



II Seminário de Formação Científica e Tecnológica

Grupo de Pesquisa em Computação Aplicada

www.gca.unijui.edu.br

19 – Maio – 2014

13:30 às 17:30

Local:

Campus Universitário da UNIJUI

Santa Rosa, RS. Brasil

Índice dos Trabalhos Apresentados

Protótipo de IDE para Desenvolvimento de Agentes Inteligentes-Redes Neurais Artificiais <i>Henrique Augusto Richter, Rogério Samuel de Moura Martins</i>	3 - 4
Modelagem e Implementação de Jogos Voltados a Sistemas Dinâmicos Simbólicos <i>Jean Rafael Reus da Silva, Rogério Samuel de Moura Martins</i>	5 - 6
Implementação de Ferramenta para Modelar Ambientes Usando JavaCC <i>Rodolfo Berlezi, Rogério Samuel de Moura Martins</i>	7
Uma Proposta de Framework de Comparação de Provedores de Computação em Nuvem <i>Igor G. Haugg, Rafael Z. Frantz</i>	8 - 9
Definição de uma Rede de Sensores Sem-Fio para a Coleta de Dados na Arquitetura AgroMobile <i>Marcos Sulzbach Morgenstern, Vinicius Maran</i>	10
Uma Arquitetura Ubíqua de Gerenciamento de Atividades no Domínio de Agricultura de Precisão <i>Roger Victor Alves, Vinicius Maran</i>	11
Desenvolvimento de uma Ferramenta Iterativa para a Visualização da Cobertura de Sinal de Antenas Baseada em Simulated Annealing <i>Gabriel Freytag, Sandro Sawicki</i>	12 - 13
Uma Ferramenta de Apoio à Edição e Validação de OVMS Textuais para dar Suporte ao Processo de Análise Automática <i>Cristiano Politowski, Fabricia Roos-Frantz</i>	14
Aplicação de Redes Neurais Recorrentes em Ambientes Parcialmente Observáveis <i>Eldair Dornelles, Rogério Samuel de Moura Martins</i>	15 - 16
Mapeamento de Informações Geomorfológicas para Ambientes Computacionais <i>Guilherme Henrique Schiefelbein Arruda, Sandro Sawicki</i>	17 - 18
Especificação Formal da Sintaxe Abstrata do Guaraná DLS <i>Mauri J. Klein, Sandro Sawicki, Fabricia Roos-Frantz, Rafael Z. Frantz</i>	19 - 21



Applied
Computing
Research Group

II SFCT

O *II Seminário de Formação Científica e Tecnológica* é um evento promovido pelo Grupo de Pesquisa em Computação Aplicada (GCA) juntamente com o Programa de Pós-Graduação em Modelagem Matemática e o curso de Ciência da Computação da UNIJUI, e tem como objetivo criar um espaço local de debate que possa contribuir positivamente sobre o trabalho que vem sendo desenvolvido pelos membros do grupo, especialmente alunos de mestrado e bolsistas de iniciação científica e tecnológica.

Comitê de Organização:

Dr. Rafael Z. Frantz

www.gca.unijui.edu.br/rzfrantz

Dr. Sandro Sawicki

www.gca.unijui.edu.br/sawicki

Dra. Fabricia Roos-Frantz

www.gca.unijui.edu.br/frfrantz

Eldair F. Dornelles

www.gca.unijui.edu.br/efdornelles

Igor G. Haugg

www.gca.unijui.edu.br/igorhaugg

Protótipo de IDE para Desenvolvimento de Agentes Inteligentes – Redes Neurais Artificiais

Henrique Augusto Richter¹, Rogério Samuel de Moura Martins

UNIJUI, Departamento de Ciências Exatas e Engenharias

Rua do Comércio, 3000. 98700-000, Ijuí – RS – Brasil

henrique.a.richter@gmail.com, rogerio.martins@unijui.edu.br

Palavras-chave: Redes Neurais; Agentes Inteligentes; Java Neural Network Simulator.

Redes Neurais Artificiais (RNAs) são sistemas computacionais inspirados nas estruturas neuronais do cérebro, que possuem a habilidade de aprender de acordo com informações recebidas do ambiente e através disso possuem uma grande capacidade de adaptação, e é por possuir esse potencial que RNAs são amplamente utilizadas em sistemas dotados de inteligência, que poderão ser utilizados para nos auxiliar em tarefas do dia-a-dia e também em tarefas específicas, como por exemplo, classificação de padrões, análise de imagens, previsão do mercado financeiro, sistemas de controle, etc.

Com o objetivo de facilitar o trabalho dos desenvolvedores na construção e treinamento de RNAs este projeto propõe o desenvolvimento de um protótipo de Ambiente Integrado de Desenvolvimento (IDE), facilitando e agilizando a implementação de RNAs utilizando apenas componentes gráficos, sem que haja necessidade de o desenvolvedor digitar grande parte do código necessário para o funcionamento da rede. Após o desenvolvimento a ferramenta estará disponível para uso nos projetos do grupo.

