

SIMULADOR EVOLUTIVO PARA CONTROLE DE SISTEMAS DINÂMICOS¹

Rodolfo Berlezi², Eldair Fabrício Dornelles³, Rogério Samuel De Moura Martins⁴, Sandro Sawicki⁵, Edson Luiz Padoin⁶, Tales Ruan Scheffler Panke⁷.

¹ Resumo expandido com os resultados do tempo de pesquisa dedicado ao projeto do grupo GCA

² Bolsista de Iniciação Científica – PIBITI/UNIJUI e Acadêmico do Curso de Ciências da Computação, rodolfo_berlezi@hotmail.com

³ Bolsista de Iniciação Científica - PROBIC/FAPERGS - e Acadêmico do Curso de Ciências da Computação, fabriciud@gmail.com

⁴ Orientador, Professor do Curso de Ciências da Computação, Mestre em Computação Aplicada, rogerio.martins@unijui.edu.br

⁵ Professor do Curso de Ciências da Computação, Mestre e Doutor em Ciências da Computação, sawicki@unijui.edu.br

⁶ Professor do Curso de Ciências da Computação, Mestre em Engenharia de Produção, padoin@unijui.edu.br

⁷ Bolsista Voluntário e Acadêmico do Curso de Ciências da Computação, talesruan@gmail.com

Introdução

Nos dias contemporâneos de hoje, as tecnologias providas pela computação estão presentes e são requisitadas na grande maioria das áreas de trabalho. A mesma, nos auxilia e facilita em inúmeras atividades de diversas formas possíveis. Uma vez sendo utilizada para a efetivação das formas de trabalho e para enxergar novas maneiras de efetuar a mesma tarefa, a tecnologia que será proposta, descomplicará um problema para uma resolução ou visualização mais prática.

Com isso em mente, foi desenvolvido um software com a capacidade de receber, evoluir e modelar de forma gráfica um modelo preposto pelo usuário. Dando possibilidade de visualizar diversos pontos da estrutura e ter a capacidade de testa-la de diversas formas virtuais antes de partir para a execução da aplicação física desejada. Assim, com a visualização anterior, será possível evitar futuros transtornos, antes desconhecidos, ao executar o modelo.

Metodologia

Como base de conhecimento fundamental para entender a composição do projeto de software criado, é necessário o esclarecimento dos seguintes conceitos: algoritmos genéticos e sua evolução, a programação genética, além de sistemas de controle, agentes inteligentes e ambientes computacionais, linguagens formais e compiladores.

Modalidade do trabalho: Relatório técnico-científico
Evento: XXII Seminário de Iniciação Científica

A começar pelos algoritmos genéticos os quais são de extrema utilidade e de fundamental importância para o desenvolvimento de um dos modelos de agente adaptativo, com o qual trabalharemos. Visto que os algoritmos genéticos são muito efetivos quando se precisa evoluir uma solução para encontrar resultados ótimos de um problema pré-estabelecido.

Os algoritmos genéticos recebem um problema e trabalham em cima do mesmo até que atinjam a melhor solução possível dentro das expectativas ou o resultado máximo, de fitness 100%, na população. Funcionam utilizando métodos comparados e similares a reprodução biológica, como o crossover e a mutação genética, em uma população de códigos de soluções e misturando-os de certo modo aleatório para tentar encontrar a solução ideal.

Sistemas de controles são compostos por três partes, a entrada, o processo e a saída. Podendo comparar qualquer ação a um modelo de sistema, como, por exemplo, o ato de respirar, onde o ar entra nos pulmões, que o processa e o modifica para então sair do corpo e novamente recomençar este sistema.

Existem dois tipos principais de sistemas, os de malha aberta e os de malha fechada. O que difere entre eles é que em uma malha fechada, a solução da saída do sistema influencia na próxima entrada do sistema podendo alterar o processo. Já em uma malha aberta, o sistema ocorre de forma linear e independente do resultado, sendo nada alterado no decorrer do processo.

A estratégia dos sistemas de controle é aplicável nos próprios algoritmos genéticos. Sendo correto afirmar que os algoritmos genéticos são uma malha fechada, visto que a última população é base para a nova população de códigos que será gerada posteriormente.

