

IV Seminário de Formação Científica e Tecnológica

Grupo de Pesquisa em Computação Aplicada

www.gca.unijui.edu.br

17 – Junho – 2016
8:30 – 12:00 e 13:30 – 17:00

Local:
Campus Santa Rosa
Santa Rosa, RS. Brasil

Índice

Estudo comparativo das ferramentas PIPE2, Mercury Tool e TimeNet baseadas em Redes de Petri <i>Dener Krebs</i>	5 - 6
Comparação das Ferramentas PIPE2 e Oris2 quanto à Simulação <i>Adriana R. Krausig, Francieli C. Welter</i>	7 - 9
Redes de Petri como Instrumento para Modelagem Matemática de um Problema de Integração da área de Publicidade e Propaganda <i>Francine Freddo</i>	10 - 12
Um Modelo Matemático para a Simulação de uma Solução de Integração baseado no Formalismo Redes de Petri <i>Alexsandro Queiroz Lencina</i>	13 - 15
Modelagem e desenvolvimento de uma solução de integração utilizando Guaraná Cloud <i>Edinaldo Gaspar da Silva</i>	16 - 17
Modelagem matemática e computacional para análise do comportamento de soluções de integração de aplicações através da criação de modelos de simulação em uma empresa de venda e aluguel de equipamentos <i>Leandro Fritzen Klem</i>	18 - 20
Framework de Comparação da Evolução do grau de Manutenibilidade da Ferramenta de Integração de Aplicações Apache Camel <i>Matheus H. Rehbein</i>	21 - 22
Modelagem de uma Solução de Integração no Contexto da Administração Pública Municipal <i>Ivan E. M. Kühne</i>	23 - 24
Execução de Soluções de Integração de Aplicações Empresariais na Nuvem: Perspectivas e Desafios <i>Daniela Freire Sellaro</i>	25 - 27
Modelo de Simulação de uma Solução de Integração Teórica utilizando a Ferramenta MATLAB/Simulink <i>Amanda Preissler</i>	28 - 29
Proposta de uma Solução para Avanços Científicos e Tecnológicos no Cultivo de Aveia: Uso de modelagem matemática e inteligência artificial na categorização das cultivares de aveia quanto a resistência genética e uso de fungicida <i>Eldair Dornelles</i>	30 - 32
Análise do Comportamento de uma Solução de Integração na Área de Reservas de Viagens utilizando Redes de Petri <i>Francisco da Silveira</i>	33 - 35
Comparação da Evolução do Grau de Manutenibilidade da Ferramenta Spring Integration <i>Thainan H. Feistel</i>	36 - 37
Análise comparativa entre Redes de Petri, Cadeias de Markov e Teoria das Filas <i>Francieli C. Welter, Adriana R. Krausig</i>	38 - 40



Applied
Computing
Research Group

IV SFCT

O *IV 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. Sandro Sawicki
UNIJUÍ, Brasil

Comitê Científico:

Dra. Fabricia Roos-Frantz
UNIJUÍ, Brasil

Dr. Rafael Zancan Frantz
UNIJUÍ, Brasil

Dra. Inma Hernandez
Universidad de Sevilla, Espanha

Dr. Carlos R. Osuna
Rochester Institute of Technology, United
States

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

Igor Gamste Haugg
UNIJUÍ, Brasil

Amanda Preissler
UNIJUÍ, Brasil

Dener Krebs
UNIJUÍ, Brasil

Estudo comparativo das ferramentas PIPE2, Mercury Tool e TimeNet baseadas em Redes de Petri

Dener Éden Krebs

Universidade Regional do Noroeste do Estado do Rio Grande do Sul

Departamento de Ciências Exatas e Engenharias

Santa Rosa, RS, Brasil

krebsdener@gmail.com.br

Resumo—Empresas geralmente possuem um grande número de aplicações distintas que não foram projetadas para compartilhar informações uns com os outros. Neste contexto surge uma alternativa que é uso de uma Enterprise Application Integration (EAI) que permite a criação de soluções de integração de maneira eficiente. Para que a solução de integração funcione de maneira eficiente é preciso garantir que não haja gargalos na modelagem da solução. Uma forma de verificar se um modelo contém gargalos é simular o seu comportamento. Para realizar as simulações é necessário desenvolver um modelo equivalente da solução de integração usando uma linguagem de modelagem e um simulador que intérprete este modelo. Este artigo apresenta três simuladores que trabalham com redes de Petri estocásticas e uma comparação das suas funcionalidades.

Palavras-chave: Enterprise Application Integration; Redes de Petri Simulação.

I. INTRODUÇÃO

Por muito tempo as empresas desenvolveram sistemas que haviam sido construídos para solucionar um problema específico e para servir a um único propósito para um grupo de usuários sem ter em mente uma futura integração com outros sistemas maiores. Enquanto essa tecnologia envelhecia o seu valor dentro da empresa permanecia crítico. Infelizmente algumas dessas tecnologias eram muito difíceis de adaptar para permitir a comunicação e compartilhar informações com outros sistemas mais avançados. Ainda que há a possibilidade de substituir esses sistemas por outros sistemas mais novos o custo para tal geralmente é proibitivo [11]. A seção II apresenta o referencial teórico. A seção III descreve as ferramentas de simulação. A seção IV aborda a descrição da modelagem de simulação. A seção V expõe as conclusões do artigo.

II. REFERENCIAL TEÓRICO

A. Integração de Aplicações Empresariais

Como a dependência das empresas na tecnologia informação tem crescido de forma mais complexa e abrangente, a necessidade de um método para integrar aplicações distintas em um conjunto unificado de processos de negócio se faz necessário. Neste contexto surge a área de (EAI - Enterprise Application Integration) que oferece um conjunto de metodologias e ferramentas para integrar distintas aplicações que per-

mitem a muitas desses sistemas hoje existentes a compartilhar tanto processos como dados, atendem essas demandas [11].

B. Guaraná DSL

O Guaraná DSL é uma linguagem de domínio específico (*Domain-Specific Language - DSL*) usada no projeto de soluções de integração empresariais em um alto nível de abstração. Como resultado, a linguagem do guaraná produz modelos independentes de plataforma que não necessitam que o engenheiro tenha conhecimentos profundos em integração de baixo nível para desenvolver soluções de integração [7]. As tarefas do guaraná são baseadas em nos padrões documentados em Enterprise Integration Patterns [9]. Os principais blocos de construção do guaraná DSL são: as portas de comunicação, processos, tarefas, e os *slots* [7].

As portas de comunicação abstraem os mecanismos de comunicação necessários para se comunicar com outras aplicações. Os processos orquestram com as aplicações, e eles precisam das tarefas para executar suas atividades. E as tarefas são blocos de construção que permitem a execução de ações nas mensagens que fluem em um processo [7].

C. Simulação

A simulação computacional utiliza modelos computacionais e matemáticos para realizar testes sem a necessidade de construir ou alterar o sistema que será simulado. Simulações computacionais utilizam-se de modelos de sistemas que são uma abstração do real comportamento do sistema, mas sempre mais simples do que o real [6].

A simulação pode ser vista como o estudo do comportamento de sistemas reais através da utilização de modelos. Um modelo incorpora características que permitem representar o comportamento do sistema real [10]. Este trabalho busca representar o modelo de simulação por meio da utilização de ferramentas baseadas em Redes de Petri.

