Área: CV ( ) CHSA ( ) ECET (X)

MODELAGEM E PROGRAMAÇÃO DE APLICAÇÕES INDUSTRIAIS USANDO LINGUAGEM DE ALTO NÍVEL

Paulo Henrique Barreto de Sousa (Bolsista ICV),Raimundo Santos Moura (Orientador, Departamento Informática e Estatística/UFPI)

Introdução

A automação industrial visa melhorar a produtividade e qualidade nos processos considerados repetitivos, estando presente no dia-a-dia das empresas para apoiar conceitos de produção tais como os Sistemas Flexíveis de Manufatura. Sob o ponto de vista produtivo, automação industrial pode ser dividida em três: a rígida, a flexível e a programável, aplicações a grande, médios e pequenos lotes de fabricação, respectivamente (Rosário, 2005).

Ainda segundo Rosário (2005), a automação industrial pode ser entendida como uma tecnologia integrada a três áreas: a eletrônica responsável pelo hardware, a mecânica na forma de dispositivos mecânico (atuadores) e a informática responsável pelo software que irá controlar todo o sistema. Desse modo, para efetivar projetos nesta área exige-se uma grande gama de conhecimento, impondo uma formação ampla e diversificada dos projetistas, ou então um trabalho de equipe bem coordenado com perfis interdisciplinares. Os grandes projetos neste campo envolvem infinidade de profissionais e os custos são suportados por grades empresas.

A forma básica de programação de Controladores Lógicos Programáveis - CLP era realizada a partir de uma linguagem oriunda do diagrama de relés (do inglês : *Ladder Diagram – LD*). De inicio, resultou numa padronização dos CLPs no ambiente industrial, mas posteriormente provocou a dependência por técnicos especializados em *Ladder*, Desta forma, linguagens, métodos e padrões de mais alto nível têm sido propostos nos meios acadêmicos e profissionais através de organizações como a *International Eletrotechnical Commission (IEC)* e PLCopen (www.plcopen.org), na tentativa de maximizar a eficiência para modelagem, construção e manutenção de sistemas de automação.

Com o objetivo de alcançar esses requisitos diversos projetos têm sido desenvolvidos pela comunidade científica e pela indústria da automação. Dois desses projetos são: Beremiz (http://www.beremiz.org) e o Editor SFC/ST.

Beremiz é uma ferramenta para o desenvolvimento de linguagem de alto nível para programação de CLP e o Beremiz usa um modelo baseado no XML TC6-XML. E o arquivo com extensão xsd é usado parar criar uma espécie de meta modelo, que definem as relações entre os objetos dentro do modelo PLCopen. É usado também um o padrão IEC 61131-3 que define elementos básicos de programação de CLP, regras sintáticas, e semânticas. para linguagens gráficas como *Diagramas Ladder, Diagramas de Blocos de Funções e Diagramas de Funções Sequenciais* e linguagens textuais como *Lista de Instruções e Texto Estruturado.*

A ferramenta Editor SFC/ST (Couto, 2009; Gameleira et. al.,2010) converte uma linguagem de alto nível para CLP's que permite usar uma combinação das linguagens SFC (*Sequential Function Chart*) e ST (*Structured Text*) compatível com o formato TC6-XML 2.0 especificado pela PLCopen (http://www.plcopen.org) . A Figura 1 apresenta a estrutura básica do tradutor. Note que a partir do formato TC6-XML é possível gerar linguagens de máquina para equipamentos de diferentes fabricantes.

Área: CV ( ) CHSA ( ) ECET (X)

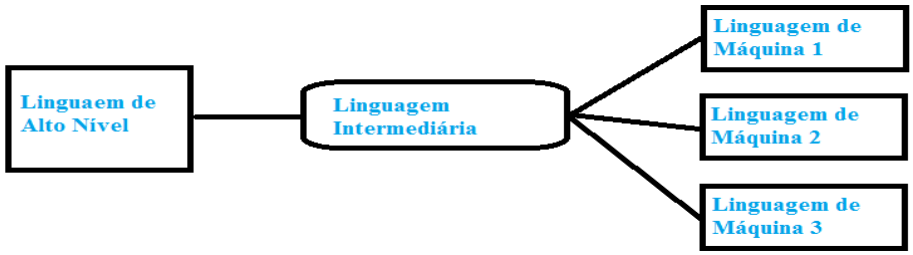

Figura 1 Tradutor de linguagem alto-nível para código executável

Apesar desses projetos seguirem o padrão definido pela PLCopen, eles possuem incompatibilidades no formato dos arquivos gerados, ou seja, o arquivo produzido em uma ferramenta não abre corretamente no outra e vice-versa. O objetivo principal deste trabalho foi discutir as incompatibilidades da codificação entre estas ferramentas para programação de CLP: Editor SFC/ST e Beremiz e propor alterações no processo de geração de código de saída da ferramenta Editor SFC/ST para resolver tais incompatibilidades.

Metodologia

Com o objetivo de identificar as incompatibilidades nos códigos gerados pelas duas ferramentas em uso neste projeto, fizeram-se vários exemplos de sistemas utilizando seus ambientes de codificação. Estes exemplos abrangem uma grande quantidade de métodos e funções que as ferramentas utilizam. A intenção era comparar os códigos gerados pelas ferramentas e identificar todas as diferenças que impedem que uma ferramenta “entenda” o código gerado pela outra.

