ISSN 2237-0234

NORMA IEC 61131-3 ORIENTADA A OBJETOS – OS REFLEXOS DA CONVERGÊNCIA TECNOLÓGICA PARA O DESENVOLVIMENTO E GESTÃO DOS SISTEMAS DE CONTROLE[1]

Marcos de Oliveira Fonseca[2]

Resumo
O desenvolvimento e gestão dos sistemas de controle industrial estão prestes a passar por um novo salto de evolução com a terceira edição da norma IEC 61131-3. A convergência tecnológica dos sistemas de TA e TI traz o paradigma da orientação a objetos na sua plenitude para a principal norma internacional para programação de controladores. A adoção dessa técnica de programação já consagrada no universo de TI abre novas possibilidades para o desenvolvimento de sistemas de controle e aponta para benefícios importantes para o negócio das empresas. Além disso, facilita a integração com novas tecnologias para controle e comunicação industrial. O presente trabalho destaca os principais aspectos da orientação a objetos previstos para a norma e analisa o seu impacto em aplicações práticas, assim como as necessidades de capacitação para as equipes de automação.
Palavras-chave: Norma IEC 61131-3; Orientação a objetos; Automação.

IEC 61131-3 OBJECT-ORIENTED – THE REFLEX OF TECHNOLOGY CONVERGENCE TO CONTROL SYSTEMS DEVELOPMENT AND MANAGEMENT

Abstract
The development and management of industrial control systems are about to undergo a new evolutionary leap with the third edition of IEC 61131-3. Technological convergence of IT and AT systems brings the paradigm of object orientated programming (OOP) in its fullness to the main international standard for controllers programming. The adoption of this programming technique which is well established in the universe of IT opens up new possibilities for the development of control systems and points to significant benefits for industrial business. It also facilitates integration with new technologies for industrial control and communication. This paper highlights key aspects of object oriented programing planned to the standard and analyzes its impacts on practical applications, as well as training needs for automation teams.
Keywords: IEC 61131-3; Object oriented; Automation.

[1] *Contribuição técnica ao 16° Seminário de Automação e TI Industrial, 18 a 21 de setembro de 2012, Belo Horizonte, MG.*

[2] *Engenheiro Eletricista, M.Sc, Senior Manager da Accenture Plant and Automation Solutions (APAS), Belo Horizonte – MG, Brasil.*

ISSN 2237-0234

1 INTRODUÇÃO

A norma IEC 61131 teve sua primeira publicação em 1992 e hoje é reconhecidamente a principal referência internacional para projeto, fabricação e programação de controladores industriais.(1) Suas diversas partes estão em constante evolução, Quadro 1, e alinhadas com as tendências e normas para controle totalmente distribuído, IEC 61499,(2) e comunicação industrial definida pelo padrão OPC UA,(3) que está em fase de publicação como norma IEC 62541-x. A terceira edição da parte 3 (Programação de Controladores) está prevista para publicação ainda em 2012 e outras duas novas partes (partes 6 e 9) estão em fase de finalização/votação.

Quadro 1 – Situação recente das partes da norma IEC 61131 e suas edições

Parte	Conteúdo	Publicação
1	Definição da terminologia e dos conceitos	2003 (2ª Ed.)
2	Requisitos de teste de verificação e fabricação eletrônica e mecânica	2007 (3ª Ed.)
3	Estrutura do software do CP, linguagens e execução de programas	2003 (2ª Ed.)
4	Orientações para seleção, instalação e manutenção de CPs	2004 (2ª Ed.)
5	Funcionalidades para comunicação com outros dispositivos	2000 (1ª Ed.)
6	Reservada para uso futuro	
7	Funcionalidades de software para tratamento de Lógica Nebulosa	2000 (1ª Ed.)
8	Orientações para implementação das linguagens IEC 61131-3	2003 (2ª Ed.)

