GERAÇÃO AUTOMÁTICA DE CÓDIGO-FONTE A PARTIR DE MODELOS SIMULINK NO CONTEXTO DOS SISTEMAS EMBARCADOS

Fernando Leite Todão (PIBIC/CNPq/UEM), Itana Maria de Souza Gimenes (Orientador), e-mail: itanagimenes@gmail.com.

Universidade Estadual de Maringá/Departamento de Informática/Maringá, PR.

Área do conhecimento: 10000003 Ciências Exatas e da Terra, 10300007 Ciência da Computação.

Palavras-chave: geração de código-fonte, sistemas embarcados, MDE.

Resumo:

No desenvolvimento de sistemas embarcados vários são os requisitos não-funcionais que devem ser considerados, tais como desempenho, consumo de potência e prazo. A engenharia dirigida por modelos propõe o desenvolvimento de software por meio das transformações de modelos em direção ao código-fonte do sistema. Este projeto avaliou a viabilidade da geração de código-fonte no contexto dos sistemas embarcados utilizando a ferramenta Simulink.

Introdução

Diversos exemplos de sistemas embarcados podem ser encontrados no cotidiano, desde celulares e dispositivos portáteis (ex. MP3, *iPod*) até eletrodomésticos e decodificadores de vídeo de alta definição. Esses sistemas são definidos como sistemas computacionais incorporados em um produto maior e normalmente não são diretamente visíveis pelos usuários (MARWEDEL, 2006). Vários requisitos específicos devem ser considerados durante as etapas de desenvolvimento de sistemas embarcados. Por exemplo, o consumo de potência é importante para os sistemas embarcados alimentados por bateria, além de desempenho, custo e tempo de projeto.
A engenharia dirigida por modelos (*Model Driven Engineering – MDE*) (PRETSCHNER, 2005, FAVRE, 2004) é uma abordagem de desenvolvimento de software na qual os modelos não são apenas usados para auxiliar a compreensão do sistema, mas são considerados os principais artefatos do processo de desenvolvimento. O objetivo principal da MDE é sistematizar o processo de desenvolvimento por meio da transformação dos

modelos (FAVRE, 2004) até a geração de um código-fonte padronizado de forma automatizada.

A transformação de modelos é classificada de acordo com o tipo de modelo de entrada e com o tipo de modelo de saída. Existem basicamente três tipos de transformação: CIM (*Computer Independent Model*) para PIM (*Platform Independent Model*), PIM para PSM (*Platform Specific Model*) e PSM para código-fonte (geração de código) (ALMEIDA, 2008). CIM são modelos de alto nível de abstração, e a cada transformação esse nível vai-se diminuindo o nível até chegar ao código da aplicação.

Materiais e métodos

Para o desenvolvimento do projeto, foi utilizada a ferramenta MATLAB (*http://www.mathworks.com/products/matlab/*) em conjunto com a ferramenta Simulink (*http://www.mathworks.com/products/simulink/*). MATLAB é um ambiente interativo para computação numérica, visualização e programação. A ferramenta Simulink integrada ao MATLAB, disponibiliza um ambiente com diagrama de blocos que foi usado para a simulação dos domínios de projeto baseados em modelos. A especificação dos parâmetros de geração automática e a geração do código-fonte em C foram feitas a partir da ferramenta Simulink Coder™ (*http://www.mathworks.com/products/simulink-coder*). Para a avaliação dos resultados, os modelos Simulink foram simulados e salvos no ambiente MATLAB para ser analisado posteriormente com os resultados obtidos pela execução dos testes para o código gerado. A geração de código para um caso real utilizou o *blockset* ArduPilot 2.0 para Simulink (*http://www.mathworks.com/matlabcentral/fileexchange/39037-apm2-simulink-blockset*) que é um *blockset* utilizado para a simulação de um VANT (veículo aéreo não tripulado) e geração de código-fonte para o hardware ArduPilot Mega 2.0. Para poder construir as funções C++ MEX da biblioteca para o ArduPilot 2.0 foram utilizados o Microsoft Visual C++ 2010 Express Edition (*https://www.visualstudio.com/pt-br/products/visual-studio-express-vs*) e Microsoft Windows SDK (*https://msdn.microsoft.com/pt-br/windows/desktop/bg162891.aspx*).

