

Um *framework* compacto e flexível em C++ para desenvolvimento modular de simulação de sistemas de hierarquia dinâmica

Jefferson William dos Santos¹

Adauto Luiz Mancini²

Luis Gustavo Barioni³

De acordo com Moore et al. (2007), um software para simulação deve permitir o reuso eficiente de modelos em diferentes contextos e escalas. Para Jones et al. (2001), modelos componentes devem poder ser adicionados, modificados e mantidos com pouco esforço. Entre as ferramentas de modelagem e simulação existentes, há *frameworks* que permitem codificar e compilar modelos com o uso de linguagens de programação genéricas. Estes têm a vantagem de permitir o uso de toda a capacidade de expressão da linguagem nativa. Além disto, o desempenho computacional é geralmente melhor do que sistemas que precisam ter seus códigos interpretados.

Sendo assim, este artigo descreve o núcleo de um *framework* de simulação orientado a objetos em fase de desenvolvimento, implementado em C++, com intuito de suportar o desenvolvimento de simulação de sistemas com hierarquias dinâmicas.

Este *framework* surgiu da necessidade de uma ferramenta para facilitar e padronizar o desenvolvimento de processos baseados em simulação de modelos em projetos de pesquisa da Empresa Brasileira de Pesquisas Agropecuárias (Embrapa), e buscou propiciar modularidade e simplicidade de código para facilitar o desenvolvimento de modelos por equipes de pesquisa multidisciplinares e sua implementação por grupo de estudantes estagiários.

¹ Fatec/Americana - jefferson.santos@colaborador.embrapa.br

² Embrapa Informática Agropecuária - {adauto.mancini, luis.barioni}@embrapa.br

Ele também provê maior desempenho em comparação às típicas ferramentas de modelagem interpretadas. Componentes do modelo, tipicamente desenvolvidos por equipes de pessoas com conhecimentos em processos específicos, podem ser desenvolvidos independentemente e depois conectados, sequencialmente ou agregados de forma hierárquica. O *framework* possibilita a compilação do simulador como bibliotecas, e provê uma interface genérica que permite que simulações sejam executadas por um aplicativo cliente (tal como uma interface gráfica de usuário, bancos de dado, pacotes matemáticos ou estatísticos) interagindo com a simulação. Em contraste com os outros *frameworks* existentes, este *framework* não armazena a trajetória das variáveis, mas permite que o aplicativo cliente obtenha os valores de saída durante a simulação por meio de uso de *callbacks* e armazene-os da maneira mais conveniente para uma finalidade específica. O *framework* suporta simulação contínua, de eventos discretos, ou simulação híbrida.

Possíveis futuras alterações neste *framework* podem ser facilmente executadas pelos estagiários, visto que seu código-fonte é bem estruturado e organizado, devido à utilização de normas para a sua padronização, bem como intensa utilização de teorias de sistemas e componentes. A Figura 1 define as classes do simulador e as relações existente entre elas.

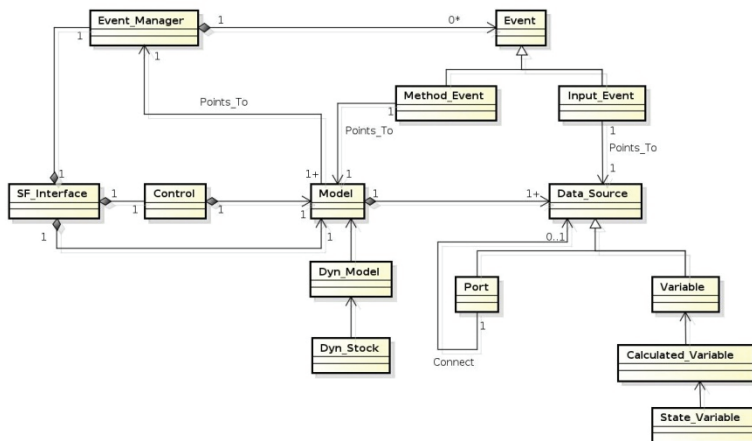


Figura 1. Diagrama de classe do *framework*.

A comunicação entre uma aplicação cliente e um simulador implementado no *framework*, possivelmente encapsulado em uma biblioteca dinâmica, é feita por *callbacks* pré-definidas. A comunicação de eventos de controle, entre os componentes sequenciais ou contidos em uma hierarquia do simulador, também é padronizada, reduzindo a introdução de erros durante o processo de implementação ou modificação do sistema pelos estagiários. A Figura 2 evidencia as *callbacks* existentes no sistema.

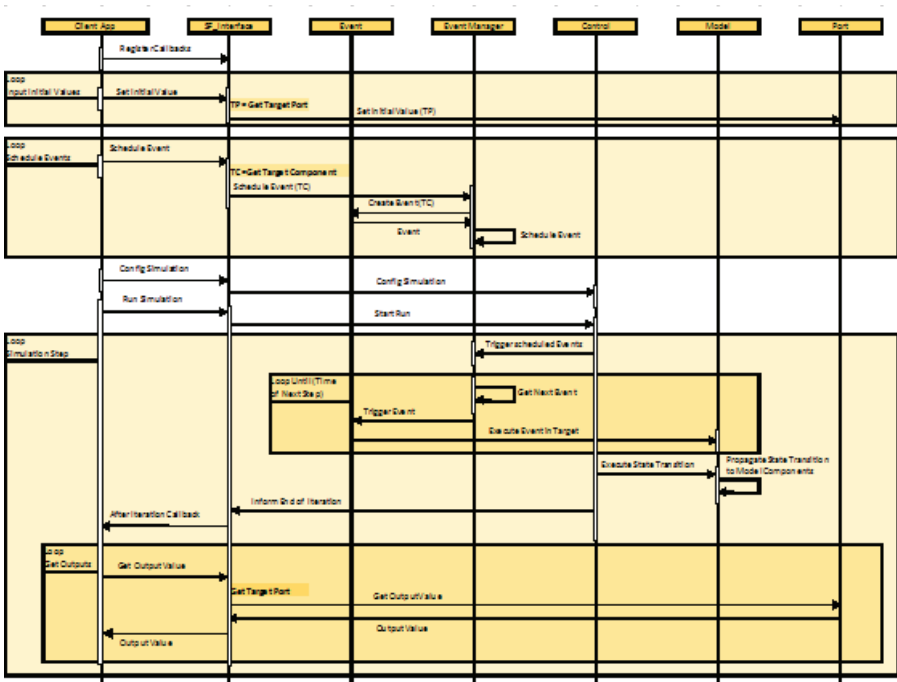


Figura 2. Diagrama de Sequencia simplificado do *framework*.

O desenvolvimento do *framework* descrito aqui é de grande importância em se tratar de apoio e padronização no desenvolvimento de modelos de simulação contínua em nossa equipe (o qual é o nosso atual objetivo). O *framework* forneceu autonomia à nossa equipe para um melhor desenvolvimento e customização de funcionalidades específicas necessárias a projetos de simulação, bem como fácil manipulação, atualização e aprendizado da parte dos estagiários envolvidos.

Referências

MOORE, A. D.; HOLZWORTH, D. P.; HERRMANN, N. I.; HUTH, N. I.; ROBERTSON, M. J. The common modelling protocol: a hierarchical framework for simulation of agricultural and environmental systems. **Agricultural Systems**, Barking, v. 95, n. 1-3, p. 37–48, Dec. 2007.

JONES, J. W.; KEATING, B. A.; PORTER, C. H. Approaches to modular model development. **Agricultural systems**, Barking, v. 70, n. 2-3, p. 421–443, Nov./Dec. 2001.