Existe a noção de Agentes Inteligentes, que são simplesmente algo que age de forma maior e mais eficiente que um programa comum e Ambientes que são a área de trabalho dos Agentes podendo aparecer com diversas propriedades pré-definidas. "Diferentes ambientes precisam de agentes diferentes para usá-los com efetividade. Isso mostra que quão mais próximo de situações reais, mais complexo e distante fica de determiná-lo" (RUSSELL; NORVING, 2004, p.45).

Uma vez que seria necessária uma linguagem de programação para a criação do software, a escolhido foi a linguagem orientada a objeto JAVA, visto que é a principal linguagem do curso e oferece inúmeras possibilidades para a criação de um programa. Somando a ferramenta NetBeans que tem por função auxiliar na tarefa de programação.

Além do próprio JAVA, um de seus derivados, utilizado para a criação de uma nova gramática composta de diferentes interpretações de símbolos, o JAVACC. JAVACC ou JAVA Compiler

Modalidade do trabalho: Relatório técnico-científico

Evento: XXII Seminário de Iniciação Científica

Compiler, é um modelo do JAVA voltado para a criação de novas gramáticas para programação. “Sabendo que um alfabeto é composto por finitos símbolos, uma linguagem é composta por infinitas palavras construídas a partir dos símbolos do alfabeto e finalmente, uma gramática é feita de finitas regras que geram as palavras e seus significados ou semânticas” (MENEZES, 1998).

Compiladores então, são programas que traduzem o código fonte que é digitado e facilmente legível pelo programador, em outras palavras, com grande nível de abstração. Traduzindo para o código objeto que é o código binário, conhecido pelo baixo nível de abstração, mas é o qual a máquina consegue ler e executar as instruções.

Todo esse processo de tradução é composto por análises de síntese, semântica e léxica. As quais verificam se a palavra no código está excepcionalmente correta e adequada ao contexto da gramática. Somente após estas análises, ocorre a real geração do novo código objeto espelhado no código fonte.

Resultados e Discussão

Finalmente, implementamos um software disposto em três grandes partes principais chamadas de Controller, Model e View. As três partes são interconectadas e devem trabalhar em harmonia para a boa utilização do programa.

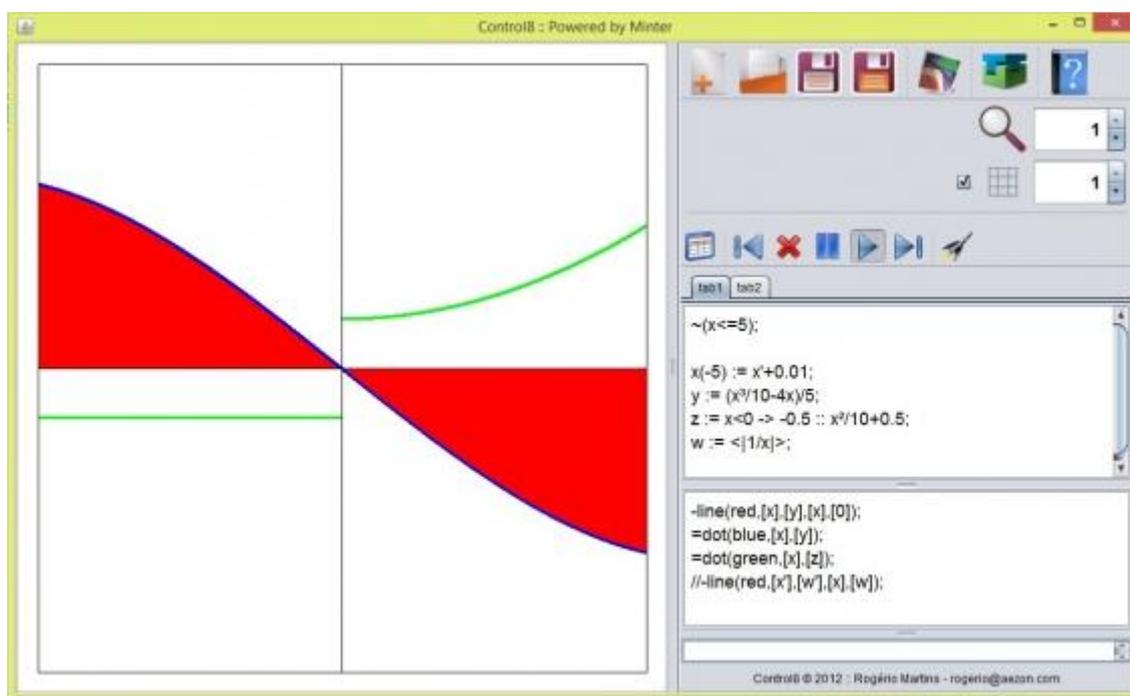
O Controller é o Agente Inteligente de característica adaptativa do software, onde acontecerá toda a leitura da equação problema, sistematizando e evoluindo através de algoritmos genéticos a equação até o máximo permitido. O Model será o Ambiente, onde o usuário poderá interagir com o software, digitando e modificando seu Ambiente para os diversos testes e espaço de adaptação do Agente Inteligente. Na View ocorrerá a transformação dos resultados obtidos pelo Agente no Ambiente em forma gráfica e de fácil legibilidade para o usuário.

