

BIBLIOTECAS DE AUTOMAÇÃO – OTIMIZANDO O DESENVOLVIMENTO DE PROJETOS

Marcos de Oliveira Fonseca¹
Constantino Seixas Filho²

Resumo

Este trabalho mostra a importância de se criar bibliotecas de objetos/componentes de automação para padronizar e aumentar a produtividade do desenvolvimento de projetos de automação. Os padrões IEC 61131 e IEC 61499 fornecem um caminho para se gerar bibliotecas que se aproximam de um modelo ideal de reutilização. Essa tendência deve mudar em definitivo o cenário do desenvolvimento de sistemas de automação, reduzindo drasticamente a expertise e o esforço necessários para se gerar uma aplicação. São apresentadas recomendações práticas para implementação e exemplos de ganhos proporcionados.

Palavras-chave: Automação; Biblioteca; IEC 61131; IEC 61499.

AUTOMATION LIBRARIES – OPTIMIZING PROJECTS DEVELOPMENT

Abstract

This paper shows the importance of building automation objects/components libraries to standardize and increase productivity in automation projects development. The IEC 61131 and IEC 61499 standards offer a way to generate libraries that are close to an ideal model or reutilization. This trend will definitively change the automation systems development scenario reducing drastically the expertise and the effort necessary to generate an application. Practical implementation recommendations and examples of provided gains are presented.

Key words: Automation; Library; IEC 61131; IEC 61499.

¹ *Contribuição técnica ao 12º Seminário de Automação de Processos, 1 a 3 de outubro de 2008, Vitória, ES*

² *1 Engenheiro Eletricista, M.Sc, Diretor da Divisão de Tecnologia da Automação da ATAN Sistemas, Belo Horizonte – MG, Brasil.*

³ *Engenheiro Eletrônico, M.Sc., Diretor de P&D da ATAN Sistemas, Belo Horizonte – MG, Brasil.*

1 INTRODUÇÃO

O grande avanço alcançado pela indústria de eletrônica nas décadas de 70 a 90 se deu graças à criação do circuito integrado, que encapsulava centenas e depois milhares e milhões de transistores em uma única pastilha. Os componentes de uma determinada família, circuitos digitais da família 74LS, por exemplo, podiam ser combinados por engenheiros de aplicação em diversas formas, dando origem a milhares de circuitos com aplicações diferentes. A criação de um circuito integrado requeria grande conhecimento de projeto de eletrônica com transistores, física de semicondutores, além da tecnologia de produção de circuitos integrados. Já a utilização dos componentes poderia ser feita por um engenheiro de aplicação com pouco mais de 60 horas de treinamento. A redução de custo dos componentes fabricados aos milhões e a simplicidade de uso foram molas propulsoras do grande desenvolvimento da engenharia eletrônica.

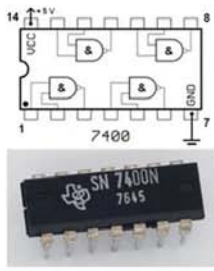
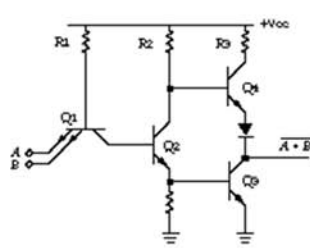

