

# INTERFACE DE COMUNICAÇÃO ENTRE AMBIENTE DE MODELAGEM ORIENTADO A OBJETOS E SISTEMAS SUPERVISÓRIOS

JAIR JONKO ARAUJO<sup>1</sup>, LEANDRO BUSS BECKER<sup>2</sup>, CARLOS EDUARDO PEREIRA<sup>3</sup>

<sup>1</sup>*Centro Federal de Educação Tecnológica de Pelotas,  
96015-360, Pelotas, RS, BRAZIL*

<sup>2</sup>*Instituto de Informática, UFRGS  
Cx. P. 15064, 91501-970, Porto Alegre, RS, BRAZIL*

<sup>3</sup>*Departamento de Engenharia Elétrica, UFRGS  
90035-190, Porto Alegre, RS, BRAZIL*

*E-mails:* jonko@cefetrs.tche.br, lbecker@inf.ufrgs.br, cpereira@delet.ufrgs.br

**Resumo**— Sistemas supervisórios são componentes importantes no contexto dos sistemas de automação industrial. Recentemente, o uso do paradigma de orientação a objetos tem se mostrado uma técnica eficiente para modelagem e implementação deste tipo de sistemas. Todavia, verifica-se uma carência em relação a interação entre ferramentas de modelagem orientadas a objetos e as ferramentas para desenvolvimento de sistemas supervisórios. O presente trabalho tem por objetivo descrever a definição e implementação de uma interface de comunicação entre um ambiente de modelagem orientado a objetos com características para tempo-real e um sistema supervisório. Esta interface possibilita a construção de sistemas supervisórios a partir dos elementos que foram definidos no ambiente de modelagem, garantindo a consistência entre o sistema implantado e o sistema que foi modelado. Para implementação da mesma, foi utilizada a tecnologia DCOM.

**Abstract**— The object-oriented paradigm has been used extensively for modeling and design of distributed real-time systems. Recent researches prove that the use of object-orientation paradigm in the development of industrial automation systems tend to produce better-structured and more reusable programs. Since supervisory tools are also a very useful element in the context of real-time industrial automation systems, this paper argues about how to link the use of object-oriented models with the design of supervisory systems. In particular, an interface that allows the link between a modeling tool and a supervisory system is presented. The goal is to enhance the consistency of the data presented during supervision, making it logically related with the designed model. The DCOM technology was used for implementing this interface.

**Keywords**— object-oriented design, supervisory systems, communication interface.

## 1 Introdução

A necessidade do aumento de produção, da redução de custos e do aumento da qualidade de bens de consumo, tem motivado a constante evolução dos sistemas de produção. Equipamentos autônomos, cooperando entre si através de redes, aliado a um fluxo contínuo de diferentes produtos pelas linhas de produção, tornam as plantas de produção cada vez mais difíceis de serem projetadas, especificadas e administradas. De modo similar, os sistemas computacionais que permitem o controle desses dispositivos também crescem em complexidade. Uma das principais características dos sistemas de automação industrial é o alto grau de acoplamento desses sistemas com os componentes situados na planta.

Nesse contexto, cresce a necessidade de ferramentas de projeto com capacidade para modelar, monitorar e intervir sobre a aplicação. Os sistemas supervisórios possuem funcionalidades para permitir a monitoração e o controle de sistemas de automação. Já os ambientes de modelagem permitem a descrição dos requisitos da aplicação, através da construção de diagramas. Todavia, estudos de casos [BEC99b] demonstram que um dos maiores problemas nas ferramentas de modelagem existentes

é a fraca interação entre modelagem, visualização e supervisão. O projetista geralmente modela e simula o sistema em uma determinada ferramenta, identificando e corrigindo os erros de projeto. Entretanto, quando este for implantado, o supervisório será construído sem nenhum vínculo com as definições do sistema modelado, o que poderá acarretar incoerência entre a modelagem e a supervisão. Este fato motivou o desenvolvimento de um ambiente para modelagem e geração de sistemas supervisórios [TIB00], derivado a partir do ambiente de modelagem SIMOO-RT [BEC99a].

