



Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica

Trabalho de Conclusão de Curso

Representação de Interfaces de Sistemas de Controle
utilizando notação XML e Web Semântica

Aluno:

Philipe Freire de Araújo Patrício
philipepatricio@gmail.com

Orientadora:

Prof^a Fátima Vieira
fatima@dee.ufcg.edu.br

Campina Grande, 31 de outubro de 2008.



Biblioteca Setorial do CDSA. Fevereiro de 2021.

Sumé - PB

ÍNDICE

1.Introdução.....	3
2.Representação de Ontologias.....	6
3.Especificação de Interfaces utilizando XML	11
4.Aplicação de Web Semântica a representação de Sistemas de Supervisão.....	19
5.Ferramentas Utilizadas.....	29
6.Considerações Finais.....	30
7.Referências.....	31

1. Introdução

Automação industrial consiste no uso de qualquer dispositivo mecânico ou eletro-eletrônico para controlar máquinas e processos. Uma contribuição importante dos sistemas de automação industrial é a conexão de sistemas de controle e supervisão, utilizados para a monitoração e controle das variáveis e dos dispositivos de sistemas de produção, permitindo o compartilhamento de dados importantes da operação dos processos, contribuindo para uma maior agilidade do processo decisório e maior confiabilidade dos dados que suportam as decisões dentro da indústria.

Uma tendência dos sistemas de automação é a supervisão remota, usando sistemas de telemetria através de tecnologias sem fios, como rádio ou satélite. Através dessas tecnologias, os sistemas de supervisão têm a capacidade de controlar processos industriais numa planta de trabalho local, ou descentralizada.

Uma questão importante advinda da supervisão remota é representação das interfaces de supervisão em dispositivos distintos, como Desktops, PDAs, Tablets ou celulares. Para um melhor aproveitamento dos recursos de cada equipamento é necessário adequar estas interfaces aos recursos gráficos dos dispositivos, como tamanho, resolução e *layout* da tela.

Uma forma de contornar o problema é a representar as informações presentes nos sistemas de supervisão de forma indiferente a sua representação gráfica. A geração das interfaces ficaria a cargo do próprio dispositivo que a apresenta ao usuário.

1.1. Representação de Sistemas de Supervisão usando XML

Visando estender a monitoração de processos industriais para ambientes fora das salas de controle através do uso de dispositivos móveis, um grupo de trabalho do Laboratório de Interface Homem-Máquina (LIHM), da Universidade Federal de Campina Grande (UFCG), propôs uma notação baseada no padrão **eXtensible Markup Language (XML)** [1] para representar informações de sistemas de produção industriais.

O padrão XML possibilita que dados sejam armazenados em um documento simples, um arquivo de texto, de forma estruturada e independente do modo como serão visualizados ou processados.

Para a integração entre os dispositivos, foi utilizada uma estrutura cliente/servidor, onde os dados de natureza dinâmica presentes nas telas gráficas, são armazenados em um Banco de Dados. O servidor se comunica com os clientes enviando ou recebendo documentos XML, através da Internet.

A especificação foi concebida inicialmente para representar um sistema de supervisão de uma indústria petroquímica e modificada de modo a estender sua aplicação a outros contextos de automação caracterizados pela transmissão, armazenamento e processamento de substâncias, objetos ou dados.

1.2. Trabalho Atual

O presente Trabalho de Conclusão de Curso toma como ponto de partida a especificação desenvolvida pelo LIHM para representar informações de telas de supervisão de um indústria petroquímica, apresentada no artigo **A Pervasividade em Ambientes de Automação Industrial** [14], que defende o uso de uma única base de dados para descrição de sistemas de supervisão, simplificando a integração dos dispositivos; oferece a possibilidade de otimização do uso dos recursos gráficos de cada dispositivo, já que cada dispositivo distinto dispõe de uma interface própria.

Com os dados referentes aos equipamentos e suas variáveis contidos em uma estrutura bem definida, constatou-se que a forma como os elementos de relacionam, seus significados, e o contexto de aplicação, embora possam ser inferidos, não estão explícitos na especificação.

O foco deste trabalho é organizar as informações presentes em sistemas industriais numa estrutura que contenha além das variáveis e elementos associados ao sistema, uma descrição dos recursos existentes. Para tanto propõe-se uma representação das informações contidas em telas de sistemas de supervisão fazendo uso de um padrão utilizado amplamente por desenvolvedores capaz de armazenar dados de maneira simples e estruturada, descrever suas relações e definir seus significados.

Foram adotadas as camadas de descrição de recursos e vocabulário ontológico, discutidas no capítulo 4, do padrão de Web Semântica proposto pelo *World Wide Web Consortium (W3C)* [2] em 2001, estando ainda em desenvolvimento.

1.3. Estrutura do Trabalho

O relatório está assim organizado:

- No capítulo dois é apresentado o conceito de ontologias e como representá-las. Apresenta o padrão XML e o conceito de Web Semântica proposto pelo W3C e a forma adota para aplicação de ontologias para representação de interfaces.
- O terceiro capítulo descreve o processo de normalização e generalização de uma notação XML utilizada para representar interfaces de sistemas de supervisão industrial.
- O capítulo quatro trata da aplicação de Web Semântica na representação de interfaces em automação industrial, relatando as alterações na especificação XML para o caso de uma indústria petroquímica; elaboração da camada de descrição de recursos; e definição do vocabulário de ontologia.
- No capítulo cinco há um resumo das ferramentas utilizadas neste trabalho.
- No sexto capítulo são discutidos os resultados obtidos e apresentadas propostas de continuidade.
- O sétimo capítulo aponta as referências utilizadas pelo autor.

2. Representação de Ontologias

A representação do conhecimento com o intuito de definir de forma declarativa os elementos integrantes de um dado ambiente, ganhou força nos últimos anos decorrente da necessidade de melhores ferramentas de busca e tratamento de informações para a Web. A idéia é usar documentos que contenham os dados necessários para a caracterização de um domínio, não definindo a forma como serão processados ou apresentados. Os principais conceitos empregados neste trabalho foram: ontologias e linguagens de marcação.

2.1.Ontologias

O termo ontologia foi definido por Aristóteles como sendo a parte da Filosofia “que trata do ser enquanto ser”, ou seja, o estudo do ser e da existência. Para a representação do conhecimento tem um sentido semelhante, trata-se da representação de uma realidade. A elaboração e utilização de ontologias é uma forma de atribuir sentido e significado a determinados termos, em dados contextos, em busca de atribuir semântica ao conteúdo dos documentos, tem sido alvo de pesquisas recentes e se mostrado uma ferramenta eficaz na representação do conhecimento.

A aplicação de ontologias deve guardar uma relação direta com o domínio modelado e permitir o reuso de um grande volume de informação desempenhando o papel de conhecimento estruturado, disponível para reuso em larga escala por sistemas e programas.

A definição de ontologia mais popular dentro da comunidade de representação de conhecimento, proposta por **Studer et al** [3], diz que “uma ontologia é uma especificação explícita e formal de uma conceitualização compartilhada”. Onde:

- Especificação explícita e formal significa que há definições declarativas, compreensível para agentes de software e sistemas de informação, de conceitos, instâncias, relações, restrições e axiomas;
- Conceitualização compartilhada se refere a um modelo abstrato de uma área de conhecimento ou de um universo limitado de discurso que utiliza uma terminologia comum da área modelada, ou acordada entre os desenvolvedores dos agentes que se comunicam.