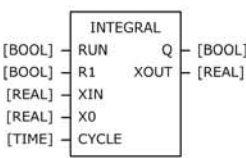
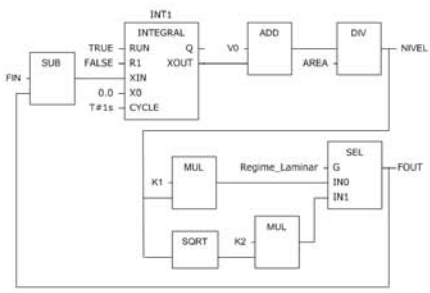
Hardware			
Software		<pre> FUNCTION_BLOCK INTEGRAL VAR_INPUT RUN: BOOL; (* Integra: 1, Mantém: 0) R1: BOOL; (* Reset *) XIN: REAL; (* Entrada *) XO: REAL; (* Vitor Inicial *) CYCLE: TIME; (* Tempo de ciclo *) END_VAR VAR_OUTPUT Q: BOOL; (* 1 = Não reset *) XOUT: REAL; (* Vitor Integrado *) END_VAR VAR XOLD: REAL; FIRST_TIME BOOL := TRUE; END_VAR Q := NOT R1; IF R1 THEN XOUT := XO; ELSF RUN THEN IF FIRST_TIME THEN XOLD = XIN; FIRST_TIME := FALSE; ELSF XOUT = XOUT + 0.5*XIN + XOLD)*TIME_TO_REAL(CYCLE); XOLD = XIN; END IF; END_FUNCTION_BLOCK </pre>	
	Interfaces	Implementação	Aplicação

Figura 1 – Uso do componente requer conhecimento apenas de sua interface. Exemplos de componentes de hardware e software

Na área de produção de programas de computador a definição de componentes reutilizáveis foi sempre o santo gral da engenharia de software (Figura 1). Os primeiros passos nesse sentido surgiram com o conceito de tipo abstrato de dados ou *Abstract Data Type* (ADT). O objetivo era encapsular dados e procedimentos em uma unidade, separando a interface da implementação. Basta conhecer a interface para poder utilizar um dado de um tipo abstrato de dado. A implementação produzida pelo programador do componente fica blindada, protegida e oculta do programador que utiliza o dado. A primeira maneira de se implementar ADT surgiu na programação orientada a objetos com o conceito de classe. Se definirmos uma classe pilha, com os métodos (funções ou procedimentos) *push* e *pop*, poderemos

inserir e retirar itens de um objeto dessa classe, apenas evocando os métodos da classe.

A programação orientada a objetos apresenta quatro princípios básicos, mostrados na Figura 2.

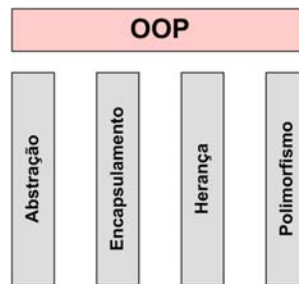


Figura 2 – Princípios da Programação Orientada a Objetos

Abstração

Abstração de dados é o processo de se eliminar todos os detalhes sem importância de um objeto, conservando apenas as características de interesse que o definem. Ao construir uma classe devemos modelar seus atributos e comportamentos.

Encapsulamento

O encapsulamento consiste em ocultar (*hiding*) os detalhes da implementação de uma classe e exibir apenas a interface. O encapsulamento blinda o código e os dados contra modificações e inibe até mesmo o acesso para leitura, protegendo sua propriedade intelectual.

Herança

Consiste em derivar novas classes a partir das classes existentes. Estas classes herdam todas as características dos objetos ancestrais, mas podem apresentar novas propriedades e comportamentos não existentes nas classes ancestrais. Quando uma classe herda de outra, buscamos duas ações: especialização ou extensão. A especialização é muito útil quando criamos classes mais específicas a partir de uma classe genérica. Na extensão, adicionam-se novos comportamentos ao componente. Seria muito interessante definir uma classe tanque bem básica e depois especializá-la em tanque cilíndrico, tanque esférico, tanque com medição de nível, etc.

Polimorfismo

Polimorfismo é a habilidade de diferentes objetos responderem, cada um de sua maneira, à mesma mensagem. Por exemplo, o objeto `moinho_de_bolas` e o objeto `correia_transportadora` ambos entendem a mensagem "liga", apesar de serem objetos de classes diferentes.

Classes evoluíram em direção a componentes. Considerando aplicativos de software, um componente encapsula toda uma aplicação complexa, como por exemplo, o Matlab ou o Excel e pode estar na mesma máquina ou remoto em relação ao programa que o utiliza. Ele pode ser construído em uma dada linguagem e ser utilizado por uma aplicação escrita em outra linguagem.

