

Adaptação de elementos reais na interface do DCB para simulação de modelos heterogêneos

Tales Marchesan Chaves¹, Bráulio A. de Mello², Luciano L. Caimi¹

¹ URI, Universidade Regional Integrada do Alto Uruguai e das Missões, Santo Ângelo, RS, DECC

² UFLA, Universidade Federal de Lavras, Lavras, MG, DCC
{tales.chaves, lcaimi}@gmail.com
bmello@dcc.ufla.br

Abstract. Heterogeneous simulation allows for the combination of real and virtual elements for simulation matters. The DCB (Distributed Cosimulation Backbone) is a cosimulation software that supports simulation of heterogeneous models. However, the adaptation of real elements into the DCB is hard and time consuming. This paper presents a framework that facilitates the adaptation process, along with the specification of a protocol used for communication of elements during simulation. Integration of real and virtual elements is advantageous when some parts of a system are hard to simulate. Some fields, such as Precision Agriculture can be benefitted from the use of heterogeneous models.

Keywords: Simulação heterogênea, adaptação, elementos reais, DCB, Agricultura de Precisão.

1 Introdução

A simulação computacional heterogênea consiste na execução de um modelo composto por diferentes elementos em computadores com o propósito de imitar um processo ou operação do mundo real. A simulação pode ser classificada como heterogênea (SH) quando combina partes distintas, como por exemplo, elementos de hardware, como sensores e atuadores, ou ainda elementos descritos em uma linguagem de descrição de hardware, e elementos de software (elementos virtuais) [4].

A quantidade de variáveis que cada elemento controla dificulta o trabalho de integração de elementos na construção de modelos. A modelagem de um sistema de controle automático de altura de corte em colhedoras é demonstrada em [10]. A construção deste modelo exigiu o uso de equações diferenciais onde foram analisadas influência da velocidade de deslocamento da máquina, da pressão de suprimento do

sistema hidráulico, da massa da plataforma, e da força de contato de referência do controlador.

Sistemas como este tornam o processo de modelagem mais complicado e custoso e exigem pessoas mais experientes na construção de modelos de simulação. O número de elementos sendo simulados ao mesmo tempo para representar fielmente o comportamento de um sistema como este também requer o uso de mais recursos computacionais (processamento, memória) quando comparado a um modelo que utilize elementos reais em conjunto com elementos virtuais. Com a utilização da simulação heterogênea, ambos os problemas podem ser amenizados. A utilização de elementos reais facilita a construção de modelos pois reduz o esforço necessário na construção de elementos simulados. Além disso, a carga de processamento e recursos utilizados pelo simulador será menor.

Contudo, apesar das vantagens da utilização de elementos reais, alguns tipos de problemas podem surgir. Por exemplo, a largura de banda de transmissão de dados entre o elemento real e o software de simulação pode ser insuficiente. Se um elemento real gera um volume grande de dados constantemente, informações importantes podem ser perdidas durante a transmissão, comprometendo os resultados da simulação. Esse problema pode ser resolvido com a utilização de um protocolo de comunicação que certifica o recebimento de um pacote através de uma primitiva que indique a ocorrência desse fato.

Outro problema a ser considerado é a dificuldade na detecção de erros causados pelo mau funcionamento do elemento real, caso ele seja um componente físico (hardware). Um protocolo pode definir um mecanismo de *timeout*, ou seja, caso o elemento real não responda em um dado instante de tempo, é disparado um alerta de mau funcionamento do elemento. Existem ainda, vários outros problemas relacionados à adaptação de um elemento real ao computador, tais como os descritos em [3].

Para que a composição de elementos reais e simulados seja possível, é necessário que essa característica esteja presente no ambiente de simulação. O DCB (Distributed Co-simulation Backbone) [1] é uma arquitetura de suporte à simulação de modelos que integram elementos heterogêneos e é utilizado como foco das estratégias e soluções desenvolvidas neste trabalho. A arquitetura do DCB permite a adição de elementos reais, porém exige que o projetista do modelo tenha conhecimento detalhado das políticas de composição de elementos e de simulação do DCB. Este trabalho tem como objetivo facilitar a integração de elementos reais utilizados por um determinado modelo com o DCB para fins de simulação. Para atingir essa meta é apresentada uma estrutura composta por uma controladora e um protocolo de comunicação entre a interface do DCB e elementos reais. Esta estrutura tem como objetivo principal facilitar a adaptação de elementos reais.