Para o desenvolvimento desse protótipo foi analisado o software já existente nomeado de Java Neural Network Simulator (JavaNNS), desenvolvido pela Universidade Everardo Carlos de Tubinga localizada na Alemanha, é um software gratuito baseado no kernel do software Stuttgart Neural Network Simulator (SNNS), tendo sua interface gráfica desenvolvida em Java, e é compatível com os sistemas Windows NT, Windows 2000, RedHat Linux 6.1, Solaris 7 e Mac OS X. Nele é possível criar as camadas da rede através de um menu onde poderá ser informada a quantidade de neurônios a serem criados nessa camada, bem como o tipo desses neurônios (entrada/saída/etc), qual a função de ativação, etc. É possível visualizar graficamente a rede criada através de uma interface de visualização que mostra todos os seus neurônios e conexões. O método de treinamento padrão é o método Backpropagation, porém além deste é possível escolher vários outros métodos disponíveis. Após configurado todos os parâmetros da rede é possível realizar o treinamento da mesma e acompanhar os erros e ajustes dos pesos através de gráficos. Também há a possibilidade de salvar a Rede Neural configurada em um arquivo, podendo, posteriormente, ser reaberta pelo software.

¹ Bolsista PIBIC/UNIJUI

O protótipo que será desenvolvido tentara apresentar uma interface mais amigável ao usuário em relação a interface do JavaNNS, e um meio mais prático para a criação de RNAs. Em relação as funcionalidades, a princípio será criado a interface do usuário do protótipo, ficando para trabalhos posteriores a implementação das funcionalidades do software. O ambiente desenvolvido será modular e permitirá a adição e integração de outros módulos futuramente. Também será integrado com o projeto do Rodolfo no qual é feita a descrição do ambiente, permitindo assim a especificação e testes de agentes inteligentes.

Modelagem e Implementação de Jogos Voltados a Sistemas Dinâmicos Simbólicos

Jean Rafael Reus da Silva¹, Rogério Samuel de Moura Martins
UNIJUÍ, Departamento de Ciências Exatas e Engenharias
Rua do Comércio, 3000. 98700-000, Ijuí – RS – Brasil
jrafaelrs.silva@gmail.com, rogerio.martins@unijui.edu.br

Palavras-chave: OpenGL; Padrões de Projeto; Implementação de Jogos.

O projeto tem por objetivo criar aplicações gráficas(jogos), que possam ser usados para realizar testes de inteligência artificial e que também possam ser usados para a aprendizagem de jogos e para melhorar o desempenho do raciocínio lógico.

Para o desenvolvimento está sendo utilizado duas linhas:

→ Programação em C++ utilizando a API gráfica OpenGL. Esta é a linha que desenvolve para computadores, com o objetivo de testar técnicas de inteligência artificial e apresentar os resultados de maneira visual.

→ Programação em java para dispositivos moveis(Android), utilizando java, XML e a API gráfica OpenGL. Que proporciona o aprendizado de jogos e também possibilita a realização de testes com inteligencia artificial.

Atualmente o foco do projeto é o desenvolvimento para Android, onde está sendo implementados jogos de tabuleiros. Esses jogos de tabuleiro são jogos de caça, que estão sendo aplicados em um projeto com parceria com a UFRGS, que busca apresentar novas formas de aprendizagem e também desenvolver o raciocino lógico dos jogadores. Como os jogos que estão na lista para serem desenvolvidos são pouco conhecidos, os aplicativos devem auxiliá-los no entendimento das regras, ou seja, dando dicas de como jogar determinado jogo e mostrando-lhe as regras gerais, que devem ser separadas por níveis, onde cada vez que uma regra for dominada o jogo passara para um novo nível desbloqueando uma nova regra mais complexa. Isso visando o processo incremental de aprendizado utilizado pela UFRS, onde também deve ser utilizada como aprendizado para agentes inteligentes.

Para iniciar o desenvolvimento para Android deve-se compreender java e um pouco de XML, instalar as ferramentas necessária, no casso Eclipse, SDK e ADT, criar um emulador ou instalar um dispositivo com Android via alguma das portas USB (a segunda opção tende a ser mais favorável para o desenvolvimento), para que os testes possam ser realizados.

Também e necessário conhecer um pouco sobre as API level, que são versões do Android onde cada uma possui diferenças na hora da programação. As level mais altas possuem mais atualizações, porém se tornam mais complexas na hora de programar.

¹ Bolsista PIBIC/UNIJUI

Para a aprendizagem do desenvolvimento foi realizado o desenvolvimento do jogo popularmente conhecido como PONG, onde existem duas barras laterais que podem ser movimentadas pelo usuário como o objetivo de não deixar com que a bolinha passe por elas, com o tempo as bolinhas vão aumentando e algumas começam a ter mais velocidade para aumentar a dificuldade do jogo.

Dentre os jogos que estão sendo desenvolvidos para o projeto o primeiro selecionado foi o da lebre e cachorros, onde o jogador tem 3 cachorros e deve impedir a lebre de passar por eles.