A proposta inicial da IEC 61131-3 (parte 3) foi de promover a estruturação, modularização e reutilização da programação de controladores. Tinha também o grande propósito de tornar o programa aplicativo do controlador portável entre diferentes produtos e plataformas de hardware e software. Devido às pressões de mercado, a portabilidade total ainda é uma promessa, mas algumas bibliotecas e partes do programa já podem ser portados entre determinados produtos de mesma plataforma de software. Apesar da limitação de portabilidade, os conceitos introduzidos pela IEC 61131-3 são adotados pela grande maioria dos fabricantes e disponibilizados nos diversos produtos para controle industrial, desde controladores para aplicações simples para uso doméstico até os controladores industriais de grande porte.

Uma das principais contribuições da edição atual da IEC 61131-3 para programação de controladores consiste na adoção de alguns dos princípios da programação orientada a objetos, do termo em inglês *Object Oriented Programming* (OOP), principalmente no que se refere ao encapsulamento de código e instanciação de blocos funcionais. A partir da utilização destes princípios, a programação moderna de controladores permite uma maior modularização e reutilização do código, o que proporciona diversos benefícios para o desempenho, qualidade e redução expressiva do esforço e custo de desenvolvimento, testes, comissionamento e manutenção de sistemas de controle. Além disso, a gestão dos sistemas e treinamento dos diversos usuários é facilitada pela utilização de bibliotecas de módulos de software, assim como a utilização de linguagens mais adequadas aos diferentes problemas de controle. Entretanto, a edição atual da IEC 61131-3 ainda não permite a aplicação de todo o potencial da orientação a objetos na programação de controladores.

Um aspecto muito relevante para a gestão de sistemas de controle e do know-how das empresas está na preservação e reutilização do conhecimento através do uso de bibliotecas de software e maior documentação dos sistemas de controle. Com o crescente envelhecimento dos profissionais de engenharia e de controle de processos, muitas empresas enfrentam grandes dificuldades em preservar seu know-how para controle do processo produtivo e desempenho operacional com a aposentadoria e rotatividade de seus profissionais mais experientes. Com a crescente evolução e utilização de sistemas, equipamentos e tecnologias de automação para controle e gestão operacional, algumas indústrias, organizações e governos estão investindo em estudos e planos para maior capacitação e preservação do conhecimento.(4) Outro exemplo é o programa de certificação de profissionais de automação da ISA (www.isa.org).

A Programação Orientada a Objetos é um paradigma de análise, projeto e programação de sistemas de software baseado na composição e interação entre diversos elementos de software chamados de objetos. O paradigma "orientação a objeto" tem bases conceituais e origem no campo de estudo da cognição, que influenciou a área de inteligência artificial e da linguística, no campo da abstração de conceitos do mundo real. Na qualidade de método de modelagem, é tida como um das principais estratégias para se eliminar o "gap semântico", dificuldade recorrente no processo de modelar o mundo real do domínio do problema em um conjunto de componentes de software que seja o mais fiel na sua representação deste domínio.

A OOP é largamente usada para o desenvolvimento de software de TI em geral e suportada pelas modernas linguagens, tais como C++, Java, etc. É também cada vez mais ensinada e difundida nos meios acadêmicos como alternativa preferida às técnicas e linguagens procedurais. Apesar de agregar maior complexidade para os ambientes de compilação, hoje em dia não representa impacto relevante para os sistemas computacionais cada vez mais poderosos.

As novas normas e padrões para sistemas de controle e comunicação industrial são concebidos e modelados utilizando as técnicas de orientação a objetos, seja para os modelos de dados e informação quanto para os modelos funcionais e sua implementação. Um bom exemplo disso é o modelo de informação do padrão OPC UA, totalmente orientado a objetos. A norma IEC 61499 também faz uso da orientação a objetos para aplicações totalmente distribuídas no seu modelo de sistemas holônicos. Neste cenário, a utilização plena dessa técnica para programação de controladores é o caminho natural trilhado pela IEC para a nova edição da parte 3 da IEC 61131. A terceira edição traz novas melhorias, mas sem dúvidas a mais relevante é a incorporação de forma plena dos principais conceitos da orientação a objetos, de forma muito semelhante ao adotado pela linguagem Java. Dessa forma, permite aos diversos usuários dos sistemas de controle em se beneficiar de um poderoso instrumento para desenvolvimento e gestão dos programas aplicativos, assim como facilitar a integração dos controladores programáveis utilizando os modernos padrões de comunicação industrial.