Este trabalho apresenta a definição e implementação de uma interface de comunicação flexível [ARA00], capaz de permitir a interação entre o SIMOO-RT e um sistema supervisório mesmo que localizados em ambientes heterogêneos. Assim, a partir da especificação do modelo, elementos criados na fase de modelagem também poderão ser utilizados no projeto do sistema supervisório. De maneira reversa, as alterações dos parâmetros desses elementos que forem efetuadas no supervisório, geram atualizações no modelo. Para implementação de uma interface com a flexibilidade de comunicação desejada foi utilizada a tecnologia DCOM [GRI98]. Como alternativa ao uso do DCOM, também poderia ser utilizado outro modelo de distribuição, como por

exemplo o CORBA [VIN99]. No entanto, escolheu-se o primeiro devido a sua grande difusão dentre os desenvolvedores de sistemas de automação. Salienta-se que o presente trabalho complementa a extensão definida em [TIB00], facilitando a importação e exportação dos elementos de supervisão definidos durante a modelagem.

As próximas seções do presente trabalho encontram-se divididas da seguinte forma: na seção 2 são descritos os elementos envolvidos na criação da interface proposta; na seção 3 detalha-se o projeto e a implementação desta interface; posteriormente, as especificações propostas são validadas através de exemplos; por fim, na seção 5 apresentam-se as conclusões deste trabalho.

## 2 Elementos Constituintes da Interface

### 2.1 Sistemas Supervisórios

Os sistemas de supervisão [ELI99, TIB00] permitem que sejam monitoradas e rastreadas informações do processo produtivo. Tais informações são primeiro coletadas através de equipamentos de aquisição de dados, seguido da manipulação e análise destes dados e posteriormente são apresentadas ao usuário.

Outra função importante é a possibilidade do sistema supervisorio executar ações baseadas em parâmetros antecipadamente informados, fazendo que o usuário participe do processo de controle apenas quando as ações exijam a intervenção humana. Para permitir isso são usadas *Tags* que são todas as variáveis numéricas ou alfanuméricas envolvidas na aplicação, podendo executar funções computacionais (operações matemáticas, lógicas, com vetores ou *strings*, etc.) ou representar pontos de entrada e saída de dados do processo que está sendo controlado. Neste caso, correspondem às variáveis do processo real (como temperatura, vazão, nível, etc.), se comportando como a ligação entre o controlador e o sistema. É com base nos valores das *Tags* que os dados coletados são apresentados para o usuário.

Um sistema supervisorio deve apresentar os dados coletados do processo para o usuário de uma forma simples e significativa. Isto é possível através da utilização de gráficos, botões e ícones coloridos. A apresentação destes gráficos é alterada de acordo com a variação dos valores das *Tags*. Como algumas sofrem mudanças frequentes, é possível criar animações de figuras com a alternância dos valores.

Uma situação de alarme é identificada quando o valor da *Tag* ultrapassa uma faixa pré-estabelecida. Esta faixa possui dois valores para o limite superior e dois valores para o limite inferior, sendo denominados altíssimo, alto, baixo e baixíssimo (*High-High*, *High*, *Low* e *Low-Low*). As ações automáticas citada acima, são programadas quando da ocorrência destes alarmes.

### 2.2 Ambiente SIMOO-RT

O ambiente SIMOO-RT [BEC99a] é uma ferramenta CASE (ferramenta que provê suporte computacional às atividades de desenvolvimento de sistemas computacionais), que tem como objetivo principal a criação de modelos orientados a objetos com restrições temporais, oferecendo recursos para permitir a simulação do modelo e a geração automática de código. Os modelos são construídos no editor de modelos (denominado MET – *Model Editor Tool*), sendo compostos por diversos diagramas, tais como o diagrama de classes, o diagrama de instâncias e o diagrama de transição de estados. Este último formaliza a especificação dos possíveis estados de execução das classes/objetos e dos eventos que provocam as transições entre estes estados.