Implementação de Ferramenta para Modelar Ambientes usando JavaCC

Rodolfo Berlezi¹, Rogério Samuel de Moura Martins

UNIJUI, Departamento de Ciências Exatas e Engenharias

Rua do Comércio, 3000. 98700-000, Ijuí – RS – Brasil

rodolfo_berlezi@hotmail.com, rogerio.martins@unijui.edu.br

Palavras-chave: JavaCC; Problema do Pendulo Invertido; Linguagens Formais.

Minha pesquisa tem como objetivo final, a produção de uma equação diferencial para controlar uma representação gráfica do problema de um pêndulo invertido. Para tal, houve a necessidade de aprender a utilizar o JavaCC, ou Java Compiler Compiler e entender os mecanismos que ele possibilita.

Para começar, tive que ler livros e artigos sobre linguagens formais, onde explicava muito sobre gramáticas, semânticas e sintaxes de uma linguagem, assim como também sobre os autômatos e em suas formas variadas, que são utilizados para manusear uma gramática, com seu alfabeto, estados e regras.

Logo após, fui buscar sobre compiladores, que são os responsáveis por fazerem a parte de tradução por trás do que vemos na linguagem, onde o compilador lê, interpreta e traduz o código fonte, que é uma linguagem de alto nível, abstrata para nós podermos entender com maior facilidade, em um código objeto, de baixo nível, composto por números binários, usualmente chamado de linguagem de máquina. Nesta tradução, é onde surge a necessidade das semânticas e sintaxes, com um analisador separado para cada uma, são estes analisadores que verificam todo o código objeto para somente depois, o traduzir com a certeza de que compilará.

Equações diferenciais são funções que aparecem sob forma de derivadas, com uma variável x , y e suas respectivas derivadas. Vasculhei também, outros projetos que utilizaram o problema do pêndulo invertido a fim de descobrir melhor o seu funcionamento, mas claro, todos resolveram por formar diferentes.

Então cheguei ao JavaCC, onde é necessário criar uma série de códigos com base no analisador Parser para criar uma gramática, utilizando os chamados Tokens, para criar palavras identificadoras de um símbolo, Skips, para ignorar certos sinais e a utilização dos métodos para manusear a semântica dos Tokens que depois serão retomados para uso no código Java.

¹ Bolsista PIBITI/UNIJUI

Uma Proposta de Framework de Comparação de Provedores de Computação em Nuvem

Igor G. Haugg¹, Rafael Z. Frantz

UNIJUÍ, Departamento de Ciências Exatas e Engenharias

Rua do Comércio, 3000. 98700-000, Ijuí – RS – Brasil

igor-haugg@hotmail.com, rzfrantz@unijui.edu.br

Palavras-chave: Integração de Aplicações Empresariais; Computação em Nuvem; Framework de Comparação.

Atualmente existe uma grande quantidade de empresas que necessitam realizar integração entre suas aplicações. Tal necessidade ocorre em função de que as empresas desejam dar suporte aos novos processos de negócios surgidos, reutilizando funcionalidades que já estão presentes nas suas aplicações atuais. Normalmente essas empresas possuem aplicações que foram desenvolvidas de forma personalizada ou adquiridas de outras empresas que não foram projetadas pensando em ser reutilizadas. Integrar aplicações empresariais é fazer com que diferentes aplicações que não foram concebidas tendo em mente sua integração, possam colaborar para dar suporte a um novo processo de negócio. Computação em nuvem é um novo paradigma de computação no qual os recursos como armazenamento e processamento podem ser acessados a um baixo custo através da internet. Pode ser dividida em três principais tipos de serviço: infraestrutura, plataforma e software como serviço. A infraestrutura de computação em nuvem é definida como a contratação de recursos computacionais como um serviço, que pode ser, servidores, roteadores ou sistemas de armazenamento. Em uma plataforma de computação em nuvem é disponibilizado uma plataforma de desenvolvimento utilizando os recursos da infraestrutura da computação em nuvem, como por exemplo, frameworks e linguagens de programação. No serviço de software são disponibilizadas pelas quais se paga por seu uso e não por sua licença e que estão disponíveis para serem assinadas através de um navegador web. Uma solução de integração pode ser implantada em servidores de computação em nuvem, utilizando o serviço de plataforma como um serviço.

Implantar soluções de integração em servidores de computação em nuvem é uma tarefa complicada, pois atualmente existem diversos provedores de computação em nuvem e cada provedor possui diferentes, tornando difícil a escolha de qual plano melhor se ajusta às necessidades da solução de integração e seja economicamente mais vantajoso. Portanto é essencial conhecer as características destes provedores e planos, reduzindo assim os custos da implantação de soluções de integração de aplicações empresariais na nuvem. Neste trabalho é proposto um framework que pode auxiliar na comparação de provedores de computação em nuvem. Foram estudadas e propostas distintas propriedades que auxiliam na comparação. Tais propriedades estão divididas em 4 grupos: básicas, de infraestrutura, de plataforma e de

¹ Bolsista PIBITI/CNPq

software. Com base no framework desenvolvido, foi possível realizar uma comparação dos principais provedores de computação em nuvem à nível mundial.

