

III Seminário de Formação Científica e Tecnológica

Grupo de Pesquisa em Computação Aplicada

www.gca.unijui.edu.br

15 – Junho – 2015
8:30 – 12:00 e 13:30 – 17:30

Local:
Departamento de Ciências Exatas e Engenharias - DCEENG
Ijuí, RS. Brasil

Índice

Modelagem Conceitual de uma Solução de Integração para o Processo de Rematrícula da Universidade Unijuí	5 - 6
<i>Igor G. Haugg, Rafael Z. Frantz</i>	
Desenvolvimento de uma Solução de Integração de Aplicações para Automatizar Reservas de Viagem	7 - 8
<i>Edinaldo G. da Silva, Fabricia Roos-Frantz</i>	
Simulação do Comportamento de uma Solução de Integração Desenvolvida para a Área de Venda Hospitalar Utilizando a Ferramenta Arena	9 - 10
<i>Gabriel Freytag, Sandro Sawicki</i>	
Modelagem Matemática e Computacional para Identificação de Gargalos de Desempenho em Soluções de Integração	11 - 12
<i>Arléte Kelm Wiesner, Rafael Z. Frantz</i>	
Comparação entre Multilayer Perceptron e Redes Neurais Lógicas no problema Presa-Predador usando Algoritmos Genéticos	13 - 14
<i>Márcia da Silva, Rogério Samuel Martins</i>	
Revisão da Literatura de Simulação de Sistemas de Eventos Discretos com Foco na Integração de Aplicações	15 - 16
<i>Roberto Saulo Cargnin, Fabricia Roos-Frantz</i>	
Uso de Unidades de Memória em Redes Neurais Recorrentes para Acelerar a Otimização	17 - 18
<i>Eldair Fabricio Dornelles, Rogério Samuel Martins</i>	
Modelagem Matemática e Computacional para Análise do Comportamento de Soluções de Integração de Aplicações Através da Criação de Modelos de Simulação com Cadeias de Markov	19 - 20
<i>Márcia Maria Horn, Sandro Sawicki</i>	
Adaptação de um Jogo Open Source para o Desenvolvimento de um Simulador de Trânsito	21 - 22
<i>Henrique A. Richter, Rogério Samuel Martins</i>	
Implementação de um Jogo de Tabuleiro para Evolução de Agentes Inteligentes: Tigres e Vacas.	23 - 24
<i>Rodolfo Berlezi, Rogério Samuel Martins</i>	
Uso de Jogos Lógicos Aplicados a Aprendizagem de Máquina	25 - 26
<i>Jean R. R. Da Silva, Rogério Samuel Martins</i>	
Desenvolvimento de um Modelo de Simulação Baseado em uma Solução de Integração Teórica Utilizando a Ferramenta PRISM	27 - 28
<i>Guilherme Henrique Schiefelbein Arruda, Sandro Sawicki</i>	



Applied
Computing
Research Group

III SFCT

O *III 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.

Coordenação Geral:

Dr. Rafael Z. Frantz
UNIJUÍ, Brasil

Comitê Científico:

Dra. Fabricia Roos-Frantz
UNIJUÍ, Brasil

Dr. Rafael Zancan Frantz
UNIJUÍ, Brasil

Dra. Inma Hernandez
Universidad de Sevilla, Espanha

MsC. Rogério S. de Moura Martins
UNIJUÍ, Brasil

Dra. Iryna Yevseyeva
Newcastle University, Reino Unido

Dr. Sandro Sawicki
UNIJUÍ, Brasil

Dr. Michael Emmerich
University of Leiden, Holanda

Dr. Vítor Manuel Basto Fernandes
Instituto Politécnico de Leiria, Portugal

Dr. Rafael Corchuelo
Universidad de Sevilla, Espanha

Comitê de Organização:

Eldair Fabrício Dornelles
UNIJUÍ, Brasil

Eliezer Prigol
UNIJUÍ, Brasil

Igor Gamste Haugg
UNIJUÍ, Brasil

Modelagem Conceitual de uma Solução de Integração para o Processo de Rematrícula da Universidade Unijuí

Igor G. Haugg*, Rafael Z. Frantz

Universidade Regional do Noroeste do Estado do Rio Grande do Sul

Departamento de Ciências Exatas e Engenharias

Ijuí, RS, Brazil

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

Resumo—Integrar aplicações tem se tornado uma necessidade constante para as empresas, pois as aplicações utilizadas são normalmente desenvolvidas por diferentes empresas e seguindo diferentes tecnologias. Integrar aplicações é fazer com que aplicações que não foram desenvolvidas com o objetivo de integração, possam colaborar e dar suporte a novos processos de negócio. Neste trabalho será apresentado um caso de estudo baseado nos processos de rematrícula de uma universidade. Ao integrar as diferentes aplicações disponíveis será possível disponibilizar de forma automática uma lista de disciplinas que o aluno ainda não cursou, a partir das disciplinas que serão ofertadas no próximo semestre. Também será apresentado um modelo conceitual utilizando a tecnologia Guaraná DSL para representar a solução deste problema. A partir da criação do modelo conceitual, podemos verificar que a realização da integração tornaria o processo de escolha de disciplinas automatizado, facilitando a rematrícula dos alunos.

Palavras-chave: Integração de Aplicações Empresariais, Linguagem de Domínio Específico, Caso de Estudo.

I. INTRODUÇÃO

Frequentemente as empresas precisam integrar as aplicações existentes em seus ecossistemas de software. A integração é necessária para dar suporte aos novos processos de negócio, reutilizando funcionalidades presentes nas suas aplicações atuais ou para sincronizar duas ou mais aplicações dentro de seus ecossistemas de software. Normalmente essas empresas possuem aplicações que foram desenvolvidas de forma independente, personalizadas ou adquiridas de terceiros, sem que tenham sido 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 e dar suporte a processos de negócio [3]. Para modelar integrações de aplicações, é possível a utilização da tecnologia Guaraná DSL, a qual é uma linguagem de domínio específico utilizada para modelar soluções de integração de aplicações empresariais com alto nível de abstração [2].

A pesquisa desenvolvida neste trabalho buscou identificar uma forma de integrar aplicações presentes no processo de rematrículas na Universidade UNIJUI. A partir desta identificação, propõe-se um modelo conceitual, utilizando a linguagem Guaraná DSL, para representar a solução de integração. O



Figura 1. Tarefas Guaraná.

objetivo de integrar estas diferentes aplicações é conseguir de forma automática disponibilizar uma lista de disciplinas que o aluno ainda não cursou através de E-mail.

II. GUARANÁ DSL

O Guaraná DSL possibilita projetar soluções de integração utilizando sintaxe gráfica. Esta linguagem disponibiliza um conjunto de ferramentas, denominadas tarefas, que são utilizadas para modelar diferentes integrações de aplicações empresariais. Uma solução de integração modelada a partir desta linguagem, é formada por um conjunto de aplicações e tarefas que cooperam para integrar as diferentes aplicações. Dentro da solução de integração desenvolvida são transmitidas mensagens, as quais contém informações que são enviadas pelas aplicações [1]. Na Figura 1 é possível visualizar algumas tarefas presentes no Guaraná DSL.

A tarefa *Correlator* analisa mensagens de entrada e produz como saída um conjunto de dados correlacionados. *Filter* realiza um filtro nas mensagens que recebe de entrada. A tarefa *Content Enricher* adiciona conteúdos na mensagem. *Replicator* realiza cópias da mensagem e encaminha para uma ou mais tarefas. *Splitter* divide a mensagem em diversas partes. *Translator* realiza a tradução do conteúdo da mensagem para um novo formato [1].

III. CASO DE ESTUDO

A solução de integração proposta foi modelada baseando-se na necessidade de melhorar o atendimento aos alunos da Universidade UNIJUI. O objetivo desta solução de integração é disponibilizar automaticamente informações referentes as possibilidades de rematrículas de cada aluno, através da geração de uma lista contendo todas as disciplinas que o aluno ainda não cursou e que serão ofertadas no próximo semestre.

* Bolsista de Iniciação Tecnológica - PIBITI/CNPq

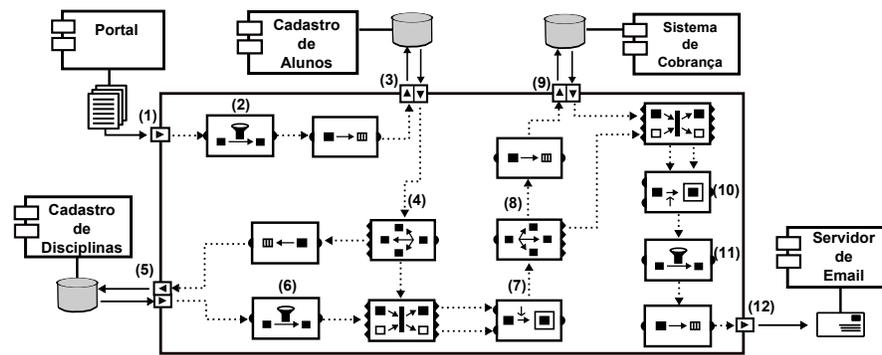


Figura 2. Solução de Integração.

A. Ecosystema

A integração de aplicações proposta envolve cinco aplicações: Portal, Cadastro de Alunos, Cadastro de Disciplinas, Sistema de Cobrança e Servidor de E-mail. Cada aplicação executa em plataformas diferentes, o Portal, as aplicações Cadastro de Alunos, Cadastro de Disciplinas e o Sistema de Cobrança são aplicações desenvolvidas pela própria universidade. Estas aplicações não foram desenvolvidas com o objetivo de serem integradas, por isso, para que possam colaborar e dar suporte ao processo de rematrículas, devem ser integradas por meio de uma solução de integração. O Servidor de E-mail oferece interfaces POP3 e SMTP. O Portal disponibiliza diversas informações para os alunos, como por exemplo, as disciplinas já cursadas, percentual do curso realizado. Também é onde ocorrem processos como as rematrículas, de forma que todos os semestres os alunos precisam solicitar através do Portal as disciplinas que desejam cursar no semestre seguinte. O Cadastro de Alunos disponibiliza um banco de dados com informações como, nome, RG de aluno e curso matriculado. Cadastro de Disciplinas disponibiliza um banco de dados que contém informações como, nome da disciplina, curso vinculado, semestre, turmas disponíveis e data. Quando um aluno for realizar sua matrícula ou rematrícula, através Portal, é possível através das informações adquiridas na aplicação Cadastro de Aluno, consultar na aplicação Cadastro de Disciplinas, quais disciplinas o aluno ainda não realizou e quais disciplinas serão ofertadas no próximo semestre. Com estas informações gerar uma lista de possíveis disciplinas a serem cursadas no próximo semestre. A partir da lista de disciplinas, é possível consultar o preço das disciplinas no Sistema de Cobrança, e gerar os preços das disciplinas contidas na lista de disciplinas. Esta lista pode ser enviada para o aluno por e-mail, utilizando o Servidor de E-mail.

B. Solução de Integração

A Figura 2 apresenta o modelo conceitual que representa a solução de integração das aplicações para este problema.

A solução toma informações do Portal por meio da porta (1), que lê mensagens contendo as solicitações dos alunos. A tarefa (2) filtra as mensagens recebidas, aceitando somente os pedidos de rematrícula que foram realizadas dentro do período

de rematrícula. Então as mensagens aceitas são encaminhadas para o Cadastro de Alunos através da porta (3), para obter informações referentes ao aluno que realizou a solicitação. A tarefa (4) replica as mensagens, uma cópia é enviada para a porta (5), na aplicação Cadastro de Disciplinas é adquirido a lista de disciplinas presentes no curso em que o aluno está matriculado. A tarefa (6) tem a função de descartar as disciplinas que o aluno já cursou e também as disciplinas que não serão ofertadas no semestre. A tarefa (7) enriquece a segunda cópia realizada pela tarefa (4), correlacionando com as informações retornadas pelo Cadastro de Disciplina. Em seguida, a tarefa (8) replica novamente a mensagem. Uma cópia é enviada para a porta (9), onde na aplicação Sistema de Cobrança será obtido o preço das disciplinas. A tarefa (10) enriquece a segunda cópia realizada pela tarefa (8), correlacionando com as informações recebidas pelo Sistema de Cobrança. As mensagens são encaminhadas primeiramente para a tarefa (11), a qual realiza um filtro para evitar que as mensagens não sejam enviadas para os alunos sem estarem completas. Em seguida, as mensagens são encaminhadas para a porta (12) que através da aplicação Servidor de E-mail encaminha para o aluno a lista de disciplinas com os preços.

IV. CONCLUSÃO

Com a realização deste trabalho foi possível aprender a utilizar o Guaraná DSL, o qual é uma ferramenta de simples entendimento e que auxilia muito na modelagem da solução de integração. Também analisamos que integrar aplicações pode trazer grandes benefícios para as empresas. No caso de estudo apresentado neste trabalho, a integração das aplicações permite que diversos processos sejam realizados de forma automática, tornando o processo de rematrícula mais simples e rápido.

REFERÊNCIAS

- [1] Rafael Z. Frantz. *Enterprise application integration: an easy-to-maintain model-driven engineering approach*. PhD thesis, Universidad de Sevilla, 2012.
- [2] Rafael Z. Frantz, Antonia M. Reina Quintero, and Rafael Corchuelo. A domain-specific language to design enterprise application integration solutions. *International Journal of Cooperative Information Systems*, 20(02):143–176, 2011.
- [3] Gregor Hohpe and Bobby Woolf. *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional, 2004.

Desenvolvimento de uma Solução de Integração de Aplicações para Automatizar Reservas de Viagem

Edinaldo G. da Silva, Fabricia Roos-Frantz
 Universidade Regional do Noroeste do Estado do Rio Grande do Sul
 Departamento de Ciências Exatas e Engenharias
 Ijuí, RS, Brazil
 edinaldogaspar@outlook.com, frfrantz@unijui.edu.br

Resumo—Atualmente o processo de Integração de Aplicações torna-se cada vez mais necessário no ecossistema de software de uma empresa. A criação de uma única ferramenta capaz de atender a todas as suas necessidades é inviável, sendo a Integração de Aplicações a melhor solução para que tais ferramentas compartilhem informações e até mesmo funcionalidades. A partir disso torna-se necessário estudar ferramentas capazes de realizar tais integrações e por consequência estudar o seu funcionamento e aplicar o conhecimento adquirido criando soluções de integração que atendam as mais diversas necessidades cotidianas.

Palavras-chave: Integração de Aplicações Empresariais; Linguagem de Domínio Específico.

I. INTRODUÇÃO

As aplicações empresarias existem há décadas e devido a heterogeneidade dos ecossistemas de software das empresas e as aplicações normalmente terem sido pensadas para funcionar de forma isolada, torna-se necessário o estudo de técnicas e ferramentas para integra-las. Além disso, a necessidade de integrar aplicações está relacionada ao fato de os processos de negócio de uma empresa evoluírem com o passar dos anos, exigindo assim a interação com novas aplicações do ecossistema de software. Pode-se citar como exemplo um software de folha de pagamento que precisa acessar os dados do sistema de vendas para saber quanto, em comissões, será anexado ao salário de determinado funcionário.

