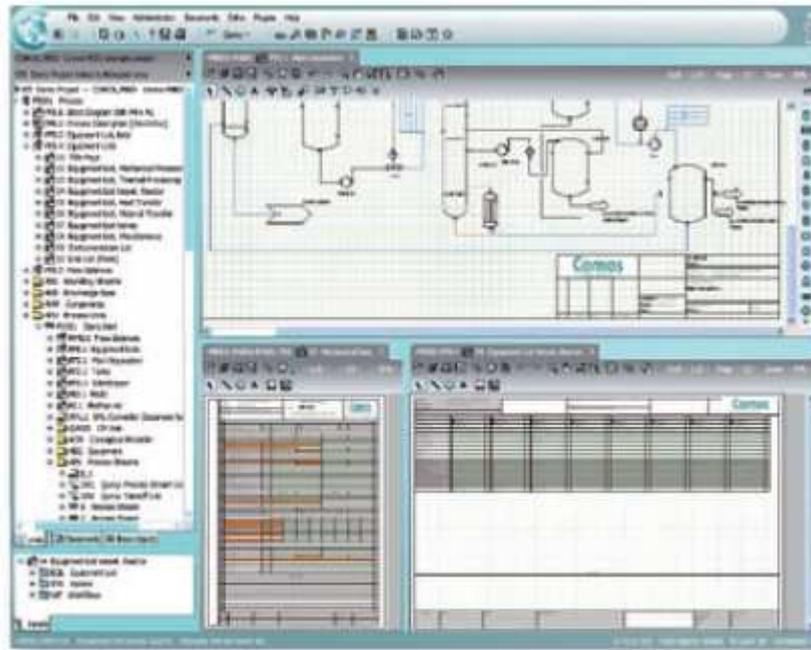


Figura 25 – Interface COMOS®.

Para permitir que o aplicativo **ORION** estabeleça comunicação com o *software* COMOS® e execute o correto intercâmbio de informações, é necessário que a biblioteca de objetos de engenharia do COMOS® esteja mapeada para as fontes de RDL da ISO-15926-4.

O COMOS® é um *software* multidisciplinar que se propõe a atender todas as fases do ciclo de vida da planta. Para fins de organização, sua biblioteca de objetos de engenharia está dividida por áreas de especialidades técnicas e fase de engenharia. Os Fluxogramas de Processo (PFDs) são elaborados no Módulo de Processo do *software*, segunda a estrutura ilustrada na Figura 26, e seus objetos de engenharia encontram-se disponíveis no ramo *@1PE Process Engineering* da biblioteca. Cada Unidade de Processo (APU) contém um único PFD e seus componentes são criados e organizados em duas pastas, uma para correntes de processo (APS) e outra para equipamentos (AEQ).

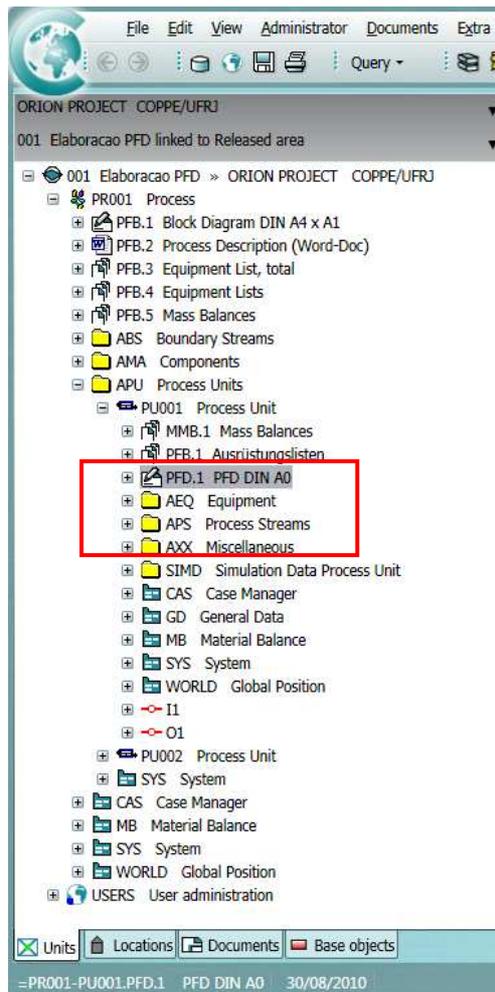


Figura 26 – Módulo de Processo do COMOS®.

Dentro da estrutura do nó *@1PE Process Engineering* da biblioteca tem-se:

- *...|DS Data structures*: Base de objetos do tipo documento, catálogo de *tabs* e atributos do projeto padrão.
- *...|PO Process objects*: Base de objetos do tipo equipamentos, correntes de processo, casos de operação e objetos para interfaces de importação.
- *...|US Unit system*: Categorias para classificação automática de objetos.

Neste caso, a transferência dos *Devices*, que constituem o diagrama de processo do EMSO, em objetos do modelo de dados do COMOS® está baseada no mapeamento da estrutura *@1PE Process Engineering|PO Process objects|EQ Equipment* para uma planilha denominada *MAP-COMOS.xls*, onde está cadastrado o relacionamento das

terminologias desta biblioteca para as RDLs da Parte 4 da ISO-15926, conforme exemplo da Tabela 7. A Figura 27 ilustra a área de equipamentos de processo da biblioteca organizada por função: transferência de calor, transferência de material, processos mecânicos, etc.

Tabela 7 - Mapeamento entre COMOS® e RDLs da ISO-15926-4

COMOS	ISO15926-4	COMANDO
Air Cooler, Heater	heater	Project.CDeviceSystem.CDevices.Item("@1PE").CDevices.Item("PO").CDevices.Item("EQ").
Centrifugal Pumps	centrifugal pump	Project.CDeviceSystem.CDevices.Item("@1PE").CDevices.Item("PO").CDevices.Item("EQ").
Pump	pump	Project.CDeviceSystem.CDevices.Item("@1PE").CDevices.Item("PO").CDevices.Item("EQ").
Centrifugal + Axial	centrifugal compressor	Project.CDeviceSystem.CDevices.Item("@1PE").CDevices.Item("PO").CDevices.Item("EQ").
Compressor	compressor	Project.CDeviceSystem.CDevices.Item("@1PE").CDevices.Item("PO").CDevices.Item("EQ").
Mixer	in-line mixer	Project.CDeviceSystem.CDevices.Item("@1PE").CDevices.Item("PO").CDevices.Item("EQ").
Vessel, vertical	flash vessel	Project.CDeviceSystem.CDevices.Item("@1PE").CDevices.Item("PO").CDevices.Item("EQ").
Reactor	batch reactor	Project.CDeviceSystem.CDevices.Item("@1PE").CDevices.Item("PO").CDevices.Item("EQ").
Vessel, horizontal	horizontal vessel	Project.CDeviceSystem.CDevices.Item("@1PE").CDevices.Item("PO").CDevices.Item("EQ").
Column	distillation column	Project.CDeviceSystem.CDevices.Item("@1PE").CDevices.Item("PO").CDevices.Item("EQ").
Tank	tank	Project.CDeviceSystem.CDevices.Item("@1PE").CDevices.Item("PO").CDevices.Item("EQ").
Battery Limit	bellows unit	Project.CDeviceSystem.CDevices.Item("@1PE").CDevices.Item("PO").CDevices.Item("XX").

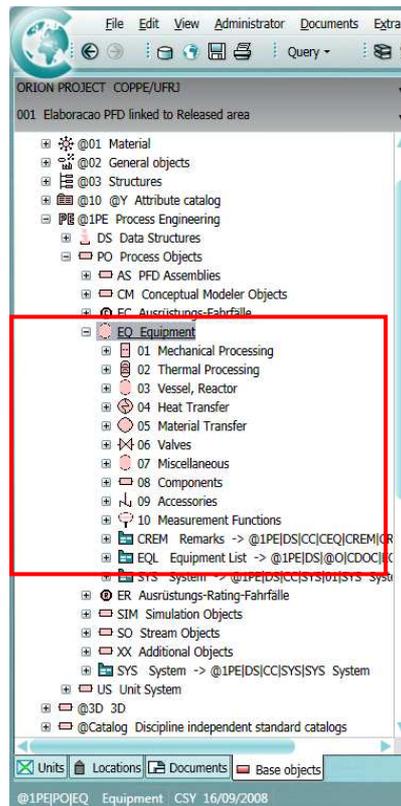


Figura 27 – Equipamentos da Biblioteca de Objetos de Processo.