Os Blocos Funcionais da norma IEC 61131-3 têm muitas semelhanças e diferenças com classes de objetos.^[1] Em primeiro lugar, blocos funcionais também encapsulam dados e comportamentos, propiciam grande reaproveitamento e podem

ser usados em diferentes aplicações uma vez construídos. Blocos Funcionais ainda têm algumas restrições quanto à herança e polimorfismo na atual versão da norma.

Este conceito mudou a maneira de se programar um controlador industrial, ou Controlador Programável (CP). Inicialmente deve-se popular uma biblioteca de controle com as principais funcionalidades de controle desejadas. A biblioteca pode ser constituída, basicamente, pelos seguintes elementos definidos pela norma IEC 61131-3:

- Tipos de Dados;
- Funções;
- Blocos Funcionais;
- Programas.

Existem funcionalidades que são genéricas como um controlador PI ou PID ou um bloco que calcula uma média móvel de uma variável. Outros blocos são específicos para controlar um dado equipamento. Podem ser definidos blocos para controlar tanques, transportador de correia, britador, moinho, etc. Muitas vezes são necessários diversos blocos para abraçar todas as variantes de um dado tipo de equipamento. Por exemplo, para controlar um circuito de moagem, precisamos definir o tipo de moinho utilizado, a estratégia de controle, se o motor da bomba da caixa para os ciclones possui ou não variador de velocidade, etc.

Uma vez constituída uma biblioteca, engenheiros de integração reutilizarão estes blocos para construir sistemas complexos. O engenheiro que projeta o bloco deve ser um especialista naquele tipo de controle ou equipamento e pode gastar várias horas na sua definição e depuração. O engenheiro de aplicação pode ter muito menos experiência específica. Ao usar o conhecimento de um especialista, encapsulado no componente, ele amplia sua capacidade de resolver problemas. Os principais benefícios proporcionados pelo uso de biblioteca dentro da norma IEC 61131 consistem na modularização do problema, reutilização dos elementos de software e estruturação da aplicação de controle.

A norma IEC 61499, publicada em 2004, surge como uma evolução natural da IEC 61131, ampliando a abrangência das melhores práticas desta última, não somente para os controladores programáveis, mas para qualquer sistema de controle inteligente totalmente distribuído. O conceito de componente é a base da IEC 61499, a qual é denominada em Inglês como *Function Blocks for Industrial Process Measurement and Control Systems* (IPMCS), ou simplesmente Blocos Funcionais. Novamente, o conceito de biblioteca de blocos funcionais atuando como componentes de automação é a abordagem utilizada.

Um dos principais motivadores para a elaboração da norma IEC 61499 é a padronização dos mecanismos e das interfaces de eventos para sincronização da execução de parte do programa de controle em diferentes componentes inteligentes interligados em rede. De forma simplificada, a norma IEC 61499 pretende padronizar a filosofia de implementação do controle totalmente distribuído já utilizado pela Fieldbus Foundation em qualquer tipo de sistema ou dispositivo (sensor, atuador, instrumento, inversor etc.).

A futura geração de sistemas de automação^[2] será caracterizada pelo uso de dispositivos totalmente inteligentes, capazes de criar uma “sociedade” e de reconhecer as capacidades de cada um na sociedade como um todo. A componentização dentro deste contexto poderá ser utilizada para a criação de sistemas de controle, nos quais cada componente poderá exibir um comportamento autônomo, capaz de reconhecer alterações no seu ambiente. Um mecanismo importante para permitir este tipo de funcionalidade é o agente, que exige o uso de

protocolos de alto nível para uma comunicação entre os agentes e também adequada ao seu comportamento autônomo. Os dispositivos de automação seriam representados como agentes.

A filosofia de componentização já é muito utilizada em sistemas de programação de computadores. Como exemplo, pode-se citar o COM/DCOM da Microsoft, o CORBA e o Modelo de Serviços da arquitetura SOA. O padrão OPC para comunicação industrial era baseado em COM/DCOM nas versões originais e agora será baseado em *web services*, também fazendo uso da componentização.