Em outras palavras, uma ontologia é formada por um vocabulário usado para descrever certo contexto e, um conjunto de fatos explícitos e aceitos que dizem respeito ao sentido pretendido para as palavras do vocabulário. Podendo-se considerar as ontologias como a materialização do nível de conhecimento.

A definição de um vocabulário explícito e um padrão para o compartilhamento da informação oferece um modelo que permite aos agentes de *software* trocar informações, possibilitando que máquinas possam fazer coisas que humanos fazem no dia-a-dia, como cruzar informações de sistemas diferentes.

Segundo **Moreira et al** [4], um dos principais objetivos do uso de ontologias na Ciência da Computação é a construção de bases de conhecimento interoperáveis e melhor estruturadas.

Nesse contexto, **Souza e Alvarenga (2004)** [5] afirmam que as ontologias se apresentam como um modelo de relacionamento de entidades e suas interações, em algum domínio particular do conhecimento ou específico a alguma atividade. O objetivo de sua construção é a necessidade de um vocabulário compartilhado para se trocar informações entre os membros de uma comunidade, sejam eles seres humanos ou agentes inteligentes.

Para **Freitas** [6], as ontologias servem não só como vocabulário de comunicação entre agentes, mas também na definição e organização apropriadas de conceitos, relações e restrições. Além disso, a utilização de ontologias permite um ganho de expressividade e flexibilidade, uma vez que o conhecimento sobre uma classe não se circunscreve a termos e palavras-chave como nos mecanismos de busca, mas a qualquer fato que diga respeito às páginas *Web*, tais como estrutura, regiões e conceitos nelas contidos.

Para a construção de ontologias, **Gruber** [7] observa que alguns princípios, se usados com precisão, garantem sua qualidade:

- **Clareza:** os programas usam diferentes modelos e abstrações na resolução de seus problemas. Na definição do conhecimento, deve-se ter a objetividade de definir apenas o que se presume ser útil na resolução da classe de problemas a ser atingida. As definições completas, com condições necessárias e suficientes devem ter precedência sobre definições parciais;
- **Legibilidade:** as definições devem corresponder com as definições correntes e informais. A ontologia deve usar um vocabulário compartilhável (geralmente o jargão e a terminologia usados por especialistas do domínio);

- **Coerência:** as inferências derivadas da ontologia definida devem ser corretas e consistentes do ponto de vista formal e informal com as definições;
- **Extensibilidade:** a ontologia deve permitir extensões e especializações monotonicamente e com coerência, sem a necessidade de uma revisão de teoria, que consiste na revisão lógica automática de uma base de conhecimento em busca de contradições;
- **Mínima codificação:** devem se especificados conceitos genéricos independente de padrões estabelecidos para mensuração, notação e codificação, garantindo a extensibilidade. Essa genericidade é limitada pela clareza;
- **Mínimo compromisso ontológico:** com a finalidade de maximizar o reuso, apenas o conhecimento essencial deve ser incluído, gerando a menor teoria possível acerca de cada conceito, e permitindo a criação de novos conceitos, mais especializados ou estendidos.

Aplicadas à representação do conhecimento, as ontologias oferecem:

- Um vocabulário para representação do conhecimento. Esse vocabulário é sustentado por uma conceituação, evitando assim interpretações ambíguas.
- Uma descrição exata do conhecimento. As ontologias, por serem escritas em linguagem formal, formam uma base para a interpretação da informação.
- Possibilidade de compartilhamento do conhecimento. Uma ontologia que modele adequadamente certo domínio de conhecimento pode ser compartilhada e usada por pessoas que desenvolvam aplicações dentro desse domínio.
- Flexibilidade. Pode ser possível estender o uso de uma ontologia genérica de forma a se adequar a um domínio específico.

2.2. Linguagens de Marcação

Uma solução para a representação das informações que caracterizam um domínio de forma estruturada e passível de serem manipuladas por softwares é o uso de linguagens de marcação, onde são usados *tags* para classificar os elementos armazenados em forma texto, com objetivo de tornar as informações tratáveis por um computador, identificando a função da informação no documento.

A utilização de padrões de marcação internacionais abertos (SGML, HTML, XHTML, XML, etc.) permite assim a criação de documentos portáteis, isto é, documentos que não são dependentes de um determinado *software*, *hardware*, ou sistema operacional. Documentos que contêm apenas texto ASCII que podem ser interpretados por aplicações presentes nos mais diversos ambientes computacionais, ao contrário de formatos binários, bastando que exista uma aplicação no ambiente que reconheça o padrão usado na criação do documento. Assim minimizam o problema de transferência de um formato de representação para outro e liberam a informação das tecnologias de informação proprietárias.

Além disso, permitem múltiplas apresentações do documento, de forma totalmente independente da mídia de veiculação, sejam desktops, celulares, PDA's, etc. A aplicação que deve tratar a informação é que se encarrega de interpretar as marcas e processá-las, para efeitos de apresentação, ou outros processamentos.

2.3. XML (*eXtensible Markup Language*)

Com o surgimento do XML, um padrão definido pela W3C (*World Wide Web Consortium*) em 1990 com a proposta de combinar a flexibilidade da SGML com a simplicidade da HTML, foram desenvolvidas diversas linguagens de marcação nele baseadas para diversos contextos de aplicação. Trata-se de uma metalinguagem, ou seja, um conjunto de regras que permitem definir outras linguagens. Considerado de grande importância na Internet e em grandes Intranets, provê grande capacidade de inter operação dos computadores por ser um padrão flexível, aberto, independente de dispositivo e permite múltiplas formas de visualização das informações.

2.4. Estrutura da linguagem XML

Uma especificação XML é formada por até três arquivos distintos, um deles contendo efetivamente as informações (I), um segundo documento utilizado para validar as informações (II) e um terceiro, opcional, descrevendo o formato de apresentação (III).

I. Documento XML

Um documento XML contém um texto formado por um conjunto estruturado de tags, não havendo restrições quanto ao número destes. No início do texto estão presentes as

informações de identificação do documento, do seu(s) validador(es) e formato de apresentação, seguidas do conteúdo se deseja representar.

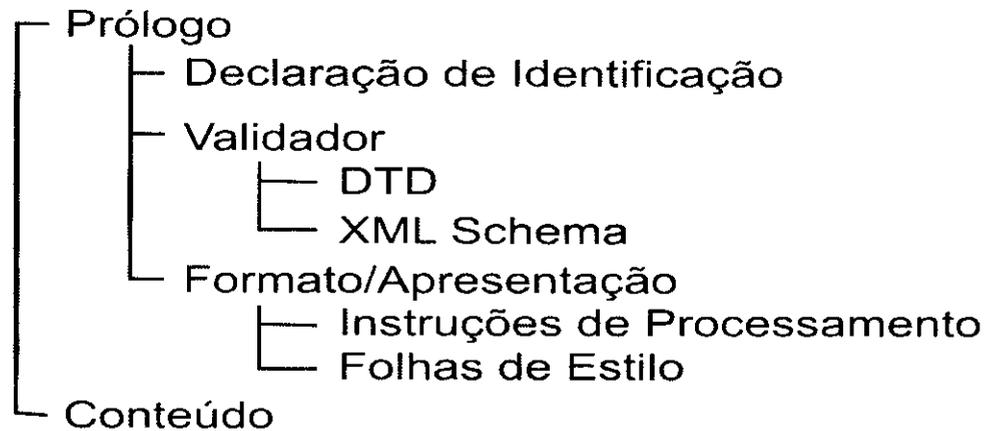


Figura 1: Estrutura de um documento XML

II. Documento de validação do XML

Uma opção para definir as regras que definem um documento XML é o uso de **DTDs** (*Document Type Definition*) [12], que tem a função de validar os dados quando a aplicação que os recebe não possui internamente uma descrição dos dados que está recebendo. Um DTD especifica a estrutura de um documento listando os elementos, atributos, entidades e notações que podem ser usados, assim como suas possíveis relações com outro documento.