Definição de uma Rede de Sensores Sem-Fio para a Coleta de Dados na Arquitetura Agromobile

Marcos Sulzbach¹, Vinícius Maran

UNIJUI, Departamento de Ciências Exatas e Engenharias

Rua do Comércio, 3000. 98700-000, Ijuí – RS – Brasil

{marcos.morgenstern, vinicius.maran}@unijui.edu.br

Palavras-chave: Agromobile; Arduíno; Análise de Solo.

Devido à demanda de qualidade e produção que o mercado cobra, muitos agricultores buscam ferramentas, sendo uma a análise de solo, aumentando a produção e qualidade da produção ou focando no monitoramento da propriedade em tempo real controlando a quantidade de nutrientes e insumos que a propriedade necessita para apresentar desempenho satisfatório. Porém, a análise de solo de forma laboratorial, além de alto custo, muitas vezes inviabilizando a adoção deste método pelo agricultor. Pode ser demorado, dependendo da quantidade de coletas, e demais fatores que influenciam no tempo de análise, podendo tornar esse processo lento, privando o agricultor da resposta rápida, ágil e precisa. Foi definido e implementado um protótipo que atua como uma rede de sensores sem fio com topologia ponto-a-ponto. Este protótipo foi desenvolvido utilizando a plataforma Arduino. Arduíno é uma plataforma de hardware livre que permite a programação e interligação de sensores para a utilização em diversos domínios. Para realizar a coleta das grandezas do ambiente e envio destas informações para a arquitetura AgroMobile, integração sensores compatíveis com a plataforma de prototipagem. A frequência entre coletas é inicialmente realizada no intervalo de 30 segundos. Após a realização da coleta, cada nó monta uma mensagem. Posteriormente esta mensagem é enviada para o próximo nó, sendo que o nó receptor concatena sua coleta com a recebida, processo realizado até esta mensagem chegar ao último nó que envia esse pacote (leitura da rede) para o coletor definido na linguagem de programação Java. Atualmente, são consideradas as informações de: Id; Ph; Temperatura; Umidade do Ambiente; Umidade do Solo; Graus; Latitude; Longitude. Para comunicar com o coletor que realiza a filtragem das informações e posteriormente armazena-as na arquitetura, a rede utiliza a tecnologia Zigbee, integrada a plataforma Arduíno. Compartilhando dados entre transmissor e receptor desde que ambos estejam conectados na mesma frequência.

¹ Bolsista PIBITI/UNIJUI

Uma Arquitetura Ubíqua de Gerenciamento de Atividades no Domínio de Agricultura de Precisão

Roger Victor Alves¹, Vinícius Maran
UNIJUÍ, Departamento de Ciências Exatas e Engenharias
Rua do Comércio, 3000. 98700-000, Ijuí – RS – Brasil
{roger.alves, vinicius.maran}@unijui.edu.br

Palavras-chave: Agricultura de Precisão; Arquitetura Ubíqua; Ontologias.

A Agricultura de Precisão trouxe para agricultura melhorias inquestionáveis em relação a produtividade. Embora não seja uma tecnologia nova, só começou a ser realmente difundida com a popularização da internet e a troca de informações e práticas agrícolas através da mesma. Com o mapeamento da lavoura advindo de sensores, o agricultor obteve um melhor controle sobre o uso de insumos, tanto em questão de quantidade, como também, em qual lugar usar.

Apesar dos avanços gerados, o mercado ainda sente falta de uma arquitetura que acompanhe a lavoura em tempo real. Através do poder das ontologias, e um banco de dados que realiza inferência¹ em tempo real, é possível criar uma arquitetura que analise os dados dos sensores distribuídos na lavoura, e através dessa análise, recomende ações ao agricultor que podem impactar positivamente se tratadas em um curto período de tempo. Ex: aumento inesperado de temperatura após chuva pode alterar a forma de irrigação de uma gleba².

A apresentação montada falará sobre o atual desenvolvimento da parte de persistência de dados e também da parte de raciocínio. Atualmente, em paralelo a contribuição do formando Daniel Rossi, a ontologia (que envolve tanto a persistência como raciocínio) está sendo modelada com alguns princípios utilizados pela empresa John Deere localizada em Horizontina. Está em desenvolvimento também o software que será utilizado para o cadastramento dos individuais³ a fim da realização dos testes e também possíveis cadastros manuais futuros, e também o sistema de log que mostrará as informações que passarão pelo banco de dados. O projeto faz a interligação entre a ontologia e o aplicativo em java através do Jena, e usará o Stardog para realizar trabalhos de banco. Para a criação do servidor web, empregaremos o Vaadin, pela seu recurso de criar aplicações web usando puramente java EE.

¹Inferência, em Lógica, é o ato ou processo de derivar conclusões lógicas de premissas conhecida ou decididamente verdadeiras

²Terreno próprio para cultura; leiva, torrão

³Objetos básicos.