D. Redes de Petri

As redes de Petri são um modelo matemático com representação gráfica que vem sendo utilizado na representação de diversos sistemas orientados a eventos discretos. As redes

de Petri são usadas principalmente para especificar, analisar o comportamento lógico, avaliar o desempenho em diversas áreas como a de comunicação, de transporte, de informação, sistemas de manufatura entre outros [12].

III. FERRAMENTAS DE SIMULAÇÃO

Existem diversas ferramentas de simulação para redes de Petri disponíveis na internet, a maior parte delas pode ser encontrada no Petri net *DataBase* [2], porém nem todas as ferramentas estão atualizadas com isso apresentam os mais variados tipos de problemas, como por exemplo, o toolbox para Matlab Petri net não recebe mais atualizações desde 2007.

Para a comparação foram escolhidas três ferramentas: PIPE2 [3], TIMENet Evaluation Tool [4] e o Mercury Tool [1]. As três ferramentas são baseadas em desenvolvidas em Java, portanto, são independentes de plataforma e as ferramentas estão disponíveis gratuitamente. A ferramenta Mercury está disponível apenas para uso acadêmico mediante a assinatura de um termo de compromisso, assim como a ferramenta TimeNet que necessita do preenchimento de cadastros para o download. Tais ferramentas suportam, entre outros tipos, redes de Petri estocásticas.

Para a comparação optou-se pela criação de uma tabela com as funcionalidades e os recursos disponíveis em cada uma das ferramentas mencionadas, conforme a tabela desenvolvida por [5]. A Tabela I, mostra a comparação das funcionalidades entre as ferramentas PIPE2, TimeNet e Mercury Tool.

Tabela I
TABELA COMPARATIVA DAS FERRAMENTAS

	PIPE2	TimeNet 4	Mercury
Interface			
Desfazer	Sim	Sim	Sim
Refazer	Sim	Não	Sim
Peso dos Arcos	Sim	Não	Sim
Funções			
Limite de capacidade	Sim	Não	Não
Prioridades	Sim	Sim	Não
Semânticas de disparo	Não	Sim	Sim
Animações dos tokens			
Passo a passo	Sim	Sim	Sim
Voltar	Sim	Não	Não
Animação continua	Sim	Sim	Sim
Análise estrutural			
Comparação de Redes	Sim	Não	Não
Lugares invariantes	Sim	Sim	Sim
Transição invariantes	Sim	Sim	Sim
Análise			
Análise Transiente	Não	Sim	Sim
Simulação transiente	Não	Sim	Sim
Análise estacionaria	Não	Sim	Sim

IV. MODELAGEM DA SOLUÇÃO DE INTEGRAÇÃO

A solução de integração utilizada foi descrita por [8], este modelo mostra como uma cafeteria processa os pedidos

dos clientes de forma que um cliente faz o pedido no caixa, que é adicionado em uma fila, e este pedido pode incluir bebidas quentes e frias que são preparadas por diferentes baristas. Assim quando os pedidos estão prontos eles podem ser entregues pelo garçom.

A construção do modelo de equivalência da solução foi desenvolvida usando como base a tabela de equivalência entre as tarefas do Guaraná e seus respectivos grafos equivalentes descritos por [13].

V. CONCLUSÃO

Este trabalho apresentou a comparação entre as ferramentas PIPE2, TimeNet4 e Mercury, baseadas em Redes de Petri. Após a comparação foi possível perceber que as ferramentas atendem as diferentes necessidades para a simulação de redes de Petri. No entanto a ferramenta PIPE2 apresenta diversas ferramentas de análise avançada apresentando, tais como na simulação, que indica o número médio de tokens por lugar assim resultando informações mais completas de como a rede de Petri se comporta.

REFERÊNCIAS

- [1] Mercury tool. www.cin.ufpe.br/~bs/MercuryTool/mercury.html. Acessado: Abril de 2016.
- [2] Petri net database. <https://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick.html>. Acessado: Dezembro de 2015.
- [3] Platform independent petri net editor 2. <http://pipe2.sourceforge.net/>. Acessado: Janeiro de 2016.
- [4] Timenet evaluation tool. <https://www.tu-ilmenau.de/sse/timenet/>. Acessado: Abril de 2016.
- [5] Pere Bonet, Catalina M. Lladó, Ramon Puijaner, and William J Knottenbelt. Pipe v2. 5: A petri net tool for performance modelling. 2007.
- [6] Leonardo Chwif, Afonso Medina, and Tecnologia Simulate. *Modelagem e Simulação de Eventos Discretos, 4ª Edição: Teoria e Aplicações*, volume 4. Elsevier Brasil, 2014.
- [7] 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.
- [8] Gregor Hohpe. Your coffee shop doesn't use two-phase commit [asynchronous messaging architecture]. *IEEE software*, 22(2):64–66, 2005.
- [9] Gregor Hohpe and Bobby Woolf. *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional, 2004.
- [10] Averill M. Law, W. David Kelton, and W. David Kelton. *Simulation modeling and analysis*, volume 2. McGraw-Hill New York, 1991.
- [11] David S. Linthicum. *Enterprise application integration*. Addison-Wesley Professional, 2000.
- [12] Norian Marranghello. *Redes de petri: Conceitos e aplicações. São Paulo: DCCE/IBILCE/UNESP*, 2005.
- [13] Fabricia Roos-Frantz, Manuel Binelo, Rafael Z. Frantz, Sandro Sawicki, and Vitor Basto-Fernandes. Using petri nets to enable the simulation of application integration solutions conceptual models.

Comparação das Ferramentas PIPE2 e Oris2 quanto à Simulação

Adriana R. Kraisig, Franciéli C. Welter
Universidade Regional do Noroeste do Estado do Rio Grande do Sul
Departamento de Ciências Exatas e Engenharias
Ijuí, RS, Brasil
akraisig@yahoo.com.br, fran-c-w@hotmail.com

Resumo—Mediante a simulação, pode-se analisar o comportamento e o desempenho de sistemas, encontrando possíveis gargalos de performance. Devido ao aumento de demandas e pesquisas realizadas, com a simulação, surge a necessidade de se fazer análises mais profundas, no que diz respeito, as diferentes ferramentas de simulação. Desta forma, este artigo, faz uma comparação entre duas ferramentas de simulação PIPE2 e Oris2, visando elencar através de um *framework*, as peculiaridades e diferenças existentes, para escolher qual destas ferramentas, pode ser a mais adequada para simulação de soluções de integração. A área da *Enterprise Application Integration* (EAI) proporciona metodologias, técnicas e ferramentas para as empresas desenvolverem soluções de integração. Uma solução de integração, busca integrar novas aplicações que inicialmente não foram pensadas em conjunto, com as já existentes.

Palavras-chave: Ferramentas de Simulação; PIPE2; Oris2.

I. INTRODUÇÃO

Normalmente, as aplicações que compõem os ecossistemas de *software* das empresas, são heterogêneas. Um dos desafios enfrentados, nos últimos anos, têm sido, o de integrar estas diferentes aplicações, fazendo com que as mesmas, possam colaborar entre si [4]. A Integração de Aplicações Empresariais (EAI), consiste em proporcionar respostas para décadas de criação de aplicações monolíticas, que não foram pensadas para trabalhar em conjunto umas com as outras. A EAI proporciona metodologias, técnicas e ferramentas, para projetar e implementar soluções de integração [4].