Desta forma, o editor suporta a descrição de aspectos estáticos e dinâmicos do modelo. Os aspectos estáticos são descritos graficamente, através dos diagramas de classe e de instâncias, enquanto que o comportamento dinâmico de cada entidade é descrito, nos diagramas de transição de estados e em *templates* na linguagem C++. A partir da descrição do modelo, o editor pode gerar automaticamente o código executável necessário para a simulação e também para a aplicação final.

O SIMOO-RT foi estendido a fim de permitir a utilização do modelo na criação de sistemas supervisorios [TIB00]. Para isso, é necessário definir, em tempo de modelagem, propriedades relacionadas com a construção de um sistema supervisorio. Como exemplo destas propriedades, cita-se os limites de valores para os atributos das classes e a possibilidade de escolher a forma de sua representação no sistema supervisorio. O elemento básico para essa tarefa é o Elemento de Visualização (EV). Um EV representa a possibilidade de apresentação gráfica de um atributo, de um estado, de um método ou mesmo uma classe. Um ou mais EVs formam uma janela no supervisorio. Os EVs que podem ser definidos na modelagem são *Gauge*, *Display*, *Bar Graph* ou *Bitmap Animation*. Cada um desses objetos será usado para um fim específico no supervisorio.

O *Bitmap Animation* têm por finalidade permitir a criação de animação no supervisorio, enquanto o *Bar Graph* possibilita a criação de elementos que possam ser representados como um gráfico de barras. Por sua vez, o *Display* possui a função de apresentar valores lidos diretamente de um coletor de dados ou simulados pelo sistema, num formato adequado. Estes EVs são compostos por um conjunto de atributos. Alguns destes atributos são comuns a todos os EVs, como por exemplo a posição, o tamanho, entre outros. Todavia, cada EV também possui um conjunto de atributos específicos, relacionados com a sua função, como por exemplo a orientação da barra, cores, título e grade no EV *Bar Graph*, o conjunto de

imagens *bitmaps* que serão apresentados no EV *Bitmap Animation*, etc.

### 2.3 Tecnologia DCOM

A partir da tecnologia DCOM (*Distributed Component Object Model*) [GRI98] diferentes programas aplicativos podem ser integrados, comunicando-se de uma maneira transparente, usando a infra-estrutura oferecida pelo sistema operacional, como se estivessem no mesmo espaço de endereçamento de memória. As plataformas Windows 95/98/NT 4.0, Apple Macintosh, Sun Solaris, AIX, MVS e Linux, possuem suporte a DCOM.

Sua arquitetura é composta, de modo simplificado, por uma interface, um servidor e um cliente. Uma interface define o comportamento ou as capacidades de um componente de software com um conjunto de métodos e propriedades, garantindo a consistência semântica do objeto que a suporta, denominado objeto servidor. As interfaces são identificadas através de um *Interface Identifier* (IID) e os objetos servidores através de *Class Identifier* (CLSIDs). Ambos são registrados pelo sistema operacional, que armazena suas características, quando executamos o processo de registro do servidor DCOM.

O objeto que utiliza os métodos e propriedades definidos na interface é denominado cliente. Quando um cliente solicita um CLSID, o DCOM localiza e ativa o servidor. O cliente recebe um ponteiro para interface e as chamadas subseqüentes são invocadas diretamente para a instância criada.

## 3 Projeto e Implementação da Interface

Para construir um sistema supervisorio a partir da ferramenta de modelagem é necessário definir, em tempo de projeto, os EVs que farão parte do modelo. Posteriormente, os atributos desses EVs são transferidos ao supervisorio através de uma interface adequada. Por fim, os EVs são associados às Tags no ambiente de desenvolvimento do supervisorio, e posteriormente usados no processo de supervisão da aplicação, a fim de informar ao usuário valores coletados em campo. A figura 1 apresenta um diagrama contendo a estrutura da interface proposta para fazer a conexão entre uma ferramenta de modelagem e um ambiente supervisorio.