Uma solução de integração consiste basicamente em uma nova aplicação que deverá ser capaz de realizar a troca de dados ou funcionalidades entre duas ou mais aplicações que não foram concebidas para tal tarefa, muitas vezes nem mesmo foram desenvolvidas na mesma linguagem, fazendo com que elas trabalhem em conjunto [2]. Infelizmente integrar aplicações por vezes se torna difícil devido a um serie de questões. Por um lado há uma dificuldade na criação de uma solução de integração porque as aplicações a serem integradas não foram pensadas para compartilhar informações, podem ter sido desenvolvidas por diferentes empresas e ainda podem utilizar diferentes linguagens de programação. Por outro lado, após a integração as aplicações devem continuar funcionando de forma independente, sendo possível que sejam adicionadas a elas novas funcionalidades [1].

II. GUARANÁ

Dentre as diversas aplicações destinadas a integração de sistemas estão Guaraná, Spring Integration, Mule, Camel.

Nesse artigo utilizaremos Guaraná, já que o mesmo possui uma linguagem para criar modelos conceituais de soluções e uma API bastante simples que permite implementar e executar tais modelos. Utilizando Guaraná será apresentado um modelo de solução de integração criado a partir do Guaraná DSL. Tal DSL prove à engenheiros de software diversas ferramentas capazes de conceber e implementar soluções de integração a um custo razoável [1]. Além do Guaraná DSL que possibilita a criação de modelos gráficos, é válido também mencionar Guaraná ESB, que se trata de uma implementação Java do Guaraná DSL. Para que possamos usar o Guaraná ESB é preciso primeiramente escolher um ambiente de programação, por exemplo NetBeans e Eclipse, sendo este último o utilizado para criar a solução de integração que irá automatizar reservas de viagens. Após isto é importante notar que como o motor do Guaraná ESB lê arquivos XML, não necessariamente tais arquivos precisem ser escritos dentro do Eclipse, podendo ser usadas outras ferramentas, como por exemplo o NotePad++, o qual foi utilizado para criar o XML em que esta contida a implementação da solução ao problema proposto.

III. CASO DE ESTUDO

A. O ecossistema

O problema a seguir é real e tem por objetivo automatizar o processo de reserva passagens junto a uma companhia aérea e de hospedagem em um hotel, sendo que tal problema envolve quatro aplicações, a primeira delas é chamada Solicitação de Reserva, é a aplicação responsável por criar o arquivo de solicitação tanto da reserva de hotel quanto da passagem aérea, de acordo com as solicitações feitas pelo cliente. Em segundo lugar temos a aplicação de hotel que precisa receber a solicitação de reserva de um ou mais quartos criada na aplicação supracitada, da mesma forma que a aplicação três, Agência de Viagens, necessita receber a solicitação de reserva de passagens. Não obstante, ainda é necessário que seja encaminhado um email ao cliente confirmando as reservas e contendo os dados das mesmas. Para que tal problema seja resolvido torna-se necessário que seja pensada uma solução de integração.

B. Modelo Conceitual

A figura 1 representa uma possível solução para o problema relatado feita a partir do Guaraná DSL. O processo de inte-

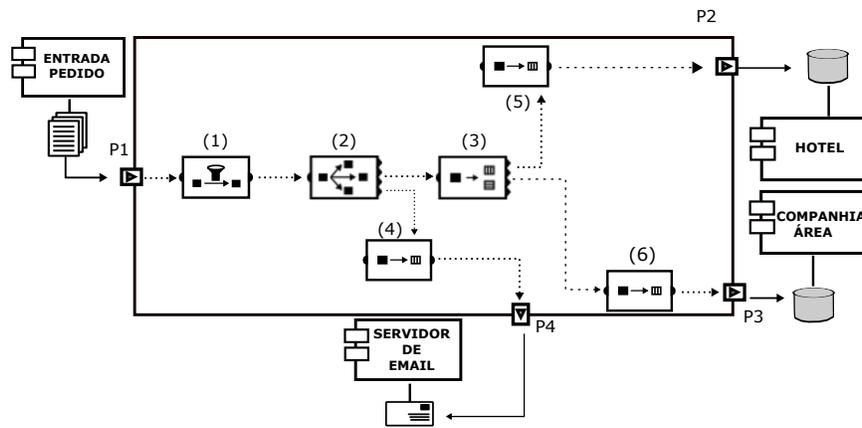


Figura 1. Modelo conceitual para a solução de integração.

gração começa na porta de entrada (P1) que encaminhará a mensagem para uma tarefa denominada Filter (1) que verificará se existem no arquivo tanto a solicitação de reserva de passagem como de hospedagem. Posteriormente a mensagem será encaminhada a outra tarefa denominada Replicator (2) que replicará a mensagem, encaminhando as cópias geradas para os seus respectivos destinos. No nosso caso de estudo serão geradas duas cópias, a primeira será encaminhada a tarefa denominada Chopper (3) e a segunda para uma tarefa chamada Translator (4). A tarefa Chopper (3) dividirá a cópia recebida em duas mensagens, é importante observar que a mensagem original possui duas partes, a primeira com a reserva de passagem e a segunda com a reserva de hospedagem, sendo que essas duas partes serão divididas pela Chopper (3). Após a divisão as mensagens resultantes serão encaminhadas as tarefas Translator (5) e (6) que converterão as mensagens para os formatos aceitos pelas aplicações destino. A segunda cópia gerada pela tarefa Replicator (2) será encaminhada a tarefa Translator (4), sendo posteriormente encaminhada ao email do cliente.

C. Implementação

Para que seja possível implementar uma solução de integração será utilizado o Guaraná ESB, uma implementação Java do Guaraná DSL que fornece suporte para a implementação de soluções de integração. A solução implementada utilizando o Guaraná ESB é composta por um arquivo XML, que representará de forma textual o modelo apresentado anteriormente. Tal modelo deverá ser escrito obedecendo um conjunto de regras, que diferem de acordo com o que se tenta fazer. A figura 2 é a implementação da porta P1 feita a partir do Guaraná ESB.

Em primeiro lugar é possível observar que é inicializada uma tag `<ports>`, sendo ambas as portas de entradas e saídas deverão estar inseridas dentro desta tag, ou seja, entre as tags `<ports>` e `</ports>`. Nota-se que o código de criação da porta de entrada P1, uma porta de entrada `<entry-port>`, possui diversas características, como por exemplo uma “ID” que identificará a porta. Outro ponto importante é o “uri” que mostrará o caminho do arquivo que será encaminhado

```
<ports>
  <entry-port id= "EntryPort1"
    uri="folder:///D:/Solicitacao/?include=*.xml"
    engine="testEngine"
    entry-pipeline="EntryPipeLine"
    slot= "input[0]"
    task= "Filter"/>
</ports>
```

Figura 2. Implementação da porta P1.

ao motor de integração do Guaraná. No exemplo da Figura 2 observa-se que a porta buscará arquivos com formato XML na pasta “Solicitacao”. Tais arquivos serão encaminhados a uma tarefa (task) chamada “Filter”, que verificará se ambas as reservas, passagem e hotel, foram solicitadas.

IV. CONCLUSÃO

Observa-se que Guaraná oferece uma solução completa, de simples entendimento e implementação a um custo razoável para resolver problemas de Integração de Aplicações como o proposto neste artigo, em que tornou-se possível integrar quatro aplicações fazendo com que um arquivo fosse compartilhado entre todas. Sendo este apenas um exemplo utilizado para apresentar o funcionamento do Guaraná, tendo em vista que esta ferramenta oferece recursos para implementar tanto pequenos como grandes problemas de integração. A partir disso o próximo passo é implementar uma solução com maior grau de complexidade, solução essa que já está sendo modelada e será apresentada em trabalhos futuros.

REFERÊNCIAS

- [1] Rafael Z. Frantz, Antonia M. Reina Quintero, and Rafael Corchuelo. A domain-specific language to design enterprise application integration solutions. *International Journal of Cooperative Information Systems*, 20(02):143–176, 2011.
- [2] Gregor Hohpe and Bobby Woolf. *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional, 2004.

Simulação do Comportamento de uma Solução de Integração Desenvolvida para a Área de Venda Hospitalar Utilizando a Ferramenta Arena

Gabriel Freytag*, Sandro Sawicki

Universidade Regional do Noroeste do Estado do Rio Grande do Sul

Departamento de Ciências Exatas e Engenharias

Ijuí, RS, Brazil

{gabriel.freytag, sawicki}@unijui.edu.br

Resumo—Empresas tipicamente possuem aplicações heterogêneas que geralmente não foram projetadas para compartilhar dados e processos. A Integração de Aplicações Empresariais (EAI) permite que essas aplicações compartilhem dados e processos entre si. Ferramentas de integração, como a Guaraná, facilitam essa tarefa. Entretanto, não permitem analisar o comportamento da solução, nem identificar otimizações. Para tornar isso possível, a técnica de Simulação pode ser empregada, pois permite analisar o comportamento de sistemas de forma virtual. Portanto, o objetivo deste trabalho é projetar uma solução de integração, criar um modelo de simulação equivalente e simulá-lo.

Palavras-chave: integração de Aplicações Empresariais, Simulação de Eventos Discretos, Tecnologia Guaraná, Software Arena

I. INTRODUÇÃO

Atualmente, empresas são compostas por centenas, senão milhares de aplicações [5]. Aplicações de diferentes gerações e fornecedores, desenvolvidas com inúmeras linguagens de programação e operando em várias plataformas. Essa heterogeneidade impede que dados e processos sejam facilmente compartilhados. EAI fornece metodologias e ferramentas para integrar aplicações heterogêneas. Para facilitar a integração de aplicações, surgiu a tecnologia Guaraná [3].

Guaraná permite criar soluções de integração de aplicações heterogêneas com um alto nível de abstração, porém, não permite visualizar o comportamento das soluções. Então, de que forma é possível visualizar seu comportamento? Utilizando a técnica de simulação, que permite modelar processos e sistemas de modo que o modelo reproduza o comportamento do sistema real para eventos que ocorrem ao longo do tempo [6]. Um dos simuladores mais populares é o Arena [9], que segundo Prado [2], é um *software* de simulação de eventos discretos que permite descrever aplicações reais.

Este trabalho propõe solucionar um problema de integração utilizando a tecnologia Guaraná, criar um modelo de simulação dessa solução e realizar simulações para identificar possíveis otimizações da solução de integração.

* Bolsista de Iniciação Científica

II. ENTERPRISE APPLICATION INTEGRATION

De acordo com Linthicum [8], EAI é um processo informal que surgiu como uma resposta a décadas de criação de aplicações de propósito específico e o resultado direto da falta de previsão arquitetônica. A maioria das aplicações existentes no ecossistema de *software* das empresas foi concebido sem ter em mente conceitos de integração. A grande maioria das empresas utiliza várias gerações de sistemas que dependem de uma ampla gama de tecnologias desenvolvidas ao longo de muitos anos [8]. EAI fornece eficiência, confiabilidade e segurança na troca de dados entre várias aplicações empresariais [5] e está no topo da lista de prioridades gerentes de TI [1].

Ferramentas como o Guaraná surgiram para facilitar a integração de aplicações. Guaraná é um projeto que objetiva fornecer ferramentas que possam ser úteis para desenvolver e implementar soluções de EAI a custos aceitáveis [4]. Possui uma linguagem DSL abstrata, um editor gráfico e um SDK para transformar os modelos em código executável.

III. SIMULAÇÃO

Simulação consiste na modelagem de um processo ou sistema de modo que o modelo reproduza as respostas do sistema real a partir de eventos que ocorrem ao longo do tempo [6]. A utilização de modelos de simulação para projetar novos sistemas e aperfeiçoar o desempenho dos sistemas existentes continua crescendo devido ao aumento da complexidade dos sistemas, redução dos custos, melhoria nos *softwares* de simulação e da disponibilidade de animação [7].

Um dos simuladores mais populares e respeitado é o Arena. Arena possui um conjunto de blocos usados para descrever aplicações reais que funcionam como comandos de uma linguagem de programação, porém projetados sob a ótica da simulação, facilitando muito a tarefa de programar [2]. Além disso, possui uma interface gráfica, que facilita a modelagem, além de analisadores de dados de entrada e de resultados.

IV. PROBLEMA EAI

Em uma empresa que presta assistência técnica, venda e aluguel de aparelhos hospitalares, existem diversas aplicações que dão suporte aos negócios da empresa. Atualmente, a

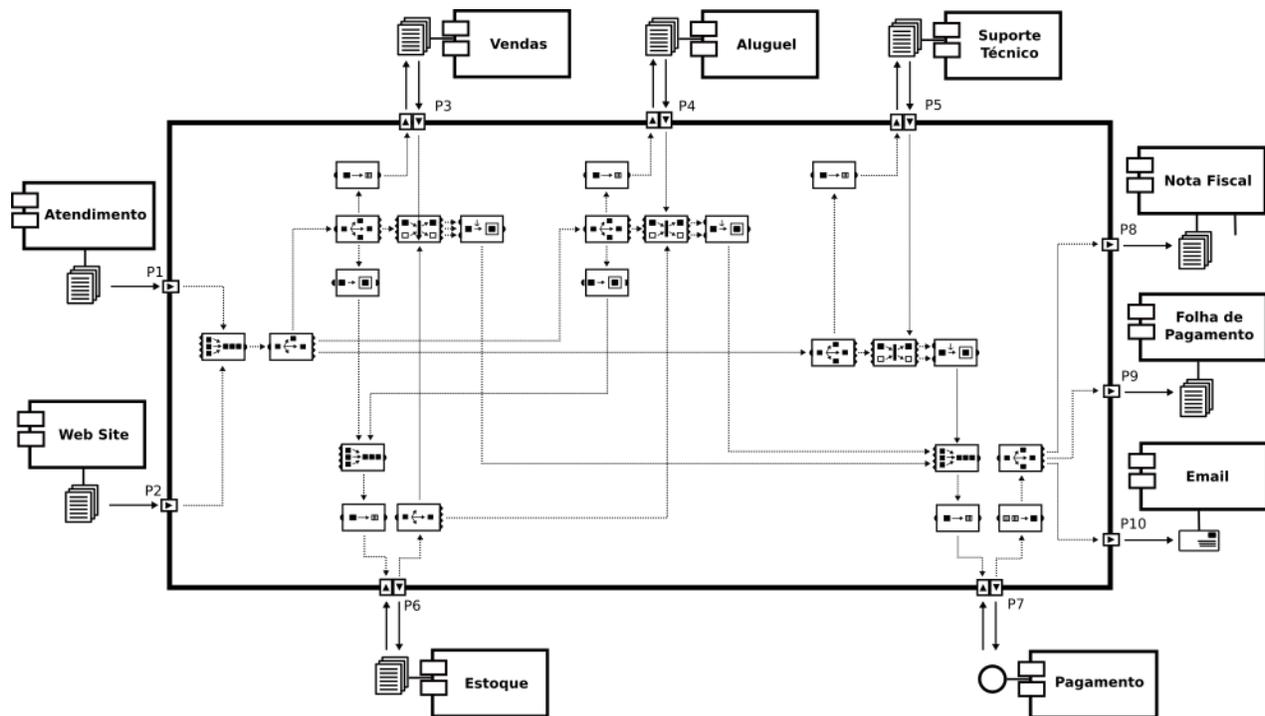


Figura 1. Solução do problema de integração

integração entre as aplicações é realizada manualmente pelos funcionários da empresa. Quando um cliente contata a central de atendimentos, por meio telefônico ou e-mail para efetuar a compra de produtos, o atendente verifica a disponibilidade dos produtos na aplicação de estoque. Se disponíveis, utiliza a aplicação de vendas para dar continuidade e, caso contrário, os produtos são encomendados por outro setor e enviados posteriormente. Atendentes recebem bonificação em sua folha de pagamento por cada venda através de um código gerado na venda e inserido pelo gerente na aplicação de folha de pagamento. A compra também pode ser realizada por meio do portal web, onde o cliente escolhe os produtos, realiza a identificação, seleciona o nome do atendente que possivelmente lhe auxiliou na compra e, então, seleciona a forma de pagamento. Além da compra, clientes podem alugar aparelhos ou solicitar suporte técnico. Para o aluguel, o cliente contata a central de atendimento, que verifica a disponibilidade na aplicação de estoque e cadastra o empréstimo na aplicação de aluguel. Já para solicitar suporte técnico, o cliente contata um atendente, que abre um novo chamado na aplicação de suporte técnico e agenda uma visita. Em ambos os casos, ao final da operação, o atendente finaliza o pagamento das solicitações, emite a nota fiscal na respectiva aplicação e envia uma cópia digital por e-mail. A Figura 1 é a solução gerada no Guarani.