A elaboração dos PFDs está baseada na criação dos componentes nos diretórios do PFD (AEQ e APS) que fazem a leitura do modelo do objeto na biblioteca através de um *link* para o *Base Object*, e posteriormente o componente é inserido no diagrama por processo *drag-drop*. Sendo assim, para que seja possível a criação automatizada dos objetos no COMOS® foi necessário inserir uma coluna com o apontamento da estrutura hierárquica da biblioteca, além do relacionamento entre os termos que definem os nomes dos componentes do domínio de plantas de processo.

A Figura 28 mostra a indicação do *link* de um trocador de calor desenhado no PFD com o modelo de objeto do *Base Object*.

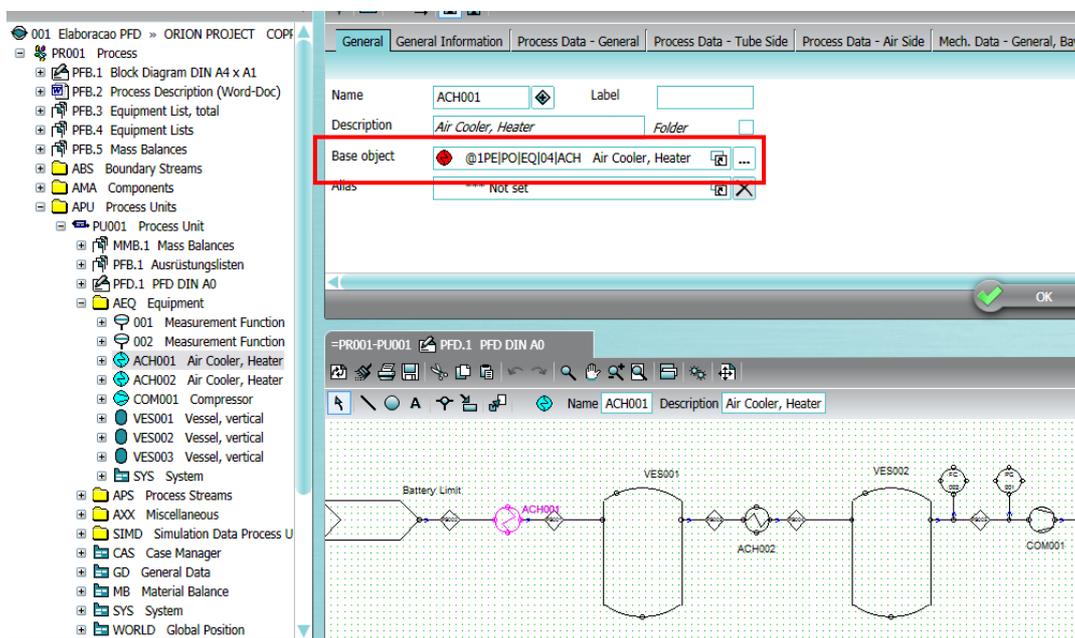


Figura 28 – *Link* do Componente do PFD com o modelo do Objeto na Biblioteca.

Além disso, cada componente do PFD possui um conjunto de propriedades físicas para especificação e geração dos documentos do tipo folha de dados de processo e listas. A transferência das condições de operação do processo oriundas da simulação tem como base o mapeamento entre o catálogo de atributos do módulo de processo do COMOS® e as respectivas RDLs da ISO-15926-4. O perfeito mapeamento destes atributos remove qualquer tipo de ambigüidade no intercâmbio das informações de simulação para os respectivos documentos da fase de projeto básico.

A Figura 29 exemplifica o catálogo de atributos da biblioteca de objetos de engenharia padrão do *software*.

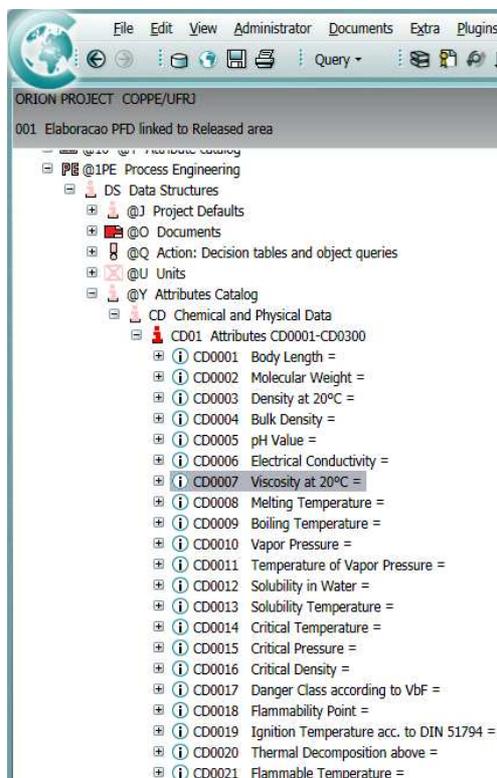


Figura 29 – Catálogo de atributos dos objetos de engenharia.

A classe *MapISO15926.vb* armazena em memória no *DataSet* as informações do mapeamento MAP-COMOS.xls para posterior utilização na leitura dos dados do XML neutro na transferência das informações para o *software* COMOS®.

3.8 LEITURA DOS DADOS NO PADRÃO ISO-15926 PARA A FERRAMENTA CAE

A leitura das informações contidas no arquivo XML de saída do aplicativo **ORION** para o modelo de dados do *software* COMOS® pode ser dividida em duas etapas: interpretação da estrutura do RDF/XML e transferência dos dados através da conexão direta com a interface do *software*.

O COMOS® suporta interface com aplicações externas para troca bidirecional de informações, incluindo interface nativa com ferramentas de simulação e cálculos de

engenharia como Aspen Plus®, CHEMCAD, EBSILON® Professional, PRO/II®, Aspen HYSYS® e UniSim® Design.

Entretanto, este trabalho visa não somente abordar a importância de automatizar o *workflow* de engenharia entre as fases de projeto, mas também discutir a necessidade de padronização do intercâmbio de dados de engenharia para permitir interoperabilidade entre sistemas. Sendo assim, o principal diferencial que levou a escolha do *software* COMOS® para este desenvolvimento foi a arquitetura aberta que suporta o desenvolvimento de códigos para importação/exportação de padrões XML e *Microsoft Office*.

A classe *ImportISOXML.vb* faz a leitura e interpretação do RDF/XML, busca as referências para terminologia e sintaxe na fonte *MAP-COMOS.xls* e executa a criação direta dos componentes do diagrama de processo na estrutura de árvore hierárquica do projeto no COMOS®.

Em uma primeira etapa são identificados todos os componentes da planta denominados *PossibleIndividual* na classificação OWL, assim como a informação do tipo como classificação em RDF para que seja executada a criação dos objetos na estrutura de projeto no COMOS®. As correntes não foram criadas como componentes do domínio planta de processo, porque a informação de conexão entre os equipamentos pode ser considerada como um indicativo de existência de uma corrente. No EMSO quando uma corrente deriva de outra são utilizados objetos para união e ramificação de correntes, ou seja, as correntes não aceitam conexão entre si. Sendo assim, as conexões entre correntes foram identificadas por este tipo de componente no diagrama de processo.

4 ESTUDO DE CASO

Neste capítulo, está apresentado um estudo de caso para avaliar a capacidade do aplicativo ORION executar a transcrição dos dados de um sistema de processo real. A aplicação do estudo de caso tem o objetivo de testar as principais funcionalidades do aplicativo e identificar possíveis falhas na implementação do código.