O artigo apresenta, na seção 2, uma discussão sobre a adaptação de elementos reais no DCB. Na seção 3 é apresentada a estrutura para adaptação de elementos e o protocolo de comunicação DCB/Elementos reais. A seção 4 apresenta um estudo de caso que utiliza as soluções descritas na seção 3 para implementação de um Sistema Eletrônico de Plantio. Este sistema utiliza um sensor de sementes (elemento real). Finalmente, a seção 4 apresenta as considerações finais.

2 Adaptação de elementos no DCB

Segundo [14] não existe uma metodologia para utilização de ambientes de co-simulação heterogênea. De acordo com [9] o maior desafio da simulação no futuro será lidar com a complexidade dos modelos. Para diminuir a complexidade de um modelo, elementos reais podem ser utilizados. A adaptação de elementos reais no DCB é possível devido ao suporte provido a heterogeneidade pela sua estrutura. Esta flexibilidade é alcançada através de um módulo componente do DCB denominado *gateway*. A adaptação de elementos reais no DCB se dá através do *gateway*. Esse processo pode ser difícil ou custoso, pois o DCB não proporciona uma infra-estrutura intuitiva e de fácil entendimento para adaptação de novos elementos reais. Isso exige do modelador o estudo da estrutura de funcionamento interna do DCB.

Um dos aspectos que afeta a adaptação de elementos reais no DCB é a necessidade de implementação de um método para gerenciar uma determinada interface de comunicação. A criação de uma biblioteca que dê suporte a mais de uma interface como, por exemplo, USB, é necessário.

No lado do DCB devem ser escritas rotinas que são responsáveis pela captura e envio dos dados para o elemento real. Além disso, é necessária a configuração de alguns parâmetros de uma determinada interface para que ela funcione corretamente com o elemento real. Alguns parâmetros de configuração são: bits de paridade, velocidade de transmissão, codificação dos caracteres.

Devido à grande diferença de elementos reais em termos de interface de comunicação, protocolo de comunicação utilizado, consumo de energia, entre outros fatores, nem sempre a adaptação de um novo elemento ao DCB resume-se a sua interligação por meio de uma interface de comunicação. Essa variedade de configurações faz com que um hardware de adaptação seja desenvolvido para cada elemento real.

O desenvolvimento de uma placa genérica que permita a adaptação de uma grande variedade de elementos reais, com funcionalidades e características diferentes, ao DCB modificação e/ou re-configuração é difícil, pois cada elemento possui a sua peculiaridade. Também depende de diferentes requisitos de hardware para funcionar de maneira correta.

A próxima seção apresenta as soluções desenvolvidas para adaptação de elementos reais no DCB.

3 Solução para adaptação de elementos reais no DCB

A estrutura desenvolvida para facilitar a adaptação de elementos reais no DCB é baseada na arquitetura formada entre as entidades envolvidas nesse processo apresentada na Figura 1. Esta arquitetura é composta pelos seguintes módulos: Elemento real, Hardware de adaptação, IO genérico e o módulo DCB. O módulo Elemento Real é formado pelo próprio elemento real. Exemplos de elementos reais são sensores e atuadores.

O módulo IO Genérico é responsável pelo gerenciamento da comunicação do elemento real com o gateway do DCB. Esse módulo oferece suporte a algumas interfaces de comunicação, tais como RS-232, porta paralela e USB. Para cada interface de comunicação foi desenvolvida uma biblioteca que trata as particularidades de cada uma, e oferece funções para configuração, envio e recebimento de dados.

O módulo hardware de adaptação é composto pela controladora descrito na seção 3.1. A seção 3.2 apresenta o protocolo de comunicação desenvolvido, o qual é utilizado para troca de mensagens entre elementos reais e o DCB.

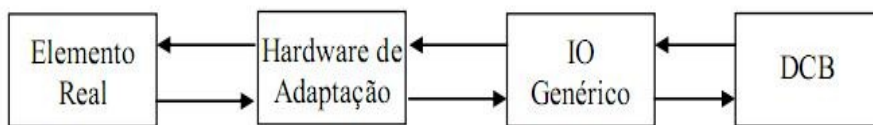


Fig. 1. Arquitetura formada a partir da composição dos elementos necessários para adaptação de elementos reais no DCB.

3.1 Controladora para elementos reais

A controladora é uma placa de circuitos digitais cujo objetivo principal é proporcionar uma interface de comunicação entre o elemento real e o DCB. Para isso é necessário que o elemento real seja acoplado à controladora. Isso pode ser feito através da utilização de um conjunto de transistores que fazem a padronização do sinal vindo de um elemento real.