V. CONCLUSÃO

A integração de aplicações permite reutilizar aplicações legadas e integrar aplicações heterogêneas, ao passo que a simulação computacional permite testar o comportamento de sistemas para determinados eventos de forma digital. Unindo

seus benefícios de ambos, é possível diminuir ainda mais os custos de integração e otimizar as soluções antes mesmo da implantação. Os próximos passos do trabalho são a modelagem do problema descrito na Figura 1, a simulação e a identificação de possíveis otimizações da solução.

REFERÊNCIAS

- [1] Kishore Channabasavaiah, Kerrie Holley, and Edward Tuggle. Migrating to a service-oriented architecture. *IBM DeveloperWorks*, 16, 2003.
- [2] Darcy Santos do Prado. Usando o arena em simulação. *Série Pesquisa Operacional*, 3(4), 2010.
- [3] Rafael Z. Frantz. *Enterprise application integration: an easy-to-maintain model-driven engineering approach*. PhD thesis, University of Seville, 2012.
- [4] Rafael Z. Frantz, Antonia M Reina Quintero, and Rafael Corchuelo. A domain-specific language to design enterprise application integration solutions. *International Journal of Cooperative Information Systems*, 20(02):143–176, 2011.
- [5] Gregor Hohpe and Bobby Woolf. *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional, 2004.
- [6] W David Kelton, Randall P Sadowski, and Deborah A Sadowski. *Simulation with ARENA*, volume 3. McGraw-Hill New York, 2002.
- [7] Averill M Law and Michael G McComas. Secrets of successful simulation studies. In *Proceedings of the 23rd conference on Winter simulation*, pages 21–27. IEEE Computer Society, 1991.
- [8] David S Linthicum. *Enterprise application integration*. Addison-Wesley Professional, 2000.
- [9] Rockwell Software Inc. *Arena: user's guide*, arena-um001b-en-p edition, Outubro 2005.

Modelagem Matemática e Computacional para Identificação de Gargalos de Desempenho em Soluções de Integração

Arléte Kelm Wiesner*, Rafael Z. Frantz

Universidade Regional do Noroeste do Estado do Rio Grande do Sul

Departamento de Ciências Exatas e Engenharias

Ijuí, RS, Brasil

arlete.kelm@gmail.com, rzfrantz@unijui.edu.br

Resumo—Normalmente as empresas possuem várias aplicações para apoiar seus processos de negócios. Estas aplicações compõem o ecossistema de software da empresa que geralmente é heterogêneo. No entanto, a maioria das aplicações foram projetadas sem a possibilidade de serem reutilizadas, necessitando de uma solução de integração, que tem como objetivo sincronizar as aplicações sem alterá-las. A análise do comportamento e a identificação de possíveis gargalos de desempenho é uma importante atividade que contribui para aumentar a qualidade das soluções entregues. Os engenheiros de software, seguem normalmente uma abordagem que requer a construção e a execução da solução de integração para analisar e prever seu comportamento. Este trabalho aborda a possibilidade de simulação dos modelos conceituais da solução de integração, ainda na fase de projeto, utilizando como base a Teoria das Filas. Com esta abordagem, o trabalho que está sendo desenvolvido busca contribuir para a redução de custos, riscos e tempo de desenvolvimento de soluções de integração.

Palavras-chave: Integração de Aplicações Empresariais; Gargalos de Desempenho; Simulação; Teoria das Filas.

I. INTRODUÇÃO

No cenário atual de negócios as empresas adquirem ou desenvolvem aplicações para apoiar a tomada de decisões e aperfeiçoar seus processos. Estas aplicações compõem o seu ecossistema de software [5], que geralmente é heterogêneo, ou seja, foram desenvolvidas com tecnologias diferentes. A ausência de integração normalmente acaba gerando redundância de dados como também a ausência. Assim o conceito de Integração de Aplicações Empresariais - do inglês *Enterprise Application Integration* (EAI), está relacionado na elaboração de soluções que resolvam estes problemas. O campo de estudos conhecido como EAI tem tido um papel muito importante a medida que proporciona metodologias, técnicas e ferramentas para que as empresas consigam desenvolver soluções de integração.

O sucesso das empresas em seus processos de negócios atualmente depende da execução correta e eficiente da solução de integração. A análise do comportamento de soluções de integração para encontrar possíveis gargalos de desempenho é considerada uma importante atividade para melhorar a qualidade das soluções construídas.

* Mestranda do PPGMM

Normalmente, a abordagem adotada pelos engenheiros de software para realizar testes frente a cenários críticos de funcionamento e recolher dados consiste na construção e execução da solução de integração. Nesse sentido, a abordagem que permite analisar o comportamento e identificar possíveis gargalos de desempenho, ainda na fase de projeto a partir do modelo conceitual, poderá reduzir custos, riscos e tempo de desenvolvimento. No trabalho que está sendo desenvolvido, pretende-se estudar um modelo conceitual de uma solução de integração projetada na tecnologia Guaraná com o objetivo de analisar o comportamento e identificar possíveis gargalos de desempenho de uma solução frente a diferentes cargas de trabalho.

Neste contexto, uma solução de integração de aplicações empresariais pode ser classificada como um sistema estocástico, dinâmico e discreto [6]. Uma solução de integração pode ser classificada como um sistema discreto, de forma que uma mensagem, elemento envolvido em uma solução de integração, mantém seu estado inalterado durante intervalos de tempo. Assim, o estado da solução de integração irá mudar quando ocorrer o evento de chegada da mensagem na tarefa, processamento e saída. A ocorrência destes eventos, provocam alterações na variável tempo e, conseqüentemente mudam o estado do sistema.

Como uma solução de integração é um sistema discreto, o modelo conceitual projetado pode ser simulado. Para a simulação de eventos discretos temos atualmente técnicas e ferramentas consolidadas. Este trabalho aborda a simulação, utilizando como base a Teoria das Filas por meio da ferramenta Arena. Isso permite que os engenheiros de software possam descobrir possíveis problemas antes da implementação e execução da solução que está sendo construída.

II. TRABALHOS RELACIONADOS

Abensur e outros [1] propõe no trabalho oferecer condições para a formulação de estratégias consistentes para o autoatendimento bancário brasileiro a partir da avaliação das variáveis de desempenho obtidas pela aplicação da teoria das filas sobre o caso estudado.

As simulações foram realizadas para dois grupos distintos de usuários, com alta e baixa habilidade de uso de caixas eletrônicos, considerando o tempo médio de ocupação dos equipamentos. Os autores desenvolveram um simulador de filas de autoatendimento em Visual Basic para Excel 2002 e os resultados mostraram que os ganhos de produtividade com a introdução de máquinas especializadas (poucas transações disponíveis) são significativos, havendo uma melhor prestação dos serviços oferecidos aos clientes com sensíveis reduções do tempo médio de fila.

Changfu e Zhenyu [2], buscam no estudo desenvolvido estabelecer modelos matemáticos e de simulação para analisar e avaliar o impacto da eficiência provocado pela variação de transações de requisição no sistema de filas do ambiente e-Business, baseado na teoria das filas. Através da manipulação matemática das fórmulas da teoria das filas, os autores afirmam que, o ajuste estrutural na manipulação do pedido de transação no sistema de filas no ambiente de e-Business pode esgotar o tempo médio de espera. Indicam também que a simplificação da documentação de entrada e saída, pode diminuir o tempo médio de espera e com isso espera-se diminuir o tempo de espera significativamente.

Baseados nos dados estatísticos de reserva de uma companhia de agenciamento marítimo, os autores estabeleceram modelos de simulação do sistema de filas por meio do MATLAB e Excel para a reserva de papel e sistemas de filas de reserva eletrônica, considerando e não considerando o tempo de documentação de entrada e saída. A conclusão teórica e os resultados da simulação mostram que o ajuste estrutural no sistema de filas no ambiente e-Business pode reduzir o tempo médio de espera do cliente, juntamente com a simplicidade de entrada e saída de documentos, e em seguida a experiência de serviço ao cliente pode ser aperfeiçoada.

III. TEORIA DAS FILAS

O modelo conceitual da solução de integração desenvolvido com a tecnologia Guaraná se assemelha ao modelo matemático de simulação utilizando como base a Teoria das Filas. Identificou-se uma equivalência entre: Filas de espera e *slot*, cliente e mensagem, servidor e tarefa [6].

Segundo Hillier e Lieberman [4], a Teoria das Filas é um método analítico que estuda a formação de filas por meio de fórmulas matemáticas e usa modelos de filas para representar os diversos tipos de sistemas de filas que surgem na prática, sendo que um sistema de filas pode ser representado por diferentes modelos, tendo elementos característicos comuns a todos que fazem parte do processo básico.

Segundo Fogliatti e Mattos [3], a utilização da Teoria das Filas permite avaliar a eficiência de um sistema por meio da análise de suas características representadas por variáveis randômicas cujos valores são utilizados como medidas de desempenho do sistema. As variáveis em relação ao sistema, processo de chegada, fila e ao processo de atendimento são respectivamente: TS - tempo médio de permanência no sistema, NS- número médio de clientes no sistema, λ - ritmo médio de chegada, IC - intervalo médio entre chegadas, TF - tempo

médio de permanência na fila, NF - número médio de clientes na fila, TA - tempo médio de atendimento e μ - ritmo médio de atendimento de cada atendente.

Os modelos de filas são obtidos de acordo com o comportamento do sistema real. A estrutura mais simples que é encontrada é aquela que existe fila única e servidor único. Quando conhecida a distribuição de probabilidade que descreve o número de clientes no sistema M/M/1 podemos determinar os valores das seguintes variáveis randômicas: número médio de clientes na fila, número médio de clientes no sistema, tempo médio durante o qual o cliente fica na fila, tempo médio durante o qual o cliente fica no sistema.

IV. CONCLUSÃO

Uma solução de integração visa reutilizar as aplicações, manter em sincronia os dados e as funcionalidades ou produzir novas funcionalidades a partir das existentes. As empresas dependem da execução correta da solução para a tomada de decisões, a coordenação e o controle de seus processos de negócios.

O desenvolvimento da solução passa pelos processos da engenharia de software e geralmente a análise do comportamento e a identificação de gargalos de desempenho em soluções de integração requer sua construção e execução. Uma solução pode ser classificada como um sistema discreto, assim, os modelos conceituais projetados podem ser simulados, e os engenheiros de software podem analisar as soluções antes de sua construção, ainda na fase de projeto, reduzindo custos, riscos e tempo de desenvolvimento.

A Teoria das Filas permite calcular e estimar resultados relacionados à performance dos sistemas, com base em propriedades mensuráveis. Elementos da estrutura básica de um sistema de filas são semelhantes aos elementos de uma solução de integração de aplicações empresariais. Neste contexto, o nosso trabalho tem como propósito transformar o modelo conceitual projetado na tecnologia Guaraná, em um modelo de simulação equivalente utilizando a ferramenta Arena e estudar as medidas de desempenho da solução por meio da Teoria das Filas.

REFERÊNCIAS

- [1] Eder Oliveira Abensur, Israel Brunstein, Adalberto Fishmann, and LINDA LEE HO. Tendências para o auto-atendimento bancário brasileiro: um enfoque estratégico baseado na teoria das filas. *Revista de Administração Mackenzie*, 4(2), 2008.
- [2] Liu Changfu and Liu Zhenyu. Research of transaction request handling queueing system in the e-business environment based on queuing theory. In *Information Processing, 2009. APCIP 2009. Asia-Pacific Conference on*, volume 2, pages 589–592, 2009.
- [3] Maria Cristina Fogliatti and Neli Maria Costa Mattos. Teoria de filas. *Rio de Janeiro: Interciência*, pages 1–20, 2007.
- [4] Frederick S Hillier and Gerald J Lieberman. *Introdução à pesquisa operacional*. McGraw Hill, 2010.
- [5] David G Messerschmitt and Clemens Szyperski. *Software ecosystem: understanding an indispensable technology and industry*. MIT Press Books, 1, 2005.
- [6] Sandro Sawicki, Rafael Z Frantz, Fabricia Roos-Frantz, Rafael Corchuelo, and Yevseyeva Iryna. *Characterising Enterprise Application Integration Solutions as Discrete-Event System*.

Comparação entre Multilayer Perceptron e Redes Neurais Lógicas no problema Presa-Predador usando Algoritmos Genéticos

Márcia da Silva, Rogério Martins

Universidade Regional do Noroeste do Estado do Rio Grande do Sul

Departamento de Ciências Exatas e Engenharias

Ijuí, RS, Brasil

marcia.dasilva, rogerio.martins@unijui.edu.br

Resumo—As redes Perceptron tem atingido bons resultados em problemas supervisionados. Por este motivo, esta estrutura de rede neural passou a ser utilizada também em computação evolutiva. Porém este uso está acontecendo sem a realização de uma validação para verificar se realmente esta estrutura é a mais indicada para se trabalhar com algoritmos genéticos. As redes neurais do tipo Multilayer Perceptron são redes que possuem estruturas genéricas para resolver qualquer tipo de problema, isto significa que a otimização destas redes é mais lenta do que se utilizarmos redes neurais específicas. Neste trabalho, realizamos duas implementações de redes neurais, uma rede Multilayer Perceptron e outra rede neural híbrida, onde modificamos a estrutura desta rede para que seja possível a utilização de funções lógicas através de neurônios McCulloch, com o objetivo de reduzir o espaço de busca da rede e por consequência reduzir o tempo de evolução da rede. As implementações foram aplicadas no problema presa-predador, e foi realizado uma comparação entre os resultados obtidos. A partir dos testes iniciais realizados, foi possível perceber que a nova rede neural proposta possui resultados melhores, realizando a evolução de forma mais rápida.