A adoção da orientação a objetos promete trazer para o mundo dos controladores programáveis muito da flexibilidade, escalabilidade e redução dos custos para desenvolvimento e gestão já consagrados nos sistemas de TI em geral, sem perder a robustez, qualidade e desempenho requeridos para as aplicações industriais. Obviamente, existe um processo de transição entre a capacitação atual e esperada para os profissionais de automação. Já existem produtos de mercado utilizando a orientação a objetos para programação de controladores com sucesso, sendo, portanto, uma prática comprovada para o ambiente industrial.(5)

2 PRINCIPAIS CONCEITOS DA PROGRAMAÇÃO ORIENTADA A OBJETOS PREVISTOS PARA A 3A EDIÇÃO DA IEC 61131-3

Na programação orientada a objetos, objeto é um elemento de software que possui seu próprio estado e comportamento, que é alteração (reação) de seus estados à chamada de seus métodos. Classe é um protótipo ou matriz para a criação de objetos. Todos os objetos são criados a partir de uma classe, sendo essa criação definida pelo processo chamado de instanciação. Na automação, podemos denominar como classe os tipos de blocos funcionais utilizados na programação de controladores. Por exemplo, um bloco funcional PID define uma classe, sendo que a instância TIC_100 é o objeto criado a partir da classe PID. Quando se faz uma especialização (ou derivação) criamos uma subclasse (filha), que mantem as características da classe mãe e pode implementar suas próprias especificidades. Na automação, um bloco funcional Ctrl_Motor é uma classe da qual podemos derivar a subclasse Ctrl_Motor_Reversivel que implementa as especificidades para reversão de rotação de um motor. Nesse caso, o objeto Ctrl_Motor_Reversivel_MR1 poderia ser um objeto criado a partir da subclasse.

Os princípios da programação orientada a objetos são: abstração, encapsulamento, herança e polimorfismo. Abstração de dados significa o processo de eliminar todos os detalhes sem importância de um objeto e ficar apenas com as características de interesse que o definem. Encapsulamento reside em esconder os detalhes da implementação de uma classe de outras partes do programa e exibir apenas a sua interface (atributos e métodos). Herança consiste em derivar novos elementos (na verdade, subclasses) a partir dos existentes (classes). Estes novos elementos ou subclasses herdam todas as características das classes ancestrais, mas podem apresentar novas propriedades e comportamentos não existentes nas ancestrais. Polimorfismo é a habilidade de diferentes objetos de responderem, cada um da sua maneira, à mesma mensagem ou chamada de método. Por exemplo, ambos os objetos moinho_de_bolas e correia_transportadora entendem a mensagem “liga”, apesar de serem objetos de classes diferentes e responderem cada um de uma forma específica. Por exemplo, o moinho ao ligar realiza uma sequencia de ligamento de todos seus subsistemas auxiliares até estar em condições de ligar o motor principal. Já a correia testa as condições de proteção e partida e depois liga o motor da correia.

Está prevista para a terceira edição da IEC 61131-3 a incorporação desses conceitos como uma extensão à forma atual (clássica) da norma,[6] de modo que é permitido ao usuário escolher pela programação da forma atual ou totalmente orientada a objetos, ou ainda uma mistura das duas. A extensão poderá ser implementada utilizando-se qualquer uma das 5 linguagens da norma, sendo multi-linguagem. Os principiais conceitos incorporados são apresentados a seguir.

2.1 Métodos

O bloco de função da IEC clássica contém uma única rotina para tratamento de todos seus estados internos. Embora sobre esses estados possam ser realizadas diversas operações como inicialização, tratamento de erros, sincronização, etc; a única forma de controlar essa rotina é através da manipulação (escrita) dos valores de suas entradas no momento da chamada da instância do bloco.