Sobre o Processo:

A corrente proveniente de um poço de perfuração *off-shore* geralmente é constituída por água, óleo e gás natural associado. A separação dessa mistura trifásica água/óleo/gás se faz necessária pelo fato da indústria ter grande interesse econômico nas frações óleo e gás. A água deve ser removida devido à sua capacidade de formar emulsões com viscosidades superiores à do petróleo desidratado e hidratos em uma corrente constituída por gás natural, formando depósitos que podem reduzir o diâmetro da tubulação. Sua remoção evita o superdimensionamento do sistema de bombeio e transferência, e danos às operações de processo nas refinarias, pois representa um volume ocioso na transferência e tancagem do petróleo e pode gerar problemas de incrustação e corrosão nos oleodutos de exportação. Todo o processo de separação da corrente trifásica é realizado em plataformas com ajuda de equipamentos como separadores trifásicos, bombas, compressores e colunas absorvedoras. O processamento do gás consiste da compressão, remoção de CO₂ e desidratação (remoção da umidade residual) para ser utilizado principalmente como gás combustível e *gás lift* nos poços de produção, sendo o excedente exportado através de tubulações. O gás excedente, ao chegar em terra, deve ser processado adequadamente em Unidades de Processamento de Gás Natural – UPGN. Nestas, o gás será desidratado e fracionado, gerando o metano e o etano, que formarão o gás natural combustível – GNC propriamente dito, e propano e butano, que formam o gás liquefeito de petróleo – GLP, e um produto denominado “gasolina natural” (SANT’ANNA *et al.*, 2005).

A Simulação do Processo:

O sistema de compressão de gás do processo de separação de óleo/água/gás descrito a cima foi simulado no *software* EMSO e o arquivo com os resultados desta simulação foi transformado pelo aplicativo ORION em uma nova saída padronizada.

O fluxograma da unidade simulada no EMSO está ilustrado na Figura 30. A simulação foi realizada em regime estacionário.

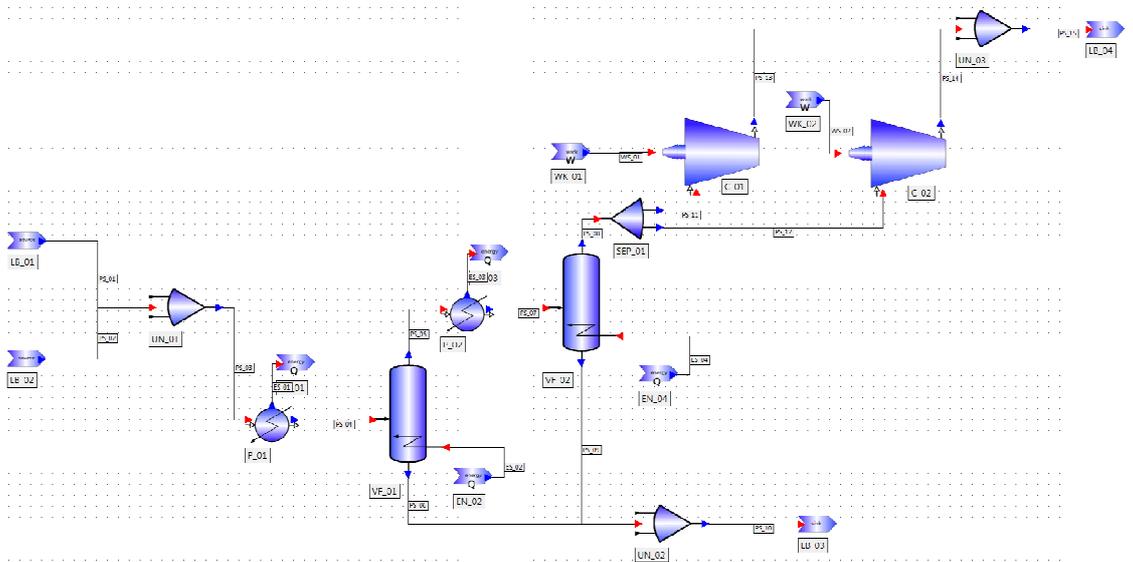


Figura 30 - Processo de Separação de Óleo – Compressão de Gás.

Para a representação do fluxograma da Figura 30 foram definidos os prefixos listados na Tabela 8.

Tabela 8 - Prefixos definidos para representação do fluxograma da unidade de reforço de compressão de gás.

Prefixo	Descrição
PS	Corrente de Processo
ES	Corrente de Energia
WS	Corrente de Trabalho
LB	Limite de Bateria
EN	Fonte de Energia
WK	Fonte de Trabalho
UN	União de Correntes
SEP	Separação de Correntes
P	Permutador de Calor
VF	Vaso de Flash
C	Compressor

A corrente PS_01 (rica em água e metano), proveniente de LB_01 (sistema de separação, coleta e bombeamento de óleo) se mistura (UN_01) com a corrente PS_02 (rica em metano), proveniente de LB_02, para em seguida serem resfriadas em P_01. A corrente resfriada passa por VF_01 para que haja separação da água presente na corrente gasosa. A corrente gasosa proveniente de VF_01 é novamente resfriada em P_02 e é feita uma segunda separação de água em VF_02. A corrente gasosa proveniente de VF_02 é dividida proporcionalmente em duas correntes (SEP_01) para que estas sejam alimentadas nos compressores C_01 e C_02 para aumento da pressão do sistema. As correntes provenientes dos compressores são misturadas (UN_03) e têm como destino LB_04 (sistema de compressão de gás). As correntes líquidas ricas em água provenientes de VF_01 e VF_02 são misturadas (UN_02) e têm como destino LB_03 (sistema de água produzida e de drenagem).

Os componentes das correntes de processo e os modelos para as fases líquida e vapor foram utilizados conforme o VRTherm, software com o qual o EMSO faz

interface através de plugin. O VRTherm é um Banco de Dados com cerca de 2000 componentes puros que faz cálculos de propriedades de mistura.

As correntes do sistema contêm 7 componentes. São eles:

- 1- Água (*water*);
- 2- Metano (*methane*);
- 3- Etano (*ethane*);
- 4- Propano (*propane*);
- 5- Isobutano (*isobutane*);
- 6- N-butano (*n-butane*);
- 7- Isopentano (*isopentane*).

Os componentes foram cadastrados no EMSO na ordem acima, como mostra a Figura 31.

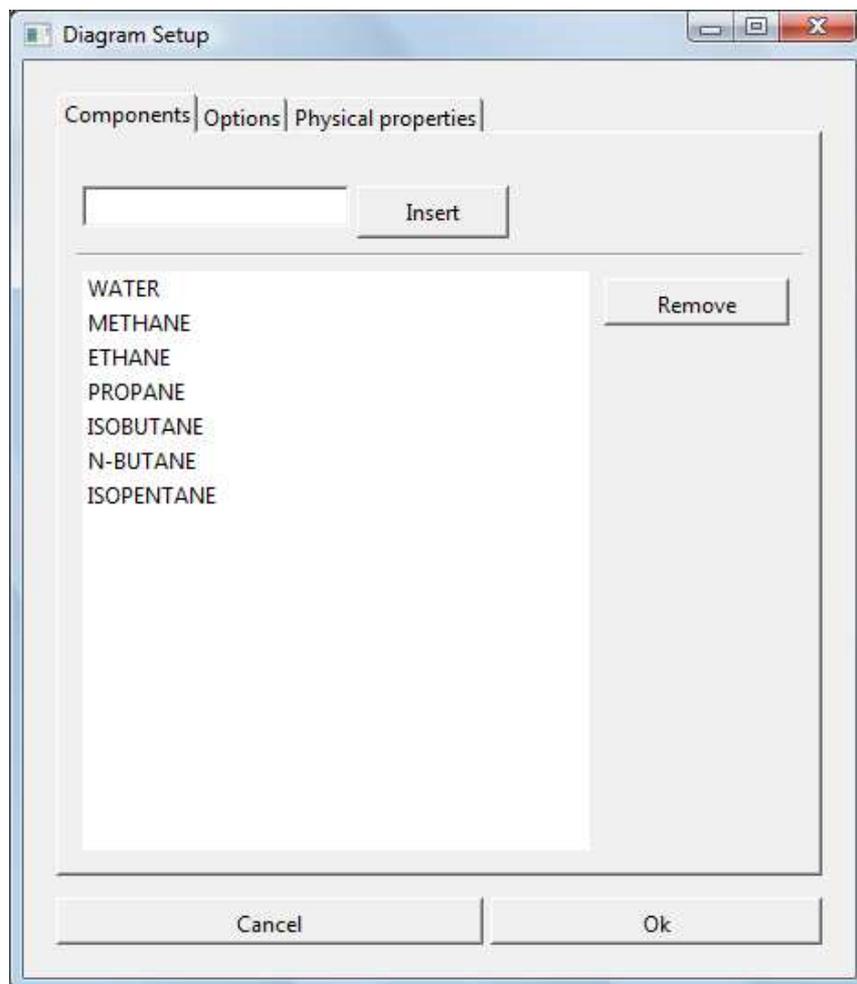


Figura 31 - Definição dos Componentes no EMSO.