Palavras-chaves: Redes Neurais Artificiais, MultiLayer Perceptron, Funções Lógicas.

I. INTRODUÇÃO

Redes neurais artificiais são modelos computacionais inspirados no sistema nervoso de seres vivos. Possuem a capacidade de aquisição e manutenção do conhecimento, e podem ser definidas como um conjunto de unidades de processamento [1].

Atualmente grande parte das implementações que utilizam redes neurais, usam como estrutura o Multilayer Perceptron (MLP). Esse tipo de rede utiliza o neurônio Perceptron, que trabalha com valores reais. Assim durante a evolução da rede, o espaço de busca é maior, e como consequência aumenta o tempo de evolução da rede, pois existem infinitas possibilidades e a rede precisa realizar a evolução até ajustar os pesos para processar a saída desejada.

Neste artigo será proposta uma alteração na estrutura da rede MLP. Para isto será necessário alterar a estrutura da rede adicionando camadas com neurônios McCulloch-Pitts, para que seja possível realizar funções lógicas. Assim foi desenvolvida uma rede híbrida, a qual é composta por neurônios perceptron e McCulloch-Pitts com o objetivo de reduzir o espaço de busca da rede e tornar a sua evolução mais rápida.

II. REFERENCIAL TEÓRICO

Neste trabalho foram estudados os conceitos de redes neurais e o problema presa-predador.

A. Redes Neurais Artificiais

Redes neurais artificiais são modelos computacionais inspirados no sistema nervoso de seres vivos. Possuem a capacidade de aquisição e manutenção do conhecimento (baseado em informações) e podem ser definidas como um conjunto de unidades de processamento, caracterizadas por neurônios artificiais, que são interligados por um grande número de interconexões (sinapses artificiais), sendo as mesmas representadas por vetores/matrizes de pesos sinápticos [1].

1) *Neurônio Biológico:* O cérebro é complexo, não-linear e computa dados paralelamente. Ele possui a habilidade de executar tarefas como padrões de reconhecimento, percepção e controle da atividade motora muito mais rápido que qualquer computador. Em adição a essas características, outras tais como a habilidade de aprender e memorizar. O principal componente de uma rede neural biológica é o neurônio. Ele é composto por um corpo celular, dendritos e um axônio. Um neurônio é totalmente interconectado. Essa interconexão acontece entre o axônio de um neurônio e o dendrito de outro. Esta interconexão tem o nome de sinapse. Os sinais propagam dos dendritos, através do corpo celular para o axônio, e a partir do axônio os sinais são propagados para todos os dendritos conectados. Um sinal é transmitido de um axônio do neurônio apenas quando a célula dispara [2].

2) *Neurônio Artificial:* Assim como as redes neurais artificiais foram criadas com base na estrutura de uma rede de neurônios biológicas, este neurônio também foi projetado utilizando a mesma ideia. Ele foi desenvolvido a partir de modelos conhecidos de sistemas nervosos biológicos, inclusive do cérebro humano. Os neurônios utilizados nos modelos de redes neurais artificiais são não-lineares, fornecem saídas tipicamente contínuas, e realizam funções simples, como coletar os sinais existentes em suas entradas, agregá-los de acordo com sua função operacional e produzir uma resposta, levando em consideração sua função de ativação inerente [1].

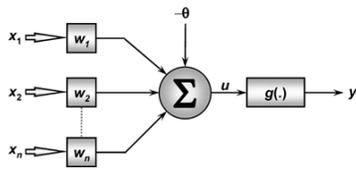


Figura 1. Representação de um neurônio artificial [1].

Na Figura 1 é possível ver uma representação gráfica de um neurônio artificial. Cada neurônio recebe sinais do ambiente, ou de outros neurônios. Após reunir estes sinais o neurônio realiza o somatório de todos os pesos e entradas e com este resultado aplica a função de ativação para produzir uma saída que é transmitida para os neurônios da camada seguinte [2].

B. Problema Presa Predador

Este problema foi proposto por Miroslav Benda em 1992. Nele há um plano de coordenadas x e y finito, um personagem chamado presa e quatro personagens denominados predadores. Os personagens podem se movimentar no sentido vertical e horizontal, não podendo ocupar a mesma posição. Os predadores tem como objetivo cercar a presa, de uma forma que a mesma não possa fazer nenhum movimento [3].

O problema Presa-Predador é um ambiente de simulação que pode ser utilizado em inteligência artificial em diferentes algoritmos para testar a qualidade dos agentes inteligentes criados, ou seja, quanto mais rápido os predadores cercarem a presa, mais eficiente será o agente que os coordenaram [3].

III. ESTRUTURA DA REDE

Nesta seção apresentamos a nova estrutura proposta para a rede neural a qual denominamos Rede Neural Lógica (RNL). A sua estrutura pode ser visualizada na Figura 3. É possível notar que a camada de limiar possui o funcionamento similar a da rede MLP (Figura 2). Ambas camadas realizam operações com números reais. A camada de limiar é responsável por fazer a separação dos elementos, ou seja, ela prepara as informações e através da sua função de ativação propaga o sinal 1 ou 0 para a camada E. Dependendo da saída dos neurônios da camada Limiar, será ativado ou não os neurônios da camada E. Na camada E, todos os neurônios são do tipo McCulloch-Pitts, onde cada neurônio será responsável por realizar um conjunto de testes lógicos do tipo E, e a camada terá a quantidade de neurônios necessária para satisfazer todas as possibilidades do problema. Após a informação ser processada pela camada E, o resultado será enviado para a camada OU. Nesta camada, se qualquer um dos neurônios da camada E propagar o resultado 1, o neurônio OU irá responder 1.

IV. RESULTADOS

A partir da implementação, executamos no problema Presa-Predador as duas redes neurais. Na Figura 4 é possível visualizar um gráfico comparativo entre o tempo de execução dos testes. Ao realizar alguns testes é possível entender que a Rede Neural Lógica sempre evolui mais rápido em relação a rede MLP, se mantém constante e obtém bons resultados. Para

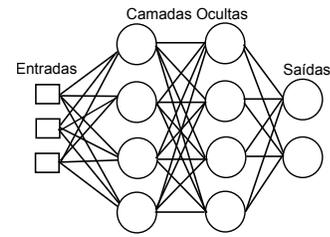


Figura 2. Rede Neural Multilayer Perceptron.

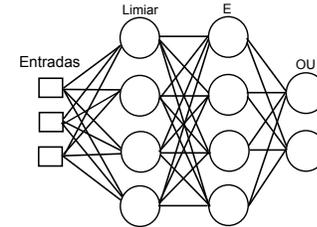


Figura 3. Rede Neural Lógica.

o problema Presa-Predador, a RNL consegue atingir o objetivo que é capturar a presa. Esta análise é baseada em um pequeno conjunto de testes, sendo necessário realizar mais rodadas de execuções para obter um conjunto maior de dados.

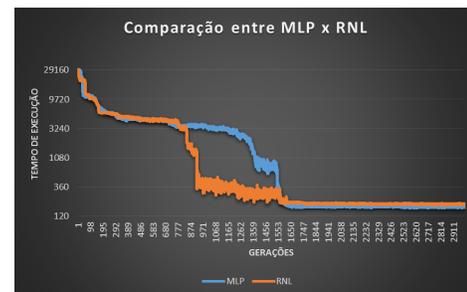


Figura 4. Comparação entre RNL e MLP.

V. CONCLUSÃO

Com a realização deste trabalho é possível notar que a Rede Neural Lógica demonstrou uma evolução superior em relação a rede MLP quando testadas para no problema Presa-Predador. O próximo passo é realizar mais testes desta estrutura em outros problemas.

AGRADECIMENTOS

Agradeço ao meu professor orientador Rogério Martins pelo auxílio no desenvolvimento do trabalho e ao grupo GCA.

REFERÊNCIAS

[1] Ivan Nunes da Silva, Danilo Hernan Spatti, and Rogério Andrade Flauzino. Redes neurais artificiais para engenharia e ciências aplicadas curso prático. *Artliber*, 2010.
 [2] Andries P. Engelbrecht. *Computational intelligence: an introduction*. John Wiley & Sons, 2007.
 [3] Richard E. Korf. A simple solution to pursuit games. Working papers of the 11th international workshop on distributed artificial intelligence, 1992.

Revisão da Literatura de Simulação de Sistemas de Eventos Discretos com Foco na Integração de Aplicações

Roberto Saulo Cargnin*, Fabricia Roos-Frantz

*Bolsista Mestrando do PPGMM

Universidade Regional do Noroeste do Estado do Rio Grande do Sul

Departamento de Ciências Exatas e Engenharias

Ijuí, RS, Brasil

cargnin7@gmail.com; frfrantz@unijui.edu.br

Resumo—O presente trabalho relata o estudo bibliográfico sobre a aplicação das redes de Petri para elaboração de modelos de simulação de sistemas de eventos discretos. Diante da importância da integração de aplicações para as corporações, estuda-se a possibilidade de aplicar as Redes de Petri para construção de modelos de simulação de soluções de integração, baseados em seus modelos conceituais com a perspectiva de analisar seu funcionamento sem a necessidade de construí-las.

Palavras-chave: Integração de Aplicações; Simulação; Redes de Petri

I. INTRODUÇÃO

O mercado corporativo está cada vez mais dependente dos aplicativos devido a sua eficiência de processamento de informações, suporte seguro e confiável aos processos de negocio. Frequentemente as empresas se deparam com a necessidade de adicionar novas funcionalidades as suas aplicações, pressionadas a acompanhar as evoluções ou perder espaços para seus concorrentes. Os aplicativos empresariais geralmente são desenvolvidos sem a preocupação de adicionar novas funcionalidades ou ainda nem permitem alterações em suas estruturas. Desta maneira a alternativa mais aplicada é desenvolver uma nova aplicação que contenha os novos recursos desejados e integra-la ao sistema em funcionamento da empresa, também denominado de sistema legado.

A dificuldade de integrar aplicações ao sistema legado esta em estabelecer a troca de informações sem interferir no funcionamento individual, manter seus dados sincronizados, adequar a incompatibilidade de formatos de arquivos compartilhados e arquiteturas desenvolvidas em diferentes tecnologias. O desenvolvimento de uma solução para a de integração de aplicações segue requisitos específicos do sistema pelo qual fará parte e por isso cresce o risco de haver Bugs (erros) em sua estrutura, podendo ainda apresentar gargalos de desempenho nos seus componentes que podem vir a comprometer seu correto funcionamento [5].

Enterprise Application Integration (EAI) propõe uma metodologia diferenciada de integração de aplicações, adicionar um terceiro componente, um servidor que terá uma solução de integração implementada e será responsável por gerenciar a troca de informação entre as aplicações envolvidas na integração com a proposta de integra-las sem gerar dependência

entre elas. As diferentes maneiras de integrar de acordo com as definições das EAI foram divididas em estilos e documentadas por Hohpe e Woolf [7].

As tecnologias atuais que desenvolvem Modelos conceituais de soluções de integração são baseadas no estilo de sistema de mensagens, possuem boa capacidade de adequação a sistemas heterogêneos e são ideais para ambientes que exigem flexibilidade [7].

Há várias tecnologias disponíveis baseadas em mensagem que possibilitam projetar modelos conceituais de soluções de integração, entre elas podemos citar o Camel, Mule, Spring e Guaraná. Cada tecnologia tem sua linguagem de domínio específico (DSL) para interpretação da solução de integração, dentro do contexto que cada uma trata. A tecnologia de desenvolvimento de modelos conceituais de solução abordada será o Guaraná, tecnologia que possibilita desenvolver modelos conceituais de soluções de integração de aplicações empresariais mantendo nível elevado de abstração por meio de uma DSL gráfica, representa todos os elementos de uma solução de integração [6].

Os modelos conceituais desenvolvidos no Guaraná são caracterizados pelo seu alto nível de detalhamento de soluções de integração, podem ser classificados como estocásticos, dinâmicos e discretos. Um sistema discreto pode ser traduzido para um modelo de simulação, simulado em ambientes específicos e obter respostas sobre a operação do sistema e analisar vários aspectos do mesmo[4].

As Redes de Petri são ferramentas matemáticas gráficas que oferecem um ambiente propício para modelagem, análise e simulação de sistemas de eventos discretos de forma a possibilitar a visualização estrutural e comportamental de forma simultânea.

II. PROPOSTA

A análise do comportamento e a identificação problemas em soluções de integração de aplicações geralmente envolve sua implementação para posterior execução e teste frente a cenários críticos projetados pelos engenheiros de software. Por demandar que as soluções sejam construídas, tal abordagem

traz consigo custos (tempo, recurso) que costumam ser elevados [8, 7].

Neste trabalho, é proposto um estudo bibliográfico sobre o redes de Petri estocásticas e sua aplicação para modelar sistemas de eventos discretos com objetivo de aplica-las como ferramenta para construção de modelos de simulação de soluções de integração, com o propósito de possibilitar a análise de comportamento com base nos seus modelos conceituais.

III. REDES DE PETRI ESTOCÁSTICAS

Redes de Petri são grafos bipartidos formados por dois tipos de nós chamados de lugares e transições, equivalem respectivamente as variáveis de estado de um sistema e às ações realizadas pelo mesmo. Lugares e transições são ligados por arcos, que ligam transições a lugares.

As Redes de Petri estocásticas tem como sua principal característica a associação de um atraso fixo para cada transição do modelo, determinadas por uma função de densidade de probabilidade exponencial negativa [2].

O conceito de tempo aliado ao formalismo de Redes de Petri permite a descrição de um comportamento dinâmico de um sistema. com a determinação de tempos aos eventos que ocorrem no sistema, é possível avaliar seu funcionamento, identificar possíveis problemas e simular alterações [9, 3].

IV. TRABALHOS RELACIONADOS

As aplicações de Redes de Petri para modelagem e simulação de sistemas de eventos discretos se expandem para diversas áreas devido a sua versatilidade de modelagem de situações. As Redes de Petri possibilitam um estudo detalhado do comportamento de sistemas, identificação de problemas e propostas de melhorias. Na sequência será abordado trabalhos que relatam o uso das Redes de Petri para simulação de sistemas de eventos discretos.