A orientação a objetos permitirá que sejam separados todos os códigos (rotinas ou funções) específicos para cada tratamento a ser realizados pelo bloco funcional.

Cada código específico é denominado como método, o qual quando chamado irá executar um tratamento sobre os estados do seu objeto e exibindo o comportamento do objeto para a chamada do método. A chamada ao método significa que será feito um acesso via ponteiro com passagem de parâmetros (entrada) e retorno dos resultados (saída) da execução de uma função, como é feito em linguagem VB, por exemplo. Essa abordagem favorece em muito a estruturação através da separação do código para cada método, assim como a facilidade de entendimento através da semântica especifica. Ou seja, quando se chama o método "Liga" de um objeto "Bomba_B1" entende-se que o comportamento esperado será o de ligar a bomba em referência. O elemento de programação da IEC 61131-3 clássica que mais se assemelha ao método é a Ação (*Action*).

2.2 Herança

A herança é caracterizada quando uma subclasse herda todos estados e métodos de uma classe superior, podendo acrescentar novos estados e métodos ou mesmo alterar os métodos herdados.

Na IEC 61131-3 clássica, quando se deseja criar uma derivação ou um novo bloco funcional (subclasse) a partir de um bloco funcional existente (classe), isso é feito através da instanciação de um objeto da classe como parte da nova subclasse. Essa forma tem restrições, pois não permite separar quais partes do código da classe se deseja utilizar na subclasse, já que é mantida uma cópia completa da classe.

2.3 Interface

A interação entre um objeto e os demais elementos de software é feito através dos seus métodos. Os métodos formam a interface do objeto com os elementos externos. De forma geral, uma interface é um grupo de métodos relacionados, sem uma implementação (estados internos ou rotina associada).

O conceito de interface é praticamente novo para a programação IEC 61131-3. Na forma clássica, um bloco funcional teria apenas uma única interface genérica. Já na orientação a objetos, a interface funciona como um bloco funcional especial que seria uma superclasse para servir de matriz para outras classes e subclasses. É, portanto, um protótipo para a derivação de blocos funcionais e interfaces. Todo bloco funcional derivado de uma interface tem que implementar todos os métodos (rotinas) herdados da interface. Um bloco funcional pode ser derivado de um único bloco funcional (subclasse), mas de um número arbitrário de interfaces.

Uma característica importante é que em um bloco funcional pode-se fazer uma declaração semântica para suas variáveis, referenciando os tipos de interfaces herdadas pelo bloco, de modo que as referências apontam para instâncias de blocos funcionais que implementam a interface referenciada. Ou seja, para uma variável declarada dessa forma, podemos atribuir à mesma diferentes apontamentos (ponteiros) para instâncias que podem apresentar diferentes implementações para a mesma interface, o que confere a capacidade de polimorfismo. Isso significa que a mesma interface pode exibir comportamento diferente em função dos diferentes blocos referenciados pela variável associada. Na prática, o controlador faz as associações às diferentes interfaces em tempo de execução do programa.

A Figura 1 exemplifica a utilização prática para alguns dos conceitos de orientação a objetos previstos para a nova edição da norma.