Portanto, a aplicação prática de bibliotecas de objetos desde o nível de hardware até os sistemas corporativos já uma realidade nos sistemas de automação e uma tendência crescente dentro da visão de componentes e de agentes. Iniciativas como o Oooneida^[3] tem buscado a utilização das melhores práticas apoiada nas tecnologias e padrões de mercado e, também, através do envolvimento com as principais entidades e organizações atuantes no mercado de automação. Nesta iniciativa, a utilização de bibliotecas de objetos e componentes em diferentes níveis é a base para sua implementação na visão de automação inteligente (Figura 3).

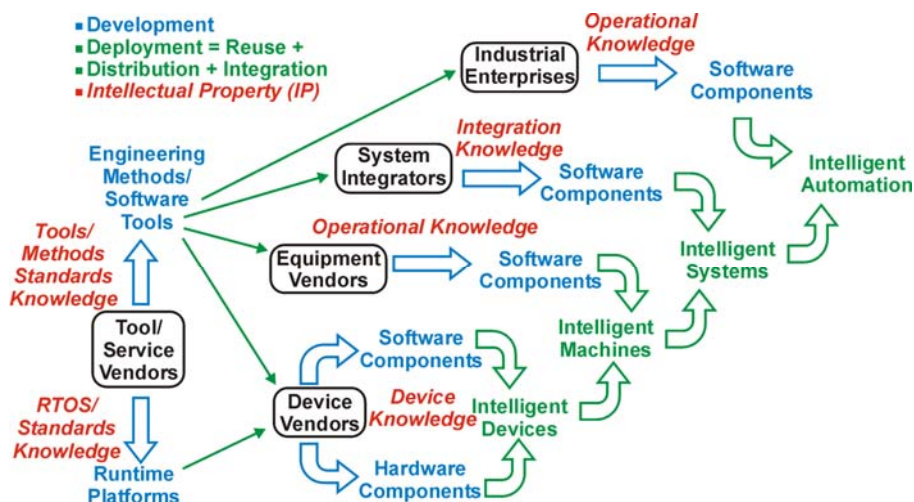


Figura 3 – Visão do OOONEIDA para a cadeia de valor na automação industrial através do uso de componentes

2 BOAS PRÁTICAS PARA UTILIZAÇÃO DE OBJETOS DE AUTOMAÇÃO E BIBLIOTECAS

A utilização de bibliotecas de objetos de automação já é uma prática relativamente conhecida, porém não totalmente aproveitada na sua potencialidade. Entendimentos incompletos dos conceitos associados e até erros de interpretação ainda são comuns na aplicação prática. Um exemplo de interpretação equivocada consiste na idéia de igualdade entre instanciação de objetos e utilização de *templates* para replicação de código. Para exemplificar como podem ser modelados os objetos mais comuns na automação para os níveis de controle e supervisão, serão apresentados a seguir exemplos práticos que se aplicam na maioria dos casos.

A partir da metodologia apresentada por Fonseca, Seixas Filho e Bottura Filho^[1] e mostrada na Figura 4, serão caracterizados os principais objetos de automação que devem compor uma biblioteca para os níveis de controle, principalmente programas e blocos funcionais para controle e simulação, e para o sistema de supervisão, principalmente janelas de operação e símbolos para animação gráfica.