Em Yamada e outros [10] os autores propõem simular as etapas de funcionamento de uma indústria com objetivo de analisar propostas de otimização e identificação de pontos vulneráveis sem interferir no funcionamento do sistema real e salientou o uso de Redes de Petri temporizadas para melhor validação do modelo.

Em Arteiro e outros [1] os autores apresentam um modelo de análise de desempenho de middlewares orientados a mensagens (MOMs) desenvolvidas em redes de Petri estocásticas para auxiliar na investigação da capacidade desse tipo de middleware por meio da simulação que possibilitou identificar o crescimento da latência, vazão do sistema e aumento da fila e a identificação de pontos de saturação do mesmo.

Em Roos-Frantz e outros [4] é proposto a simulação de modelos conceituais de soluções de integração desenvolvidos na tecnologia guaraná, modelados por Redes de Petri estocásticas que possibilitou observar o comportamento do modelo nas etapas anteriores a sua implementação para obter informações sobre o desempenho da solução de integração, possíveis gargalos e formação de filas, contribuindo minimizar

custos e diminuir riscos de falas nos projetos de integração de aplicações.

Em Carvalho e outros [3] é proposto um modelo de análise de desempenho de trabalho multifuncional em linhas de produção usando o formalismo das Redes de Petri temporizadas. Aborda a problemática da relação de tempo de execução das tarefas que compõem o sistema sua interferência no fluxo produtivo do sistema, ociosidade do trabalho e comprometimento da meta de produção.

V. CONCLUSÃO

As aplicações empresariais são de grande importância para dar suporte aos processos de negocio e necessitam se adequar a novos processos. O risco e o custo de integrar aplicações tende a diminuir, com a aplicação de técnicas de simulação para identificação de problemas em etapas precoces de sua integração.

O formalismo das Redes de Petri é usado frequentemente para modelar sistemas de eventos discretos, como é o caso das soluções de integração. A possibilidade de simular um modelo conceitual desenvolvido no Guaraná com as Redes de Petri pode vir a simplificar a identificação características importantes de uma solução de integração anterior a implementação.

AGRADECIMENTOS

Agradeço aos professores, aos colegas do mestrado e do grupo GCA e ao CNPQ pela bolsa de estudos.

REFERÊNCIAS

- [1] Roberto D. Arteiro and Paulo R.M. Maciel Fábio N Souza, Nelson S. Rosa. Utilizando redes de petri para modelagem de desempenho de middleware orientado a mensagem, 2007.
- [2] João Paulo Barros. Cppnets: uma classe de redes de petri de alto-nível: implementação de um sistema de suporte à sua aplicação e análise. 1996.
- [3] HJR CARVALHO et al. Análise de desempenho do trabalho multifuncional em linhas de produção, em forma de u pela modelagem e simulação usando redes de petri temporizadas. *XXXVII-SBPO-Simpósio Brasileiro de Pesquisa Operacional*, 2005.
- [4] Rafael Z. Frantz Sandro Sawicki Vitor Basto-Fernandes Fabricia Roos-Frantz, Manuel Bino. Using petri nets to enable the simulation of application integration solutions conceptual models. In *17th International Conference on Enterprise Information Systems (ICEIS)*, pages 87–97, 2015.
- [5] Rafael Z. FRANTZ. A dsl for enterprise application integration.
- [6] Rafael Z. Frantz and Rafael Corchuelo Reina Quintero, Antonia M. A domain-specific language to design enterprise application integration solutions. *International Journal of Cooperative Information Systems*, 20(02):143–176, 2011.
- [7] Gregor Hohpe and Woolf Bobby. *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional, 2004.
- [8] David S. Linthicum. *Enterprise application integration*. Addison-Wesley Professional, 2000.
- [9] Norian Marranghello. Redes de petri: Conceitos e aplicações. *São Paulo: DCCE/IBILCE/UNESP*, 2005.
- [10] Mamoru Carlos Yamada and Ricardo Yassushi Inamasu Arthur José Vieira Porto. Aplicação dos conceitos de modelagem e de redes de petri na análise do processo produtivo da indústria sucroalcooleira. *Revista Pesquisa Agropecuária Brasileira*. Brasília, 37(6):809–820, 2002.

Uso de Unidades de Memória em Redes Neurais Recorrentes para Acelerar a Otimização

Eldair F. Dornelles*, Rogério S. M. Martins
 Universidade Regional do Noroeste do Estado do Rio Grande do Sul
 Departamento de Ciências Exatas e Engenharias
 Ijuí, RS, Brasil
 eldair.dornelles@gmail.com, rogerio.martins@unijui.edu.br

Resumo—Neste trabalho foi realizado a modelagem de uma unidade de memória para ser incorporada a estrutura de uma rede neural artificial, visando acelerar a otimização de um agente inteligente em ambientes parcialmente observáveis.

Palavras-chave: Redes Neurais; Algoritmos Genéticos; Presa Predador.

I. INTRODUÇÃO

Na modelagem da estrutura de uma rede neural que tem por objetivo evoluir um programa que irá representar a inteligência de um agente, deve ser amplamente considerado o ambiente em qual o mesmo irá atuar. Em determinados ambientes, tais como os parcialmente observáveis, a rede neural pode apresentar dificuldades ou ser incapaz de evoluir, devido ao conceito de memória de curto prazo utilizado por elas.

Neste trabalho foi realizado a modelagem e proposta de uma unidade de memória para ser incorporada a estrutura de uma rede neural artificial visando acelerar a otimização de um programa para representar a inteligência de um agente, que tem por objetivo atuar em ambientes parcialmente observáveis. A rede neural modelada neste trabalho foi aplicada a uma versão modificada do problema presa predador. A otimização desta rede foi realizada através de técnicas evolutivas, para isso foi implementado um *framework* de algoritmo genético.

Ao final deste trabalho foi feita uma análise do desempenho deste agente com unidades de memória aplicado ao problema citado anteriormente e inferido o quão eficiente foi o seu desempenho.

II. REFERENCIAL TEÓRICO

A. Redes Neurais

Redes neurais artificiais são modelos computacionais inspirados no sistema nervoso dos seres vivos. Possuem capacidade de memorização e aprendizado, podem ser definidas como um conjunto de unidades de processamento, caracterizadas por neurônios artificiais, os quais são interligados[1]. A rede neural recorrente utilizado neste trabalho é constituída de uma única camada, em que todos os neurônios artificiais (unidades processadoras) são interligados, de forma que a saída de cada neurônio alimenta as entradas de todos os outros neurônios e a própria entrada. Tal rede neural se destaca por seu comportamento tipicamente dinâmico, capacidade de memorizar relacionamentos e armazenamento de informações[1].

* Bolsista de Iniciação Científica

B. Algoritmo Genético

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 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 (medida que identifica a capacidade de solução oferecida pelo indivíduo ao problema)[2].

C. Unidade de Memória

A unidade de memória modelada neste trabalho é composta por dois módulos, um deles define o valor a ser armazenado e o outro define se o novo valor vai ser armazenado ou se vai ser descartado, caso este novo valor seja descartado a unidade mantém o valor anterior na memória. O valor a ser retornado pela unidade de memória é sempre o valor que estiver armazenado. A figura 1 ilustra a estrutura da unidade de memória modelada neste estudo.

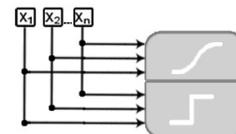


Figura 1. Estrutura da unidade de memória.

D. Ambiente de Testes

O problema abordado neste trabalho é uma versão modificada do tradicional “Presa Predador”. Nesta versão existe apenas um predador, uma presa, dois arbustos venenosos e um arbusto não venoso. O objetivo do predador neste ambiente modificado é capturar a presa sem encostar nos arbustos venenosos. Ao iniciar o jogo, os componentes são posicionados de forma aleatória a partir da segunda coluna, exceto o predador que é inicializado na posição (0,0) da matriz-ambiente. Os movimentos da presa e do predador são intercalados,

a cada ação executada pelo predador, a presa executa uma ação também. Os movimentos da presa são todos realizados buscando se posicionar junto ao arbusto não venoso, com o objetivo de se esconder do predador. O predador recebe as informações da localização da presa desde a inicialização do jogo até o momento que ela se esconde, a partir deste instante a única informação recebida é o posicionamento dos três arbustos, porém sem o conhecimento de suas características. Uma ilustração do ambiente é apresentada na figura 2.

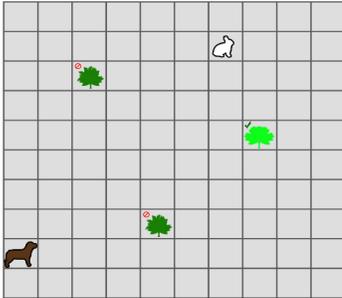


Figura 2. Representação do ambiente presa predador (modificado).

Seguem descritas as especificações do ambiente de tarefa para esse jogo, seguindo alguns conceitos básicos, tais como: medida de desempenho, atuadores e sensores, conforme classifica RUSSEL e NORVIG [3].

Ambiente de tarefa: O problema presa predador modificado consiste em um ambiente representado por uma matriz de ordem 10. Este ambiente é constituído por 2 arbustos venosos, 1 arbusto não venoso, uma presa e um predador.

Medida de desempenho: O predador é pontuado positivamente pela captura ou pela proximidade que o mesmo se encontra em relação a presa no momento em que o jogo é encerrado. Sendo que ao capturar a presa recebe +100, e a cada passo distante da presa é descontado 5. Caso o predador encoste em um arbusto venoso a pontuação obtida é dividida por 2.

Atuadores: O predador possui 4 ações possíveis para interagir no meio ambiente, são elas: se movimentar para frente, para trás, para esquerda e para a direita.

Sensores: As percepções são dadas ao agente em forma de lista, esta lista contém as informações referente a posição (x,y) de cada componente do ambiente.

III. METODOLOGIA

Para o desenvolvimento deste trabalho, buscou-se primeiramente conhecimentos teóricos sobre as técnicas a serem utilizadas. Em seguida deu-se início ao processo de modelagem da unidade de memória e estrutura da rede neural, seguido de sua implementação, a qual foi realizada em linguagem de programação C++, fazendo uso da IDE Eclipse. Tendo concluído a implementação deu-se início ao processo de testes, e por fim, fez-se a análise e discussão dos resultados obtidos.

O agente inteligente implementado neste projeto foi estruturado através de uma rede neural recorrente. O número de neurônios e de unidade de memória não foi fixado, podendo

assim, variar os mesmos, visando encontrar uma combinação que proporcione uma convergência mais rápida aos resultados desejados. Os pesos sinápticos utilizados pela rede neural recorrente são fornecidos pelo algoritmo genético, o qual gera uma população de indivíduos, onde cada indivíduo é formado por um vetor que contém uma combinação de pesos (números reais), de acordo com o tamanho da rede neural.

O processo de otimização ocorre da seguinte maneira: para cada indivíduo da população do algoritmo genético são definidos os pesos na rede neural, configurando assim a inteligência do agente a ser testado. Para cada um dos agentes é inicializado o jogo Presa-Predador de modo que o jogo é finalizado se o agente encostar em um arbusto venoso, encontrar a presa ou atingir o número limite de iterações. Para cada agente é realizado 100 execuções com o jogo, visando obter uma melhor precisão do desempenho, a cada execução é gerado um novo mundo de forma aleatória. Ao final das 100 execuções é retornado para o indivíduo do algoritmo genético a pontuação média obtida pelo agente. Essa pontuação será utilizada como base para definir a eficiência deste indivíduo, sendo esse valor atribuído à sua aptidão. No passo seguinte o Algoritmo Genético se encarrega da otimização dos pesos, seguindo os seus critérios de evolução.

Para o treinamento foi definido um agente com 4 neurônios e 2 unidades de memória, a otimização foi realizada através de 2000 gerações do algoritmo genético, com uma população de 500 indivíduos. Tendo o agente treinado, deu-se início aos testes de desempenho.

IV. RESULTADOS

Na fase de testes do agente já treinado, foi realizado 10.000 execuções, sendo cada execução um novo jogo. Destas 10.000 execuções, 61% das vezes o agente conseguiu capturar a presa, 15 % das vezes morreu por encostar em um dos arbustos venosos e em 24% das jogadas o agente atingiu o número máximo de iterações e o jogo foi encerrado.

V. CONCLUSÃO

Através da análise dos resultados obtidos, perceber-se que o agente obteve um bom aprendizado, pois o número de captura da presa, realizada pelo agente, foi quatro vezes maior que a quantidade de vezes que ele encostou nos arbustos venosos. Levando em consideração que as chances de encostar em um arbusto venoso é o dobro em relação a captura da presa, comprova-se então o bom aprendizado do agente.

REFERÊNCIAS

- [1] Ivan Nunes da Silva, Danilo Hernane Spatti, and Rogério Andrade Flauzino. *Redes Neurais Artificiais para Engenharia e Ciências Aplicadas - Curso Prático*. Rio de Janeiro, 2010.
- [2] Ricardo Linden. *Algoritmos Genéticos, 2a edição*. Brasport, 2012.
- [3] Peter Norvig and Stuart Russell. *Inteligência Artificial, 3a Edição*. Elsevier Brasil, 2014.

Modelagem Matemática e Computacional para Análise do Comportamento de Soluções de Integração de Aplicações Através da Criação de Modelos de Simulação com Cadeias de Markov

Márcia Maria Horn*, Sandro Sawicki

Universidade Regional do Noroeste do Estado do Rio Grande do Sul

Departamento de Ciências Exatas e Engenharias

Ijuí, RS, Brazil

marcia.horn@hotmail.com, sawicki@unijui.edu.br

Resumo—É de senso comum que, para auxiliar seus processos de negócio, as empresas utilizam heterogêneas aplicações de software. Neste cenário, a área de integração de aplicações empresariais tornou-se fundamental para a gestão da informação, pois proporciona metodologias e ferramentas para projetar e implementar soluções de integração, sem afetar a estrutura de dados e aplicações existentes. A tecnologia Guaraná é ferramenta que proporciona o desenvolvimento de soluções de integração possibilitando projetar soluções em um alto nível de abstração, utilizando uma sintaxe concreta gráfica e muito intuitiva. Este trabalho propõe analisar o comportamento e identificar gargalos de performance em soluções de integração de aplicações, buscando o desenvolvimento de um modelo matemático de simulação a partir de modelo conceitual utilizando Cadeias de Markov. O modelo será verificado utilizando técnicas de verificação formal, existentes na literatura.

Palavras-chave: Solução de Integração de Aplicações Empresariais, Guaraná - DSL, Cadeias de Markov, Simulação.

I. INTRODUÇÃO

Geralmente as empresas possuem um ecossistema de software para dar suporte a seus processos de negócios [7]. Um ecossistema de software é composto por um conjunto de aplicações, habitualmente heterogêneo e que geralmente foram desenvolvidas sem o propósito de integração de informações. A integração de aplicações é necessária quando uma aplicação isolada não é mais capaz de gerir ou suportar certos processos de negócio. A integração de aplicações possibilita reutilizar aplicações já existentes tendo como objetivo orquestrar essas aplicações para mantê-las sincronizadas ou proporcionar novas funcionalidades que possam ser construídas a partir daquelas já existentes [4].