Schemas são documentos que definem a validade de documentos XML, semelhante ao DTD. Foram criadas várias linguagens schema, muitas delas estão fora de uso. A mais aceita atualmente é a *W3C XML Schema Language*, criada pelo *W3C XML Schema Working Group* [8], que possui uma grande capacidade de especificar conteúdos de texto e atributos.

III. Folha de estilos

Uma folha de estilo define como um documento XML deve ser apresentado em determinada interface.

3. Especificação de Interfaces utilizando XML

O projeto tomado como base para elaboração deste trabalho de conclusão de curso teve uma primeira etapa realizada entre agosto de 2006 e maio de 2007, onde propôs um padrão de representação das informações apresentadas em telas de um sistema de supervisão de uma indústria petroquímica de forma ergonômica e ubíqua (que está ao mesmo tempo em toda parte) para os diferentes dispositivos utilizados no contexto de supervisão e controle.

Para a representação das informações foi escolhida a linguagem XML devido a sua facilidade de aplicação a qualquer nível de complexidade, além de prover grande capacidade de inter operação entre dispositivos por ser flexível, aberta e independente de equipamento ou plataforma. A marcação pode ser alterada de uma forma mais geral para uma mais detalhada.

Por possuir um maior nível de especificação e ser amplamente usado por desenvolvedores o validador escolhido foi o *W3C XML Schema Language*.

Sistemas de supervisão contêm diversos objetos gráficos como botoeiras, armazenadores e motores. Devido a sua limitação de recursos para edição de imagens, as folhas de estilo não foram usadas para representação das telas de supervisão.

Neste trabalho a especificação XML proposta inicialmente para um caso particular foi modificada de modo a estender sua aplicação a outros contextos de automação.

3.1. Arquitetura do sistema

Para a integração entre os dispositivos, foi utilizada uma estrutura cliente/servidor, onde os dados de natureza dinâmica apresentados nas telas (alterados durante a execução do sistema de supervisão) foram armazenados em um Banco de Dados presente no servidor, que se comunica com os clientes representando os dados em documentos XML, através da Internet.

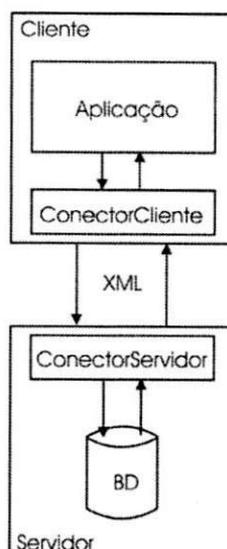


Figura 2: Arquitetura de comunicação

3.2. Especificação XML

Após estudo de alguns padrões baseados em XML usados em trabalhos semelhantes ao realizado, como XIML e UIML, foi decidida a criação de uma notação XML própria.

A figura abaixo mostra a estrutura elementar da notação.

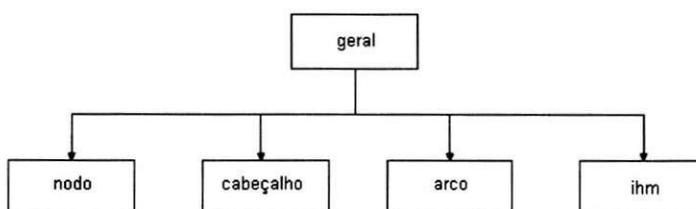


Figura 3: Estrutura elementar da notação particular proposta

Este conjunto contém o elemento “geral” como raiz e os demais tendo a seguinte função/relação:

- **cabeçalho**: elemento que contém os dados identificadores da tela (e.g. título, subtítulo);
- **nodo**: elemento que encapsula um equipamento representado na interface de supervisão (e.g. tanque, bomba, válvula, entre outros). Cada nodo possui um identificador único;

- **objetos de interface:** elementos que permite a representação de objetos compostos (e.g. quadro e painel);
- **arco:** elemento que tem por fim representar as tubulações e, por conseguinte, a relação entre os diversos elementos conectados por tubulações.

Derivados do elemento **nodo** foram definidos os elementos que representam os objetos contidos nas telas de supervisão. São eles:

- **armazenador:** unidade de armazenamento a exemplo tanques, vasos, baterias ou discos rígidos. Podem conter dispositivos de medição e alerta;
- **atuador:** chave ou regulador de fluxos, como válvulas ou relés. Podem conter dispositivos de medição e alerta;
- **terminal:** origem ou destino daquilo que é transportado, podendo se referenciar a outra parte do sistema ou mesmo a outro sistema;
- **conversor:** elementos com uma ou mais entrada e uma ou mais saídas, como transformadores de energia elétrica ou roteadores. Dispositivos de medição e alerta podem ser associados ao conversor e a cada entrada ou saída;
- **sensor:** monitoram grandezas não relacionadas a outros objetos como detectores de gás ou medidores de temperatura e umidade do ambiente. Podem conter dispositivos de alerta;
- **sinalização:** alguns símbolos são usados para, por exemplo, indicar o sentido de fluxo ou lógica aplicada ao sistema;
- **objeto simples:** objeto meramente ilustrativo.

Os **objetos de interface** podem ser definidos como:

- **quadro:** tabela com informação do sistema;
- **painel:** tabela interativa com informações do sistema;
- **botoeira:** pode alterar o modo de funcionamento da planta ou da forma como ela é visualizada;
- **alerta:** mensagem de alerta ou recomendações de procedimentos expostos na tela.

3.3. Normalização da Especificação XML

Definida a estrutura da especificação, foram aplicadas algumas regras na construção da notação de modo a normalizá-la e facilitar sua compreensão:

- Todos os tags contêm apenas letras minúsculas;
- Não utilizar preposições;
- Atributos e elementos filhos referenciam os elementos pais imediatamente;
- Não são usados acentuação ou hífen por não serem aceitos em alguns aplicativos;
- Usa-se subscrito “_” para separar palavras.

3.4. Generalização da Especificação XML

A notação XML desenvolvida para o sistema de supervisão petroquímico foi estendida para outros sistemas mostrou que a especificação tornando-se aplicável a ambientes de automação industrial caracterizados pela presença de armazenamento e transferência de dados, energia ou substâncias.

Foi criada uma especificação genérica baseada no XML Schema específico da indústria petroquímica, chamada de Schema Generalizado. Com a generalização houve perda de especificidade em alguns atributos, como exemplo três atributos que informavam se havia injetor de espuma, faixa verde-amarela e revestimento nos tanques foram substituídos por um único atributo, subtipo_armazenador.