A equação de Peng/Robinson (PR), desenvolvida especificamente para cálculos de equilíbrio líquido/vapor (Smith, 2005) e opção default do VRTherm, foi utilizada para as fases líquida e vapor, como ilustrado na Figura 32.

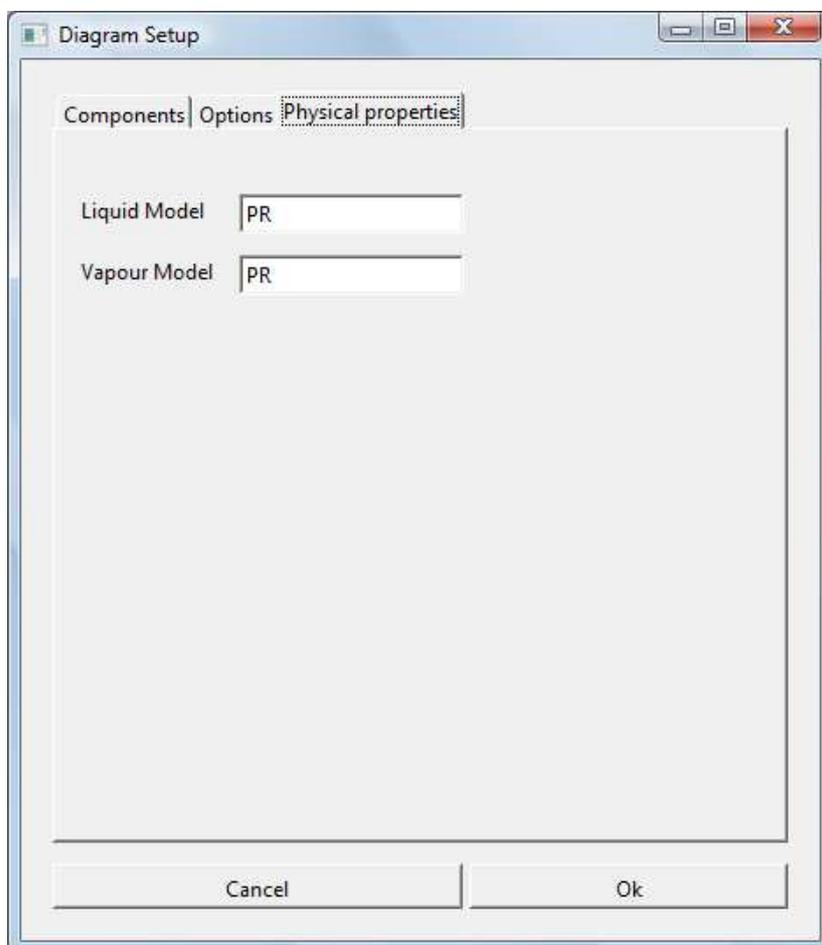


Figura 32 - Definição dos modelos das fases líquida e vapor no EMSO.

Para que o sistema fique sem graus de liberdade, sem referências circulares, e a simulação possa convergir, alguns dados precisam ser definidos nos objetos presentes no fluxograma a ser simulado no EMSO. Quando a simulação convergir e o sistema for solucionado, os demais dados do sistema serão calculados.

Para representar os limites de bateria (LB_01 e LB_02), que geram as correntes iniciais da unidade, foi utilizado o objeto "source". Para este objeto foram definidos os seguintes dados:

- Composição molar global de $n - 1$ (6) componentes da mistura;

- Vazão molar;
- Vazão mássica;
- Temperatura;
- Pressão.

Para representar os trocadores de calor (P_01 e P_02), que são resfriadores, foi utilizado o objeto “cooler”. Para este objeto foram definidos os seguintes dados:

- Temperatura da corrente de saída;
- Pressão da corrente de saída.

Para representar os vasos de flash (VF_01 e VF_02) foi utilizado o objeto “flash_steady”, que opera em estado estacionário. Para este objeto foram definidos os seguintes dados:

- Temperatura da corrente de saída;
- Queda de pressão no vaso.

Para representar as fontes de energia (EN_01, EN_02, EN_03 e EN_04) foi utilizado o objeto “energy_source”. Com as definições feitas para os trocadores de calor e vasos de flash, a quantidade de energia fornecida pelas fontes foi calculada pela simulação.

Para representar os compressores (C_01 e C_02) foi utilizado o objeto “centrifugal_compressor”. Para este objeto foram definidos os seguintes dados:

- Temperatura da corrente de saída;
- Pressão da corrente de saída;
- Trabalho.

Para representar as fontes de trabalho (WK_01 e WK_02) foi utilizado o objeto “work_source”. Com as definições feitas para os compressores, a quantidade de trabalho fornecida pelas fontes foi calculada pela simulação.

Para representar a separação de correntes (SEP_01) foi utilizado o objeto “splitter”. Para este objeto foi definida a fração de separação entre as correntes.

Para representar a mistura de correntes (UN_01, UN_02 e UN_03) foi utilizado o objeto “mixer”. Para este objeto não foi necessário definir nenhum dado.

Para representar os limites de bateria (LB_03 e LB_04) que recebem correntes finais da unidade foi utilizado o objeto “sink”. Para este objeto não foi necessário definir nenhum dado.

Como resultado final da simulação, é gerado um arquivo na extensão XML contendo todos os dados referentes à unidade.

Transferência para Fase de Projeto:

O aplicativo **ORION** executou a correta interpretação do arquivo de saída do simulador de processos EMSO e gerou com sucesso o arquivo XML de saída no padrão ISO-15926 para a simulação descrita acima. A Figura 33 ilustra um trecho da saída padronizada.