A área de Integração de Aplicações Empresariais – EAI (do inglês Enterprise Application Integration) utiliza técnicas e ferramentas computacionais para que as empresas possam integrar dados e funcionalidades oferecidas por distintas aplicações. Dessa forma as empresas desenvolvem soluções de integração visando reutilizar suas aplicações, e dar suporte às novas demandas que surgem com a evolução dos seus

processos de negócio. Por tanto, uma solução de integração engloba um conjunto de aplicações, ou para orquestra-las de forma a mantê-las sincronizadas ou para proporcionar novas funcionalidades já oferecidas por essas aplicações.

Para integrar aplicações, os engenheiros de software geralmente seguem um determinado estilo de integração. Os diferentes estilos de integração existentes levam em consideração alguns critérios, tais como acoplamento das aplicações, simplicidade na integração, tecnologia usada pelas aplicações e formato dos dados. Os estilos de integração são divididos em quatro principais categorias: Arquivo, Banco de dados, RPC e Sistemas de mensagens.

Algumas tecnologias utilizam padrões de integração como Camel (Ibsen e Anstey, 2010), Spring Integration (Fisher et al., 2010), Mule (Dossot e D’Emic, 2009) e Guaraná (Frantz et al., 2011). Essas tecnologias possuem uma linguagem de domínio específico baseadas em mensagens e permitem desenvolver soluções de integração em uma plataforma. A tecnologia Guaraná DSL é uma das ferramentas que fornecem este suporte, pois é uma proposta para construir soluções de integração acessível, apresenta como solução uma construção gráfica.

A análise do comportamento e a identificação de gargalos de performance em soluções de integração de aplicações geralmente envolve sua construção para posterior execução e teste frente a cenários críticos projetados pelos engenheiros de software. Por demandar que as soluções sejam construídas, tal abordagem traz consigo custos (tempo, recurso) e riscos (bugs) inerentes que costumam ser elevados. A proposta de pesquisa é motivada pela possibilidade de análise do comportamento e identificação de gargalos de performance ainda na fase de projeto, a partir dos modelos conceituais das soluções. A partir dos modelos conceituais propõe-se um modelo matemático para representar formalmente uma solução de integração utilizando Cadeias de Markov. A verificação da equivalência do modelo conceitual com o modelo matemático proposto irá ser feito por meio de uma técnica de verificação.

*Mestranda em Modelagem Matemática da UNIJUI

II. TRABALHOS RELACIONADOS

Ábrahám e outros [1] apresentam um algoritmo de verificação de modelos utilizando Cadeias de Markov de Tempo Discreto. O algoritmo tem como base a detecção e captação de componentes fortemente ligadas, oferece contra-exemplos, que podem ser refinados interativamente pelo usuário. Durante a verificação do algoritmo todas as informações para as fases da recursividade são armazenadas. Isso permite que um usuário que precisa obter um contra-exemplo para uma determinada propriedade tenha acessibilidade para formular um contra-exemplo abstrato. No final, este produz de maneira sucinta um contra-exemplos que são construídos de acordo com a estrutura da Cadeias de Markov de Tempo Discreto.

Basagiannis e outros [2] introduzem a verificação de um modelo probabilístico como uma abordagem ferramenta-assistida viável para quantificar sistematicamente DoS (Navegação de Serviço) para ameaças de segurança. Ameaças DoS são expressas como propriedades acessibilidade probabilísticos que são automaticamente verificados através de uma Cadeia de Markov de Tempo Discreto que representa os participantes de protocolo e do modelos invasor. A análise de verificação do modelo probabilístico é realizada no software PRISM [6]. A verificação do modelo probabilístico é baseada em transições entre os estados de rotulagem modelo com informações sobre a probabilidade de virem a ocorrer.

Gomes e Wanke [5] propõem aplicar o conceito de Cadeias de Markov na gestão de peças de reposição. Os autores ressaltam que em uma Cadeia de Markov de Tempo Discreto aplicada à gestão de estoques, os estados possíveis denotam as diferentes posições de estoques que podem ocorrer ao longo do tempo. Neste estudo os autores buscaram modelar o consumo e a política de estoque de peças de reposição através de cadeias de Markov. Através da propriedade de convergência das Cadeias de Markov, é possível inferir a distribuição de probabilidades da posição de estoque, e a partir destes valores, determinar indicadores como estoque médio, probabilidade de falta.

Clement e outros [3] propõem um estudo a partir da observação de uniformidade dos modelos PP (Protocolos de População) que permite aplicar um contador de abstração. Cada agente no sistema é modelado como uma máquina de estado finito, que representa o programa do agente. Estes programas são uniformes: cada agente executa a mesma máquina de estado finito, de tal forma a não depender do número de agentes do sistema. Para a verificação os autores utilizaram o software PRISM, podendo lidar de forma muito eficiente com o modelo de verificação dessa propriedade sobre as abstrações do modelo PP. Um modelo probabilístico de verificação como PRISM permite calcular o tempo médio esperado de convergência, ou seja, o número de interações que ocorrem antes da convergência.

A proposta de pesquisa diferencia-se dos trabalhos descritos, pois visa à integração de aplicação com o uso de simulação em Cadeias de Markov. Sendo uma área em desenvolvimento e requer muita pesquisa para identificar os gargalos de per-

mance.

III. CONCLUSÃO

Soluções de integração podem ser caracterizado como sistemas discretos para a razão que todos os componentes envolvidos em uma solução de integração consumir um tempo de execução específico quando ocorre um evento [8]. Portanto, é possível utilizar modelos computacionais e matemáticos, e técnicas de simulação de eventos discretos para analisar o comportamento e identificar gargalos de performance que podem surgir em soluções de integração de aplicações quando submetidas à cenários críticos de funcionamento.

A partir de um sistema real de integração o modelo conceitual é projetado no Guaraná – DSL, e utilizando as Cadeias de Markov em Tempos Discretos será feito a verificação através da simulação. Nas simulações em eventos discretos as variáveis devem ser equacionadas em um modelo, cujas mudanças de estado ocorrem em pontos discretos de tempo, dirigidas por eventos isolados e, em geral, assíncronos. A escolha da utilização das Cadeias de Markov para este estudo é por ser um caso particular de processos estocásticos e, portanto utilizado para sistemas de eventos discretos, estudando a formação de filas. Os eventos geram as transições de estados representada por uma matriz de transição. Para a simulação propõe-se a utilização da ferramenta PRISM que é uma ferramenta para modelagem formal, a análise é feita por meio de um modelo de verificação dos sistemas que exibem comportamento aleatório ou probabilístico.

REFERÊNCIAS

- [1] Erika Abrahám, Nils Jansen, Ralf Wimmer, Joost-Pieter Katoen, and Bernd Becker. Dtmc model checking by scc reduction. In *QEST*, pages 37–46, 2010.
- [2] Stylianos Basagiannis, Panagiotis Katsaros, Andrew Pombortsis, and Nikolaos Alexiou. A probabilistic attacker model for quantitative verification of dos security threats. In *Computer Software and Applications, 2008. COMPSAC'08. 32nd Annual IEEE International*, pages 12–19. IEEE, 2008.
- [3] Julien Clément, Carole Delporte-Gallet, Hugues Fauconnier, and Mihaela Sighireanu. Guidelines for the verification of population protocols. In *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, pages 215–224. IEEE, 2011.
- [4] Rafael Z Frantz, Sandro Sawicki, Fabricia Roos-Frantz, Rafael Corchuelo, Vitor Basto-Fernandes, and Inma Hernández. Desafios para a implantação de soluções de integração de aplicações empresariais em provedores de computação em nuvem. *Salão do Conhecimento*, 2(01), 2014.
- [5] Antonio Vinicius Pimpão Gomes and Peter Wanke. Modelagem da gestão de estoques de peças de reposição através de cadeias de markov. *Gestão & Produção*, 15(1):57–72, 2008.
- [6] Marta Kwiatkowska, Gethin Norman, and David Parker. Prism 2.0: A tool for probabilistic model checking. In *null*, pages 322–323. IEEE, 2004.
- [7] David G Messerschmitt and Clemens Szyperski. Software ecosystem: understanding an indispensable technology and industry. *MIT Press Books*, 1, 2005.
- [8] S. Sawicki, R. Z. Frantz, V. M. B. Fernandes, F. C. R. Frantz, I. Yevseyeva, and R. C. Gil. Characterising enterprise application integration solutions as discrete-event system [no prelo]. In *Research on Computational Simulation and Modeling in Engineering. Ied.Hershey, Pennsylvania: IGI Globa*, pages 1–30. Francisco Miranda; Carlos Abreu. (Org.), 2015.

Adaptação de um Jogo Open Source para o Desenvolvimento de um Simulador de Trânsito

Henrique A. Richter*, Rogério S. de M. Martins

Departamento de Ciências Exatas e Engenharias

UNIJUÍ

Ijuí, Brasil

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

Resumo—Este projeto é um subprojeto do Física para todos e descreve a adaptação de um jogo de código fonte aberto para o desenvolvimento de um simulador de trânsito que possibilitará a coleta de dados das interações do usuário, que serão utilizados para traçar o perfil dos condutores de veículos.

Palavras-chave—simulador; jogos; educação para o trânsito;

I. INTRODUÇÃO

Segundo dados estatísticos do Departamento Autônomo de Estradas de Rodagem (DAER), no Rio Grande do Sul, o condutor é identificado como o maior causador dos acidentes de trânsito, com 90.15% dos acidentes causados por comportamento inadequado, em um total de 12.868 acidentes de trânsito ocorridos em 2012. A tabela 1 exibe a porcentagem de acidentes de 2010 a 2012 de acordo com a causa [1].

TABELA I. ACIDENTES OCORRIDOS E SUAS CAUSAS

CAUSA DO ACIDENTE	2010	2011	2012
Animais	2,26%	2,07%	2,37%
Condutor	88,96%	89,60%	90,15%
Pedestre	1,75%	1,70%	1,48%
Rodovia	1,04%	0,86%	0,53%
Veículo	2,38%	1,98%	1,71%
Não Constatado		0,89%	1,07%
Não Apurados-Outros	3,67%	2,87%	2,65%
TOTAL	11.939	12.632	12.868

É através destes dados que surgem incentivos para que os condutores recebam uma educação sobre o trânsito que possa informar os perigos da condução imprudente, e se conscientizar que é necessário respeitar as leis de trânsito, e dirigir com cuidado.

Visando essa necessidade na formação dos condutores, uma das possibilidades para que os condutores recebam treinamento sem comprometer a própria segurança e a segurança da sociedade, é a de utilizar simuladores de trânsito para realizar o

ensino de futuros ou atuais motoristas, facilitando a coleta de dados para posterior análise da forma como o condutor se comportou em determinadas situações de trânsito.

II. PROJETO

Este projeto é um subprojeto do Física para todos e utilizará um protótipo que já foi desenvolvido pelo grupo, onde o sistema mede o tempo de reação do condutor entre retirar o pé do acelerador e pisar no freio, porém não há imersão do motorista em um ambiente que seja próximo ao de estar realmente dirigindo o veículo. Para criar algo mais próximo da realidade será utilizado a adaptação de um jogo já existente, poupando tempo e trabalho na etapa de desenvolvimento, sendo necessário apenas adaptar o código fonte do jogo para receber dados da parte física do simulador.

Tendo um software que permite a imersão do usuário e que seja próximo a realidade é possível retirar dados mais precisos, como por exemplo, tempo de reação do motorista ao executar determinados comandos, que podem ser, desviar de obstáculos, acionar o freio ao ver obstrução na pista, reconhecimento de sinalizações, entre outros.

III. JOGOS ANALISADOS

Para decidir qual jogo utilizar para o desenvolvimento do simulador de trânsito foram analisados alguns jogos de corrida de código fonte aberto, dentre estes podemos destacar, Speed Dreams [2], The Open Racing Car Simulator (Torcs) [3] e Rigs of Rods [4].

Após comparações da interface gráfica, testes de jogabilidade e física, verificou-se que o jogo que mais se enquadra no projeto é Speed Dreams, devido a sua melhor jogabilidade, melhor física e interface gráfica mais amigável e realista.

IV. SPEED DREAMS

Speed Dreams é um jogo de simulação automobilística, com código fonte aberto (open source), liberado sobre a licença GNU General Public License (GPL), teve origem na bifurcação do jogo Torcs, tendo seu desenvolvimento iniciado em 2008 e sua última versão foi disponibilizada em julho de 2014, com o objetivo de melhorar diversas funções já existentes além do acréscimo de novidades. No momento as plataformas

*Bolsista de Iniciação Científica PIBIC/UNIJUÍ.

suportadas são Linux e Windows, além dessas o suporte para Mac OS X está sendo finalizado [2].

Entre as características que se destacam no Speed Dream podemos citar:

- Clima: o jogo possui diversos tipos de clima, como por exemplo, ensolarado, chuvoso, neblina, etc;
- Hora do dia: é possível escolher entre dia e noite;
- Câmera: posicionamento da câmera em diversas posições dentro e fora do veículo;
- Física: implementada visando o máximo de realismo;
- Open Source: possibilitando alterações no código fonte.

Além destas características o jogo também se destaca por não necessitar de um hardware potente para ser executado, um processador de 1.5 GHz, 1 Gb de memória RAM e uma placa de vídeo de ao menos 128 Mb é o suficiente para configurar o jogo na qualidade mínima.

Na figura 2 pode-se verificar umas das possíveis posições da câmera durante o jogo.



Figura 2. Câmera sobre o capô do carro.

Para realizar alterações no código fonte foi utilizado o Sistema Operacional Linux, sendo necessário a instalação de algumas bibliotecas que o jogo necessita para ser compilado, como por exemplo bibliotecas de áudio, vídeo, comunicação com periféricos, etc.

V. CONSIDERAÇÕES FINAIS

Por se tratar de um projeto de código fonte aberto, Speed Dreams é desenvolvido com ajuda da comunidade, e apesar de vários anos sendo desenvolvido a qualidade dos gráficos está longe de jogos comerciais atuais, já que seu desenvolvimento não é financiado por nenhuma grande empresa, porém isto não é empecilho para este projeto, já que o jogo fornece o necessário para o desenvolvimento do simulador.

Após a análise, escolha do jogo, instalação e configuração do ambiente de desenvolvimento, os próximos passos serão a adaptação do código fonte do jogo para receber comandos provenientes do volante e pedais do simulador, que serão desenvolvidos pelo grupo, para então preparar o restante do simulador para as próximas etapas do projeto, que envolvem ajustes no sistema de captura de dados e ajustes no jogo.

AGRADECIMENTOS

Ao PIBIC e a UNIJUÍ pela bolsa de iniciação científica e ao GCA pela oportunidade de participar do grupo.