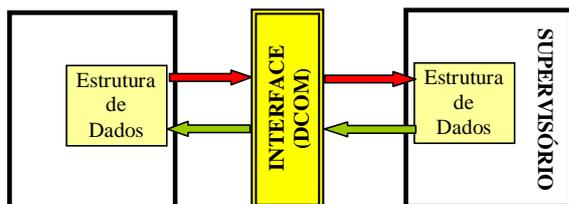


Figura 1 - Esquema da proposta

Os EVs são associados a atributos de usuários, os quais são definidos nas classes durante a construção do modelo. Cada atributo poderá ter um

ou mais EVs, do mesmo tipo ou de tipos diferentes. Não é permitido dois EVs do mesmo tipo, referentes ao mesmo atributo e a mesma classe, com o mesmo nome. Na figura 2 pode-se observar a janela do SIMOO-RT, acessada quando uma classe é selecionada no MET, onde são definidos os EVs de cada atributo.

Para construir a interface buscou-se uma solução de comunicação entre os programas que permitisse a troca de informação durante sua execução, com a transferência de dados já carregados em memória, usando recursos oferecidos pelo sistema operacional. Desta forma, foi adotada a tecnologia DCOM da Microsoft que, além de suportar as funcionalidades desejadas, é um padrão bastante difundido entre os desenvolvedores de sistemas de automação.

A fim de permitir o envio e o recebimento dos dados dos EVs entre os dois programas, o objeto servidor foi definido com uma propriedade. Essa propriedade é capaz de ler dados de um objeto, o qual contém os atributos de um EV selecionado no modelo, e que será transferido ao supervisorio. Também é capaz de escrever outro objeto com os atributos de um EV enviado pelo supervisorio, o qual poderá ser incluído no modelo.

Todos os tipos de dados usados devem ser previamente mapeados para tipo padrão DCOM, pois é o sistema operacional que movimenta os dados e a interface deve acessível por programas escritos em qualquer linguagem.

O resultado da implementação da interface e do objeto servidor é um arquivo, padrão DCOM, com as informações de tipo e com descrição dos elementos disponibilizados pelo servidor. Esse arquivo será conectado ao SIMOO-RT transformado-o em um servidor DCOM. Posteriormente, ao implementar um supervisorio como um cliente DCOM, ambos serão capazes de trocar informações entre si.

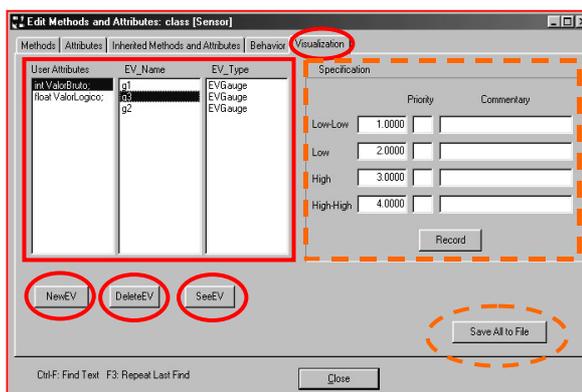


Figura 2 - Janela de Definição dos EVs

Quando o sistema supervisorio abrir o servidor para estabelecer a comunicação, o SIMOO-RT será carregado na memória. Deve-se abrir o arquivo do modelo que contenha os dados que serão transferidos ao supervisorio. A seguir, através de um botão localizado na barra de ferramentas do editor o formulário de interface com o usuário deverá ser

aberto para seleção dos EVs. Todas as tarefas de preparação dos dados dos EVs, que serão acessados pelo objeto servidor, para envio ao sistema supervisorio e o tratamento dos dados vindos do mesmo, serão realizados por métodos associados a esse formulário.

Na figura 3 observa-se o formulário de interface com o usuário que é composto por quatro janelas, as quais são preenchidas a partir da estrutura de dados SIMOO-RT. Na primeira delas é mostrado, em forma de árvore, as classes definidas para o modelo no MET. Quando uma classe da árvore é selecionada, os seus atributos são mostrados na janela seguinte. Ao selecionar um atributo dessa janela, se existir EVs definidos para essa classe, seu nome será mostrado na terceira janela, com o tipo correspondente exibido na quarta. Caso contrário uma mensagem informa ao usuário que não há EV definido para essa classe.