Em seguida foi criado um documento XML para descrição de uma subestação energética a partir de seu diagrama unifilar e, tomando como exemplo um modelo de rede *http* em software simulador de redes de computadores foi desenvolvido um documento XML para representar os dados apresentados na simulação.

Após a primeira etapa de generalização do XML Schema poucas alterações foram necessárias para aplicações fora do contexto petroquímico. Foram elas:

- Inserção sub-elemento de conversor lista_rotas, uma lista dos possíveis destinos dos pacotes encaminhados pelos roteadores;
- Os sub-elementos de conversor passaram a ser opcionais;

- Adição de dois atributos ao elemento arco: taxa_entrada, taxa_saida.

O processo de generalização foi concluído com a incorporação de algumas declarações restritivas que haviam sido retiradas. Estas restrições não causam prejuízos a generalização por especificarem atributos que podem assumir um mesmo conjunto de valores independentemente do contexto de aplicação. São elas:

- Estado dos atuadores: aberto, fechado, em manutenção, sem comunicação, ligado, desligado;
- Modo de acionamento dos atuadores: solenóide, pneumático, manual;
- Tipos de terminação: entrada, saída, entrada/saída.

Segue abaixo um trecho de código referente a especificação de um elemento armazenador em XML.

- **Especificação em XML Schema do objeto tanque cilíndrico:**

```
<xs:element name="nodo">
<xs:complexType>
<xs:choice minOccurs="0" maxOccurs="unbounded">
<xs:element name="armazenador">
<xs:complexType>
<xs:sequence>
<xs:element name="medidor_num_armazenador" minOccurs="0"
maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="alarme_medidor_num_armazenador" minOccurs="0"
maxOccurs="unbounded">
<xs:complexType>
<xs:attribute name="id_alarme_medidor_numarmazenador" type="xs:string"
use="required"/>
<xs:attribute name="mensagem_alarme" type="xs:string" use="optional"/>
<xs:attribute name="estado_alarme" type="xs:boolean" use="required"/>
<xs:attribute name="tipo_alarme" type="xs:string" use="optional"/>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="id_medidor_num" type="xs:string" use="required"/>
<xs:attribute name="tipo_medidor_num" type="xs:string" use="required"/>
<xs:attribute name="rotulo_medidor_num" type="xs:string" use="optional"/>
<xs:attribute name="unidade_medidor_num" type="xs:string" use="required"/>
<xs:attribute name="valor_absoluto_medidor_num" type="xs:float"
use="required"/>
</xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
```

```

</xs:complexType>
</xs:element>
<xs:element name="medidor_graf_armazenador" minOccurs="0"
maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="alarme_armazenador" minOccurs="0" maxOccurs="unbounded">
<xs:complexType>
<xs:attribute name="id_alarme_medidor_graf_armazenador" type="xs:string"
use="required"/>
<xs:attribute name="mensagem_alarme" type="xs:string" use="optional"/>
<xs:attribute name="estado_alarme" type="xs:boolean" use="required"/>
<xs:attribute name="tipo_alarme" type="xs:string" use="optional"/>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="id_medidor_graf_armazenador" type="xs:string"
use="required"/>
<xs:attribute name="conteudo_medidor_graf" type="xs:string" use="required"/
>
<xs:attribute name="valor_relativo_medidor_graf" type="xs:float"
use="required"/>
<xs:attribute name="rotulo_medidor_graf" type="xs:string" use="optional"/>
<xs:attribute name="tipo_medidor_graf" type="xs:string" use="optional"/>
</xs:complexType>
</xs:element>
<xs:element name="alarme_armazenador" minOccurs="0" maxOccurs="unbounded">
<xs:complexType>
<xs:attribute name="id_alarme_armazenador" type="xs:string" use="required"/
>
<xs:attribute name="mensagem_alarme" type="xs:string" use="optional"/>
<xs:attribute name="estado_alarme" type="xs:boolean" use="required"/>
<xs:attribute name="tipo_alarme" type="xs:string" use="optional"/>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="id_armazenador" type="xs:string" use="required"/>
<xs:attribute name="conteudo_armazenador" type="xs:string" use="optional"/>
<xs:attribute name="rotulo_armazenador" type="xs:string" use="optional"/>
<xs:attribute name="tipo_armazenador" type="xs:string" use="required"/>
<xs:attribute name="subtipo_armazenador" type="xs:string" use="optional"/>
</xs:complexType>
</xs:element>

```

A definição de um tanque contido em uma tela de supervisão ilustra a aplicação da especificação XML.