O objetivo de uma solução de integração é orquestrar duas ou mais aplicações, que inicialmente não foram pensadas para trabalhar em conjunto [5]. Quando essa integração de aplicações é feita corretamente, diminuem-se os custos de integração, tempo e riscos de falhas. O uso de ferramentas de simulação, permite, encontrar falhas na solução de integração; promove uma melhor compreensão de sistemas e vêm sendo amplamente utilizadas em diversas áreas, incluindo na computação [3].

Diante disso, procurando facilitar o uso da simulação na área de EAI, é importante escolher ferramentas que mais se ajustem a área. Para isto, à pesquisa propõe um *framework* de comparação, que pode ser utilizado como guia neste processo, a partir de um conjunto de propriedades, como: tipo de Rede de Petri, componentes, ambientes, funções, *interface* de trabalho, interpretações dos resultados da simulação e características do editor, a partir da área de EAI.

Este artigo propõe um simples *framework* de comparação que permite avaliar ferramentas de simulação, a partir da perspectiva de integração de aplicações. Com isso, será possível compreender melhor o funcionamento das ferramentas PIPE2 e Oris2, buscando encontrar peculiaridades e diferenças, quanto à simulação. O restante do artigo, está organizado da seguinte maneira: a Seção II, apresenta o *framework* de comparação, a Seção III, trata da discussão dos dados e por fim as conclusões na Seção IV.

II. FRAMEWORK DE COMPARAÇÃO

O *framework* de comparação, elenca um conjunto de sete propriedades que permitem analisar as ferramentas de simulação sob a perspectiva da EAI. Seguem abaixo, as propriedades utilizadas a nível de comparação:

Tipos de Redes de Petri: É possível com as ferramentas de simulação, criar Redes de Petri, para posteriormente simular. A pesquisa, elenca quais tipos de Redes de Petri, podem ser utilizadas em cada ferramenta. Destaca-se a Rede de Petri estocástica (RE), que se caracteriza, por incorporar aspectos temporais não determinísticos. Rede de Petri Temporizada (RT), que incorpora aspectos temporais determinísticos aos modelos. Rede de Petri Colorida (RC), que se caracteriza por possuir cores ou marcas. Rede de Petri Hierárquica (RH), que tem por objetivo permitir a representação de modelos de sistemas complexos de forma mais compreensível pelo modelador.

Componentes: Cada ferramenta, possui seus componentes específicos, com os quais, é possível realizar tarefas. Neste sentido, podem apresentar, editores gráficos (EG), que são programas de computador que têm como objetivo facilitar a alteração e criação de imagens digitais. Jogo de *tokens* (JT), oferece um animador para que o usuário possa experimentar manualmente, disparando qualquer uma das transições habilitadas em cada estado. Simulação gráfica (SG), em que a simulação é construída por um conjunto de *interfaces* gráficas. Estados (E), gera gráficos de acessibilidade. Lugar *invariant* (LI), calcula os vetores das invariantes de lugar e de transições com precisão e eficiência. Análise estrutural (AE), identifica topologias especiais e marcação de gráficos. Análise de desempenho simples (ADS), disponibiliza suporte de ferramentas com a simulação, que utiliza tempo. Formato de arquivo *interchange* (FAI), exporta e importa diagramas para outras

ferramentas. Análise de módulos extensivos (AME), disponibiliza suporte para a geração de um protótipo, que funciona, com base em um modelo. Análise de formatos de arquivo (AFA), gerencia o fluxo de trabalho e análise de desempenho avançado (ADA), proporciona o suporte de ferramentas, tais como cadeias Markovianas.

Ambientes: Para que cada ferramenta, possa executar suas tarefas, é necessário que seja implementado, um ou mais ambientes. Dentre os ambientes, está o Java (J), o qual, possui um ambiente necessário para desenvolver e executar aplicativos em Java. PC Linux (PCL), possui código aberto e pode ser escrito e distribuído por qualquer tipo de usuário na *internet*, dá suporte a placas, cd-rom e outros dispositivos mais ultrapassados e/ou avançados. PC Windows XP (PCXP), funciona de duas formas: como um sistema operacional virtual e como uma maneira de abrir programas com o *windows 7*. *Macintosh Mac OSX* (MOSX), apresenta qualidade na *interface* gráfica do computador, comandos diferenciados em suas últimas versões, como permissão de múltiplos toques e uma navegação baseada na intuição do usuário.

Função: As ferramentas, possuem funções, as quais, são utilizadas para disparo. A função de distribuição exponencial negativa (FDEN), é um tipo de distribuição contínua de probabilidade, representada por um parâmetro. A função determinante (FD), é uma função matricial. Função de probabilidade polinomial (FPP), associa, a cada possível ocorrência de uma variável aleatória discreta, uma probabilidade.

Interface de trabalho: As ferramentas de simulação estudadas, possuem diferentes *interfaces* de trabalho. Estas ferramentas, são diferentes entre si, pois, cada uma delas, é composta por funções específicas. *Interface* acessível (A), sendo possível, desenhar e executar Redes de Petri, de maneira, rápida e eficaz. *Interface* de trabalho complexa (C), sendo difícil manusear as funções proporcionadas pela mesma.

Resultados da simulação: Ambas as ferramentas, geram os resultados após a simulação. Resultados simples (RS), que apresenta uma resolução mais simples. Resultados complexos (RC), de difícil compreensão.

Características do editor: Estas ferramentas de simulação, possuem um editor, que apresenta diferentes funções, como: zoom (ZO), que é responsável pelo controle do enquadramento de uma imagem. Exportação (EX), é usado para gerar arquivos com a imagem em vários formatos de arquivo. Edição (ED), permite criar e alterar Redes de Petri. Animação (AN), mecanismo para selecionar e controlar os atributos da cena necessários para produzir os efeitos visuais desejados. Desfazer (DE), permite desfazer o último atributo. Recortar e colar (RC), permite transferir uma notação, de um lugar para outro fazendo com que essa notação deixe de existir no local onde estava. Grade (GR), série de barras verticais paralelas espaçadas, podendo ter outras barras entrecruzadas. Alinhamento (AL), ato ou efeito de pôr em linha reta ou paralela. Espaçamento (ES), permite a escolha de espaços desejados entre as notações. Notas auto-adesivas (NAA), é utilizada para criar lista de tarefas.

III. DISCUSSÃO

Nesta pesquisa, estudou-se duas ferramentas de simulação, que possibilitam elaborar e analisar Redes de Petri. Ambas, foram comparadas, dando origem à Tabela 1:

Conjunto de Variáveis	PIPE 2	ORIS 2
Tipo de Rede de Petri	RE RT RC RH	RE RT
Componentes	EG JT SG E LI AE ADS FAI AME AFA	EG E ADA JT
Ambientes	J	PCL PCXP MOSX J
Função	FDEN	FD FPP
Interface de trabalho	A	C
Resultado da simulação	RS	RC
Editor	ZO EX ED AN	DE RC ZO GR AL ES EX NAA

Figura 1. Tabela de Comparação

A partir da Tabela 1, constatou-se que a ferramenta PIPE2 é independente de plataforma e possui uma *interface* gráfica que permite criar e analisar Redes de Petri, de forma rápida e eficaz. O disparo das Redes de Petri, ocorre através de uma função de distribuição exponencial negativa. Sua implementação é feita em Java, para garantir a independência de plataforma. Além disso, possui módulos de análise, para verificar propriedades comportamentais, produzir estatísticas de desempenho e comparar dados [1].