Quando um EV for selecionado na terceira janela, seus dados tornam-se disponíveis para serem acessados pelo objeto servidor, podendo ser transferido para o cliente. Essa seleção deve ser feita individualmente.

Quando o usuário fechar essa janela os dados que foram enviados pelo cliente, o supervisorio, serão salvos automaticamente no modelo. Se o cliente enviou várias vezes o mesmo EV, serão armazenadas no modelo as alterações que foram enviadas na última vez.

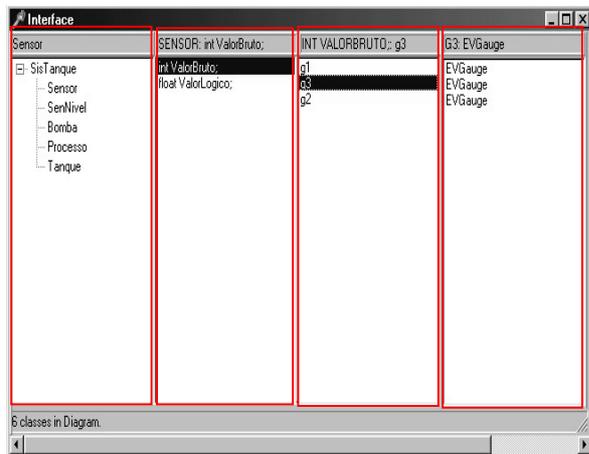


Figura 3 – Janela de Exportação dos EVs

#### 4 Validação da Interface

Como o cliente DCOM do SIMOO-RT será um software de supervisão, foi implementado um software supervisorio genérico, somente com as funcionalidades necessárias para validar a interface descrita na seção anterior. A comunicação com o processo externo, característica típica de um sistema supervisorio, não foi implementada por fugir ao escopo deste trabalho. Este software, que é totalmente independente da ferramenta de modelagem, utiliza os recursos da tecnologia DCOM para se comunicar com o SIMOO-RT. Assim, o sistema supervisorio pode encontra-se localizado em

um nodo de execução distinto, independente de plataforma.

A janela principal do sistema supervisorio implementado é mostrada na figura 4. Dentre as suas funções, destaca-se a capacidade de edição de EVs e telas. As telas representam diferentes visões do sistema, permitindo adequar a maneira de mostrar as informações conforme a necessidade do operador. Por exemplo, a um operário de chão de fábrica interessado na dinâmica do processo seria mais interessante fornecer uma janela gráfica com componentes animados. Já para um gerente, planilhas com informações estatísticas sobre a produção seriam mais adequadas. Os EVs são os elementos básicos que compõem as telas. Com isto, a partir da janela principal do supervisorio, é possível derivar-se as seguintes janelas:

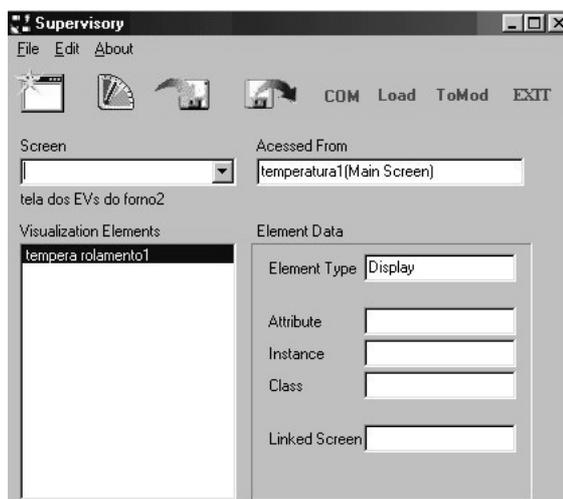


Figura 4 – Janela principal do software supervisorio de teste

**Screen:** permite a seleção de uma tela já existente. Mostra a tela que está atualmente selecionada;

**Visualization Elements:** mostra os EVs que foram editados na tela atual, ou seja, a tela que está selecionada na janela *Screen*;