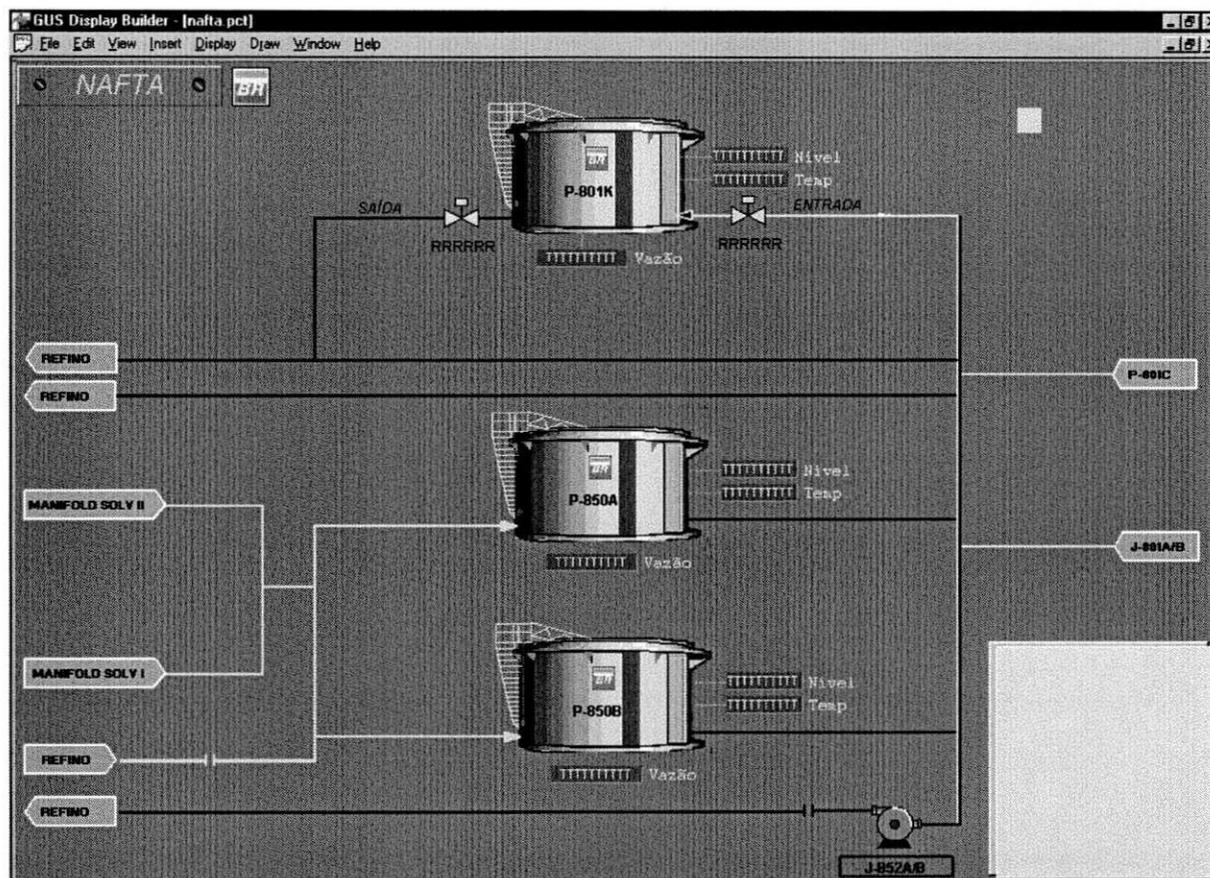


Figura 4: Tela de supervisão de uma industria petroquímica

➤ **Exemplo de definição de um tanque cilíndrico em XML:**

```
<nodo id_nodo="49">
<armazenador tipo_armazenador="tanque_cilindrico" id_armazenador="P-801K"
subtipo_armazenador="normal">
<medidor_num_armazenador id_medidor_num="1" tipo_medidor_num="temperatura"
unidade_medidor_num="oC" valor_absoluto_medidor_num="35"
rotulo_medidor_num="Temp."/>
<medidor_num_armazenador id_medidor_num="2" tipo_medidor_num="nivel"
unidade_medidor_num="m3" valor_absoluto_medidor_num="125"
rotulo_medidor_num="Nivel"/>
<medidor_num_armazenador id_medidor_num="3" tipo_medidor_num="vazao"
unidade_medidor_num="m3/s" valor_absoluto_medidor_num="75"
rotulo_medidor_num="Vazao"/>
<medidor_graf_armazenador id_medidor_graf_armazenador="1"
conteudo_medidor_graf="azul" valor_relativo_medidor_graf="50"/>
</armazenador>
</nodo>
```

A especificação genérica busca ser o mais abrangente possível sem perder a capacidade de especificação de interface, razão pela qual apresenta uma grande quantidade de atributos e elementos opcionais. Como consequência o XML Schema Genérico é extenso, pois deve contemplar todos os possíveis elementos e atributos do sistema que se deseja descrever. Já um documento XML para uma tela particular é mais enxuto, contendo apenas as informações presentes numa tela particular.

Não havendo modificações no XML Schema, o que é um fato pouco freqüente, sua transmissão é realizada apenas quando um dispositivo entra na rede de supervisão. Assim, seu tamanho não é uma questão de grande importância, mesmo quando se deseja um volume de tráfego de dados reduzido.

4. Aplicação de Web Semântica a representação de Sistemas de Supervisão

A maior parte das páginas *World Wide Web* é codificada em HTML, uma linguagem de marcação usada para apresentação visual de objetos (em geral textos e imagens) em um *browser*. As informações contidas em páginas HTML podem ser lidas e interpretadas por seres humanos, mas não existe uma descrição do significado dos objetos apresentados.

As ontologias fornecem o vocabulário necessário para a comunicação entre os agentes e as páginas da Web, definindo as relações entre os conceitos, como salientam **Dziekaniak e Kirinus** [9]. Para as autoras, na prática, uma ontologia define termos associados aos textos que descreve, o que os mesmos significam e axiomas formais que restringem a interpretação e o uso dos termos.

Basicamente o que se deseja é organizar os dados onde eles estejam descritos e ligados de tal maneira que seja possível além da apresentação, a manipulação desse conhecimento por computadores. A idéia é que se evolua de uma Web na qual temos recursos e ligações sem descrição para uma Web onde se tenha recursos e relacionamentos identificáveis.

4.1. Estrutura da Web Semântica

A W3C tem sugerido uma série de tecnologias organizadas em camadas, onde se tem definidas desde a codificação de caracteres até as camadas lógicas no topo.

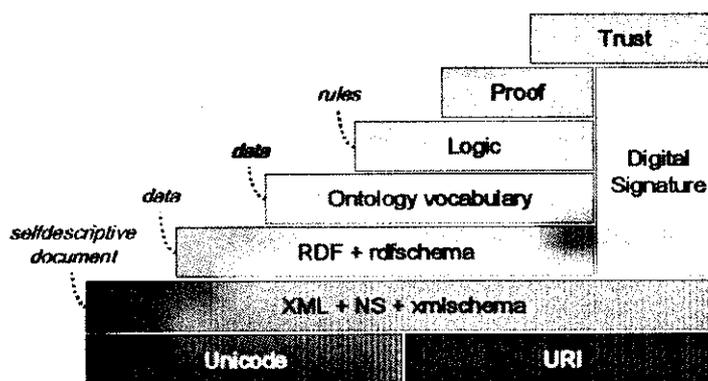


Figura 5: Estrutura de camadas da Web Semântica - fonte W3C Web Semantic Activity [2]

As camadas mais baixas, Unicode e URI (*Uniform Resource Indicator*), são responsáveis, respectivamente, pelo uso padronizado do mesmo conjunto de caracteres e pela forma unívoca de identificação e localização de páginas.

A linguagem XML foi escolhida para codificação para a Web Semântica e tem a função de padronizar a publicação e troca de dados entre aplicativos. Nela são descritas as informações em uma estrutura hierárquica de marcadores. O XML foi escolhido, pois estende a funcionalidade do HTML, projetado para apresentação de dados e definição de links entre documentos, sendo bem mais flexível.

RDF (*Resource Description Framework*) é um modelo de dados para referenciar objetos e como eles estão relacionados. Nesta camada são descritos recursos através de declarações no formato de triplas (sujeito, predicado, objeto).

Por recursos entende-se qualquer coisa do mundo que esteja sendo descrita. São caracterizados por propriedades que possuem um valor. Um valor pode ser literal ou outro recurso.

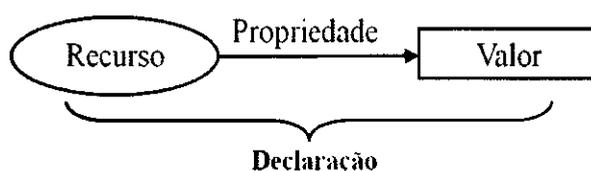


Figura 6: Declaração RDF

A vantagem de RDF como linguagem de descrição de recursos sobre DTDs e esquemas XML reside na liberdade de ignorar as imposições da estrutura do documento, referindo-se apenas aos dados sobre o conteúdo. Para uma padronização de uso de RDF, foram criados os esquemas RDF (RDF-Schemas ou RDFS), que fornecem tipos básicos para a criação de esquemas voltados a aplicações específicas. As primitivas a serem usadas para modelar novos esquemas RDF incluem classe, subclasse (herança), propriedade, subpropriedade (para construir hierarquias de propriedades), instância e restrição.

RDFS é o vocabulário para descrever os recursos na forma de hierarquia de classes e propriedades. Através de classes é possível especificar quais propriedades devem aparecer na descrição de um recurso.

A camada de Vocabulário Ontológico é a mais importante e pesquisada da Web semântica. Ela é responsável por oferecer a expressividade necessária à representação de ontologias. Isso é feito aproveitando a extensibilidade de RDFS para definir restrições

complexas e outras construções que implementam características de *frames* e lógica de descrições.