Os módulos de análise incluem: análise (*invariant*); análise estado-espaco, que permite observar as propriedades qualitativas das Redes de Petri, como limitabilidade, segurança e auto-travamento; e análise de simulação, que calcula o número médio de marcas (*tokens*) por lugar, com intervalo de confiança de 95%. A ferramenta realiza o fluxo de acordo com as regras de disparo das transições, permitindo a visualização do fluxo de *tokens* e identificação dos estados alcançáveis do sistema modelado. Ressalta-se que, o princípio chave do projeto é desenvolver uma *interface* extensível do módulo, permitindo que novos módulos de análise sejam facilmente implementados.

Com relação a ferramenta Oris2, verificou-se através da Tabela 1, que a mesma, cria e edita graficamente Redes de Petri, além de permitir a modelagem de extensões estocásticas e temporizadas, sendo utilizada para verificação qualitativa e quantitativa de sistemas. A ferramenta associa transições aos

lugares no decorrer do fluxo, utilizando para isso, uma função determinante ou polinomial de probabilidade, sendo que sua implementação pode ocorrer em: *PC, Linux; PC, Windows XP; Macintosh, Mac OSX* ou Java [2]. O editor da ferramenta, permite, desfazer, recortar, colar, *zoom*, grade, alinhamento, espaçamento uniforme de elementos, exportação SVG e notas autoadesivas. O gráfico das Redes de Petri estocásticas ou temporizadas pode ser elaborado e visualizado. No caso das estocásticas, pode destacar pontos de regeneração e excluir disparos que tenham probabilidade nula. A probabilidade pode ser calculada através da enumeração após cada disparo, dentro de um tempo pré-determinado.

Neste sentido, através da Tabela 1, foi possível constatar, que ambas as ferramentas, possuem suas próprias peculiaridades e diferenças. No que diz respeito, às Redes de Petri, têm em comum, o fato de representarem Redes de Petri estocásticas e temporizadas, no entanto, a ferramenta PIPE2, também representa outros tipos de Redes de Petri, como por exemplo, hierárquicas e coloridas. Quanto aos seus componentes, ambas apresentam animação: jogo de *tokens*; editor gráfico e estados. A ferramenta Oris2 também conta com análise de desempenho avançado, enquanto a ferramenta PIPE2 tem simulação rápida; lugar (*invariant*); transição (*invariant*); análise estrutural; análise de desempenho simples; formato de arquivo (*interchange*); análise de módulo extensiva e formatos de arquivo. Em relação aos ambientes de implementação, as ferramentas, possuem em comum o Java, porém, a ferramenta Oris2, também, pode ser implementada em: *PC, Linux; PC, Windows XP e Macintosh, Mac OSX*. Quanto às funções, na ferramenta PIPE2 as Redes de Petri disparam por distribuição exponencial negativa e na ferramenta Oris2 têm-se o uso de função determinante ou função de probabilidade polinomial. No que tange, a *interface* de trabalho, o PIPE2, possui uma *interface* acessível. Por sua vez, a ferramenta Oris2, possui uma *interface* de trabalho complexa. No que diz respeito, aos resultados obtidos com a simulação. Com o PIPE2 é possível compreendê-los melhor, pois, gera gráficos e tabelas de dados, gerados através da simulação, os quais são de fácil entendimento, por apresentar uma resolução simples. Já o Oris2, apresenta resultados mais complexos, de difícil compreensão. Quanto ao editor, o PIPE2 possui: *zoom*, exportação, edição e animação. Já o Oris2, possui, além do *zoom*, as seguintes funções: desfazer, recortar e cortar, grade, alinhamento, espaçamento, exportação e notas autoadesivas.

IV. CONCLUSÃO

Concluiu-se, através da pesquisa entre as ferramentas, que PIPE2 é melhor quando busca-se criar modelos de simulação de soluções de integração. Isto se deve, ao fato de que esta ferramenta abrange mais tipos de Redes de Petri, possui mais componentes e sua *interface* de trabalho é mais acessível. A função, que a ferramenta utiliza para o disparo da simulação, é aceitável. Quanto, a interpretação dos resultados, o PIPE2, também se destaca positivamente, por apresentar gráficos e tabelas de dados de fácil compreensão. Os pontos negativos, é com relação ao ambiente de implementação e o editor, pois,

PIPE2 pode ser implementado somente em Java e seu editor possui menos funções, enquanto Oris2, possui outras opções de implementação e seu editor possui mais funções. Além disso, pode-se observar que através da simulação é possível analisar a solução de integração, para desta forma, encontrar gargalos de performance.

REFERÊNCIAS

- [1] Pere Bonet, Catalina M. Lladó, Ramon Puijaner, and William J. Knottenbelt. Pipe v2. 5: A petri net tool for performance modeling. In *Proc. 23rd Latin American Conference on Informatics (CLEI 2007)*, 2007.
- [2] Giacomo Bucci, Laura Carnevali, Lorenzo Ridi, and Enrico Vicario. Oris: a tool for modeling, verification and evaluation of real-time systems. *International journal on software tools for technology transfer*, 12(5):391–403, 2010.
- [3] Leonardo Chwif and Afonso Medina. *Modelagem e Simulação de Eventos Discretos, 4ª Edição: Teoria e Aplicações*, volume 4. Elsevier Brasil, 2014.
- [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.

Redes de Petri como Instrumento para Modelagem Matemática de um Problema de Integração da área de Publicidade e Propaganda

Francine Freddo

Universidade Regional do Noroeste do Estado do Rio Grande do Sul
Departamento de Ciências Exatas e Engenharias
Ijuí, RS, Brasil
francine.freddo@hotmail.com

Resumo—Com o avanço tecnológico, as empresas que possuem aplicações para gerenciar seus processos de negócios necessitam, normalmente, trocar informações entre seus diferentes tipos de sistemas. Muitas vezes estas aplicações foram adquiridas por diferentes fornecedores em diferentes épocas, ocasionando dificuldades para trabalharem de forma única e sincronizada. Neste contexto, surge a área de Integração de Aplicações Empresariais que visa proporcionar metodologias, técnicas e ferramentas para a concepção e a implementação de soluções de integração. Para tanto, quando há uma grande demanda de recursos computacionais, esta solução pode apresentar problemas, sendo o mais comum o acúmulo de mensagens, o que ocasiona gargalos de performance. Para a elaboração de uma solução de integração algumas fases precisam ser seguidas: levantamento de requisitos, análise, projeto, implementação e testes. Atualmente, os gargalos de performance são encontrados após a implementação e teste do sistema, entretanto, isso demanda tempo e custo. Este trabalho propõe a análise do comportamento e identificação de possíveis gargalos de performance da solução de integração, ainda na fase de projeto, através do desenvolvimento de um modelo formal de simulação a partir de um modelo conceitual usando Redes de Petri. A solução de integração analisada insere-se na área de publicidade e propaganda e foi desenvolvida pela tecnologia de integração Guaraná.