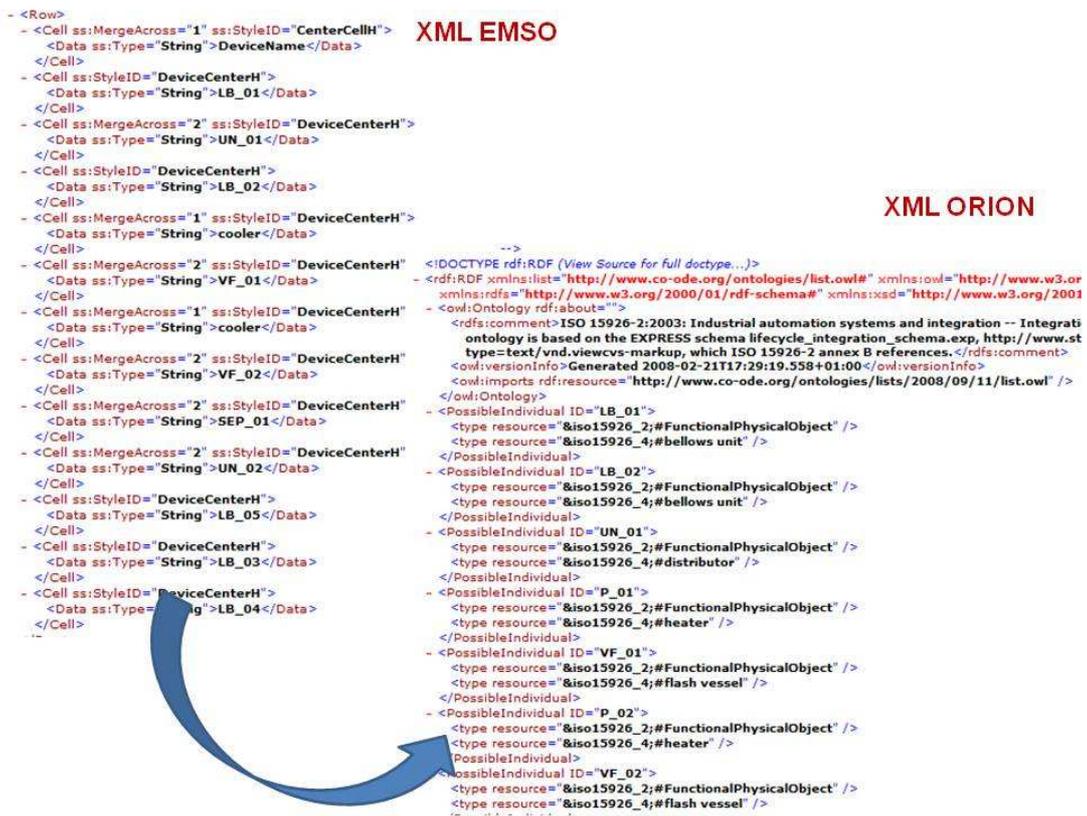


Figura 33 – Conversão XML do EMSO para XML Padrão do ORION.

A importação do arquivo de saída do **ORION** para o COMOS® executou a criação automática dos equipamentos do diagrama de processo exposto na Figura 30 dentro do diretório *AEQ – Equipment* do projeto em questão.

Para cada informação de conexão entre dois equipamentos uma corrente de processo foi criada no diretório APS – *Process Streams* para representação gráfica deste fluxo. Os conectores de entrada e saída deste objeto corrente foram definidos via código como conectados aos equipamentos em questão.

A Figura 34 e Figura 35 mostram os equipamentos e correntes criados, com as respectivas referências de conexão definidas.

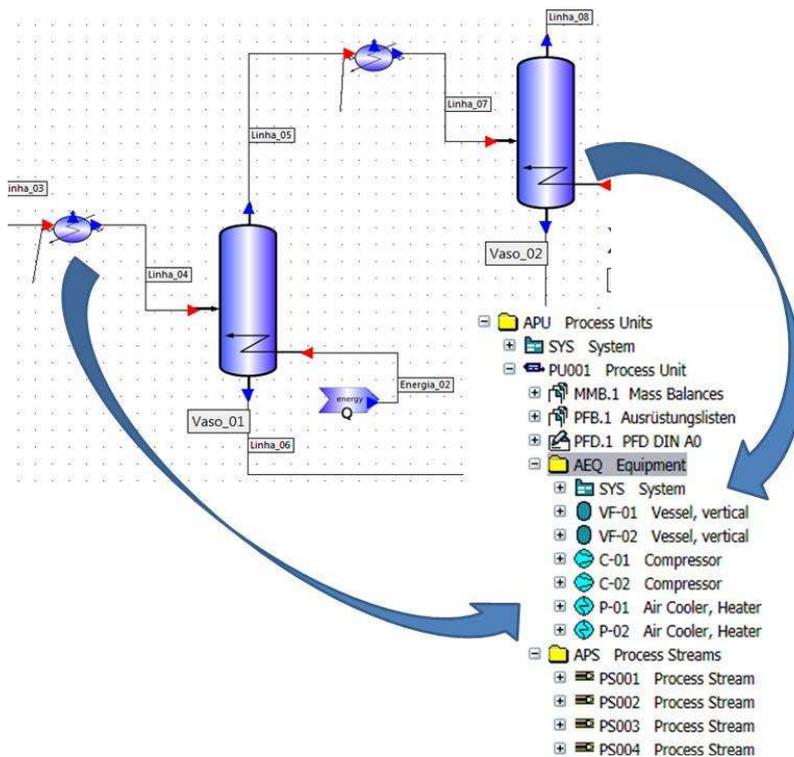


Figura 34 – Criação dos Equipamentos do Processo.

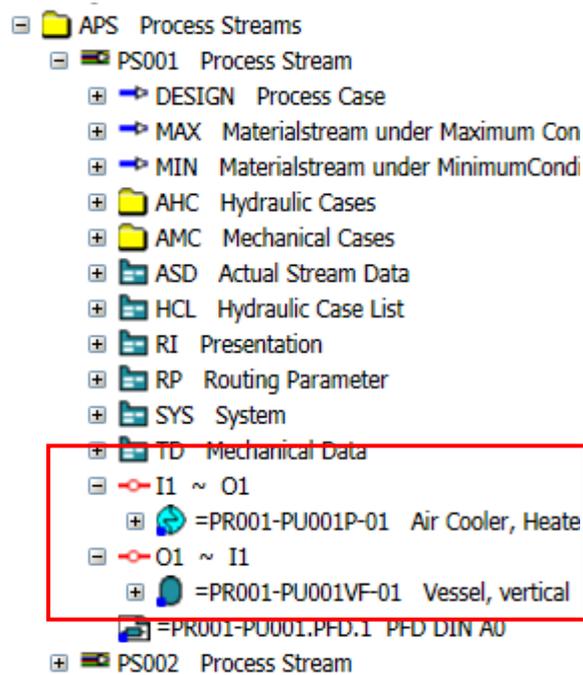


Figura 35 – Conectores das Correntes de Processo.

As conexões são indicadas na Figura 35 como I1 e O1 e podem ser consideradas propriedades de cada componente da planta que armazenam as informações de ligação com outros componentes. Neste exemplo, a corrente de processo PS001 está conectada com o Air Cooler P-01 na entrada e com o Vaso de Flash VF-01 na saída. Uma vez que as referências de conexão entre os objetos (equipamentos e correntes) existem na estrutura de cada objeto, é possível utilizar a opção nativa do *software* “*Connect Automatically*” para construção do PFD. Este comando insere automaticamente a corrente de processo no diagrama e efetua as conexões gráficas.

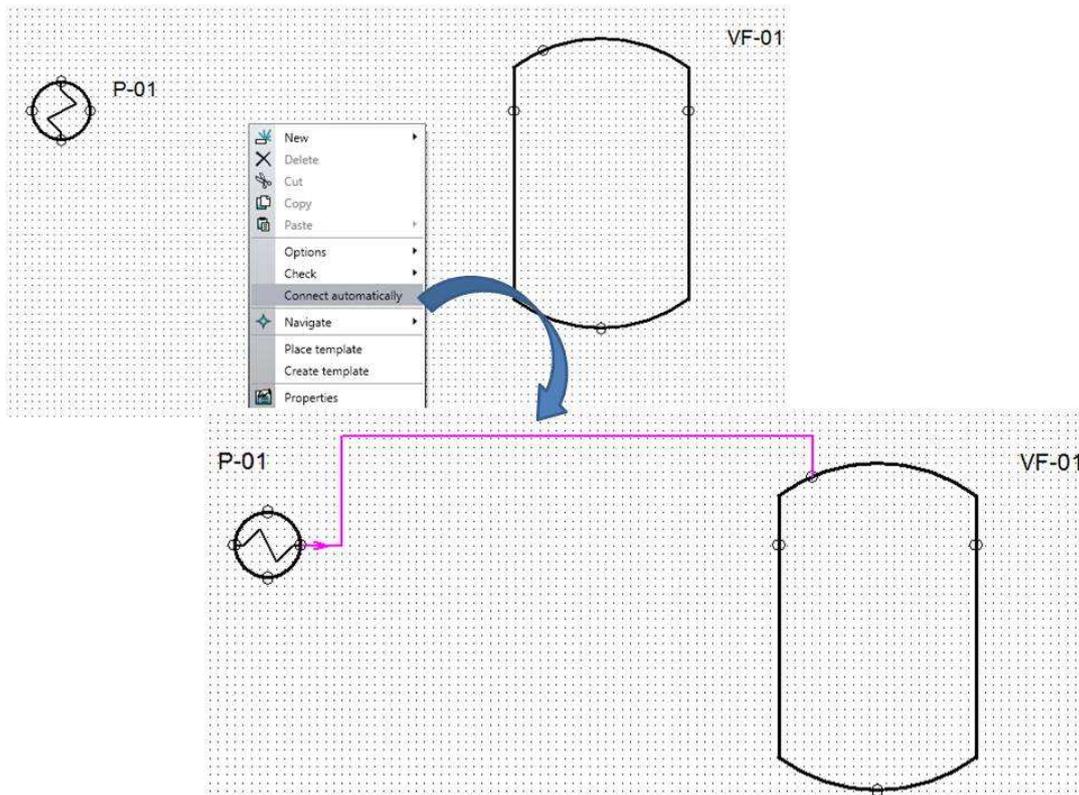


Figura 36 – Conexão Automática no PFD.