¹ Bolsista PIBIC/CNPq

Desenvolvimento de uma Ferramenta Iterativa para a Visualização da Cobertura de Sinal de Antenas Baseada em Simulated Annealing

Gabriel Freytag¹, Sandro Sawicki

UNIJUÍ, Departamento de Ciências Exatas e Engenharias

Rua do Comércio, 3000. 98700-000, Ijuí – RS – Brasil

gabrielfrtg@gmail.com, sawicki@unijui.edu.br

Palavras-chave: Simulated Annealing; Distribuição de Antenas.

No posicionamento de antenas em uma determinada área, a cobertura ideal da área se dá tendo um determinado número de antenas cobrindo o máximo possível da área. As antenas precisam estar distribuídas de modo a cobrirem esta sem que duas ou mais cubram uma mesma parte simultaneamente. Portanto, todas as antenas numa área, precisam ser trocadas de posição até que estas consigam cobrir a área sem sobreposições.

Porém, a cobertura de sinal ideal da área se dá através do posicionamento das antenas cobrindo parte da área simultaneamente. O posicionamento precisa ser feito de modo que duas ou mais antenas cubram uma determinada parte da área, sem que esta parte seja quase ou totalmente coberta simultaneamente. Desta forma, as antenas precisam ser posicionadas tendo em vista a cobertura da área com uma boa cobertura de sinal. Ambos os modos devem levar em consideração obstáculos, como rios, lagos e florestas, onde as antenas não podem estar posicionadas.

A ferramenta em desenvolvimento tem por objetivo simular a cobertura de sinal de antenas, seguindo estes dois modos de cobertura da área. No posicionamento das antenas, são utilizados três métodos: Livre, que busca cobrir ao máximo a área por meio da troca de posição de todas as antenas a cada iteração; Simples, também busca a cobertura máxima da área, porém, trocando apenas uma antena de lugar; e UDG, que tem como objetivo encontrar a melhor cobertura de sinal da área, trocando uma antena por iteração, com base em uma segunda antena para manter uma determinada proporção de sobreposição. Através destes três métodos de posicionamento, é possível observar e comparar o tempo de execução e cobertura de área e sinal entre eles.

Para a simulação do posicionamento, esta ferramenta disponibiliza de uma interface para a definição dos parâmetros de execução e outra cujo posicionamento é mostrado ao final da execução. Na interface de definição dos parâmetros, é preciso que se escolha o arquivo em que estão as informações da(s) barreira(s) e o local de gravação dos resultados. A altura e a largura da área onde será feito o posicionamento das antenas. O tipo de raio, igual ou diferente para todas as antenas, o valor do raio, método de posicionamento e modo de avaliação da meta- heurística, clássico ou guloso. Ao final da execução, é exibida uma

¹ Bolsista PIBIC/CNPq

interface onde são mostradas as posições das antenas, além da(s) barreira(s) inseridas nesta área.

Até o momento, a ferramenta possui grande parte de suas funcionalidades implementadas, restando apenas a implementação da consideração das barreiras e a iteratividade do posicionamento. Os próximos passos serão a implementação desta funcionalidade, onde a ferramenta identifique o local onde as barreiras estão e não posicione antenas nestes locais. Além disto, implementar a iteratividade do posicionamento, encontrando uma forma de fazê-la com a linguagem usada. Terminados estes passos e após alguns ajustes e melhorias tanto nas interfaces quanto no funcionamento ferramenta, os objetivos traçados terão sido alcançados.

Uma Ferramenta de Apoio à Edição e Validação de OVMs Textuais para dar Suporte ao Processo de Análise Automática

Cristiano Politowski¹, Fabrícia Roos-Frantz
UNIJUÍ, Departamento de Ciências Exatas e Engenharias
Rua do Comércio, 3000. 98700-000, Ijuí – RS – Brasil
crispolitowski@gmail.com, frfrantz@unijui.edu.br

Palavras-chave: Modelo de Variabilidade; Linguagens Específicas de Domínio.

A Engenharia de Linha de Produtos de Software (do inglês SPLE) é um paradigma de desenvolvimento de software voltado ao reuso de artefatos comuns, sendo seu principal elemento a Variabilidade, definida por um Modelo de Variabilidade, o qual pode ser expresso em diferentes notações. O Modelo de Variabilidade Ortogonal (do inglês OVM) é um desses modelos, cujo objetivo é tratar dos detalhes entre os elementos da variabilidade, ou seja, restrições e relações. O tamanho desses modelos cresce exponencialmente a medida que se acrescentam elementos, tornando impossível a análise manual dos mesmos. A Análise Automática de Modelos de Variabilidade (do inglês AAVM) tem o propósito de extrair informações destes modelos de forma automática facilitando a gestão da SPLE. FaMa-OVM é uma ferramenta de análise automática de OVMs, a qual recebe como entrada um modelo OVM escrito com a sintaxe da linguagem textual OVM, juntamente com operações de análise que são interpretadas e processadas resultando em respostas do tipo Verdadeiro ou Falso, um Produto, vários Produtos, entre outros. Porém, caso haja erros no modelo de entrada, a ferramenta não informará e o processamento resultará em erro.