Palavras-chave: Integração de Aplicações, Modelagem Computacional, Simulação, Redes de Petri.

I. INTRODUÇÃO

No contexto econômico e industrial atual, é necessário que muitos sistemas empresariais sejam reprojatados para atender a demanda do mercado, tanto em razão de mudanças em requisitos funcionais como pela evolução tecnológica. O grande desafio, nesse contexto, é conciliar a criação de novos sistemas empresariais com as tecnologias atuais, ao mesmo tempo fazendo com que os sistemas antigos se mantenham operantes em novas plataformas ou trabalhem em conjunto. Problemas de integração não ocorreriam se os sistemas fossem totalmente independentes. Percebe-se, contudo, que isso não ocorre na realidade, pois na maioria dos casos cada novo sistema depende das informações já existentes.

A partir dessa necessidade surgiu um campo de pesquisa conhecido, atualmente, como Integração de Aplicações Empresariais. Segundo [6] o termo EAI ou *Enterprise Application*

Integration "é o compartilhamento irrestrito de dados e processos de negócios entre quaisquer aplicações conectadas e fontes de dados na empresa." O Processo de Negócio refere-se ao conjunto de atividades que produzem um serviço específico para seus clientes. EAI é ainda, o termo formal que contempla a integração de aplicações corporativas e um conjunto de ferramentas e tecnologias que propõe uma solução. As aplicações que a empresa foi adquirindo ao longo do tempo e que dispõe para dar suporte aos seus processos de negócios são denominadas ecossistema de *software*.

Uma solução de integração tem o objetivo de "orquestrar" um conjunto de aplicações para mantê-las harmonizadas ou oportunizar novas funcionalidades a partir daquelas existentes [5]. Para a sua elaboração é necessário seguir algumas etapas tais como levantamento de requisitos, análise, projeto, implementação e testes, no qual as duas últimas possuem um custo elevado. Mesmo seguindo todas essas fases, ainda podem ocorrer erros ou gargalos de performance em situações de grande demanda de processamento, que são detectados, somente na fase de implementação e testes. Com o intuito de diminuir custos, percebe-se a importância de verificar esses erros e gargalos ainda na fase de projeto, por meio de modelos conceituais de solução e modelos de simulação.

Os modelos conceituais podem ser construídos a partir de uma DSL (Linguagem de Domínio Específico) que é uma linguagem própria para interpretação da solução de integração, neste trabalho será utilizada a tecnologia Guaraná [4]. Os modelos de simulação podem ser representados por modelos matemáticos, neste caso, Redes de Petri.

Este trabalho propõe a análise do comportamento e identificação de possíveis gargalos de performance da solução de integração, ainda na fase de projeto, através do desenvolvimento de um modelo formal de simulação a partir de um modelo conceitual usando Redes de Petri, para um caso na área de publicidade e propaganda.

II. CASO DE ESTUDO

O problema de integração refere-se à uma empresa real cujo ramo de atividade esta voltado para a área de publicidade e

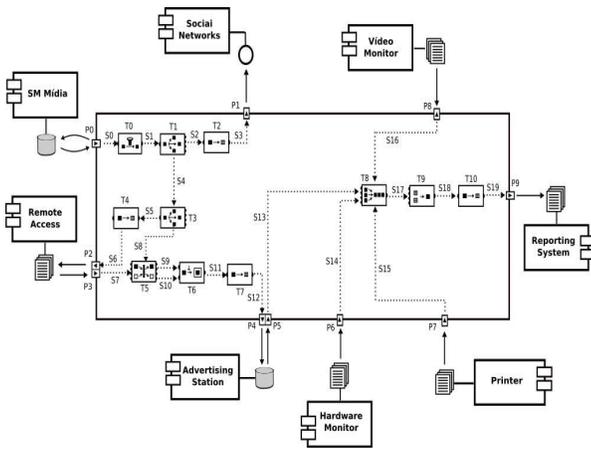


Figura 1. Modelo conceitual da solução de integração.

propaganda. Esta empresa visa divulgar comerciais de vários estabelecimentos do comércio da região, utilizando estações de propagandas, chamada de "totens". Cada estação possui um monitor LCD Full HD de 20", um mini gabinete e uma impressora em seu interior. A empresa possui diversos totens localizados nas principais cidades do Rio Grande do Sul, onde há um intenso fluxo de pessoas.

Deste modo, a empresa precisa selecionar os comerciais que serão inseridos em cada um dos totens, pois em cada cidade existe um conjunto de estabelecimentos diferentes. Em cada local existe uma rede *wireless* em que o totem está conectado. O problema consiste em uma série de pacotes que envolvem diferentes aplicações e que trabalham individualmente, na qual a empresa precisa verificar cada um deles para obter informações concretas sobre os componentes da estação. Segundo a empresa, perde-se muito tempo realizando estas verificações, tempo este que poderia ser utilizado para personalizar mais anúncios por dia ou buscar novos estabelecimentos que estejam dispostos a divulgar sua propaganda.

O ecossistema de *software* é composto por 8 aplicações, divididas em 5 pacotes. No pacote 1 temos as mídias digitais ou e-mail, no pacote 2 as aplicações banco de dados e redes sociais, no pacote 3 o *software* de acesso remoto, a estação de propaganda e o *software* que reproduz vídeos, no pacote 4 os *softwares* de monitoramento do *status* da rede, da impressora, da temperatura e no pacote 5 a impressora.

O objetivo então, é automatizar o controle de monitoramento que a empresa utiliza para verificar o comportamento de cada estação de propaganda. Diante disso, [1] criou um modelo conceitual da solução de integração responsável por coordenar as atividades dos *softwares* que serão integrados, como mostra a Figura 1, utilizando a tecnologia Guaraná.

III. SIMULAÇÃO E REDES DE PETRI

Simulação é um termo muito amplo, porém pode-se definir basicamente como sendo o processo de elaboração de um modelo real e a condução de experimentos com a finalidade de entender o comportamento de um sistema ou avaliar sua operação [12], ou seja, permite a geração de cenários, a partir

dos quais pode-se orientar o processo de decisão, fazer análises e avaliações e propor melhorias no sistema.

A simulação, segundo Aalst, citado em [11] é composta por onze fases, sendo elas: definição do problema, modelagem, modelo conceitual, realização, modelo executável, verificação e validação, modelo validado, experimentando, resultados da simulação, interpretação e respondendo as soluções. A definição do problema precisa considerar as metas para saber o que vai e o que não vai fazer parte da simulação. A modelagem, é onde o modelo conceitual é elaborado com as propriedades mais importantes do sistema. A fase de realização consiste em transformar o modelo conceitual em um modelo executável. O modelo executável pode ser simulado no computador, por isso necessita de uma ferramenta de simulação. O próximo passo é verificar e validar o modelo, comparando com o modelo real. Com o modelo validado, os experimentos podem ser realizados, sendo conduzidos de maneira eficiente para que os resultados sejam confiáveis. Os resultados da simulação precisam ser interpretados para permitir um entendimento sobre o problema. E, por fim, é realizado um relatório com as respostas para as questões da definição do problema no qual as propostas de soluções são criadas [11].