**Accessed From:** mostra qual o EV que permite o acesso à tela atual e em que tela ele se encontra;

**Element Data:** mostra alguns dados de um EV selecionado, relacionados com o modelo. Apenas EVs que tenham sido obtidos a partir do SIMOO-RT possuem todos os campos *attributes*, *instance* e *class* preenchidos e, por isso, podem ser reenviados ao modelo. Os EVs criados no sistema supervisorio, não podem ser incorporados ao modelo.

Através do menu da parte superior pode-se, além das tarefas executadas pelos botões, editar e apagar EVs ou telas selecionados.

Além da implementação das funções básicas de criação e edição de EVs e telas, o sistema proposto foi transformado em um cliente DCOM para poder receber e transferir EVs do SIMOO-RT. Desta forma, quando o botão *COM* é pressionado, o objeto servidor é instanciado e a propriedade que foi definida para o servidor pode ser acessada como se estivesse definida localmente. Se o SIMOO-RT não

estiver em memória ele será carregado no momento de instanciar o objeto. A partir de então, os dados dos EVs podem ser transferidos entre ambos os programas, como se estivessem acessando variáveis comuns. Nesse protótipo isso é feito através dos botões *Load* e *ToMod*.

Conforme já citado no item 3, as informações capturadas pelo sistema supervisorio, através dos EVs, provêm da estrutura estática do modelo OO, ou seja, dos atributos pertencentes aos objetos da aplicação.

Na figura 5 pode-se ter a visão de todos os elementos envolvidos na transferência de dados do SIMOO-RT para o sistema supervisorio. Na parte superior da figura observam-se as janelas do SIMOO-RT e na inferior as janelas do supervisorio de teste. Estas janelas ilustram, junto com a interface proposta, uma transferência de dados entre os dois softwares.

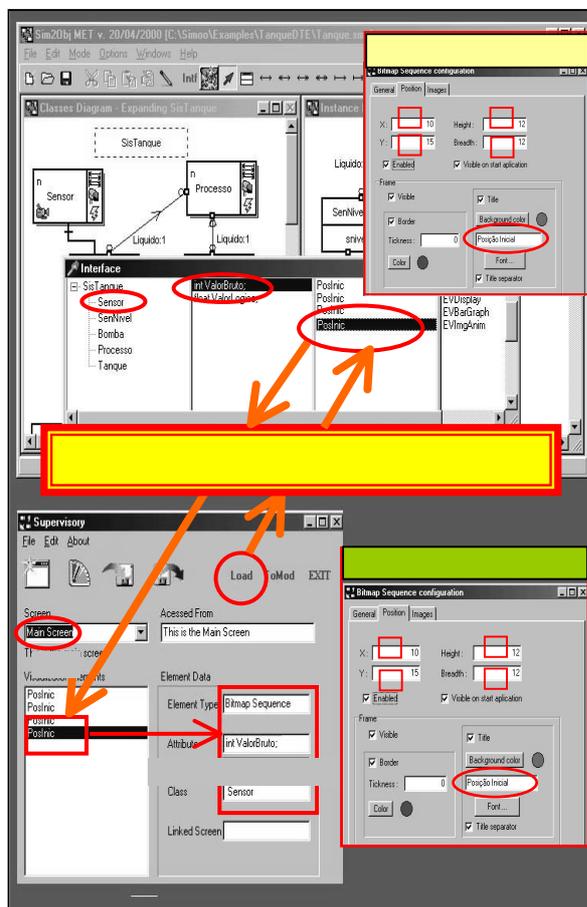


Figura 5 - Transferência de dados em o SIMOO-RT e o supervisorio de teste

Quando for pressionado o botão *Load* no supervisorio, os atributos do EV selecionado na janela de exportação de EVs do SIMOO-RT (figura 3) são transferidos ao supervisorio, usando os recursos da tecnologia DCOM, conforme descrito nos itens 2.3 e 3. Este EV será incorporado na tela selecionada no supervisorio, sendo os atributos mapeados para a sua estrutura de dados.