REFERÊNCIAS

- [1] E. J. Masiero, "Acidentalidade em Rodovias Estaduais do Rio Grande do Sul: Acidentalidade no Ano de 2012", disponível em: http://www.daer.rs.gov.br/site/controle_estudos_estatisticos_acidentes_transito.php, acesso em Junho, 2015.
- [2] Speed Dreams, disponível em: <http://www.speed-dreams.org/>, acesso em Junho, 2015.
- [3] The Open Racing Car Simulator, disponível em: <http://torcs.sourceforge.net/>, acesso em Junho, 2015.
- [4] Rigs of Roads, disponível em: <http://www.rigsofrods.com/content/>, acesso em Junho, 2015.

Implementação de um Jogo de Tabuleiro para evolução de agentes inteligentes: Tigres e Vacas

Rodolfo Berlezi

Unijuí, DCEEng

GCA, Grupo de Pesquisa em Computação Aplicada

Ijuí, Brasil

rodolfo_berlezi@hotmail.com

Rogério Samuel Martins

Unijuí, DCEEng

GCA, Grupo de Pesquisa em Computação Aplicada

Ijuí, Brasil

rogerio@aezon.com

Abstract— we will develop an application for artificial intelligent tests, using the knowledge achieved by studying the programming language JAVA3D and the useful tool to create tridimensional universes, Unity3D.

Palavras-Chaves— aplicação;jogo; universo; tridimensional; linguagem;

I. INTRODUÇÃO

Aprendendo a manipular as utilidades disponíveis da linguagem de programação JAVA3D, criaremos um universo tridimensional para testes de agentes inteligentes. Também explicaremos a decisão do ambiente tridimensional usado, no caso um jogo de tabuleiro, Tigres e Vacas. Com o JAVA3D entendemos a organização destes universos, mas para a criação do ambiente de testes do agente inteligente, será necessário uma ferramenta mais completa, escolhemos o Unity3D.

II. DESENVOLVIMENTO

A começar pelo estudo da linguagem de programação JAVA3D, que nos disponibiliza uma forma de mais baixo nível de abstração para trabalhar com universos tridimensionais, foi essencial para a total compreensão do funcionamento e posicionamento das formas criadas no universo.

Primeiramente, temos de entender a esquematização dos componentes do universo tridimensional, que se dispõem no formato de um grafo de árvore.

A grande raiz da árvore, a base para a inicialização de um projeto, é o objeto do universo virtual onde tudo mais será alocado em diferentes níveis de camadas. O primeiro nó da árvore, logo após o universo, é o nó do objeto de Canvas, o quadro, retrato, onde será devidamente posicionado as próximas peças, de uma forma visível para o usuário os objetos geométricos que serão criados nos próximos nodos do grafo de árvore.

Os objetos tridimensionais geométricos são os fundamentais itens que serão utilizados para trabalhar na aplicação e interação do usuário. São eles classificados como

os grupos de galhos das árvores, ou como dito no JAVA3D, são o BranchGroup.

O nosso objeto de BranchGroup é dado como uma forma neutra, imagine uma massa que precisa ser moldada para adquirir o formato que será utilizado, mas também são os únicos objetos que são aceitos na associação com o objeto Canvas. Para possibilitar a moldagem do objeto galho, temos que descer mais um nível na árvore e utilizar um objeto da classe Transform3D, que somente dentro dela teremos os métodos necessários para a criação e execução de uma figura geométrica tal como um cubo, uma esfera, um cilindro.

Agora para além de termos o objeto puro, podemos especializar ele, descendo mais nodos do grafo de árvore, temos como manipular o comportamento da figura e sua aparência, com texturas que a aproximam de um objeto real. Os métodos de Behavior variam de fazer a figura girar ao redor de um eixo no universo, a poder movimentá-la com cliques do mouse.

Nos nodos iniciais do grafo de árvore, em paralelo, podemos ter incontáveis galhos ou objetos de BranchGroup como o universo é infinito, e nisto, os objetos podem se dar de formas diferentes como um galho nova aparência para o quadro ou um galho de luz, que ilumina toda ou parte da cena do nosso Canvas, dados por classes como AmbientLight ou PointLight.

Todavia essas técnicas de implementação tem como finalidade em um universo tridimensional criar a aproximação da noção de realismo, interação e animação para o usuário. A excelente reprodução destes três atributos fazem com que o universo seja muito mais próximo, intuitivo e vivo, dando a ideia de ter sido criado um verdadeiro universo paralelo ao real.

III. CASOS DE USO

Uma vez que tenhamos aprendido técnicas suficientes para compreender a essência de um universo tridimensional, teríamos de escolher um jogo para criarmos e que fosse de maior relevância para testarmos futuros agentes inteligentes. Dada a situação nos informamos sobre seis

diferentes módulos de jogos de tabuleiros, cada um com uma complexidade superior a outra.

O primeiro módulo é o mais simples e arcaico de todos, temos que pensar que os jogos de tabuleiros vem de muitos anos A.C., assim, no começo os jogos tinham como visão o bloqueio das poucas peças adversárias e alinhamento das suas. Este modelo de jogo era muito limitado visto que utilizavam pequenos tabuleiros com poucas peças, as jogadas se tornavam repetitivas e muito eficientes. Simples demais para um agente bem treinado.

Por segundo já temos a adição do deslocamento das peças, com objetivos de movimentar todas suas peças até o outro lado do tabuleiro, percorrendo percursos livres e curtos. Por mais que o tamanho do tabuleiro e a quantidade de peças tenha aumentado, os jogos ainda não eram complexos o bastante, poucas regras e jogadas ensaiadas muito práticas.

Agora com os jogos de posicionamento no terceiro módulo, temos uma verdadeira maior complexidade, há jogos simples demais ainda, como o famoso Jogo-da-Velha, mas a outros de uma complexidade singular como o GO. No caso do GO, temos um tabuleiro enorme e um número que é muito variante de peças que podem ser posicionadas durante uma única partida. Mas por mais que seja um jogo de infinitas possibilidades, é um jogo também já muito explorado e poderíamos escolher algo mais singular.

No quarto módulo temos uma nova ação dos jogadores, a captura e promoção de peças, o jogo mais conhecido do módulo é o de Damas, onde uma peça pode se mover um passo por vez, saltar por uma peça adversária para captura lá e quando chegar ao final do tabuleiro receber uma promoção. Também é um jogo já muito saturado e não o optamos.

O quinto módulo é também o provável mais desconhecido pela sociedade, os jogos de caça, onde temos um pouco de cada outro módulo junto e com um grande diferencial que o torna um jogo aparentemente desbalanceado, cada jogador não tem necessariamente o mesmo número de peças, e as peças tem funções diferentes. Exemplos desse jogo são o jogo da Onça, Raposa e Galinhas, Urubu e Corvos, Tigres e Vacas. Como os nomes sugerem, cada jogador controlará um tipo de peças que tem comportamentos únicos e objetivos de capturar ou proteger as peças.

Este módulo é muito mais inteligente e complexo que todos os demais apresentados, e principalmente, é muito exclusivo, visto que são jogos despopularizados pela sociedade, provavelmente devido a este suposto desbalanceamento.

O último módulo é aquele que abrange todas as complexidades dos anteriores, onde estão jogos como o Xadrez e o Shogi. São jogos com bloqueio, deslocamento, posicionamento, captura, promoção e singularidade nas peças.

Estes porém são jogos de uma complexidade muito maior que os de caça, dado a grande variedade de peças e possibilidades infinitas do jogo.

Sendo assim, o módulo escolhido para trabalharmos um simulador de agentes inteligentes, foi o quinto, e o jogo será o Tigres e Vacas, um jogo que tem por disposição, dois tigres para um jogador e dez vacas para o outro. O objetivo dos tigres é capturar as vacas e o das vacas é incapacitar o movimento dos tigres.

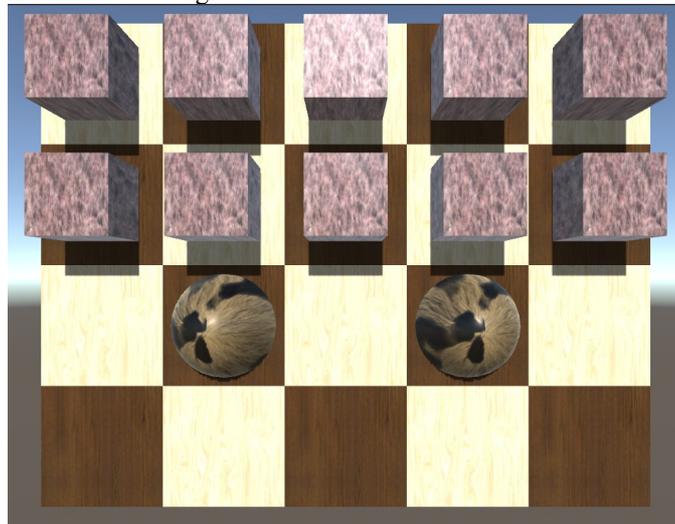


Figura 1: Tabuleiro de Tigres e Vacas produzido com a ferramenta Unity3D.

IV. CONCLUSÃO

A implementação do jogo utilizando a ferramenta de criação de universos tridimensionais Unity3D fica significativamente mais rápida e prática de ser construída do que puramente pela linguagem de baixo nível JAVA3D. O jogo está atualmente em fase de implementação, temos o tabuleiro com as peças posicionadas e um simples movimento delas por todo o mesmo. Ao finalizar a implementação das regras de captura e finalização do jogo, o mesmo estará concluído e pronto para ser testado utilizando futuros agentes inteligentes.

REFERÊNCIAS

- [1] Devmedia, de Eduardo Oliveira Spinola, disponível em <<http://www.devmedia.com.br/introducao-a-java-3d-api/1608>>. Acesso em Setembro de 2014.
- [2] Introdução à JAVA3D, de Isabel Harb Manssour, disponível em <<https://www.inf.pucrio.br/manssour/Java3D/>>. Acesso em Outubro de 2014.
- [3] Unity Tutoriais, disponível em <<http://unity3d.com/pt/learn/tutorials/modules>>. Acesso em Março de 2015.
- [4] Learning Geek Blog, disponível em <<https://learninggeekblog.wordpress.com/>>. Acesso em Abril de 2015.

Uso de jogos lógicos aplicados a aprendizagem de máquina

Jean R. R. da Silva, Rogério S. de M. Martins
 Universidade Regional do Noroeste do Estado do Rio Grande do Sul
 Departamento de Ciências Exatas e Engenharias
 Ijuí, RS, Brazil
 {jrafaelrs.silva}@unijui.edu.br, {rogerio.martins}@unijui.edu.br

Resumo—O projeto tem por finalidade desenvolver um aplicativo android, nesse caso um conjunto de jogos que possam ser utilizados na educação e como ambiente de testes para inteligência artificial.

Palavras-chave: Jogos; Educação; Android.

I. INTRODUÇÃO

Este trabalho faz parte de dois projetos, sendo esses o projeto LoboGames desenvolvido pela UFRGS e o projeto Sistema de Decisão Adaptativo Baseado em Redes Neurais para Uma Arquitetura Aplicada a Agricultura de Precisão desenvolvido pela UNIJUI.

O projeto LoboGames tem por objetivo divulgar e disseminar o interesse pelos jogos lógicos de tabuleiro, promovendo seu aprendizado e onde um dos seus benefícios é o exercício do raciocínio lógico.

Assim o projeto tem por objetivo o desenvolvimento de jogos de raciocínio lógico, mais especificamente jogos de caça, sendo desenvolvidos na forma de aplicativos para a plataforma android.

Este aplicativo final poderá então ser usado em ambos os projetos, em um deles na forma de jogos educativos e no outro para o estudo e análise técnicas de inteligência artificial.

II. APLICATIVO

Os jogos estão sendo desenvolvidos para a plataforma android, utilizando para isso o ambiente de desenvolvimento Android Studio.

O projeto já conta com quatro aplicativos desenvolvidos de um total de sete, onde estes possuem uma interface padronizada, o que facilitara na hora de integrar esses jogos em um único aplicativo, onde será buscado o aperfeiçoamento da interface gráfica.

O primeiro aplicativo desenvolvido no projeto foi o jogo “Lebre x Cachorros”, onde o objetivo dos cachorros é trancar a lebre em qualquer parte do tabuleiro, enquanto que o objetivo da lebre é atravessar pelos cachorros para poder escapar.

A seguir as duas figuras mostram respectivamente a tela inicial do jogo, a qual é padronizada em todos os jogos, e o tabuleiro do jogo.

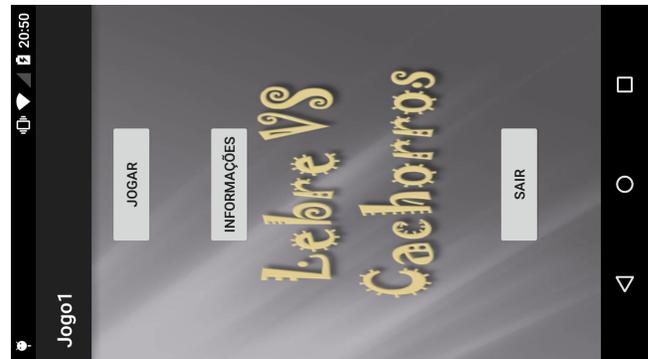


Figura 1. Menu dos jogos!

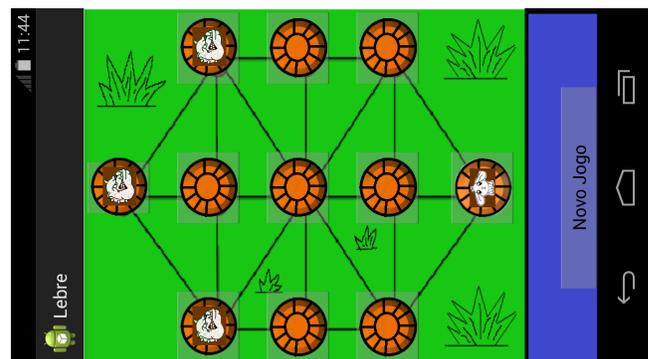


Figura 2. Tabuleiro primeiro jogo!

Nesse ambiente o jogador controla sempre os caçadores e terá seus possíveis movimentos indicados pelo aplicativo, com isso o jogo e para apenas um jogador por vez.

Os jogos desenvolvidos na sequencia sofreram algumas alterações, de modo que passou a ser jogado por dois jogadores ou ainda por 1 jogador e um agente neural, além dessa modificação também foi dada uma liberdade maior aos jogadores, onde ele não terá mais a sensação de ter seus movimentos restringidos pelo aplicativo.

A programação do jogo continuara controlando os movimentos para que não ocorram eventuais enganos dos jogadores como por exemplo o de jogar duas vezes consecutivas, nesses casos o jogo emitira um alerta como na figura a seguir.