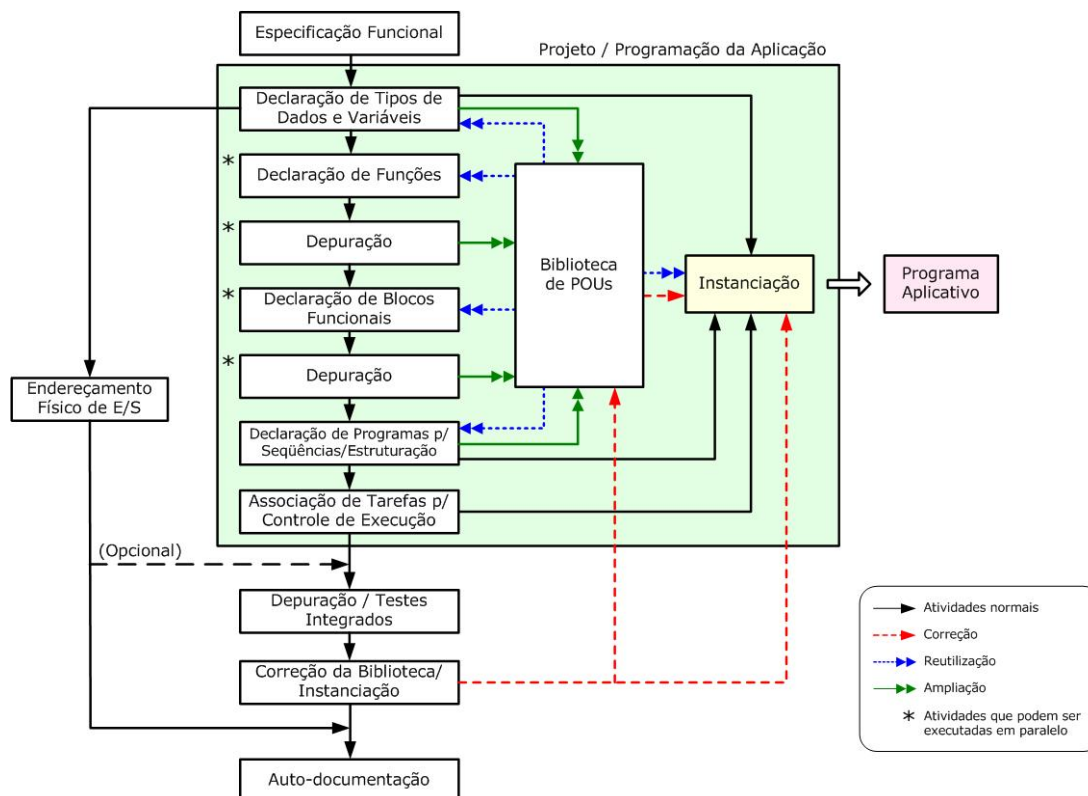


Figura 4 – Metodologia para desenvolvimento de projetos de controle baseada na norma IEC 61131

Funções – Programação de CP

São basicamente rotinas ou procedimentos para tratamento de dados passados como parâmetros. Apesar de não serem objetos propriamente ditos, pois não possuem instâncias, são muito úteis na reutilização de procedimentos repetitivos, como cálculos, manipulação de dados, verificações, etc.

Blocos Funcionais – Programação de CP

São os principais elementos de reutilização de software definidos pela norma IEC 61131-3 e a base para a filosofia definida pela norma IEC 61499. São objetos ou até componentes que possuem métodos (funções ou procedimentos) e uma estrutura de dados (instância), sendo usados para implementar os controles de equipamentos reais de processo ou entidades de programação (blocos de comunicação, por exemplo).

Programas – Programação de CP

Pela norma IEC 61131 podem ser tratados como objetos no nível de controle. Entretanto, a maioria dos ambientes de desenvolvimento disponíveis no mercado ainda não implementa esta característica. Apresentam características semelhantes aos blocos funcionais, mas destinados a aplicações mais complexas no âmbito do controlador e normalmente com vinculação de endereçamento físico.

Simulação

São objetos destinados ao teste e validação de outros objetos, normalmente no nível de controle. Apesar de poderem ser implementados dentro do próprio CP ou como parte dos próprios blocos funcionais, recomenda-se que sejam implementados num sistema à parte destinado apenas à simulação, com integração através da

comunicação OPC, preferencialmente. A implementação no CP é indesejável, pois, normalmente, implica numa competição pelos recursos computacionais, o que mascara o desempenho do sistema testado. Pode ser utilizado também para testes de outros sistemas.