O sistema real pode ser representado por meio de um modelo, sendo físico (protótipo) ou matemático, no qual este último pode ser verificado analiticamente ou utilizando-se a simulação. A maioria dos sistemas reais são melhor representados por modelos estocásticos, dinâmicos e discretos, pois mudam aleatoriamente no tempo, assim será tratado o modelo de simulação de eventos discretos.

Neste contexto, este trabalho utiliza Redes de Petri para criar o modelo de simulação proposto, pois é um formalismo que permite a modelagem de sistemas dinâmicos discretos com grande poder de expressividade, permitindo representar com facilidade todas as relações de causalidade entre processos em situações de: sequencialidade, conflito, concorrência e a sincronização [8] [10].

De acordo com [2]: "A Rede de Petri é uma ferramenta gráfica e matemática que se adapta bem a um grande número de aplicações em que as noções de eventos e de evoluções simultâneas são importantes." Dentre estas aplicações estão: avaliação de desempenho, análise e verificação formal em eventos discretos, protocolos de comunicação, sistemas de informação (organização de empresas), sistemas de transporte, logística, gerenciamento de base de dados, interface homem-máquina e multimídia [2].

Uma Rede de Petri é vista também como um tipo particular de grafo orientado que permite modelar as propriedades estáticas de um sistema de eventos discretos, constituído de dois tipos de nós: as transições (que correspondem aos eventos que caracterizam as mudanças de estado do sistema), e os lugares (que correspondem às condições que devem ser certificadas para os eventos acontecerem) interligados por arcos direcionados ponderados [9].

Sendo assim, a representação gráfica de uma Rede de Petri consiste de lugar, transição e ficha (*token*). Segundo [2] cada elemento pode ser definido como (Figura 2):

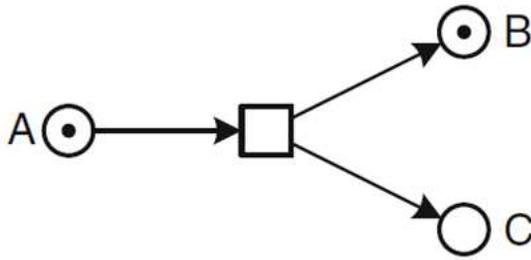


Figura 2. Rede de Petri [3].

- Lugar: (representado por um círculo) pode ser interpretado como uma condição, em estado parcial, uma espera, um procedimento.
- Transição: (representada por barra ou retângulo) é associada a um evento que ocorre no sistema.
- Ficha: (representado por um ponto em um lugar, *token*) é um indicador significando que a condição associada ao lugar é verificada. Pode representar um objeto, uma estrutura de dados.

Os lugares e as transições são conectadas por arcos (setas). Lugares podem conter fichas e a quantidade de fichas em um lugar é chamado de marcação. Um arco de entrada conecta um lugar para uma transição e um arco de saída conecta uma transição para um lugar. Quando uma transição é acionada, as fichas nos lugares ligadas ao arco de entrada são removidas e os *tokens* são adicionados aos lugares ligados aos arcos de saída. Uma transição está ativa e pode ser emitida se a quantidade de *tokens* determinada pelos arcos de entrada existe nos respectivos lugares. A quantidade de símbolos gerados pelos arcos de saída não são necessariamente o mesmo que o removido por os arcos de entrada [10].

Existem vários tipos de Redes de Petri, dentre elas temos as Redes de Petri Coloridas, as temporizadas, as estocásticas. Sendo que as Redes de Petri Estocásticas, segundo [7] surgiram como um modelo de performance, sendo a principal ferramenta para sistemas distribuídos e de simulação de eventos discretos, tornando-se uma ótima alternativa para a sua avaliação de desempenho.

IV. CONCLUSÃO

Este trabalho apresentou uma proposta para criar um modelo formal de simulação ainda na fase de projeto utilizando Redes de Petri. Utilizou-se, como estudo de caso o modelo conceitual de um problema voltado à área de publicidade e propaganda.

Busca-se com este trabalho compreender o funcionamento de uma solução de integração e encontrar gargalos de performance, por meio da simulação de eventos discretos, com o intuito de reduzir o tempo e custos, pois antecipa a identificação de gargalos.

A evolução tecnológica rápida e constante faz com que as empresas tenham a necessidade de atualizar seu ecossistema de *software* com frequência, em consequência as

suas aplicações tornam-se heterogêneas e seu funcionamento isolado das demais aplicações. Diante disso, percebe-se a necessidade da Integração de Aplicações Empresariais, que busca metodologias, ferramentas e tecnologias para desenvolver uma solução de integração, que visa facilitar a interação das aplicações, fazendo com que estas troquem informações, dados e oportunizem novas funcionalidades.

REFERÊNCIAS

- [1] Guilherme Henrique Schiefelbein Arruda, Sandro Sawicki, Rafael Z. Frantz, and Fabricia Roos-Frantz. Desenvolvimento de um modelo de simulação baseado em uma solução de integração real utilizando a ferramenta prism. *Salão do Conhecimento*, 1(1), 2015.
- [2] Janette Cardoso and Robert Valette. Redes de petri. 1997.
- [3] Jörg Desel and Wolfgang Reisig. The concepts of petri nets. *Software and Systems Modeling (SoSyM)*, 14(2):669–683, 2015.
- [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] 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. 2014.
- [6] David S. Linthicum. *Enterprise application integration*. Addison-Wesley Professional, 2000.
- [7] Y Narahari, K Suryanarayanan, and NV Subba Reddy. Discrete event simulation of distributed systems using stochastic petri nets. In *TENCON'89. Fourth IEEE Region 10 International Conference*, pages 622–625. IEEE, 1989.
- [8] Reynaldo Chile Palomino. *Uma Abordagem para a Modelagem, Análise e Controle de Sistemas de Produção Utilizando Redes de Petri. 120 f.* PhD thesis, Dissertação. Engenharia de Produção. Universidade Federal de Santa Catarina, 1995.
- [9] James L Peterson. Petri net theory and the modeling of systems. 1981.
- [10] Fabricia Roos-Frantz, Manuel Binelo, Rafael Z. Frantz, Sandro Sawicki, and Vitor Basto-Fernandes. Using petri nets to enable the simulation of application integration solutions conceptual models. 2015.
- [11] Sandro Sawicki, Rafael Z. Frantz, Vitor Manuel Basto Fernandes, Fabricia Roos-Frantz, Iryna Yevseyeva, and Rafael Corchuelo. Characterising enterprise application integration solutions as discreteevent system. *IGI Global*, 2015.
- [12] Robert E. Shannon. *Systems simulation: the art and science*, volume 1. Prentice-Hall Englewood Cliffs, NJ, 1975.