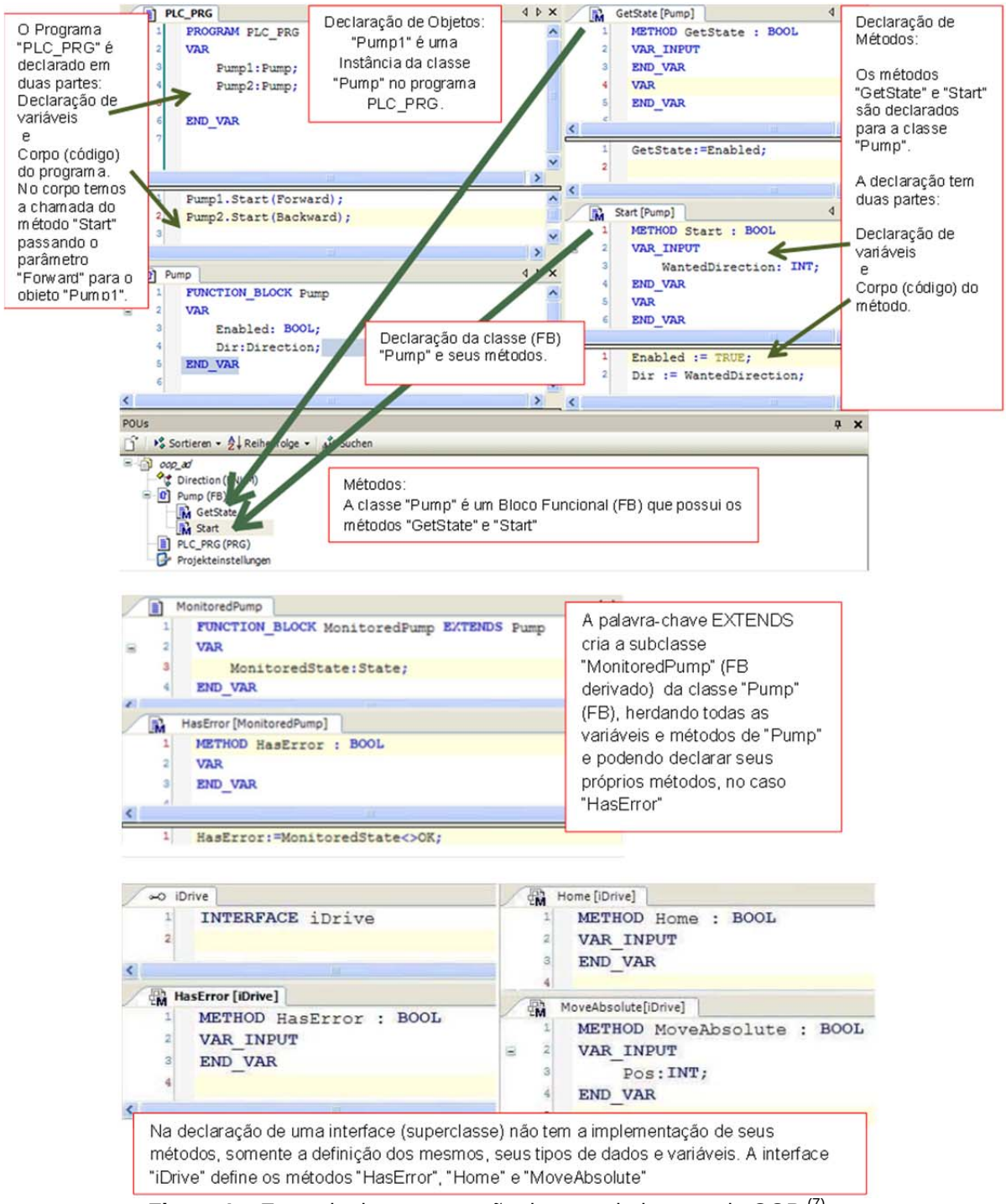

Figura 1 – Exemplo de programação de controlador usando OOP.[7]

O quadro a seguir resume as principais características da programação orientada a objetos (OOP) previstas para a nova edição da norma e sua comparação com a edição atual e com outras linguagens de programação.

ISSN 2237-0234

Quadro 2 – Comparativo básico das principais características de OOP entre linguagens.[7]

Característica	IEC 61131-3 Atual	IEC 61131-3 3ª Ed.	C++	Java	C#
Multi-linguagem	+	+	-	-	-
Mistura de programação OO e Procedural	-	+	+	-	-
Classes	~ (FB)	+	+	+	+
Métodos	~ (Ação)	+	+	+	+
Interfaces	-	+	-	+	+
Polimorfismo	-	+	+/-	+	+
Referência Semântica	-	+ (Interfaces)	-	+	+
Construção / Destruição de objetos	-	+	+	+	+
Propriedades	-	+	-	-	+
Memória dinâmica (“new”)	-	-	+	+	+
Controle de Acesso	~ (Variáveis)	~ (Variáveis)	+	+	+
Legenda: **-** (tem pouco ou nenhum suporte) | **+** (tem suporte) | **+/-** (tem suporte parcial) | **~** (tem suporte equivalente)					