A linguagem e estrutura básica do software foi implementada com sucesso, isso inclui, a gramática feita através do JAVACC, classes com diversas utilidades específicas, mas ainda não totalmente interligadas. Também já foi construído um protótipo da interface gráfica. Na Figura 1, mostra a atual interface gráfica do software desenvolvido. Duas das três partes estão funcionando adequadamente.

O Ambiente, definido como o Model, pode ser visto na “tab1”, localizada no lado direito da interface, com uma equação diferencial de exemplo. Já logo abaixo está localizada a View e o gráfico ao lado representa o resultado obtido desta equação. O Agente Inteligente adaptativo ainda está para ser implementado, sendo ele o Controller que não é visível ao usuário.

Modalidade do trabalho: Relatório técnico-científico
Evento: XXII Seminário de Iniciação Científica

Dado o fato de o Agente ainda está para ser desenvolvido, o software funciona coerentemente apenas com este exemplo fixo para testes.



(Figura 1, Interface Gráfica do Usuário)

Conclusão

A proposta de criação deste software é válida, já que a utilizaremos para testes de diversos outros projetos dentro do grupo de pesquisa GCA. O andamento da implementação do código está de acordo com o planejado mas, ainda faltam alguns detalhes a serem resolvidos e construir a resolução do Agente que é o cérebro do programa. Uma vez que o Agente Inteligente adaptativo será a parte chave para o bom desempenho e resultados do software para o usuário.

Então, como trabalhos futuros temos a terminar o Agente Inteligente adaptativo, melhorar a leitura do Model para ampliar o número de possibilidades de utilização dos usuários, conectar classes e métodos específicos de formas de visualização, além da possibilidade de acrescentar uma versão 3D dos gráficos da View usando Java3D.

Palavras-Chave: Agente Inteligente Adaptativo; Software; Ambiente.

Modalidade do trabalho: Relatório técnico-científico
Evento: XXII Seminário de Iniciação Científica

Agradecimentos

Gostaria de agradecer a UNIJUÍ por apoiar a pesquisa com as bolsas de Iniciação Científica, meu Orientador por ter escolhido e confiado a mim, mesmo achando que não fui totalmente exemplar. No mais meus colegas do grupo GCA pelo apoio e ajuda em diversas questões no trabalho e minha família por tudo o resto.

Referências Bibliográficas

RUSSELL, Stuart; NORVING, Peter. Inteligência Artificial. 2ª ed. Campus, 2004.

LINDEN, Ricardo. Algoritmos Genéticos. 3ª ed. Ciência Moderna, 2011.

NISE, Norman S. Engenharia de Sistemas de Controle. 6ª ed. LCT, 2012.

MENEZES, Paulo Blauth. Linguagens Formais e Autômatos. 5ª ed. Artmed. 1998.

LOUDEN, Kenneth C. Compiladores – Princípios e Práticas. Pioneira Thomson Learning, 2004.