Um Modelo Matemático para a Simulação de uma Solução de Integração baseado no Formalismo Redes de Petri

Alexsandro Queiroz Lencina

Universidade Regional do Noroeste do Estado do Rio Grande do Sul

Departamento de Ciências Exatas e Engenharias

Ijuí, RS, Brasil

alexsandro.queiroz@iffarroupilha.edu.br

Resumo—O ecossistema de software de uma empresa é composto por diversas aplicações, que originalmente não foram criadas para trabalharem conjuntamente. Esta heterogeneidade dificulta a integração entre as aplicações, pois geralmente são desenvolvidas sem a preocupação de integração. A área de integração de aplicações empresariais trata da integração dessas aplicações existentes no ecossistema de software das empresas, por meio de uma solução de integração. Neste artigo, propomos um modelo matemático de simulação equivalente ao modelo conceitual da solução de integração em Guaraná DSL, em que os elementos do modelo conceitual são traduzidos para uma Rede de Petri. As Redes de Petri oferecem uma linguagem simples e de poucos elementos, sua representação matemática é formal e possui ferramentas de simulação. O desenvolvimento deste modelo possibilita fazer futuras análises por meio da simulação.

Palavras-chave: Modelagem Matemática; Integração de Aplicações Empresariais; Redes de Petri.

I. INTRODUÇÃO

Hoje em dia, uma quantidade expressiva de empresas baseia seu trabalho em sistemas informatizados, os quais constituem o chamado "ecossistema de software". A porcentagem do trabalho que corresponde a estes sistemas ainda é variável, mas tem apresentado uma tendência ao crescimento na maioria das organizações, as quais precisam utilizar os seus ecossistemas de software para apoiar e aperfeiçoar os seus processos de negócios [1]. Esses ecossistemas são compostos de muitas aplicações, normalmente concebidas sem levar em conta sua possível integração, podendo apoiar campos diversos, tais como a contabilidade, a comunicação com clientes e a gestão de recursos humanos.

Dentro da área de Engenharia de Software, o campo de estudos conhecido como Integração de Aplicações Empresariais [2] busca proporcionar metodologias, técnicas e ferramentas para a concepção e a implementação de soluções de integração. Em termos gerais, uma solução de integração tem como objetivo orquestrar um conjunto de aplicações para mantê-las sincronizadas ou proporcionar novas funcionalidades que possam ser construídas a partir daquelas já existentes. Uma solução de integração é composta por processos que contêm lógica de integração e portas de comunicação, que conectam processos ou aplicações do ecossistema à solução de integração.

Conforme a Figura 1, uma solução de integração está composta por processos que contêm lógica de integração e portas de comunicação, que conectam processos ou aplicações do ecossistema à solução de integração.

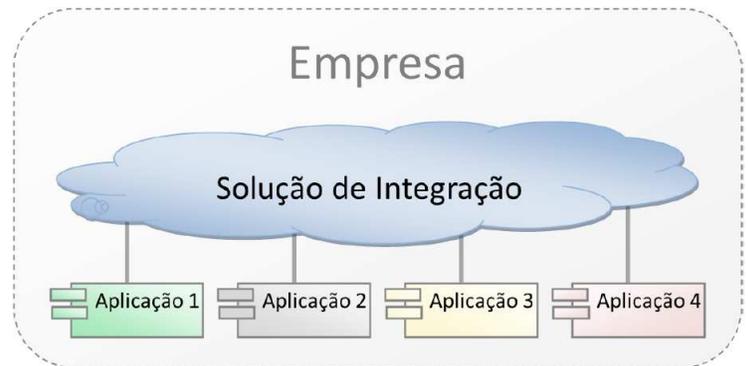


Figura 1. Solução de Integração.

Neste artigo, propomos a tradução do modelo em Guaraná em um modelo de Redes de Petri Estocásticas, esta tradução traz vários benefícios, o primeiro é o fato que as Redes de Petri são um modelo matemático consolidado, que pode ser traduzido em outros modelos formais além de fornecer várias análises do comportamento do modelo, outro é que Redes de Petri tem componentes simples (lugares, transições e arcos), são muito generalizáveis e existem inúmeras ferramentas disponíveis para análise e simulação.

II. ESTUDO DE CASO

O Estudo de caso consiste num problema de integração do sistema real de currículos da agência nacional e tecnologia de Portugal, a qual envolve a interação de quatro aplicações externas, constituídas pela caracterização da unidade de pesquisa local, Plataforma DeGois, Aplicação CMS e Web Of Science. A solução de integração proposta, está apresentado na Figura 2 é modelada utilizando o Guaraná DSL. Uma entrada XML do documento chamado Pesquisadores.XML está presente em "A caracterização da unidade de pesquisa local" que refere-se a entrada inicial e principal para a solução de integração.

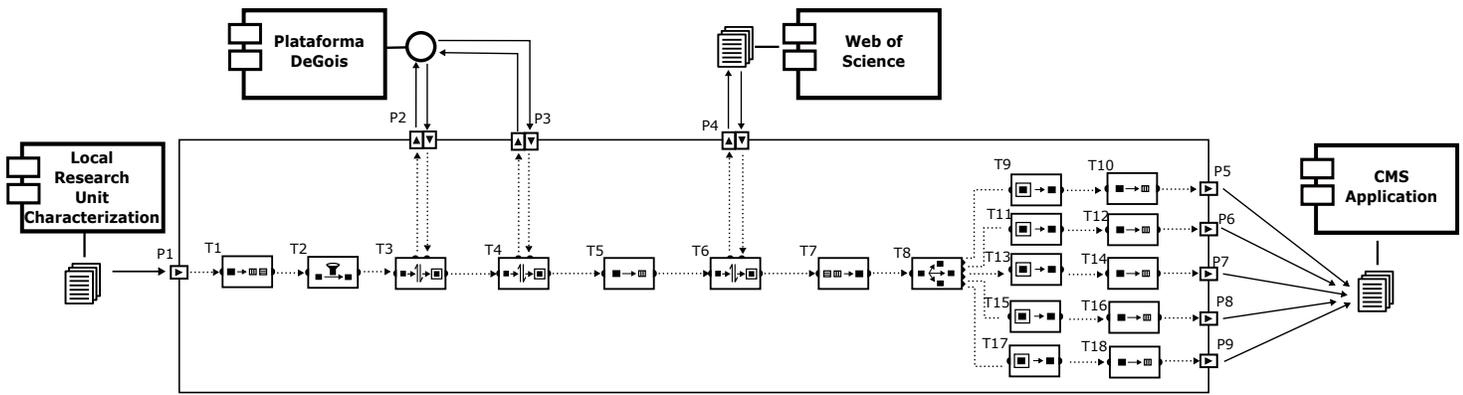


Figura 2. Modelo Conceitual.

Este documento irá conter informações sobre pesquisadores (tag “Pesquisadores”), nomeadamente o seu estado (atributo “Status”), o grupo de pesquisa a que ele ou ela pertencem (atributo “Grupo”), o nome do pesquisador para fazer uma busca do seu CV em “PlataformaDeGois®” (atributo “NameForDeGoisSearch”), etc. Este documento é atualizado com dados da “PlataformaDeGois®”(por exemplo, pesquisador CV última atualização em “PlataformaDeGois®”), e suas informações sobre pesquisadores alimenta e provoca todas as transformações que estão ocorrendo dentro da solução de integração.

A saída da solução de integração consiste em um conjunto de documentos HTML que são encaminhados para uma instância “Joomla®” CMS, sob a forma de artigos “Joomla®”, conforme [5].