A generalidade da controladora não é muito extensa, pois diferentes elementos reais possuem requisitos de hardware diferentes, o que torna a construção de um padrão genérico de controladora, quase impossível.

O trabalho atual desenvolveu uma controladora simples que suporta a adaptação de um sensor de sementes composto de um conjunto de leds infravermelhos. Estes leds geram uma tensão positiva quando detectam a passagem de uma semente. A implementação atual é suficiente para realização de testes com o sensor de sementes e valida a proposta da controladora.

Existem algumas funcionalidades que podem ser incluídas na controladora, independente do tipo de elemento real acoplado a ela. Por exemplo, um módulo, implementado em hardware (onde micro-controladores, como os da família PIC podem ser utilizados), que seja capaz de fazer o roteamento de mensagens sem a intervenção do DCB. A inserção de tal funcionalidade na controladora reduz o *overhead* do DCB, uma vez que mensagens não precisam ser recebidas e redistribuídas por ele. O estudo e desenvolvimento de tais funcionalidades é sugerido como tema para trabalhos futuros.

3.2 Protocolo de comunicação entre elementos reais e o DCB

O protocolo utilizado para a troca de dados foi desenvolvido tendo como referência o protocolo MODBUS. Seu principal objetivo é padronizar trocas de informações entre o DCB e o elemento real adaptado a ele através da controladora. Além disso, outro objetivo é oferecer mais controle sobre a comunicação, através de técnicas como a inclusão de mensagens de confirmação (ACKs), por exemplo. O elemento real pode incorporar o protocolo de comunicação em um módulo de hardware especializado, ou como uma aplicação que funciona no computador hospedeiro de uma determinada instância do DCB.

O protocolo implementado permite tanto o envio e recebimento de dados, como também a execução de funções. O código da função transmitido em cada pacote define que tipo de ação/reação deve ser tomada pelo receptor do pacote. O pacote resposta pode conter dados solicitados, ou apenas confirmar que a execução da função solicitada foi atendida. Caso ocorra um erro na execução da função, o campo *Data* irá conter um código de erro que será utilizado para decidir qual a próxima ação a ser tomada.

As funções implementadas no protocolo atual são descritas abaixo:

Tabela 1. Funções disponíveis na implementação atual do protocolo de transferência de dados elemento real/DCB.

Código da função	Nome da função	Descrição
0x04	Read Input Register	Lê dados de um registrador
0x06	Write Single Register	Escreve dados em um registrador
0x10	Write Multiple Registers	Escreve dados em vários registradores

O formato das mensagens trocadas entre os elementos reais e o DCB é mostrada na figura abaixo.

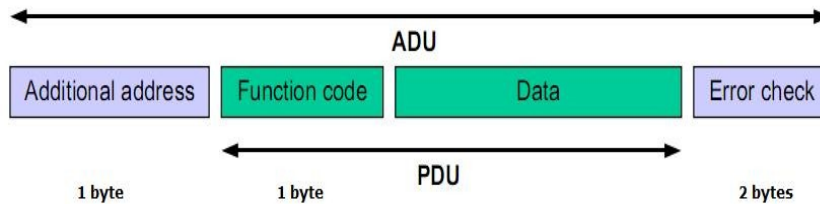


Fig. 2. Formato da mensagem definida pelo padrão Modbus [5].

O primeiro campo da mensagem contém o endereço do elemento real ao qual a mensagem se destina. O campo *function code* ou código da função carrega o código da função que será executada. O campo *data* ou dados contém os dados necessários

para a execução da função. Algumas funções não necessitam de dados. Além desses campos, também existe o campo *error check* ou campo de verificação de erros, que contém um código que é utilizado para verificar a integridade do pacote através do cálculo do CRC (Cyclic Redundance Check) [12].

A utilização de um protocolo para comunicação traz um série de vantagens como a maior facilidade para a integração de elementos reais, pois uma vez definidas as interfaces e os padrões de cada mensagem, novas implementações podem utilizar códigos criados por elementos anteriores. Além disso, a utilização de um protocolo facilita o oferecimento de suporte a mais de um elemento real em um determinado gateway pois o protocolo incorpora endereçamento de elementos, de modo que mensagens possam ser enviadas a um determinado elemento de acordo com seu endereço. O uso do protocolo possibilita a redução do overhead do DCB na troca de mensagens pois um módulo em hardware que seja capaz de manter uma tabela de endereçamento, semelhante a de um roteador, pode enviar mensagens de um elemento real X para um Y sem a intervenção do DCB.