Adiciona mais vocabulário para descrever propriedades e classes: entre outros, relações entre classes, cardinalidade, igualdade, características de propriedades e classes de enumeração. De acordo com **Smith et al 2003** [10], já existe uma linguagem padrão, a OWL (*Web Ontology Language*), que deriva de um consenso entre duas propostas, a européia OIL e a DAML. A OWL é subdividida em três sub linguagens, de acordo com sua expressividade:

- **OWL Lite:** abrange a expressividade de *frames* e lógica de descrições, com algumas restrições. Por exemplo, a cardinalidade máxima ou mínima assume apenas os valores 0 ou 1. Apesar disso, a linguagem é dotada de riqueza semântica, sendo, por isto, ideal para usuários iniciantes e desenvolvedores que preferem *frames* a lógica de descrições.
- **OWL DL:** garante completude, decidibilidade e toda a expressividade da lógica de descrições, almejando satisfazer engenheiros de conhecimento familiarizados com esta tecnologia. A expressividade torna-se ainda maior do que em OWL Lite: classes podem ser construídas por união, interseção e complemento, pela enumeração de instâncias e podem ter disjunções. Tipos são mantidos cuidadosamente separados (por exemplo, uma classe não pode ser instância e propriedade ao mesmo tempo).
- **OWL Full:** fornece a expressividade de OWL e a liberdade de usar RDF, inclusive permitindo novas metaclasses, já que elas são subclasses definidas em RDFS. Fazendo este uso mais complexo, não há garantia de computabilidade. Aqui, não cabem as restrições de separação de tipos da versão anterior e é possível manipular e modificar metaclasses.

As linguagens menos expressivas (OWL Lite e DL) estão contidas dentro das mais expressivas (OWL DL e Full), de maneira que uma ontologia definida numa linguagem menos expressiva é aceita por uma linguagem mais expressiva; a recíproca, naturalmente, não é verdadeira.

As camadas mais altas da proposta de Web semântica ainda não tomaram corpo. A camada lógica permite a especificação de regras que atuam sobre instâncias e recursos, enquanto a camada de prova as executa, e a de confiança avalia se a prova está correta. Para que essas camadas entrem em operação, as camadas inferiores devem estar bem sedimentadas, o que ainda está acontecendo. Além do mais, sob o ponto de vista ontológico,

não é interessante antecipar o uso de ontologias com regras, pois isto pode restringir a sua aplicabilidade.

Na Assinatura Digital é verificada a autoria de um documento. Está fora do escopo deste projeto, o desenvolvimento desta tarefa pode ser um tópico para trabalhos posteriores.

4.2. Aplicação de Web Semântica a Representação de Interfaces

Neste trabalho são abordadas as camadas de ontologia e descrição de dados: XML/XMLS, RDF/RDFS e Vocabulário Ontológico.

Atualmente, com diversos estudos sobre a Web Semântica, diversas linguagens e ferramentas para definição e edição de ontologias têm sido propostas. Recomendadas pelo W3C, as principais linguagens são o RDF/RDFS e OWL.

4.3. Ferramentas de Desenvolvimento

Existem diversas ferramentas para desenvolvimento de ontologias incluindo modelagem, construção de bases de conhecimento, visualização ou gerenciamento de projetos. Muitas dessas ferramentas foram desenvolvidas para um projeto particular de algum instituto de pesquisa. Dentre as principais temos: Protégé; OntoEdit; Ontolingua; Altova SemanticWorks 2008; OilEd e WebODE.

Protégé é uma ferramenta grátis e “*open source*” de desenvolvimento. Consiste em um editor intuitivo para ontologias e tem diversos plug-ins disponíveis para a construção. Pode exportar ontologias em vários formatos, incluindo RDF, RDF Schema, OWL e XML Schema.

OntoEdit é um editor de ontologias que suporta o desenvolvimento colaborativo, permitindo aos usuários a construção de ontologias de acordo com a metodologia proposta por Bruno Bachimont.

OilED é um editor gratuito que permite ao usuário a construção e manipulação de ontologias e que utiliza uma ferramenta de software capaz de inferir conseqüências lógicas a partir de um conjunto de fatos afirmados ou axiomas, para classificar e verificar a coerência das ontologias.

WebODE (Web Ontology Development Environment) é um editor com suporte para múltiplos usuários e para a metodologia Menthontology, onde são oferecidas orientações para

especificação de ontologias no nível do conhecimento, como a especificação de uma conceituação.

A ferramenta Ontolingua foi construída para facilitar o desenvolvimento de ontologias com uma interface com base em formulário Web. É uma ferramenta que suporta edição e criação distribuída e colaborativa de ontologias. O usuário deve ter alguma noção de KIF (Knowledge Interchange Format) e da linguagem Ontolingua. A interface não apresenta boa ergonomia.

Altova Semantic Works é um editor visual comercial para Web Semântica fácil de usar para a criação e edição de ontologias. Pode ser baixado gratuitamente e utilizado por um período de 30 dias. O usuário pode imprimir representações gráficas de RDF e OWL afim criar documentação para implementações de Web Semântica. É possível alternar da forma gráfica do RDF ou OWL para a visualização em formato texto para ver como documentos estão sendo construídos nos formatos RDF/XML, OWL ou N-triplas. O código é gerado automaticamente. O editor é capaz de trabalhar com arquivos XML, OWL, RDF, RDFS e N-triplas. Foi a ferramenta escolhida por ser de fácil uso e atender plenamente as necessidades do trabalho.

4.4. Construção da camada OWL

Na camada de vocabulário ontológico os elementos que compõem as interfaces de supervisão são definidos através de uma hierarquia em classes. Também são definidas as propriedades de cada classe. A sub linguagem OWL escolhida foi a OWL DL por garantir expressividade e não possuir complexidade muito elevada como a OWL Full.

A estrutura de classes foi construída de forma semelhante a estrutura elementar da especificação desenvolvida em XML, apresentada no item 3 deste trabalho, e está representada na Figura 5.

As propriedades definidas para cada classe equivalem aos atributos definidos na notação XML Genérica. Buscou-se também aplicar um vocabulário ontológico utilizado por outros desenvolvedores e padronizado por uma organização, onde são descritos um conjunto de elementos para caracterização de um domínio.

Foi utilizado um vocabulário definido pela a *Dublin Core Metadata Initiative* (DCMI) [13] é uma organização dedicada a promover a adoção generalizada de metadados inter

operáveis e o desenvolvimento de um vocabulário padrão para descrever recursos que permitam sistemas de busca de informação mais inteligentes.

Contudo, foi necessário definir algumas propriedades adicionais para os medidores e alarmes, elementos que indicam variáveis características do processo produtivo. Foram elas: unidade (unidade de engenharia), valor (valor de uma variável medida), estado (condição de operação).