Para auxiliar neste processo de modelagem e análise de OVMs, se faz necessário um editor com recursos de validação de sintaxe bem como uma gramática que valide a linguagem OVM existente. Sendo a Linguagem de OVM Específica de Domínio, o uso de Language Workbenches para sua construção é pertinente. Uma das mais conhecidas e completas ferramentas existentes na comunidade open source é a Xtext, possuindo, entre outras funcionalidades, suporte à criação da gramática com notação BNF, validação, formatação, syntax highlighting além dos recursos providos pela plataforma Eclipse, onde o Xtext funciona como plugin.

O trabalho realizado consistiu-se em duas etapas. Primeiramente foi feito um estudo dos conceitos de SPLE e DSL, trabalho este apresentado na primeira edição do Seminário de Formação Científica e Tecnológica. Posteriormente, foi feita a implementação da gramática utilizando a ferramenta escolhida, sua validação e integração com a ferramenta de análise FaMa-OVM.

¹ Bolsista PIBITI/CNPq

Aplicação de Redes Neurais Recorrentes em Ambientes Parcialmente Observáveis

Eldair F. Dornelles¹, Rogério Samuel de Moura Martins

UNIJUÍ, Departamento de Ciências Exatas e Engenharias

Rua do Comércio, 3000. 98700-000, Ijuí – RS – Brasil

{eldair.dornelles, rogerio.martins}@unijui.edu.br

Palavras-chave: Redes Neurais; Algoritmos Genéticos; Agentes Inteligentes.

Os ambientes parcialmente observáveis são ambientes que apresentam problemas complexos para a construção de um agente inteligente que possa interagir nesse meio, devido ao fato de não se ter informações completas sobre o ambiente o qual está interagindo. Para que um agente possa compreender e tomar as decisões adequadas, visando atingir um bom resultado, faz-se uso de algumas técnicas de aprendizado.

Neste trabalho foi modelado e implementado um agente inteligente, utilizando como estrutura computacional redes neurais recorrentes, o aprendizado desse agente ocorre através da utilização da meta-heurística algoritmos genético. O código está implementado em um framework de algoritmo genético e uma biblioteca de redes neurais recorrentes.

O ambiente escolhido para a modelagem do agente neste projeto foi o mundo de wumpus; um jogo aparentemente simples, porém exige que jogador/agente possua a habilidade de memorizar eventos passados para a tomada de decisão. O cenário desse jogo é uma caverna que consiste em salas conectadas por passagens. Com exceção da sala onde está posicionado o agente, entre as demais estão distribuídos: um monstro/wumpus que emite fedor às suas salas adjacentes, cada sala possui 20% de chances de conter um poço, os quais emitem brisa às salas adjacentes, e em uma das salas tem ouro que emite brilho apenas na sala em que se encontra. Este ambiente pode ser representado por uma matriz 4x4, cada endereço da matriz equivale a uma sala, e o agente se encontra posicionado no cruzamento entre a primeira coluna e última linha, voltado para direita. O agente deve encontrar o ouro sem cair nos poços e nem ser pego pelo wumpus, podendo este usar as percepções de brisa e fedor como norteadores para a tomada de decisões [1].

As redes neurais recorrentes “Arquitetura de Hopfield” são constituídas de uma única camada, em que todos os neurônios artificiais (unidades processadoras) são interligados, de forma que a saída de cada neurônio alimentam as entradas de todos os outros neurônios e a própria. Tal rede neural se destaca por seu comportamento tipicamente dinâmico, capacidade de memorizar relacionamentos, possibilidade de armazenamento de informações [2].

Algoritmos Genéticos é uma meta-heurística, a qual busca uma otimização global. Baseia-se nos mecanismos de seleção natural e da genética. Basicamente é gerada uma população com n indivíduos (cromossomos), cada indivíduo contém informações que geram

¹ Bolsista PROBIC/FAPERGS

uma determinada solução para o problema analisado, enquanto não for encontrado um indivíduo que contenha uma solução satisfatória é feito o cruzamento entre os indivíduos, o cruzamento é feito de dois em dois e geram outros dois indivíduos, o critério de avaliação para selecionar cada dois indivíduos que irão cruzar, é o seu fitness, ou seja, quanto maior o fitness, maiores as chances de ele ser selecionado mais vezes para cruzar, quanto maior o fitness de um indivíduo, melhor é a solução para o problema [3].

Referências

- [1] RUSSEL, Stuart; NORVIG, Peter. Inteligência Artificial: Tradução da Segunda Edição. Tradução de PubliCare Consultoria. Rio de Janeiro: Elsevier, 2004, p. 193.
- [2] SILVA, Ivan Nunes da; SPATTI, Danilo Hernan; Flauzino, Rogério Adrade. Redes Neurais Artificiais para Engenharia e Ciências Aplicadas. São Paulo: Artiliber, 2010, p. 201.
- [3] LINDEN, Ricardo. Algoritmos Genéticos. 3. ed. Rio de Janeiro: Ciência Moderna, 2011, p. 46.