4 Estudo de caso

Um das áreas que pode ser beneficiada com uso da simulação heterogênea é a Agricultura de Precisão (AP). A AP é uma técnica de gerenciamento sistêmico e otimizado do sistema de produção através do domínio da informação [13]. A essência da AP é a contínua obtenção de informações detalhadas da cultura seguida da utilização adequada dessas informações para otimizar o manejo [2]. Pesquisas na área vem tentando encontrar novas soluções voltadas para os equipamentos utilizados na agricultura. Esta afirmação justifica o desenvolvimento deste estudo de caso.

Adotando a estrutura para adaptação de elementos reais, desenvolvida por este trabalho, foi feita a adaptação de um sensor de contagem de sementes. Este sensor é componente de um Supervisor Eletrônico de Plantio, ilustrado na Figura 3. O sensor consiste de um conjunto de *leds* infravermelhos que detectam a passagem de sementes individualmente. A cada nova semente detectada é gerado um pulso elétrico que pode ser contabilizado como um incremento no número de sementes detectadas.

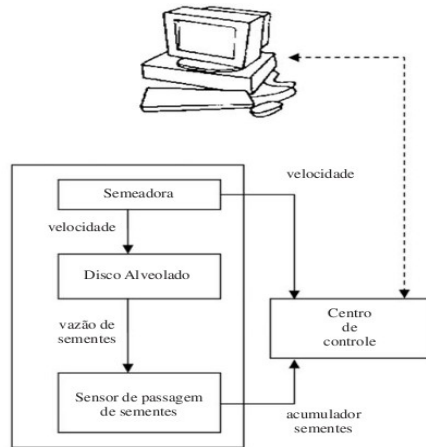


Fig. 3. Modelagem de um Sistema de Plantio Eletrônico (SEP) proposto por [1].

Para coleta dos pulsos elétricos que são enviados através da porta paralela, foi desenvolvido um módulo de software (*driver*) na linguagem C. Este *driver* é utilizado por um módulo, escrito em Java, através de um mecanismo de comunicação entre processos fornecido nativamente pela linguagem Java. O módulo em Java faz a comunicação direta com o *gateway* do DCB. A utilização da linguagem C deve-se à facilidade do tratamento dos dados, devido a sua característica de linguagem de baixo nível. Abaixo, é apresentado o código responsável pela comunicação entre o *driver* e o *gateway* do DCB. Os aplicativos executados através da primitiva *exec*, *pp* e *OpenCom*, são responsáveis pela manipulação dos pacotes referentes à porta paralela e serial, respectivamente.

Parte do código, retirado da implementação atual da estrutura, que demonstra a comunicação entre o *driver* (desenvolvido em C) e o *gateway* do DCB (desenvolvido em java).

```

Runtime runtime = Runtime.getRuntime();

Process proc = null;

try
{
    if (interface_comm == PARALELA)
        proc = runtime.exec("pp read");
    else if (interface_comm == SERIAL)
        proc = runtime.exec("OpenCom read");

    ...
}

```

Os testes realizados utilizam o estudo de caso proposto por [Mello 2005] e visam apenas a validação da estrutura proposta, e não uma avaliação do desempenho da mesma. A avaliação de desempenho deste e de outros protocolos a serem implementados para comunicação de elementos reais e o DCB é tema de pesquisas futuras.

O modelo SEP (Figura 3) caracteriza-se como heterogêneo devido a presença do sensor de sementes (elemento real). Os dados de entrada do modelo são: velocidade (em km/h); número furos do disco; e metros por volta do disco; os quais correspondem aos números 1, 2 e 3 da Figura 4, respectivamente. Os dados de saída do modelo, visualizados através da simulação são: sementes por metro esperadas; média de sementes por metro – sensor; velocidade (em metros/s); e sementes por metro instantâneo; os quais correspondem aos números 4, 5, 6 e 7 da Figura 4, respectivamente.

O sensor que fornece o número de sementes capturas em um dado instante de tempo é um elemento real (sensor de sementes). Esse elemento, a cada interrupção de hardware gerada, incrementa o contador de sementes. Como é possível perceber na Figura 4 existe um campo chamado de “Sementes por metro instantâneo – sensor” que mostra o número de sementes por metros que estão passando pelo sensor atualmente. Os valores resultantes da simulação do modelo SEP são mostrados na Figura 4.



Fig. 4. Resultados obtidos na simulação do Sistema Eletrônico de Plantio com a utilização do elemento real adaptado ao DCB.