3 ASPECTOS PRÁTICOS

Considerando a prática atual para desenvolvimento e gestão dos sistemas de controle baseado na IEC clássica, os principais aspectos relativos aos impactos esperados pela adoção da programação orientada a objetos são apresentados a seguir.

- **Compatibilidade retroativa**: Com a abordagem adotada pela IEC de incluir a orientação a objetos como uma extensão às características atuais da norma, considera-se que haverá compatibilidade plena com as implementações atuais, de forma que ambas as implementações irão coexistir por um período de transição. Os usuários poderão se planejar para uma transição suave nesse processo.
- **Integração através de chamadas a métodos**: Para usufruir da maior flexibilidade e escalabilidade da OOP, a integração entre controladores e outros sistemas, principalmente sistemas de supervisão (SCADA, SDCD ou Híbrido), será mais eficiente através da chamada a métodos. Esse mecanismo potencializará a utilização das interfaces e também aumentará a robustez do sistema, uma vez que a técnica atual normalmente expõe todas as variáveis internas e parâmetros dos blocos funcionais. Utilizando chamadas a métodos, as variáveis internas serão de gerenciamento exclusivo dos blocos funcionais, sendo que somente solicitações aceitas pelos métodos serão tratadas. Como o novo padrão de comunicação OPC UA já suporta a chamada a métodos, não se espera dificuldades na adoção desta forma de integração.
- **Racionalização do uso de memória e comunicação**. A utilização da herança de forma plena da OOP e do polimorfismo evitará que sejam criadas cópias de instâncias da classe mãe na derivação de blocos funcionais como é feito atualmente. Isso reduzirá a duplicação desnecessária de variáveis internas dos blocos instanciados, racionalizando o uso de memória dos controladores e eliminando o tráfego de dados na comunicação entre sistemas.
- **Maior estruturação e componentização**: A utilização de bibliotecas de software ainda é muito tímida entre os usuários de sistemas de controle. Boa parte disso

devido ao desconhecimento das modernas técnicas disponíveis, mas também devido às limitações atuais para modelamento e componentização do software de controladores programáveis. A adoção plena da OOP potencializará uma evolução no desenvolvimento e gestão de software para controladores programáveis, na mesma linha de crescimento observada para os softwares de uso geral, onde a utilização de componentes contribuiu em muito para a escalabilidade de soluções e consequente redução de custos. Pode-se até esperar que essa evolução siga na direção de uma arquitetura orientada a serviços e até mesmo de *cloud computing*, mas isso depende muito da quebra de paradigmas e de uma forte transformação cultural do mercado de automação. Independentemente disso, as empresas estão precisando urgentemente investir na preservação e reutilização do conhecimento e know-how, e a utilização de bibliotecas é um caminho natural a ser seguido desde já.