Mapeamento de Informações Geomorfológicas para Ambientes Computacionais

Guilherme Henrique S. Arruda¹, Sandro Sawicki
UNIJUÍ, Departamento de Ciências Exatas e Engenharias
Rua do Comércio, 3000. 98700-000, Ijuí – RS – Brasil
guilherme_arruda@live.com, sawicki@unijui.edu.br

Palavras-chave: Mapeamento Geomorfológico; Fast Light Tool Kit.

Nesta nova etapa do projeto, foi dado início aos passos que foram destacados na última apresentação. As funções desenvolvidas foram fundamentais para a utilização de outros métodos de detecção, obtendo objetos, áreas e formas diferentes dos que estavam sendo analisados no princípio da aplicação. A razão da busca por novos objetos surgiu devido às áreas utilizadas serem muito amplas, o que facilitava a detecção de contornos, mas dificultava a adição do obstáculo na aplicação de cobertura de sinais devido ao tamanho da imagem resultante e da quantidade de coordenadas obtidas. Isso apresentou resultados pouco satisfatórios, pois, com áreas mais distantes e objetos menores, os contornos ficavam muito próximos uns dos outros, tornando a saída irregular.

A partir disso, surgiu a necessidade de reorganizar a aplicação no que diz respeito à sua estrutura. As funções e variáveis desta foram desenvolvidos estruturalmente, tanto por uma questão de testes, pois se tratava de uma biblioteca nova, quanto pelo motivo de que a organização não era prioridade no início do projeto. Deste modo, o poder do C++ não estava sendo aproveitado. Toda a parte estrutural foi então dividida em classes, facilitando a identificação dos processos e separando algumas partes do projeto, objetivando o encapsulamento. A orientação a objetos também foi adicionada, que é outro recurso poderoso do C++.

Outra mudança positiva no projeto foi a adição de interface gráfica para facilitar a execução da aplicação. Primeiramente, foi feito um estudo em relação à parte gráfica na linguagem C++. Optou-se por utilizar FLTK (Fast Light Tool Kit) pelo fato de ser simples e fácil de utilizar, compatível com OpenGL e possuir funções que se adaptavam ao nosso projeto. Assim, a parte gráfica foi pré-modelada e então desenvolvida com os recursos disponíveis. A interface possui um explorador de arquivos, onde é possível escolher graficamente uma imagem a ser manipulada.

Ao fazer isso, uma segunda janela é aberta, e a imagem é carregada em um espaço próprio, sendo este integrado com ferramentas para manipulá-la. As funções, que antes precisavam ser escolhidas via comando através do terminal, agora possuem seu próprio botão. Quando um destes é pressionado, os efeitos na imagem são executados em tempo real, o que facilita muito a identificação das mudanças ocorridas. A adição da parte gráfica

¹ Bolsista PIBIC/UNIJUI

possibilitou a integração de funções, o que antes não era possível fazer porque apenas uma função de cada vez poderia ser executada através do terminal. Esta integração tornou o processo de obtenção dos obstáculos mais fácil e preciso. As coordenadas obtidas podem ser dinamicamente salvas ao pressionar seu botão característico, processo que é feito graficamente.

Portanto, é fácil perceber o quanto a aplicação mudou diante do que foi apresentado. São inúmeros os benefícios resultantes, tanto para o tempo de execução quanto para o resultado final. Os próximos passos serão destinados à adição de mais funções de manipulação de imagem na interface e a integração destas, possibilitando trabalhar melhor com áreas mais afastadas.

Especificação Formal da Linguagem Guaraná Utilizando a Notação Z

Mauri J. Klein¹, Sandro Sawicki, Fabrícia Roos-Frantz, Rafael Z. Frantz

UNIJUÍ, Departamento de Ciências Exatas e Engenharias

Rua do Comércio, 3000. 98700-000, Ijuí – RS – Brasil

{mauri.klein, sawicki, frfrantz, rzfrantz}@unijui.edu.br

Palavras-chave: Integração de Aplicações Empresariais; Tolerância a Falhas; Notação Z.

Atualmente grande parte das corporações, contam com aplicações em seu ecossistema de software como suporte para os seus processos de negócio [9]. Estes são compostos, em sua maioria, por aplicações legadas, pacotes adquiridos de terceiros ou desenvolvidos especificamente para resolver um problema particular, o que dificulta a sua reutilização. Os ecossistemas de software são sistemas heterogêneos compostos por aplicativos que, normalmente, não foram projetados levando em conta a sua integração. A integração, porém, é necessária, principalmente porque permite a reutilização de duas ou mais aplicações, com o objetivo de otimizar processos de negócios já existentes ou fornecer suporte a novos processos.

A *Integração de Aplicações Empresariais* (EAI) fornece metodologias e ferramentas para desenvolver e implementar soluções de integração. O objetivo de uma solução de integração é manter dados sincronizados entre diferentes aplicações ou desenvolver novas funcionalidades sobre as já existentes [7].