Tendo toda a topologia do processo construída na estrutura de árvore do COMOS® via os comandos executados pelo aplicativo **ORION**, o PFD é obtido arrastando todos os equipamentos do diretório AEQ para área branca do desenho e ao executar o comando todas as correntes são desenhadas e conectadas aos equipamentos automaticamente.

Completadas as conexões o resultado final do PFD no software de projeto encontra-se na Figura 37.

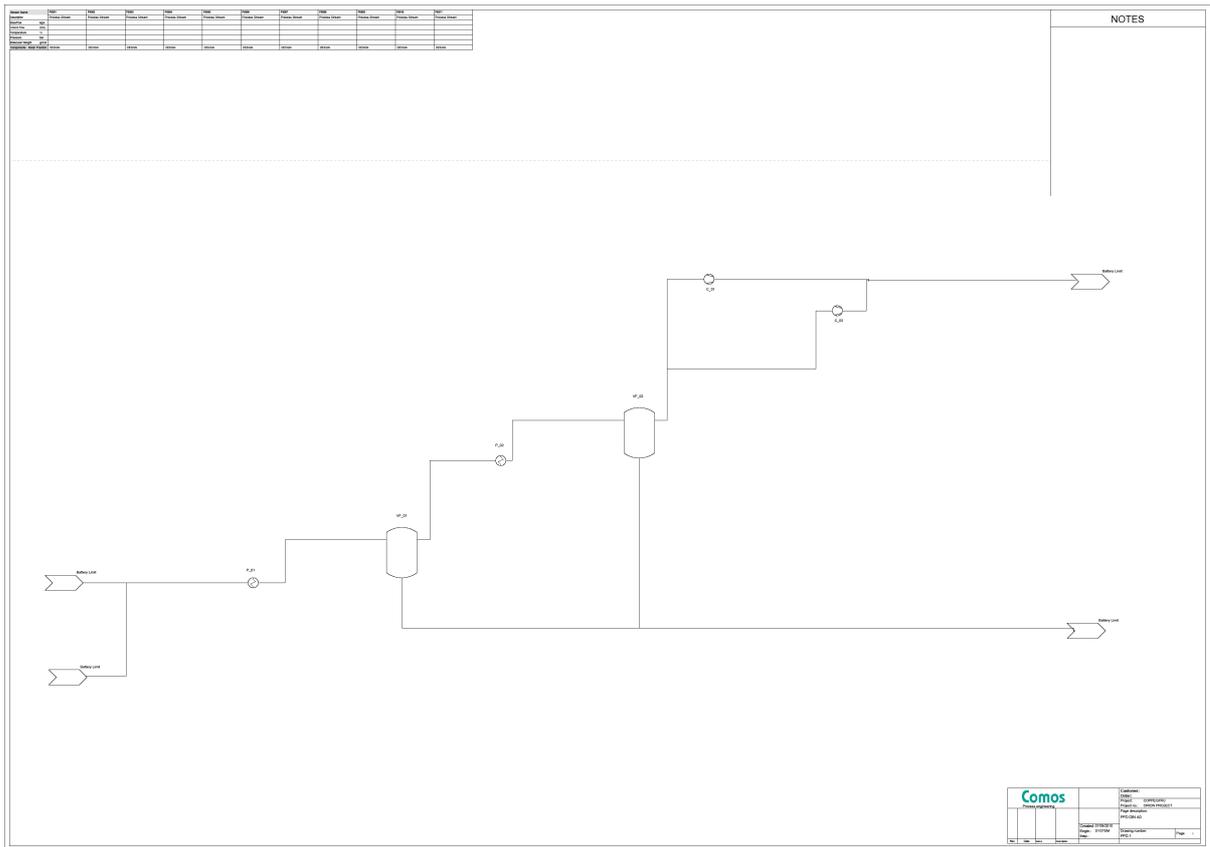


Figura 37 - Fluxograma de Processo no COMOS

As propriedades que descrevem a condição de operação dos equipamentos foram atualizadas, em termos de quantidades físicas e escala de medida, automaticamente no banco de dados do COMOS® através do aplicativo, conforme pode ser visualizado na tabela de balanço de massa do PFD gerado em comparação com a saída da simulação.

A Figura 38 mostra a atualização dos parâmetros Pressão, Temperatura e Vazão Molar das correntes de entrada e saída do vaso VF_01, de acordo com os valores resultantes da simulação ilustrados na Tabela 9.

Tabela 9 – Resultado da Simulação para o Vaso VF_01.

DeviceName		VF_01		
StreamName		Inlet	OutletL	OutletV
F	kmol/h	675,7000	346,7500	328,9500
T	K	329,8870	329,8870	329,8870
P	atm	1,3551	1,3551	1,3551
h	kJ/kmol	-21435,0000	-43190,2000	1497,2600

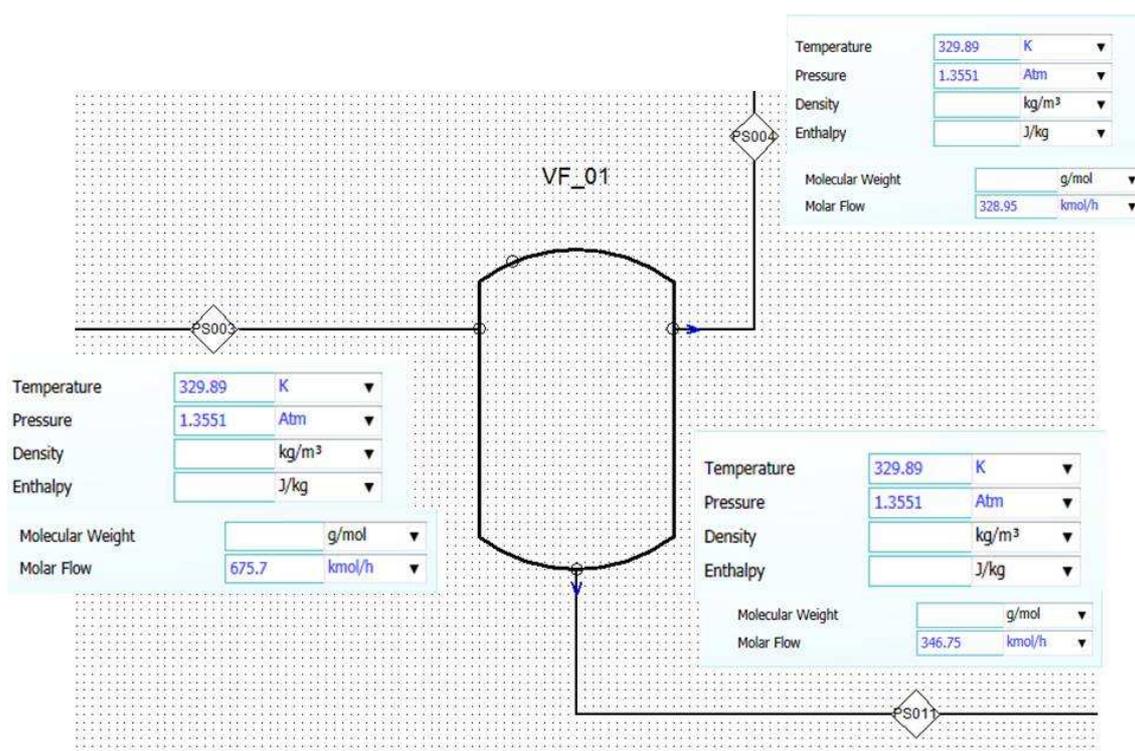


Figura 38 – Importação Dados de Simulação do VF_01 para o software CAE.