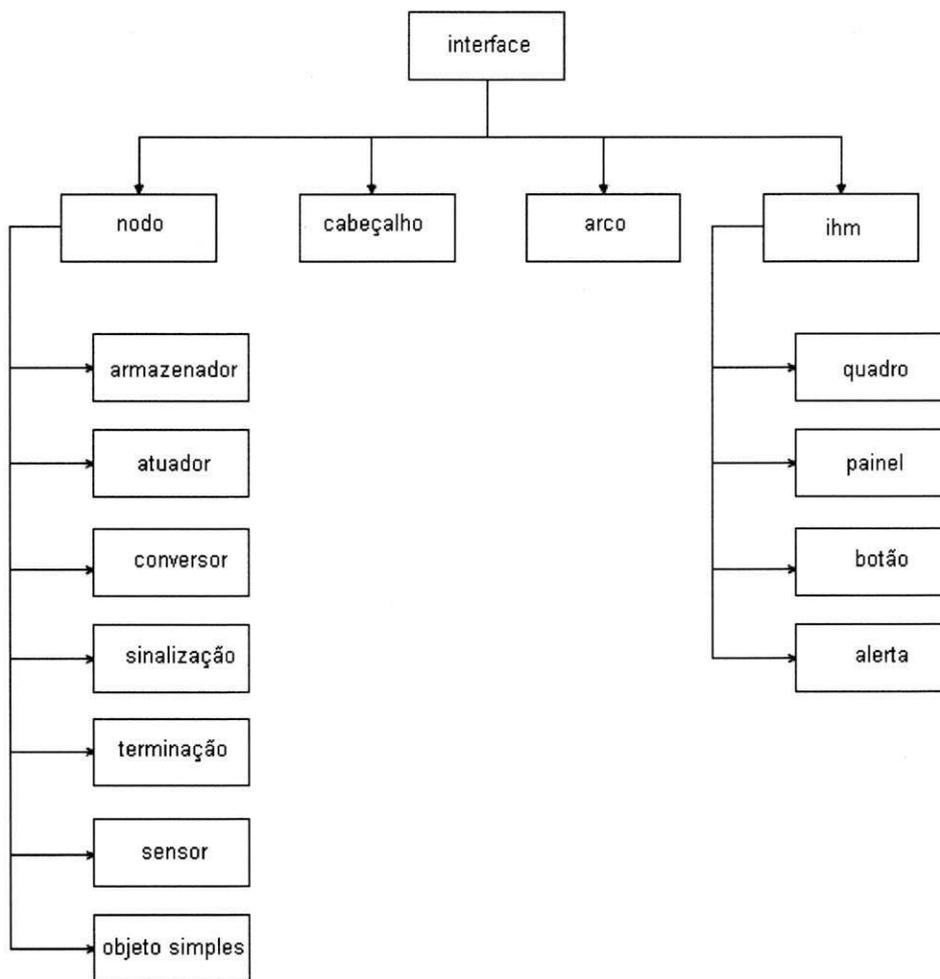


Figura 7: Classes OWL

4.5. Construção da camada RDF

Na camada de descrição dos recursos são definidos os elementos que compõem cada interface de supervisão. Para cada tela de interface é criado um arquivo com extensão .rdf, contendo a descrição de seus elementos, como armazenadores ou atuadores.

As informações definidas na camada de vocabulário ontológico (OWL) podem ser referenciadas pelo documento RDF, tornando os recursos (classes, propriedades e instâncias) disponíveis para serem usados.

A seguir é apresentado um trecho de código para descrição de um armazenador contidos na tela Nafta (Figura 4) e um grafo correspondente a descrição (Figura 7).

➤ Exemplo de descrição de um tanque em RDF:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:itf="http://www.web/interface#" xmlns:nft="http://www.web/nafta#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://
www.w3.org/2000/01/rdf-schema#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
<rdf:Description rdf:about="http://www.web/interface#armazenador">
<rdf:nil>
<rdf:Description>
<dc:format
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">cilindrico_normal</d
c:format>
<dc:identifier
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">P-801K</dc:identifie
r>
<dc:type
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">tanque</dc:type>
</rdf:Description>
</rdf:nil>
</rdf:Description>
<rdf:Description rdf:about="http://www.web/interface#medidor_graf">
<rdf:nil>
<rdf:Description>
<dc:identifier
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">LI_04</dc:identifie
r>
<dc:type
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">azul</dc:type>
<dc:relation
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">P-801K</dc:relation>
<itf:unidade rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
%</itf:unidade>
<itf:valor
rdf:datatype="http://www.w3.org/2001/XMLSchema#short">50</itf:valor>
</rdf:Description>
</rdf:nil>
</rdf:Description>
```