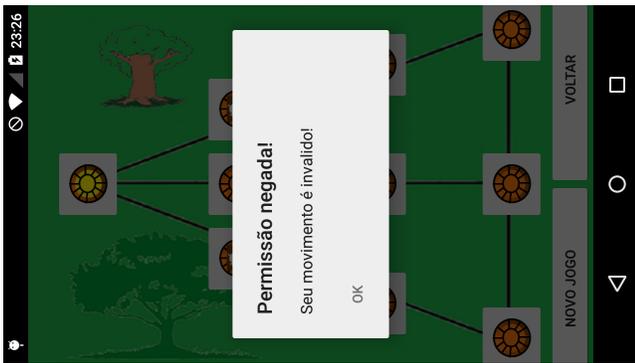


Figura 3. Aviso do jogo!

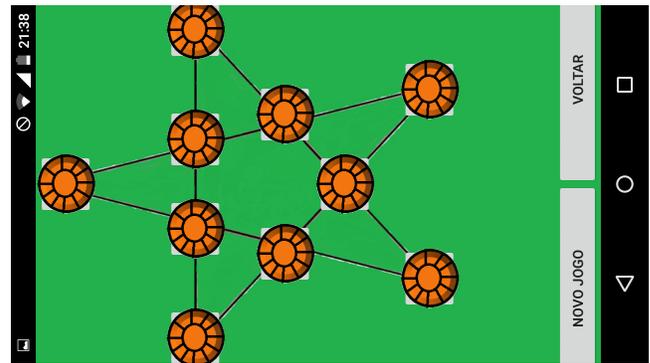


Figura 5. Urubu e Corvos!

Com essa nova organização o jogador terá uma maior sensação de liberdade no tabuleiro, isso transmitira uma ideia similar a de jogar em um tabuleiro físico, porém se utilizando da tecnologia para fazê-lo.

Os próximos dois jogos os quais são “Leopardo e Caçador” e “Urubu e Corvos”, que já estão desenvolvidos nesse novo parâmetro, ambos possuem uma ideia semelhante de jogabilidade mesmo utilizando-se de tabuleiros diferentes.

As regras são quase as mesmas, os dois jogos possuem uma presa e sete caçadores, tendo os caçadores o objetivo de trancar a presa em qualquer lugar do tabuleiro e a presa tem que capturar três dos caçadores.

No primeiro desses a presa tem uma posição fixa para iniciar a partida e no segundo ela pode sair da posição que o jogador julgar melhor, enquanto que em ambos os jogos todos os caçadores são livres para entrarem no tabuleiro.

O quarto e ultimo jogo desenvolvido até o momento é o “Raposa e Cachorros” que possui as mesmas regras e objetivos que o primeiro jogo apresentado, porém foi implementado para dois jogadores.

Para que a lista desses sete jogos seja finalizada ainda faltam os seguintes jogos “Jogo da Onça”, “Raposa e Gansos” e “Tigres e Vacas”.

A seguir e possível visualizar a imagem dos jogos citados anteriormente, a parte de seus tabuleiros:

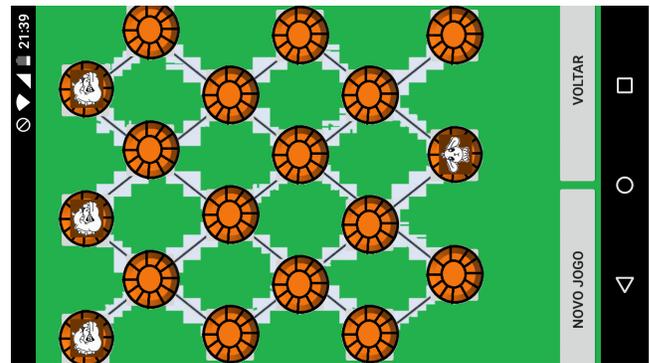


Figura 6. Lebre e Cachorros!

III. CONCLUSÃO

Como projetos futuros ainda temos que finalizar os jogos e juntá-los em um único aplicativo, para que possamos assim usá-los imediatamente, ou deixa-los a disposição, nos projetos como jogos educativos e também para o estudo de técnicas de inteligência artificial, onde o mesmo poderá ainda ser utilizado em outros projetos.

IV. REFERENCIAS

- 1 Projeto LoboGames, acessado em 14 de junho de 2015, <http://www.inf.ufrgs.br/lobogames>
- 2 Projeto Sistema de Decisão Adaptativo Baseado em Redes Neurais para Uma Arquitetura Aplicada a Agricultura de Precisão, acessado em 14 de junho de 2015, <https://www.unijui.edu.br>
- 3 Site Android Studio, acessado em 13 de junho de 2015, <http://developer.android.com/tools/studio>

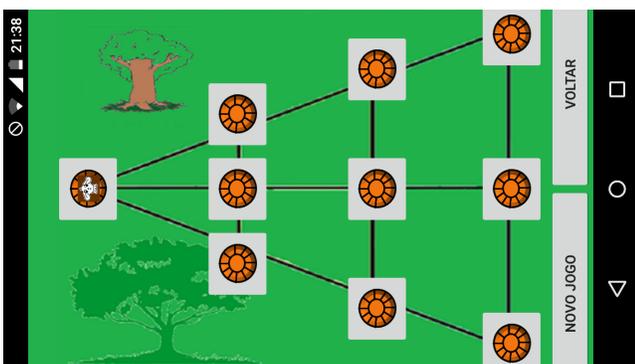


Figura 4. Leopardo e caçadores!

Desenvolvimento de um Modelo de Simulação baseado em uma Solução de Integração Teórica utilizando a Ferramenta PRISM

Guilherme Henrique Schiefelbein Arruda*, Sandro Sawicki
 Universidade Regional do Noroeste do Estado do Rio Grande do Sul
 Departamento de Ciências Exatas e Engenharias
 Ijuí, RS, Brazil
 {guilherme.arruda, sawicki}@unijui.edu.br

Resumo— A maioria das empresas possuem vários tipos de softwares que foram desenvolvidos individualmente, sem que houvesse integração, tornando complicada a comunicação e o compartilhamento de informações entre eles. Uma alternativa para a empresa seria adotar uma tecnologia de Enterprise Application Integration (EAI), a qual permite desenhar soluções de integrações de maneira simples e eficiente. Entretanto, antes de aplicar uma solução de integração em uma empresa, esta deve ser analisada e testada para garantir que se adequará com o ambiente empresarial. Diante disso, este *paper* apresenta os resultados obtidos com uma solução de integração que já estava definida, porém não foi implementada. Esta foi utilizada apenas como referência para verificar a sua equivalência com o modelo de simulação criado e prever seu comportamento com base nos testes efetuados.

Palavras-chave: Enterprise Application Integration; Simulação.

I. INTRODUÇÃO

Por vários anos, os softwares têm sido construídos para um único propósito, para um único conjunto de usuários, sem pensar na possibilidade de integrar estes sistemas e transformá-los em grandes softwares com múltiplas aplicações. Diversos tipos destes sistemas são vistos como um gargalo para a integração, pois são desenvolvidos independentemente e sem um padrão comum de armazenamento de dados ou tecnologia de aplicação [8].

A. Enterprise Application Integration

Uma solução para este tipo de problema, segundo [8], é analisar dois pontos fundamentais. O primeiro, é que a empresa precisa conhecer a arquitetura que possui, bem como os processos de negócio e os dados dentro do ambiente empresarial. Em segundo lugar, é necessário encontrar uma tecnologia de integração de aplicações empresariais (EAI) que consiga resolver o problema e seja compatível com a arquitetura estudada anteriormente.

Dessa maneira, os softwares utilizados pela empresa estariam conectados constantemente e seriam controlados por uma aplicação EAI centralizada, que permitiria o compartilhamento de informações e o tratamento dos dados de cada software de

maneira simples e precisa, tornando o ambiente empresarial organizado e estável. Além disso, as aplicações não precisam ser modificadas para serem integradas com a aplicação EAI. Neste trabalho, será utilizada a ferramenta Guaraná [4] para efetuar a criação do modelo de uma solução de integração.

B. Guaraná DSL

O Guaraná foi desenvolvido para facilitar a modelagem de soluções de integração. Para isso, faz uso de uma linguagem específica de domínio (DSL), que permite desenhar soluções independentes de plataforma utilizando uma interface gráfica simples e intuitiva. Para isso, existem alguns componentes básicos que fazem parte dos construtores do Guaraná como, por exemplo, mensagens, tarefas, portas, slots, entre outros [5].

Todos estes elementos estão inseridos em um processo de integração que executa operações sobre as mensagens, dentro de uma lógica de integração. Isso permite que sejam utilizados filtros e canais para realizar a comunicação com as aplicações que estão sendo integradas [5].

C. Simulação

Uma vez que o problema de integração está bem definido, é necessário criar um modelo conceitual, pois este permite descrever o comportamento de um sistema real, além de ser uma alternativa para problemas muito caros de se resolver através de um experimento, ou muito complexos para aplicar técnicas analíticas. Criar um modelo conceitual significa descrever e transcrevê-lo de uma forma que o computador possa processá-lo.

Para isso, deve haver uma ferramenta capaz de realizar estas tarefas como, por exemplo, software de simulação [3]. A simulação permite utilizar um modelo computacional a fim de realizar experimentos baseados em técnicas matemáticas para avaliar seu comportamento. Assim, é possível prever comportamentos futuros, baseado nos efeitos produzidos através de alterações no sistema ou nos seus processos.

Estes efeitos podem ser observados quando ocorre uma mudança em algum evento do sistema, dentro de certo período de tempo. Isso caracteriza os eventos como sendo discretos. Caso as mudanças partam de decisões aleatórias, então o

* Bolsista de Iniciação Científica PIBIC/UNIJUÍ

sistema é considerado estocástico [2]. A definição da proposta deste trabalho é explicada na seção seguinte.

D. Definição da Proposta

Neste trabalho, serão considerados apenas modelos estocásticos e discretos, pois são compatíveis com os problemas de integração, onde o comportamento do sistema integrador depende totalmente das aplicações envolvidas. Será utilizado um simulador probabilístico, para suportar este tipo de modelo. Assim, será possível inserir o modelo no simulador e realizar testes com o mesmo, a fim de analisar sua equivalência com o Guaraná. O resultado disso mostrará o quanto a solução de integração será compatível para resolver o problema em questão.

II. DEFINIÇÃO DO MODELO DE UMA SOLUÇÃO DE INTEGRAÇÃO

O primeiro problema teórico utilizado foi o descrito por [6], o qual não foi implementado por nenhuma ferramenta. Este mostra como uma cafeteria processa os pedidos de seus clientes. O objetivo da solução de integração é retirar pedidos da fila, encaminhá-los para o devido barista e notificar o garçom quando o pedido foi concluído.

Neste modelo existem duas filas, uma para os pedidos que entram no sistema e outra para os pedidos concluídos. Considerando que estes dois elementos estão interligados, seria possível também classificá-los como uma rede de filas Markovianas, pois, além da interconexão, as mensagens precisam passar pela primeira fila e receber seus serviços, e então prosseguir para a próxima fila até sair do sistema.

III. FERRAMENTA DE SIMULAÇÃO PRISM *Model Checker*

O próximo passo, a partir da obtenção do modelo conceitual, é utilizar uma ferramenta de simulação, criar um modelo equivalente e testar seu comportamento diante de determinadas situações. A ferramenta utilizada para isso foi o PRISM *Model Checker* [7], que é adequada para fazer a modelagem e verificação formal de modelos probabilísticos. Para efetuar a construção de um modelo, este deve estar especificado na linguagem PRISM, a qual é baseada em estados e módulos de Alur e Henzinger [1].

Cada módulo é composto por suas variáveis, que representam o estado em que o módulo pode estar, e seus comandos, que descrevem o comportamento dos módulos. Depois que o modelo for construído, recomenda-se descrever suas propriedades para que a ferramenta possa avaliá-lo [7]. Os resultados da verificação da solução de integração presente neste trabalho são mostrados na próxima seção.

IV. PRINCIPAIS RESULTADOS OBTIDOS

A partir da obtenção do modelo conceitual da solução de integração, efetuou-se a criação de um modelo no PRISM e, posteriormente, foi feita a simulação deste modelo utilizando cadeias de Markov com tempo discreto (DTMC). Dentro da ferramenta PRISM, foram descritos os módulos e estados que representam a solução de integração. A maioria dos processos

possui 100% de probabilidade de seguir para o próximo estado, pois representam passos determinísticos que devem ser seguidos para chegar ao resultado final.

Em contra partida, existem estados em que o sistema deve realizar uma escolha não determinística, onde o próximo estado pode direcionar o caminho para um resultado positivo ou negativo. Neste modelo, o comando para estes estados foi criado utilizando probabilidades distintas, em vez utilizar comandos não determinísticos. Dessa maneira, foi possível mudar os valores do simulador manualmente para analisar o comportamento do sistema. Posteriormente, utilizou-se o simulador para verificar o caminho percorrido pelo sistema com base nos possíveis estados disponíveis.

V. CONCLUSÃO

Diante dos resultados obtidos, a ferramenta PRISM permite construir modelos probabilísticos equivalentes com os modelos de solução de integração desenhados pela ferramenta Guaraná. Analisar o comportamento do sistema durante a simulação auxilia a prever quais caminhos serão tomados e qual será o resultado final. Caso o resultado não for o esperado, basta fazer alterações nos modelos e executar novos testes, a fim de encontrar o melhor resultado e inserir o novo modelo na empresa, integrando todas as aplicações necessárias, sem gastar tempo e recursos.

Como neste *paper* foi utilizada uma solução de integração que já estava definida, será criada uma nova solução baseada em um problema de integração real. Esta será descrita e desenhada, utilizando a ferramenta Guaraná DSL. Posteriormente, será criado um modelo de simulação equivalente no PRISM para avaliar o comportamento e prever possíveis problemas que a solução possa apresentar.

REFERÊNCIAS

- [1] Rajeev Alur and Thomas A Henzinger. Reactive modules. *Formal Methods in System Design*, 15(1):7–48, 1999.
- [2] Paulo José de Freitas Filho. *Introdução à modelagem e simulação de sistemas: com aplicações em Arena*. Visual Books, 2001.
- [3] Bráulio Adriano de Mello. Modelagem e simulação de sistemas. *Ciência da Computação/Sistemas de informação. Universidade Regional Integrada do Alto Uruguai e das Missões, Departamento de Engenharias e Ciência da Computação*, 2001.
- [4] Rafael Z Frantz and Rafael Corchuelo. A software development kit to implement integration solutions. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 1647–1652. ACM, 2012.
- [5] Rafael Z Frantz, Rafael Corchuelo, and Jesús González. Advances in a dsl for application integration. *Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos*, 2(2), 2008.
- [6] Gregor Hohpe. Your coffee shop doesn't use two-phase commit [asynchronous messaging architecture]. *Software, IEEE*, 22(2):64–66, 2005.
- [7] Marta Kwiatkowska, Gethin Norman, and David Parker. Prism: Probabilistic symbolic model checker. In *Computer performance evaluation: modelling techniques and tools*, pages 200–204. Springer, 2002.
- [8] David S Linthicum. *Enterprise application integration*. Addison-Wesley Professional, 2000.