Nos últimos anos, a demanda pela integração tem motivado o surgimento de ferramentas para o projeto e implementação de soluções de integração, como Camel [8], Spring Integration [3], Mule [2] e Guaraná [5]. Estas ferramentas facilitam a construção de soluções. No entanto, semelhante às aplicações convencionais, uma solução de integração pode apresentar falhas em qualquer parte de sua execução.

Um problema no software pode ser permanente (defeito) ou transiente (recurso como rede ou serviço temporariamente indisponível) e pode ou não ocorrer durante uma execução. Caso ocorra, gera-se um erro levando o sistema para um estado que, se não for tratado, pode resultar em um erro que então será perceptível aos usuários finais [1] [11].

Neste caso, uma característica desejável em ferramentas de integração é a Tolerância a Falhas (TF). Assim a solução de integração continuará funcionando corretamente apesar da ocorrência de falhas.

São quatro os estágios que envolvem a Tolerância a Falhas. O *Event Reporting*: reporta eventos sobre o funcionamento de uma solução de integração; *Error Monitoring*: monitora e detecta possíveis erros emitindo notificações sobre sua ocorrência; *Error Diagnosing*: analisa as notificações para conhecer as causas do erro e as partes da solução que foram afetadas;

¹ Bolsista CAPES/PROSUP

Error Recovery: tenta-se recuperar do erro [4]. Estes estágios são fundamentais para que a solução de integração apresente uma confiabilidade em sua execução. As tecnologias para construir soluções de integração como Camel, Spring Integration e Mule, fornecem um mecanismo de detecção de erro baseados principalmente na utilização de *try-catch* [6]. Dentre as propostas que estudamos, Guaraná é a única que proporciona um mecanismo de detecção de erro com base em um sistema de monitoramento que pode ser configurado usando uma linguagem baseada em regras [4].

Neste contexto, para validar a cobertura das regras escritas pelos Engenheiros de Software ou gerá-las automaticamente, este trabalho propõe especificar formalmente, através da Notação Z, a sintaxe abstrata, primeiro passo para a formalização do Guaraná DSL.

Na literatura existem inúmeros trabalhos que utilizam a especificação formal para modelar linguagens, dos quais destacamos a formalização de diagramas UML utilizando a Notação Z [10].

Contudo, a Notação Z é utilizada para a especificação formal de sistemas e baseia-se em princípios da matemática discreta e da lógica de primeira ordem. Emprega estruturas matemáticas como: conjuntos, relações e funções como forma de expressar o estado, comportamento e propriedades de um sistema. Possui sintaxe e semântica precisas que resultam em notações completas, coerentes, concisas e legíveis capazes de validar simbolicamente as propriedades do sistema sem a necessidade de compilá-los e testá-los.

Outra característica importante da especificação em Z, é que a mesma é formada por esquemas que são decomposições da especificação em partes menores e que podem ser combinados e usados em outros esquemas. Além disso, a notação Z utiliza-se da definição de tipos para cada objeto definido, sendo possível a checagem de tipos em Z.

Portanto, através da especificação formal utilizando a Notação Z, pode-se representar com clareza e precisão todos os tipos, esquemas e restrições do Guaraná DSL contribuindo para gerar automaticamente as regras da solução EAI e validá-las, permitindo verificar que sua especificação atende aos requisitos da solução de integração.

Referências

- [1] Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans. on Depend. and Secure Comp.*, Jan-Mar 2004
- [2] Dossot, D. and D'Emic, J. (2009). *Mule in Action*. Manning.
- [3] Fisher, M., Partner, J., Bogoevici, M., and Fuld, I. (2010). *Spring Integration in Action*. Manning.
- [4] Frantz, R. Z., Corchuelo, R., and Molina-Jiménez, C. (2012). A proposal to detect errors in Enterprise Application Integration solutions. *Journal of Systems and Software*, 85(3):480–497.
- [5] Frantz, R. Z., Quintero, A.M. R., and Corchuelo, R. (2011). A Domain-Specific Language to Design Enterprise Application Integration Solutions. *Int. J. Cooperative Inf. Syst.*, 20(2):143–176.
- [6] Goodenough, J. B. (1975). Exception handling: Issues and proposed notation. *Communications of the ACM*, 18(12):683–696.

- [7] Hohpe, G. and Woolf, B. (2003). *Enterprise Integration Patterns - Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley.
- [8] Ibsen, C. and Anstey, J. (2010). *Camel in Action*. Manning.
- [9] Messerschmitt, D. and Szyperski, C. A. (2003). *Software Ecosystem: Understanding an Indispensable Technology and Industry*. MIT Press.
- [10] Mostafa, A. M., Ismail, M. A., Bolok, H. E., and Saad, E. M. (2007). Toward a Formalization of UML2.0 Metamodel using Z Specifications. In *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on*, volume 1, pages 694–701.
- [11] R. Campbell and B. Randell. Error recovery in asynchronous systems. *IEEE Trans. Soft. Eng.*, 1986