Nesse caso, pode-se observar a correspondência entre os valores dos atributos através de janelas de

visualização de EVs. A janela superior, onde está escrito "Atributos do EV", pode ser acessada a partir da janela de definição dos EVs (figura 2), no SIMOO-RT, para visualizar os atributos definidos para um EV na modelagem. A janela inferior é acessada no supervisorio, para visualizar os dados de um EV incorporado em uma tela do mesmo. A semelhança entre as duas deve-se apenas ao fato de se ter aproveitado um formulário já existente na implementação do supervisorio.

## 5 Conclusões

Este trabalho apresenta uma estratégia para realizar a integração efetiva entre ambientes de modelagem e sistemas supervisorios. Os resultados obtivos comprovam a viabilidade de transferência de dados entre ferramentas distintas, permitindo a criação de supervisorios a partir de atributos especificados no modelo orientado a objetos, minimizando o tempo de projeto e garantindo coerência entre o que foi modelado e o que será supervisionado. A interface implementada permite a comunicação do SIMOO-RT com qualquer sistema supervisorio que seja implementado como cliente DCOM, permitindo, inclusive, a comunicação com sistemas que rodem em plataformas não Windows.

É importante salientar que a proposta apresentada não é necessariamente dependente do SIMOO-RT, sendo que a restrição para implementá-la em outro ambiente de modelagem é ter-se acesso ao modelo de dados e diagramas armazenados. A recente definição por parte do *Object Management Group* (OMG) do padrão XML, para armazenagem e intercâmbio de diagramas orientados a objetos, tende a facilitar a migração da interface para outros ambientes.

Como sugestão, verifica-se que algumas modificações técnicas podem ser feitas para melhorar a definição adotada para a interface. Por exemplo, poderia ser possível a seleção múltipla de EVs na interface e o envio simultâneo de vários EVs ao sistema supervisorio. Todavia, acrescenta-se que o sistema foi validado através de diversos estudos de caso, encontrando-se totalmente operacional.

## Referências Bibliográficas

- [ARA00] ARAUJO, Jair Jonko. **Interface de Comunicação entre Ambiente de Modelagem e Sistema Supervisorio**. Monografia (bacharelado) –UFPEL, Pelotas, 2000.
- [BEC99a] BECKER, Leandro Buss & PEREIRA, Carlos Eduardo. **SIMOO-RT: Ambiente Integrado para Desenvolvimento de Sistemas de Tempo Real Distribuídos usando o Paradigma de Orientação a Objetos**. 2º. Workshop Brasileiro de Sistemas de Tempo Real. Salvador, BA, 1999. p.68-75.

- [BEC99b] BECKER, Leandro Buss; PARDI Jr., Wilson & PEREIRA, Carlos Eduardo. **Proposal of an Integrated Object-Oriented Environment for the Design of Supervisory Software for Real-Time Industrial Automation Systems.** IEEE Workshop on Object-Oriented Real-Time Dependable Systems, Santa Barbara, USA, 1999. pp.357-368.
- [ELI99] ELIPSE Software Ltda.. **Sistema de Supervisão e Controle: Manual do Usuário.** 1999, disponível em <<http://elipse.com.br>> Acesso em: 30 ago. 2000.
- [GRI98] GRIMES, Richard. **Professional DCOM Programming.** Wrox Press. Olton, UK. 1998.
- [VIN99] VINOSKI, Steve, Advanced CORBA Programming with C++. Addison-Wesley, Reading, Massachusetts, 1999.
- [TIB00] TIBOLA, Leandro Rosniak. Geração de Sistemas Supervisórios a partir de Modelos Orientados a Objetos de Aplicações Industriais. Dissertação de Mestrado – UFRGS, Porto Alegre, 2000.