Alguns exemplos foram elaborados para testar especificidades da linguagem SFC e outros foram adquiridos na pagina web http://www.dca.ufrn.br/~maitelli/FTP/clp/Experi%EAncias_20072.pdf.

Em resumo simulam um braço mecânico que avança e recua com chaves de acionamento respectivo e com chave de recuo e avanço automático regulado por sensores; Um distribuidor de esferas que tem sensores para verificar material presente ou faltando; Sensores de um acionamento de uma bomba d’água regulando nível alto e baixo de forma automática ou manual. Fora exemplos que foram desenvolvidos especificamente para verificar divergências e convergências na linguagem SFC.

Resultados e Discussão

Foi feita a analise dos códigos dos exemplos relacionados a partir da leitura dos códigos, de padrão TC6-XML, gerados pelas duas ferramentas, identificou-se que o maior problema é a versão do XML utilizada pela ferramenta Beremiz. Ela utiliza uma versão anterior a do Editor SFC/ST. Com base na análises feitas nos exemplos, citados, foi encontrado algumas diferenças que são motivo de incompatibilidade.

- **Diferenças listadas e explicadas abaixo:**
  - **CABEÇALHO:** O cabeçalho, onde se encontra todas as informações sobre a ferramenta e outras definições do programa gerada na ferramenta, teve que ser alterado por completo, pois cada ferramenta tem seu método para definir estas informações;
  - **DOCUMENTAÇÃO:** O campo documentação gerado pelo Editor SFC/ST necessita expressa seus dados em um campo CDATA, é um campo que o

interpretador do XML não faz sua analise;

- **INLINE:** O campo inline necessita do mesmo campo CDATA;
- **BLOCO DE AÇÃO:** No Editor SFC/ST optou-se não cria-los graficamente, chamados de "*actionblock"*, vez que as ações especificadas no domínio da automação de poços são bastante extensivas. As solução encontrada foi abrir uma janela para manipular tais ações.
- **VERSÃO DO XML:** A ferramenta Beremiz utiliza a versão 1.0, enquanto a ferramenta Editor SFC/ST utiliza a versão 2.0.
- **ERRO NA FERRAMENTA:** Ferramenta Beremiz é instável; no momento da programação gera erros e encerra o programa sozinho.
- **ERRO NA LINGUAGEM:** A ferramenta Beremiz na utilização da linguagem SFC aceita ligações de um passo para outro passo diretamente. Sendo que a especificação da linguagem não permite.
- **ERRO AO SALVAR ARQUIVOS EXISTENTES:** A ferramenta Beremiz ao reabrir programas salvos anteriormente corrompe o arquivo ao tentar edita-lo e salvar novamente.

Conclusão

Concluiu-se que as incompatibilidades apresentadas entre as ferramentas são relacionadas as versões utilizadas no padrão TC6-XML e na interface gráfica criada por cada desenvolvedor especifico das ferramentas utilizadas nesse trabalho. Bem como na especificidade de cada ferramenta, sendo que a Editor SFC/ST foi criada para a programação de alto nível de CLPs dedicados a poços de petróleo, enquanto que a Beremiz foi criada para ser uma ferramenta genérica que atendesse a maioria dos problemas da programação de alto nível para CLPs.

Referências

BEREMIZ – Open Source Software for Automation. Disponível em http://www.beremiz.org/. Último acesso: novembro, 2011.

FONSECA, M. O.; SEIXAS FILHO, C.; BOTTURA FILHO, J. A., Aplicando a Norma IEC 61131 na automação de Processos; Editado por ISA – Sociedade de Instrumentação, Sistema e Automação/Distrito 4 – América do Sul.

GAMELEIRA, T. A.; COUTO, F. C. A.; MOURA, R. S. ; GUEDES, L. A., **Ferramenta Para Programação de CLP's em Alto Nível**. In: Congresso Brasileiro de Automática, 2010, Bonito-MS. Anais do Congresso Brasileiro de Automática, 2010. p. 1-6.

PLCopen for Efficiency in Automation. Disponível em http://www.plcopen.org/. Último Acesso: novembro, 2010.

ROSÁRIO, J. M., *Princípios de Mecatrônica* – Editora Pearson Pretice Hall, São Paulo, 2005.

COUTO, F. C. A., Ambiente Computacional para Desenvolvimento de Controladores Lógicos Programáveis baseado em Linguagem SFC e ST. Dissertação de Mestrado, PPGEEC - UFRN, 2009.

BATISTA, D. A. S. P., Automação da Linha de Fabrico Flexível do DEEC, 2012.

FERNANDES, V. A. M. N., 'Driver' Modbus para Ambiente de Desenvolvimento IEC 61131-3, 2009.

MAITELLI, Andre Laurindo, http://www.dca.ufrn.br/~maitelli/FTP/clp/Experi%EAncias_20072.pdf, fevereiro de 2012.

Palavras-chave: Controlador Lógico Programável, Automação Industrial. Incompatibilidades.