Os componentes das correntes também foram cadastrados com sucesso no software CAE, conforme a leitura do arquivo XML padrão. A atualização das frações molares de acordo com o resultado da simulação complementa as informações da tabela de balanço de massa do PFD. A Figura 39 e 39 ilustram estes resultados exemplificando um trecho da tabela de balanço de massa do PDF no COMOS®.

Plugins	
Name	Value
Brief	Physical Properties
Type	PP
Components	WATER
	METHANE
	ETHANE
	PROPANE
	ISOBUTANE
	N-BUTANE
	ISOPENTANE

- 002 - Importacao PFD » ORION PROJECT C
 - PR001 Process
 - PFB.1 Block Diagram DIN A4 x A1
 - PFB.2 Process Description (Word-Doc)
 - PFB.3 Equipment List, total
 - PFB.4 Equipment Lists
 - PFB.5 Mass Balances
 - ABS Boundary Streams
 - AMA Components
 - MAL Component List
 - MA0017 Methane
 - MA0022 n-Butane
 - MA0023 Propane
 - MA0033 Water
 - MA0034 IsoButane
 - MA0035 IsoPentane

Figura 39 – Importação dos Componentes das Correntes no Software CAE.

DeviceName	LB_01	LB_02	cooler		VF_01		
StreamName	Outlet	Outlet	Inlet	Outlet	Inlet	OutletL	OutletV
F	618,9000	56,8000	675,7000	675,7000	675,7000	346,7500	328,9500
T	402,0000	311,0000	395,0470	329,8870	329,8870	329,8870	329,8870
P	1,4518	1,4518	1,4518	1,3551	1,3551	1,3551	1,3551
h	4481,0700	405,0230	4138,4300	-21435,0000	-21435,0000	-43190,2000	1497,2600
v	1,0000	1,0000	1,0000	0,4868	0,4868	0,0000	1,0000
z(1)	0,6207	0,0000	0,5686	0,5686	0,5686	1,0000	0,1138
z(2)	0,2305	0,9112	0,2877	0,2877	0,2877	0,0000	0,5910
z(3)	0,0479	0,0504	0,0481	0,0481	0,0481	0,0000	0,0989
z(4)	0,0248	0,0144	0,0239	0,0239	0,0239	0,0000	0,0491
z(5)	0,0117	0,0052	0,0111	0,0111	0,0111	0,0000	0,0229
z(6)	0,0138	0,0057	0,0131	0,0131	0,0131	0,0000	0,0269
z(7)	0,0506	0,0130	0,0474	0,0474	0,0474	0,0000	0,0975

Stream Name	PS001	PS002	PS003	PS004	F
Description	Process Stream	Process Stream	Process Stream	Process Stream	F
MassFlow	kg/s				
Volume Flow	m³/s				
Temperature	°C	128.85	37.85	56.737	56.737
Pressure	bar	1.4710	1.4710	1.3731	1.3731
Molecular Weight	g/mol				
Components Molar Fraction	DESIGN	DESIGN	DESIGN	DESIGN	I
Methane		0.2305	0.9112	0.2877	0.5910
n-Butane		0.0138	0.0057	0.0131	0.0269
Propane		0.0248	0.0144	0.0239	0.0491
Water		0.6207	0	0.5686	0.1138
IsoButane		0.0117	0.0052	0.0111	0.0229
IsoPentane		0.0506	0.0130	0.0474	0.0975
Ethane		0.0479	0.0504	0.0481	0.0989

Figura 40 – Importação das Frações Molares dos Componentes.

A estratégia de modelagem orientada a objeto do *software* COMOS® permite transformar diagramas PFD em representação de P&ID automaticamente, transferindo com segurança as informações de engenharia para a fase seguinte do projeto através das referências entre os objetos.

5 CONCLUSÕES E PERSPECTIVAS

5.1 CONCLUSÕES

Este trabalho apresentou o desenvolvimento do aplicativo **ORION** para promover integração entre ambientes de simulação e projeto, baseado na norma ISO-15926 para padronização do intercâmbio de dados digitais de engenharia.

O estudo de caso evidencia que as ferramentas de simulação de processos e construção dos documentos de projeto que estão disponíveis no mercado são altamente sofisticadas e atendem às necessidades específicas da indústria de processos. Entretanto, as terminologias e sintaxe utilizadas em cada modelo não permitem a transferência da informação coerente e consistente, além de demandar um elevado grau de esforço na construção da comunicação entre os modelos distintos.

A ausência da informação do tipo de equipamento no arquivo de saída do simulador EMSO impossibilita a automatização da transferência dos dados para fase de projeto e a inclusão ou alterações nos modelos construídos via código podem implicar em falhas no mapeamento do EMSO devido a mudanças nos nomes das propriedades físicas. Esta constatação mostra que para garantir a completa interoperabilidade entre as ferramentas segundo a norma ISO-15926, algum desenvolvimento padronizado nos softwares de simulação é necessário.

A norma ISO-15926 foi capaz de transcrever as informações da simulação de processo estática com sucesso, salvo algumas propriedades físicas pendentes citadas ao longo do capítulo Metodologia. Entretanto, a norma apresenta um grau elevado de complexidade que pode inviabilizar seu uso em larga escala.

Os mapeamentos construídos entre os termos semânticos e classes da ISO-15926 e os *softwares* EMSO e COMOS foram eficientes para transcrição das informações entre as duas ferramentas, garantindo o intercâmbio de dados com segurança.

O trabalho comprova a necessidade de um padrão específico para o gerenciamento e intercâmbio de informações de plantas de processos industriais, já que sem esta tradução desenvolvida não seria possível a importação dos dados entre as duas ferramentas de forma reutilizável.

5.2 PERSPECTIVAS

O mercado de *software* para engenharia está bastante aquecido e existe atualmente um interesse mundial pela interoperabilidade entre sistemas. Espera-se que este trabalho seja apenas o início da aproximação entre as universidades, fornecedores de *software* e operadores de planta para o desenvolvimento de ambientes padronizados e que suportem a perfeita comunicação entre si.

O desenvolvimento da exportação do diagrama de processo do simulador de processos EMSO para um arquivo RDF/XML ISO-15926 é uma opção bastante atrativa para competitividade do *software* e alinhada com sua proposta acadêmica e de arquitetura aberta.

A transferência dos dados de simulação dinâmica complementaria este trabalho com informações relevantes para especificação dos instrumentos.

O **ORION** como qualquer software está sujeito a falhas quando testado em outros cenários de simulação. Sendo assim, o aplicativo pode ser aprimorado via testes de conceito com outros tipos de simulação para atender a conversão de qualquer saída do EMSO sem perdas. Também seria interessante, mapear a biblioteca completa de objetos de engenharia do software COMOS® para atender conversões de outras áreas técnicas além do módulo *PR – Process* e com outros simuladores de processo.