```

<rdf:Description rdf:about="http://www.web/interface#medidor_num">
<rdf:nil>
<rdf:Description>
<itf:unidade
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">oC</itf:unidade>
<itf:valor
rdf:datatype="http://www.w3.org/2001/XMLSchema#short">35.0</itf:valor>
<dc:identifier
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">TI_01</dc:identifier
>
<dc:relation
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">P-801K</dc:relation>
<dc:title
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">TEMP.</dc:title>
<dc:type
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">temperatura</dc:type
>
</rdf:Description>
</rdf:nil>
<rdf:nil>
<rdf:Description>
<rdf:type>nivel</rdf:type>
<itf:unidade
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">m3</itf:unidade>
<itf:valor
rdf:datatype="http://www.w3.org/2001/XMLSchema#short">125.0</itf:valor>
<dc:identifier
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">LI_01</dc:identifier
>
<dc:relation
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">P-801K</dc:relation>
<dc:title
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Nivel</dc:title>
</rdf:Description>
</rdf:nil>
<rdf:nil>
<rdf:Description>
<itf:unidade rdf:datatype="http://www.w3.org/2001/XMLSchema#string">m3/
h</itf:unidade>
<itf:valor
rdf:datatype="http://www.w3.org/2001/XMLSchema#short">75.0</itf:valor>
<dc:identifier
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">FI_01</dc:identifier
>
<dc:relation
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">P-801K</dc:relation>
<dc:title
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Vazao</dc:title>
<dc:type
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">vazao</dc:type>
</rdf:Description>
</rdf:nil>
</rdf:Description>
</rdf:RDF>

```

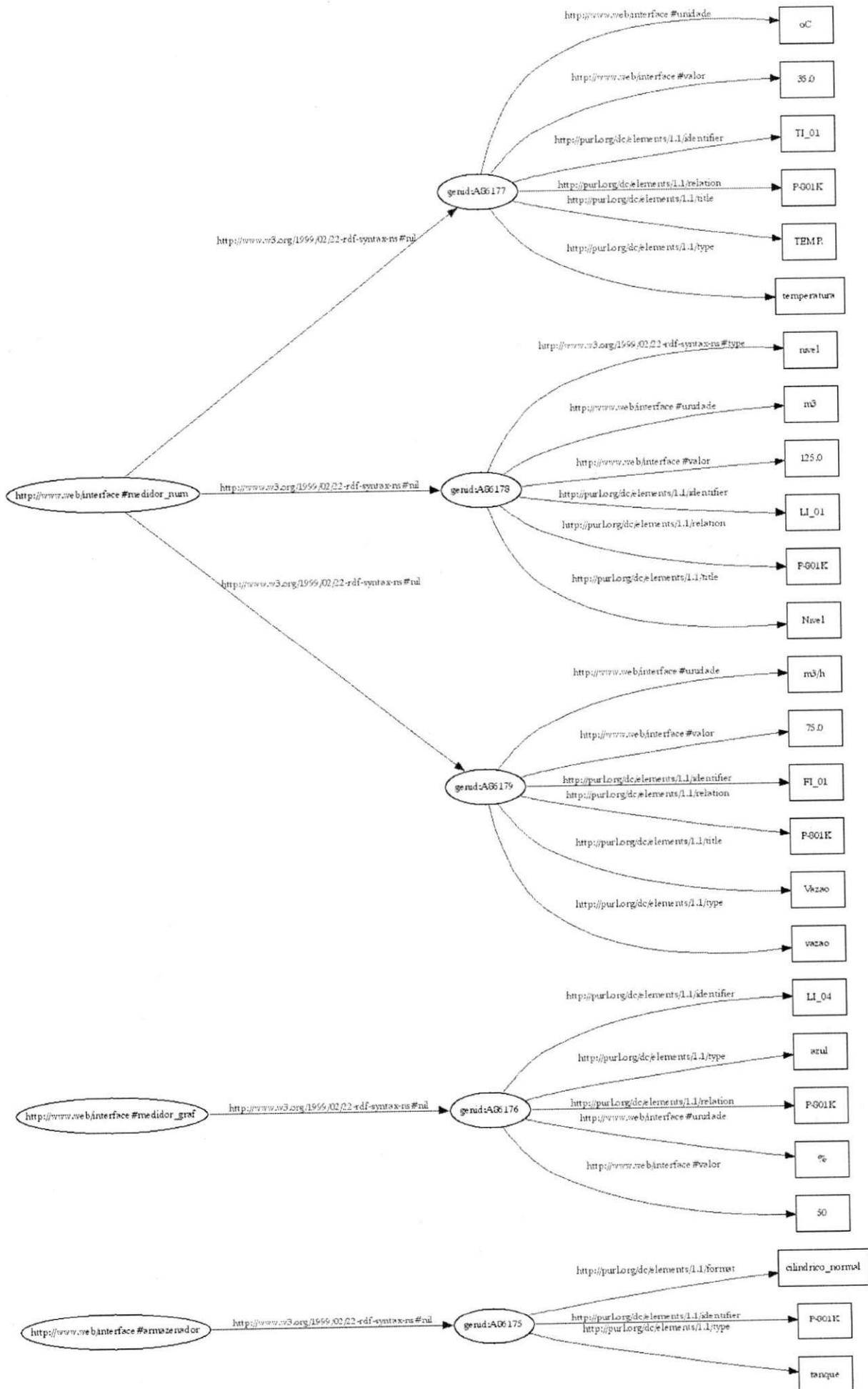


Figure 8: Grafo representando a definição do tanque P-801K em RDF

A adaptação da notação XML genérica a padrões de Web Semântica ofereceu um formalismo para especificação e um vocabulário ontológico definido por uma iniciativa internacional. O uso de um vocabulário compartilhado é um fato de grande importância para o compartilhamento dos dados na Internet com outras aplicações Web.

Em termos de capacidade de descrição, não houve diferença significativa da migração da notação XML pura para as linguagens OWL e RDF.

5. Ferramentas Utilizadas

- **Altova XML Spy 2008**

Disponível em: http://www.altova.com/products/xmlspy/xml_editor.html

- **Altova Semantic 2008**

Disponível em: http://www.altova.com/products/semanticworks/semantic_web_rdf_owl_editor.html

- **RDF Validation Service**

Disponível em: <http://www.w3.org/RDF/Validator/>

6. Considerações Finais

A utilização de um padrão XML como meio para representar informações referentes às interfaces de supervisão é uma forma viável para integração de dispositivos com capacidades distintas de exibição da informação. A interface pode ser descrita em um simples arquivo de texto e a tela de supervisão ou controle pode ser apresentada ao usuário de acordo com os recursos do equipamento disponível.

A generalização da notação XML anteriormente concebida para plantas petroquímicas foi generalizada para outros ambientes de automação industrial, também caracterizados pelo armazenamento e transferência de itens tais como: dados, energia elétrica ou substâncias químicas. A especificação genérica foi aplicada de forma satisfatória aos contextos: petroquímico, redes de computadores e sistemas de distribuição de energia elétrica, exceto pela perda de especificidade decorrente do processo de generalização, a qual pode ser contornada com a adição de um ou mais atributos específicos ao elemento representado, quando necessário. Nos casos abordados não houve necessidade.

A característica de compartilhamento da informação é muito desejada na Web, para agentes de busca e comércio eletrônico. Nas indústrias a definição formal de um vocabulário ontológico pode significar a possibilidade de maior integração entre aplicações de diferentes fabricantes. A utilização de um conjunto de elementos definidos em língua estrangeira resultou numa especificação definida em dois idiomas. Em uma possível continuação deste trabalho é recomendável migrar totalmente a notação para a língua inglesa, a mais utilizada em aplicações Web.

O estudo sobre atualização dos dados representados de acordo com alterações sofridas pelas variáveis que caracterizam os processos industriais ou por mudanças nas plantas de produção, e a implantação de protocolos de segurança, definindo restrições, privilégios e prioridades no acesso à informação são possíveis temas para trabalhos posteriores.

7. Referências

1. eXtensible Markup Language
Disponível em <http://www.w3.org/XML/>
Acesso em 31 de outubro de 2008
2. W3C Web Semantic Activity
Disponível em <http://www.w3.org/2001/sw/>
Acesso em 31 de outubro de 2008
3. Studer, R.; Benjamins, R.; Fensel, D.. 1998. Knowledge Engineering: Principles and Methods. *Data and Knowledge Engineering*. 25(1998):161-197.
4. Moreira, A.; Alvarenga, L.; Oliveira, A. P.. O nível do conhecimento e os instrumentos de representação: tesouros e ontologias. *Datagramazero: Revista de Ciência da Informação*, v. 5, n. 6, dez. 2004.
Disponível em http://www.dgzero.org/dez04/Art_01.htm
Acesso em 23 de abril de 2008
5. Souza, R. R.; Alvarenga, L.. A Web Semântica e suas contribuições para a ciência da informação. *Ci. Inf.*, Brasília, v. 33, n. 1, p. 132-141, jan./abr. 2004.
6. Freitas, F. L. G.. Ontologias e a web semântica. Universidade Católica de Santos. 2004.
7. Gruber, T. R.. 1995. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human and Computer Studies*, 43(5/6): 907-928.
8. XML Schema Working Group
Disponível em http://www.w3.org/XML/Schema_
Acesso em 31 de outubro de 2008
9. Dziekaniak, G. V.; Kirinus, J. B.. Web Semântica. *Enc. Bibli: R. Eletr. Bibliotecon. Ci. Inf.*, Florianópolis, n. 18, p. 20-39, 2004.
Disponível em http://www.encontros-bibli.ufsc.br/Edicao_18/2_Web_Semantica.pdf
Acesso em 31 de outubro de 2008

10. Smith, M., Welty, C., McGuinness, D.; 2003. *Web Ontology Language (OWL) Guide Version 1.0*.
Disponível em <http://www.w3.org/TR/2003/WD-owl-guide-20030210/>
Acesso em 31 de outubro de 2008
11. Puerta, A., Eisenstein, J.. Developing a Multiple User Interface Representation Framework for Industry. A. Seffah e H. Javahery (eds) *Multiple User Interfaces: Engineering and Application Frameworks*. John Wiley and Sons, 2003.
12. Harold, E.R.. *XML 1.1 Bible*. Wiley Publishing Inc. 2004.
13. Dublin Core Metadata Initiative
Disponível em <http://dublincore.org/>
Acesso em 30 de outubro de 2008
14. Vieira, M. F.; Netto, A.; Scherer, D.; Araújo, T.; Patrício, P. A Pervasividade de Dispositivos de Operação em Ambientes de Automação Industrial. XVII Congresso Brasileiro de Automática, 2008.