- **Investimento na capacitação de equipes**: Observa-se que o mercado de automação ainda não absorveu todas as técnicas e tecnologias disponíveis nos sistemas de controle compatíveis com a IEC 61131-3 clássica. Isto se deve principalmente pela dificuldade de capacitação de equipes associada ao processo de modernização dos sistemas existentes, mas também devido a uma falta de política e incentivo das empresas na evolução dos seus processos de desenvolvimento e gestão de sistemas de controle. Ainda são implantados sistemas de última geração utilizando conceitos e técnicas de programação de controladores anteriores à IEC 61131-3. Se as empresas quiserem usufruir de todos os ganhos e benefícios proporcionados pelas técnicas e tecnologias atuais, faz-se necessário rever sua política e práticas de automação, assim como investir em programas para capacitação de suas equipes e de gestão da mudança, objetivando uma transformação da cultura de automação da organização.
- **Desempenho dos sistemas de controle**: O desempenho é sempre um item impactante quando se incorporam maiores sofisticações e complexidades tecnológicas nos sistemas informatizados. Entretanto, a utilização da OOP já está bastante madura nos sistemas de TI em geral, de forma que a automação irá receber um pacote de soluções já bastante evoluídas. Além disso, o crescente aumento de capacidade computacional dos sistemas de controle associado com arquiteturas distribuídas e redes de alta velocidade permite esperar que não sejam causados impactos relevantes. Alguns produtos de mercado relatam desempenho superior de controladores programados utilizando OOP em relação à programação IEC clássica para aplicações de maior porte.[8]
- **Harmonização entre padrões e tecnologias**: A utilização das técnicas de orientação a objetos é utilizada em diversos sistemas e soluções para controle e comunicação industrial, dos quais podemos citar o controle distribuído definido pela IEC 61499 e também o padrão OPC UA para comunicação industrial. Dessa forma, evoluir para a adoção da OOP na programação de controladores é o caminho natural para facilitar a colaboração e integração entre os sistemas industriais, habilitando também a utilização de ferramentas e soluções já consagradas para os sistemas de software em geral. Um bom exemplo disso é a utilização da linguagem UML (*Unified Modeling Language*) para modelagem do sistema, associada a geradores de código e metodologias para testes e validação.

4 CONCLUSÕES

Tendo em vista todos os aspectos práticos e tendências do mercado para os sistemas de TA e TI, assim como a convergência tecnológica, pode-se concluir que a programação de controladores seguirá os passos da práticas líderes e tecnologias de sistemas de software em geral, adotando a programação orientada a objetos como novo paradigma para os sistemas de controle. Novamente, a norma IEC 61131-3 em sua terceira edição será uma forte referência para o mercado de automação se alinhar no desenvolvimento de novos e produtos e soluções buscando uma nova evolução das práticas líderes para desenvolvimento e gestão de sistemas de controle.

Os principais benefícios esperados para a adoção da OOP na programação de controladores são:

- maior estruturação e robustez da programação;
- maior reutilização de código e de conhecimento;
- facilidade de manutenção e gestão dos sistemas de controle;
- maior flexibilidade e escalabilidade da aplicação;
- redução de custos pela utilização de ferramentas e práticas líderes de ti para o desenvolvimento de sistemas de controle.

As empresas, profissionais e academia voltados para o mercado de automação devem se preparar para o desenvolvimento da capacitação requerida para utilização da programação orientada a objetos no desenvolvimento de sistemas de controle. A adaptação das metodologias atuais buscando o suporte das práticas líderes, já consagradas para os sistemas de software em geral, serão fatores importantes para a mudança do paradigma atual e para o processo de transformação cultural do mercado.

Referências

1 Fonseca, M. O; Seixas Filho, C; Bottura Filho, J. A; “**Aplicando a Norma IEC 61131 na Automação de Processos**”, ISA América do Sul, 2008, 568p.

2 Vyatkin, V. “**IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design**”, OOONEIDA - ISA, 2007.

3 Mahnke, W; Leitner, S; Damm, M; “**OPC Unified Architecture**” Springer, 2009, 339p.

4 McAree R, Lever P. “**Automation for success**”, Mining Industry Skills Centre Inc; 2010.

5 Schünemann, U. “**Programming PLCs with an Object-Oriented Approach**”, Automation Technology in Practice - International, vol. 2, p.59 a 63, 2007.

6 Werner, B. “**Object-Oriented Extensions for IEC 61131-3**”, IEEE Industrial Electronics Magazine, p. 36 a 39, 2009.

7 Van der Wal, E. “**The 3rdEdition of IEC 61131-3**”, PLCopen, disponível em www.iestcfa.org, consultado em Maio/2012.

8 “Object-Oriented Programming in CoDeSys”, 3-S Software, disponível em http://www.3s-software.com/index.shtml?en_v3_oop, consultado em Maio/2012.