Símbolos para animação gráfica

São objetos gráficos utilizados pelos sistemas de supervisão para possibilitar a visualização e monitoração dos estados e condições dos objetos de processo. Normalmente, são implementados utilizando-se os recursos nativos do ambiente de supervisão, fazendo conexão direta com a estrutura de dados dos objetos do CP via comunicação OPC, preferencialmente.

Janelas de Operação - Faceplates

São objetos gráficos utilizados pelos sistemas de supervisão para possibilitar a operação e monitoração dos estados, condições e diagnósticos dos objetos de controle e de processo. Normalmente, são implementados utilizando-se os recursos nativos do ambiente de supervisão, fazendo conexão direta com a estrutura de dados dos objetos do CP via comunicação OPC, preferencialmente.

A Figura 5 apresenta um esquema com os principais objetos de automação.

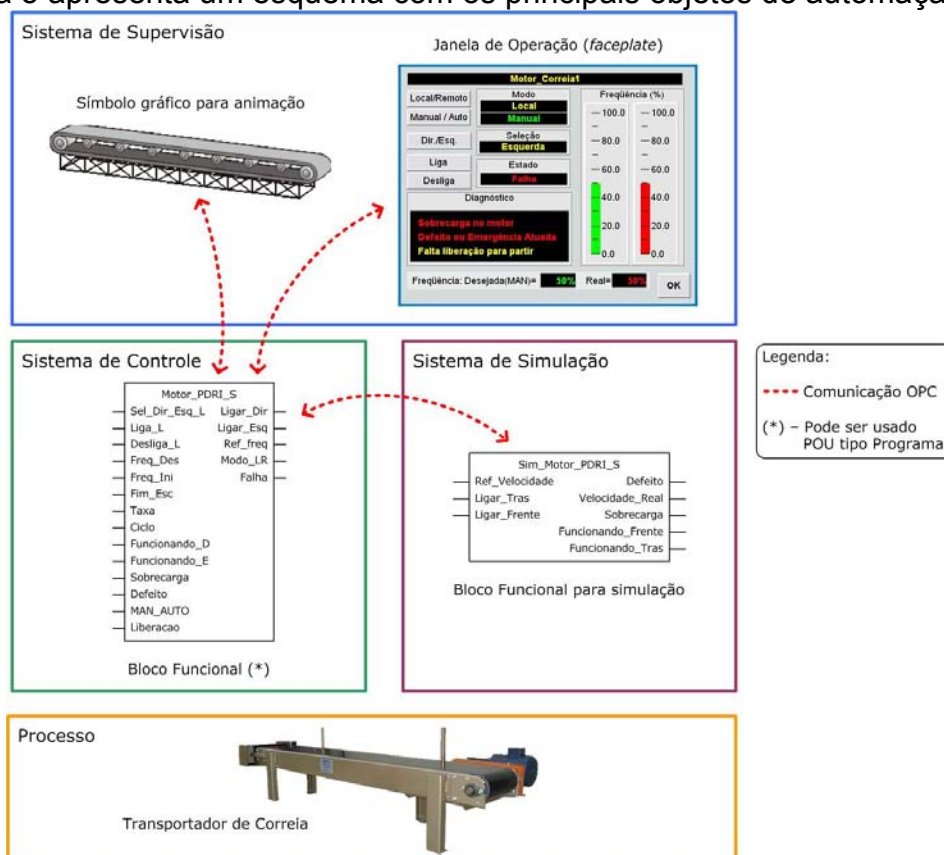


Figura 5 – Principais objetos que compõem a biblioteca de automação

3 VIABILIDADE DO USO DE BIBLIOTECAS

A utilização de bibliotecas proporciona ganhos expressivos em todos os projetos de automação. Dentre os ganhos proporcionados, podemos destacar o aumento da qualidade de software, a preservação do conhecimento empregado na criação do

objeto, a redução dos custos de desenvolvimento e a facilidade de treinamento e manutenção. De forma geral, os ganhos quantitativos típicos proporcionados pelo uso de bibliotecas de objetos para atividades específicas em sistemas de automação convencionais são apresentados na Tabela 1.