6 REFERÊNCIAS BIBLIOGRÁFICAS

- AMERICAN CHEMICAL SOCIETY. **Technology vision 2020: The US Chemical Industry**. Washington, 1996.
- BATRES R.; AOYAMA A.; NAKA Y. A life-cycle approach for model reuse and Exchange, **Computers and Chemical Engineering**, v. 26, n. 4-5, p. 487 - 498, 2002.
- BATRES, R. *et al.* An upper ontology based on ISO 15926, **Computers and Chemical Engineering**, v. 31, n. 5-6, p. 519 - 534, 2007.
- BAYER, B.; MARQUARDT, W. Towards integrated information models for data and documents, **Computers and Chemical Engineering**, v. 28, n. 8, p. 1249 - 1266, 2004.
- BECKER, A. **Web Services e XML - um novo paradigma da computação distribuída**. 2001. Trabalho de Conclusão de Curso (Ciência da Computação) - Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis.
- BEETZ, J.; VAN LEEUWEN J. P.; VRIES, B. An ontology web language notation of the industry foundation classes. In: Proceedings of the **22nd CIB W78 Conference on Information Technology in Construction**. Dresden. 2005.
- BERNAUER M. *et al.* Specification of interorganizational workflows - a comparison of approaches, In: Proceedings of the **7th World Multiconference on Systemics, Cybernects and Informatics**. Orlando, 2003.
- COMOS Industry Solutions. **The Comos object-oriented way of thinking**. Disponível em <www.innotec.de/objektgedanke.html?&L=1>. Consultado em 07/09/2010.
- CORCHO, O.; FERNÁNDEZ-LÓPEZ, M.; GÓMEZ-PÉREZ, A. Methodologies, Tools and languages for building ontologies. Where is their meeting point?, **Data & Knowledge Engineering**, v. 46, n.1, p. 41 - 64, 2003.
- DOUGLAS, J. **Conceptual Design of Chemical Processes**. McGraw-Hill, 1988.
- FERNANDES, P. **Modelagem e simulação dinâmica do ciclo de refrigeração a propano de uma unidade de processamento de gás natural**. 2009. Rio de Janeiro: Escola de Química, Universidade Federal do Rio de Janeiro (Monografia de Conclusão de Curso).
- FERNÁNDEZ-LÓPEZ, M.; GÓMEZ-PÉREZ, A. Overview and analysis of methodologies for building ontologies, **The Knowledge Engineering Review**, v. 17, n. 2, p. 129 - 156, 2002.
- GALLAHER, M. *et al.* **Cost analysis of inadequate interoperability in the U.S. capital facilities industry**. 2004. Gaithersburg: National Institute of Standards and Technology (Report NIST GCR 04-867).
- GIELINGH, W. An assessment of the current state of product data technologies, **Computer-Aided Design**, v. 40, n. 7, p. 750 - 759, 2008.

- GRUBER, T. **Toward principles for the design of ontologies. Knowledge Systems Laboratory.** 1993. Stanford: Stanford University (Technical Report KSL 93-04).
- GUARINO, N. *Formal Ontology and Information Systems*. In: Formal Ontology in Information Systems. Proceedings of **FOIS'98**. Trento, 1998.
- GULLA, J.; TOMASSEN, S.; STRASUNSKAS, D. (2006) Semantic interoperability in the norwegian petroleum industry. In: Karagiannis, D., Mayer, H.C. (eds.): **5th International Conference on Information Systems Technology and its Applications**. ISTA, 2006.
- KELLNER, M.; MADACHY, R.; RAFFO, D. Software process simulation modeling: Why? What? How?, **The Journal of Systems and Software**, v. 46, n. 2-3, p. 91 - 105, 1999.
- LEAL, D. ISO 15926 "Life Cycle Data for Process Plant": An Overview. Oil & Gas Science and Technology, **Rev. IFP**, v. 60, N. 4, p. 629 - 637, 2005.
- MARQUARDT, W.; NAGL, M. Workflow and information centered support of design processes - the IMPROVE perspective. **Computers and Chemical Engineering**, v. 29, n.1, p. 65 - 82, 2004.
- MORBACH, J.; YANG, A.; MARQUARDT, W. Ontocape - A large-scale ontology for chemical process engineering. **Engineering Applications of Artificial Intelligence**, v. 20, n. 2, p. 147 - 161, 2007.
- NAGL, M.; WESTFECHTEL, B.; SCHNEIDER, R. Tool support for the management of design processes in chemical engineering, **Computers and Chemical Engineering**, v. 27, n. 2, p. 175 - 197, 2003.
- PCA - POSC Caesar Association. **A mapping of ISO 15926-2 in EXPRESS to OWL**. Disponível em <<https://www.posccaesar.org/wiki/ISO15926inOWLtranslateEXPRESStoOWL#EXPRESSattributes>>. Consultado em 10/08/10.
- PCA - POSC Caesar Association. **ISO 15926 Integration of lifecycle data for process plant including oil and gas production facilities**. Disponível em <<https://www.posccaesar.org/wiki/ISO15926>>. Consultado em 13/06/2010.
- PCA - POSC Caesar Association. **Metaphor – ISO 15926 is like Esperanto on your cell phone**. Disponível em <https://www.posccaesar.org/wiki/ISO15926Primer_Metaphor_Cell_Phone> Consultado em 10/08/2010.
- QIU, H. *et al.* Research on workflow modeling methods for collaborative product development. **Advanced Materials Research**, v. 44-46, p. 247 - 252, 2008.
-
- SANT'ANNA, A.; MEDEIROS, J.; ARAÚJO, O. Simulação de processamento de gás natural em plataforma off-shore, In: **3º Congresso Brasileiro de P&D em Petróleo e Gás** . Instituto Brasileiro de Petróleo e Gás – IBP, 2005.
- SCHNEIDER, R.; MARQUARDT, W. Information technology support in the chemical process design life cycle, **Chemical Engineering Science**, v. 57, n. 10, p. 1763 - 1792, 2002.
- SMITH, J.; VAN NESS, H.; ABBOTT, M. **Introdução à Termodinâmica da Engenharia Química**,

7ª edição, LTC, 2005.

SOARES, R. **EMSO Manual**. Agosto de 2007.

SOARES, R. **Desenvolvimento de um simulador genérico de processos dinâmicos**. 2003.

Dissertação (Mestrado em Engenharia Química) – Programa de Pós-Graduação em Engenharia Química, Universidade Federal do Rio Grande do Sul, Porto Alegre.

SOARES, R.; SECCHI, A. EMSO: A new environment for modeling, simulation and optimization, In: **ESCAPE 13th**. [S.I.]: Elsevier Science Publishes, v. 1, p. 947 – 952, 2003.

STUDER, R.; BENJAMINS V.; FENSEL, D. Knowledge engineering: principles and methods. **IEEE Transactions on Data and Knowledge Engineering**, v. 25, n. 1-2; p. 161 - 197, 1998.

STEGWEE, R.; RUKANOVA, B. Identification of different types of standards for domain-specific interoperability, In: Proceedings of the **Workshop on Standard Making: A Critical Research Frontier for Information Systems**. Seattle, 2003.

TAYLOR, J. Understanding and combating design error in process plant design. **Safety Science**, v. 45, n. 1-2, p. 75 - 105, 2007.

USCHOLD, M.; GRUNINGER, M. Ontologies: principles, methods and applications, **Knowledge Engineering Review**, v. 11, n. 2, p. 93 - 155, 1996.

W3C. **Extensible Markup Language (XML) 1.0 (Fifth Edition)**. Disponível em <www.w3c.org/TR/REC-xml>. Consultado em 18/07/2010.

ISO 15926-1. **ISO 15926 Integration of lifecycle data for process plant including oil and gas production facilities: Part 1. Overview and fundamental principles**. 2004. ISO 15926-1.

ISO 15926-2. **ISO 15926 Integration of lifecycle data for process plant including oil and gas production facilities: Part 2. Data model**. 2003. ISO 15926-2.

ISO 15926-4. **ISO 15926 Integration of lifecycle data for process plant including oil and gas production facilities: Part 4. Initial Reference Data**. 2007. ISO 15926-4.

W3C. **OWL Web Ontology Language Guide**. Disponível em <<http://www.w3.org/TR/2004/REC-owl-guide-20040210>>. Consultado em Junho de 2010.

WASSERMAN, A. Tool integration in software engineering environments, In: Proceedings of the **International workshop on environments on Software engineering environments**. Chinon, 1990.

YANG, X., MCGREAVY, C. Requirements for sharing process data in the life cycle of process plants. **Computers and Chemical Engineering**, v. 20, n. Supplement 1, p. 363 - 368, 1996.

YOGUI, R. ISO 15926 - Padrão internacional para integração e automação no PLM (Plant Lifecycle Management), In: **V Congresso Rio Automação**, Instituto Brasileiro de Petróleo, Gás e Biocombustíveis – IBP. Rio de Janeiro, 2009.