O processo de adaptação do elemento real no DCB foi simples devido à estrutura desenvolvida. A controladora permitiu ao sensor enviar os pulsos elétricos gerados

através da porta paralela. O protocolo de comunicação entre elementos reais e o DCB fez com que nenhuma informação fosse perdida durante a simulação. A biblioteca de funções para diferentes interfaces de comunicação tornou o desenvolvimento de um módulo em Java para comunicação com o *gateway* simples e evitou a necessidade da manipulação de funções internas do DCB.

5 Conclusão

A adaptação de elementos reais no DCB pode ser facilitada com o desenvolvimento de uma estrutura que proporcione recursos apropriados para adaptação de elementos reais. Através da estrutura proposta neste trabalho, o processo de adaptação de um elemento real ao DCB é facilitado devido à utilização de uma controladora que, de um lado, oferece uma interface reconfigurável para cooperação com o elemento real e do outro coopera com o ambiente computacional. A cooperação é gerenciada por regras implementadas pelo protocolo específico para integração de elementos reais a modelos simulados.

A utilização do protocolo de comunicação entre elementos reais e o DCB possibilita uma melhora no controle da troca de mensagens. Isso ocorre com a implementação de mecanismos que controlam o fluxo de mensagens. Por exemplo, o envio de uma mensagem de confirmação a cada nova mensagem recebida. Outra vantagem oferecida pela utilização do protocolo é a utilização de mais de um elemento real. O uso de um campo de endereçamento contido no pacote gerenciado pelo protocolo permite que mais de um elemento real possa ser integrado ao modelo simulado usando uma única controladora

Os resultados deste trabalho podem ser utilizados, por exemplo, no projeto de sistemas computacionais no contexto da Agricultura de Precisão. A principal vantagem é a maior independência do uso de bancadas de testes para validação das partes computacionais que são acopladas às partes mecânicas dos equipamentos.

A próxima etapa do trabalho prevê a construção e validação de um novo conjunto de controladora e protocolo mais flexível. O objetivo é a composição de múltiplos elementos reais usando a mesma controladora.

Referências

1. Mello, B.A.: Co-simulação distribuída de sistemas heterogeneos. Tese (Doutorando em Ciência da Computação). Instituto de Informática, UFRGS. (2005).
2. Saraiva, A. M.: TI no agronegócio e biodiversidade. Escola Politécnica da USP, São Paulo. (2003).
3. Reynolds, P. F.: Heterogeneous distributed simulation. Proceedings of the 1988 Winter Simulation Conference. Washington. (1988).
4. Heiko, H.: A survey of HW/SW cosimulation techniques and tools. Royal Institute of Technology. Stockholm. Junho. (1998).
5. Modbus: Modbus Application Protocol Specification. Capturado em: www.modbus.org

6. Holzmann, G. J.: Design and Validation of Computer Protocols. Prentice Hall, 512 pgs. (1991).
7. Umezu, C. K, Cappelli, N. L. Desenvolvimento e avaliação de um controlador eletrônico para equipamentos de aplicação de insumos. Revista brasileira de engenharia agrícola ambiental. Vol 10, Março. (2006).
8. Rafull, L., Queiroz, L., ET AL.: Modelagem e análise de um sistema de controle automático da altura de corte em colhedoras. Revista brasileira de engenharia agrícola ambiental. Vol 10, Junho. (2006).
9. Oraw, B.;Choudhary, V.; Ayyanar, R.: A Cosimulation Approach to Model-Based Design for Complex Power Eletronics and Digital Control Systems. Arizone State University. Tempe. USA. SCSC. (2007).
10. Landi, D. C.: Proposta de Implementação de Terminal Virtual e Controladores de tarefas da Norma ISO 11783 em dispositivos computacionais portáteis. São Paulo. USP. (2004).
11. Edwards, S., Lavagno, L., Lee, E. A., Sangiovanni-Vincentelli, A.: Design of embedded systems: formal methods, validation, and synthesis. Proceedings of IEEE, Vol. 85, No. 3, Março. (1997).
12. Griffiths, G., Stones, G. C.: The Tea-Leaf Reader Algorithm: An Efficient Implementation of CRC-16 and CRC-32. Communications of the ACM. v.30, n.7, pp.617-620. (1987).
13. McBratney, A.; Bouma, J.; Whelan, B.; Ancev, T.: Future directions of precision agriculture. Precision agriculture, Springer Netherlands Publisher, v.6, n.1, p.7-23. (2005).
14. Fummi, F., Loghi, M., Poncino, M., and Pravadelli, G. 2009. A cosimulation methodology for hw/sw validation and performance estimation. ACM Trans. Des. Autom. Elect. Syst., 14, 2, Article 23. (2009).