Tabela 1 – Ganhos típicos da utilização de biblioteca de objetos em sistemas convencionais

Atividades/Áreas	Ganho Global Esperado
Aplicativo de controle	5 a 30%
Aplicativo da estação de operação	10 a 40%
Testes	10 a 25%
Comissionamento de sistemas	5 a 15%
Qualidade de software	10 a 50%

Para exemplificar a redução de custo proporcionada pelo uso de bibliotecas de automação para o desenvolvimento de um projeto real comparado com projetos que já fazem uso de ambientes de desenvolvimento modernos, mas sem a utilização de bibliotecas de automação, foram estimados quantitativos reais para projetos de uma empresa do setor de mineração, mostrados na Tabela 2.

Tabela 2 – Redução do custo global estimado para utilização de biblioteca de objetos em sistemas modernos.

Dados de Referência de Ganhos Percentuais Típicos para Projetos de Controle e Supervisão									
Porte	Num. Pontos	Entradas Digitais	Saídas Digitais	Entradas Analógicas	Saídas Analógicas	Custo Projeto Normalizado Sem Biblioteca	Custo Projeto Normalizado Com Biblioteca	Redução devido à Biblioteca	Redução Média
Pequeno	500	325	125	40	10	1	0.972	2.8%	3.9%
Médio	2500	1625	625	200	50	1	0.940	6.0%	
Grande	7000	4550	1750	560	140	1	0.971	2.9%	

Os ganhos mostrados na Tabela 2 refletem a escala de desenvolvimento para um projeto real, contemplando todas as atividades para os sistemas de controle e de supervisão e as atividades de teste e implantação. Neste contexto, mesmo comparado com ambientes que já oferecem os modernos recursos de desenvolvimento, a utilização de uma biblioteca de automação desde o início do projeto implica numa redução média de 3,9% do custo global do projeto. Esta mesma estimativa foi utilizada na análise de retorno e proporcionou uma TIR da ordem de 95% para o investimento necessário para a criação e documentação de uma biblioteca de automação com 150 conjuntos de objetos. Cada conjunto de objetos é composto por um objeto de automação, um de simulação, um símbolo para animação gráfica e uma janela de operação.

4 CONCLUSÕES

A utilização de bibliotecas de automação é uma realidade para o mercado, mas ainda não totalmente incorporada pelos usuários e pelas empresas. Os benefícios qualitativos e quantitativos apresentados neste trabalho, juntamente com o exemplo de viabilidade, devem ser considerados de forma efetiva para a adoção desta prática.

A evolução proporcionada pela norma IEC 61131 e esperada para a norma IEC 61499 exigirão uma adoção cada vez maior para o uso de bibliotecas de objetos/componentes. Sendo um caminho promissor para uma automação cada vez mais eficiente e produtiva. Os esforços de entidades como a OOONEIDA, em conjunto com outras organizações, prometem contribuir em muito para uma adoção rápida das novas soluções. Já existem produtos no mercado que fazem uso das soluções mais modernas baseadas em componentes de automação no nível de controle.

De forma semelhante, os sistemas computacionais para supervisão do processo e gestão de produção também já fazem uso intensivo de componentes de software, e mais recentemente da orientação a serviços, de forma que a adoção de bibliotecas é a mola mestra para a rápida evolução destes sistemas.

Os usuários de automação devem se informar sobre como colocar em prática a utilização de bibliotecas e buscar o máximo de benefícios proporcionados.

REFERÊNCIAS

- 1 FONSECA, M. O; SEIXAS FILHO, C; BOTTURA FILHO, J. A; “**Aplicando a Norma IEC 61131 na Automação de Processos**”, ISA, 568p. 2008.
- 2 VYATKIN, V; “**IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design**” ISA – The Instrumentation, Systems and Automation Society, 279p. 2007.
- 3 Web site da OOONEIDA, <http://www.ooneida.org>, visitado em Junho de 2008.