Resultados e Discussão

Foram criados três Diagramas de Blocos Simulink para serem utilizados para geração de código-fonte e desse modo foram feitas simulações no ambiente Simulink salvando os dados de saída referentes a cada diagrama. Com os dados de saída das simulações salvos, o código-fonte foi gerado e no ambiente MATLAB foi executado gerando um arquivo MAT-file com os dados de saída referentes a execução do código. Os dois

arquivos contendo os dados de saída foram comparados para verificar a corretude do código gerado, assim foi verificado que o processo de geração de código mantém a corretude esperada na simulação do modelo, portanto há evidências de que processo de geração é viável. A Figura 1 ilustra a comparação dos resultados:

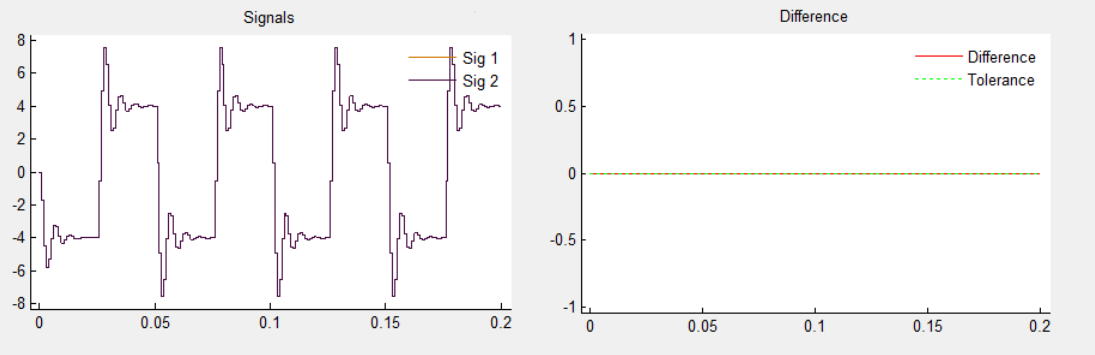

Figura 1 - Comparação dos sinais analógicos dos resultados da simulação do código gerado e do Diagrama de Blocos.

No primeiro gráfico da Figura 2 o sinal de saída do código gerado não aparece, pois é o mesmo resultado da simulação do Diagrama de blocos, o que pode ser confirmado pelo segundo gráfico que mostra a diferença entre os sinais.
A ferramenta Simulink possui uma biblioteca com vários blocos padrões, mas para casos muito específicos, como por exemplo, a geração de código para o ArduPilot Mega 2.0 é necessário que blocos personalizados sejam usados, com isso um pequeno guia de como especificar um bloco com outras funções foi feito.

Conclusões

A partir da simulação e testes realizados com três Diagramas de Blocos Simulink foi possível verificar que este processo é vantajoso na implementação de sistemas embarcados, pois ele reduz a necessidade de um conhecimento específico em programação bem como aumentam as possibilidades de geração de códigos corretores e eficazes. O código-fonte gerado é documentado o que torna a leitura do código mais acessível. Além disso, um documento é fornecido após o processo de geração que oferece informações importantes em relação à estrutura do código. Para casos em que os blocos disponibilizados pelo Simulink não oferecem suporte, é necessário a especificação de novos blocos e consequentemente a implementação de suas funções, resultando em um acréscimo de trabalho no desenvolvimento do sistema, porém ainda sendo viável caso outros projetos semelhantes sejam desenvolvidos posteriormente. No caso do

blockset utilizado é importante ressaltar que ele possui vários *wrappers* e links para códigos prontos, previamente implementado. Isso ocorre porque é uma aplicação bem específica e a geração não consegue atingir o total da aplicação, portanto e necessário o uso desses links. No entanto a utilização é válida, pois facilita tanto a simulação como reutilização para novos projetos.

Agradecimentos

Agradecemos ao Programa de Iniciação Científica CNPq-FA-UEM por incentivar e financiar a realização deste trabalho.

Referências

ALMEIDA P., "**MDA – Model Driven Architecture: Improving Software Development Productivity in Large-Scale Enterprise Applications**". University of Fribourg, Switzerland, 2008, p.108. Disponível em: http://diuf.unifr.ch/main/softeng/teaching/studentprojects/dealmeida.

FAVRE, Jean-marie. **Towards a basic theory to model: Model Driven Engineering**. 3rd Workshop in Software Model Engineering, WiSME. França, 10 nov. 2004.

FRAGAL, V. H. **Engenharia de aplicação para sistemas embarcados: Transformando especificações SysML em Simulink**. 2013. 104f. Dissertação de Mestrado - Universidade Estadual de Maringá. Maringá,

P. MARVEDEL. **Embedded System Design**. Netherland: Springer, 2006. ISBN 978-0-387-29237-3.

PRETSCHNER, A. - **Software Engineering**, 2005. ICSE 2005. Proceedings. 27th International Conference.