III. REDES DE PETRI ESTOCÁSTICAS

Redes de Petri são grafos bipartidos formados por dois tipos de nós chamados de lugar e transições [3]. A representação dos lugares são dadas por círculos ou elipses que equivalem as variáveis de estado de um sistema e as transições por barras que equivalem às ações realizadas pelo mesmo, porém sem considerar o tempo. As Redes de Petri Estocásticas determinam um atraso nas suas transições, associando às transições uma probabilidade de disparo e altera sua frequência de disparo, assumindo como temporizadas todas as transições.

A representação algébrica das Redes de Petri Estocásticas é semelhante às Redes de Petri, descritas por conjunto de transições, lugares, arcos e marcação inicial, anexando-se conjunto das taxas de disparo de cada transição. A representação gráfica também é semelhante, com alteração nas transições, que continuam sendo representados por retângulos, porém com interior branco para diferenciar das não estocásticas.

A Figura 3 representa a tabela de equivalência entre o Guaraná DSL e Redes de Petri.

IV. EQUIVALÊNCIA ENTRE O MODELO CONCEITUAL E O MODELO DE SIMULAÇÃO

A proposta de tradução do modelo conceitual de solução de integração em modelos em Redes de Petri é possível pela

Guaraná	Grupo/Nome	Funcionalidade	RdP
	Porta de Entrada	Insere mensagens na solução de integração	
	Porta de Saída	Envia mensagens a uma aplicação integrada	
	Porta de Solicitação	Solicita informações à uma aplicação integrada	
	Transformador/Splitter	Adequa formato da mensagem	
	Roteador/Filter	Remove mensagens do fluxo ou permite a sua passagem	
	Enquirer	Tarefa Composta por um correlacionador, enriquecedor de conteúdo e transformador	
	Transformador/Aggregator	Adequa formato da mensagem	
	Roteador/Replicator	Faz cópias idênticas de uma mensagem	
	Modificador/Slimmer	Adiciona conteúdo a uma mensagem	
	Transformador/Traslator	Adequa formato da mensagem	
	Correlacionador	Processa juntamente mensagens com alguma relação	
	Enriquecedor de Conteúdo	Adiciona conteúdo a uma mensagem	

Figura 3. Tabela de Equivalência

relação análoga entre seus componentes e pela semelhança do modelo de execução. Os tokens das Redes de Petri são análogos às mensagens da solução de integração, e os lugares são equivalentes aos slots, que atuam como buffers, mesmo havendo uma equivalência entre componentes, cada tarefa do Guaraná tem uma semântica implementada que não pode ser traduzida.

A tradução de todos os componentes do Guaraná é possível, porém muitos elementos são mapeados para Redes de Petri idênticos conforme aborda [4], assim a Figura 4 mostra a Rede de Petri que modela o estudo de caso apresentado na Figura 2. A rede foi construída seguindo a tradução das tarefas

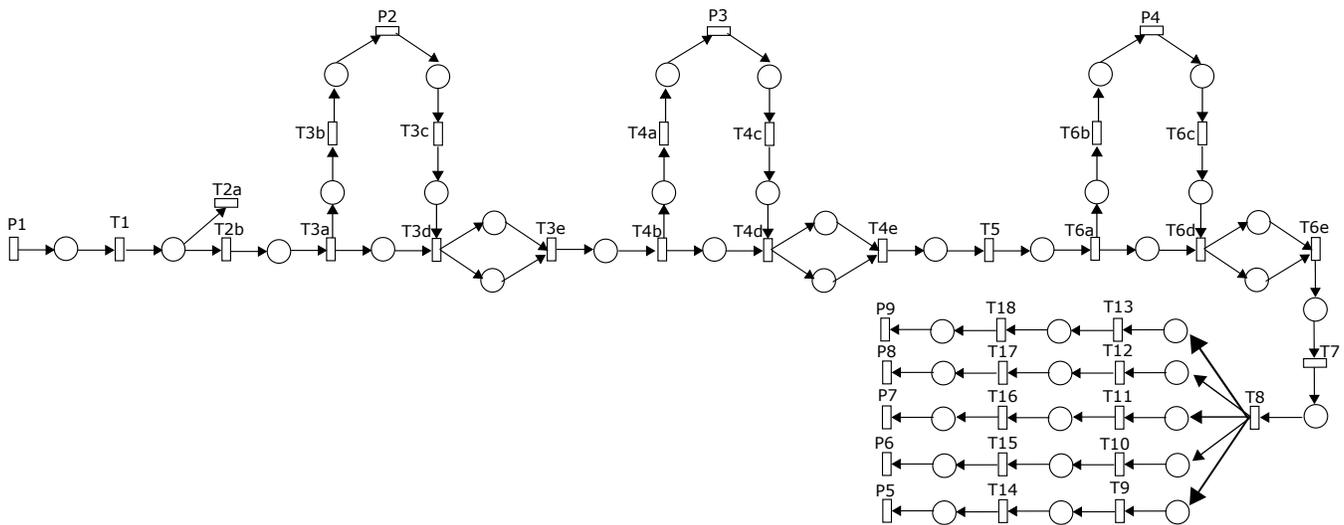


Figura 4. Modelo de Simulação

em Guaraná DSL para as respectivas Redes de Petri. Estas redes estavam ligadas para cada estado na mesma forma das interligações das tarefas mostrado na Figura 3. Na Rede de Petri resultante, as transcrições foram denominadas de acordo com suas tarefas correspondentes e os lugares foram nomeados de acordo com seus respectivos slots.

V. CONCLUSÃO

O Guaraná oferece um conjunto de ferramentas e metodologias para apoiar a construção de soluções de integração de aplicações empresariais. Uma vez que para analisar uma solução de integração dependem de várias atividades relacionadas com a construção, execução, e recolhimento de dados de tal execução, associado o alto custo, risco e tempo de desenvolvimento gasto associado a essas atividades, além do risco de conter gargalos de desempenho em seus componentes. Neste artigo foi proposto a tradução de um modelo conceitual no Guaraná DSL para um modelo de simulação em Redes de Petri, possibilitando assim a sua implementação em uma ferramenta de simulação para uma análise do comportamento e a verificação de possíveis gargalos de desempenho.

Em trabalhos futuros, será proposto aplicar o modelo de simulação equivalente ao modelo conceitual a uma ferramenta de simulação de Redes de Petri Estocásticas, podendo ser utilizado funções de distribuição probabilística que simulam entradas, simulando sob cenários de diferentes cargas de entrada a fim de possibilitar a identificação de gargalos de desempenho que possam vir a ocorrer na solução em cenários de grande demanda.

REFERÊNCIAS

[1] 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. *XIX Jornada de Pesquisa*, pages 1–11, 2014.

[2] Gregor Hohpe and Bobby Woolf. *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional, 2004.

[3] Carl Adam Petri. *Kommunikation mit automaten*. 1962.

[4] Fabricia Roos-Frantz, Manuel Binelo, Rafael Z Frantz, Sandro Sawicki, and Vitor Basto-Fernandes. Using petri nets to enable the simulation of application integration solutions conceptual models.

[5] Fernando Rosa Sequeira, Rafael Z. Frantz, Iryna Yevseyeva, Michael TM Emmerich, and Vitor Basto-Fernandes. An eai based integration solution for science and research outcomes information management. *Procedia Computer Science*, 64:894–901, 2015.

Modelagem e desenvolvimento de uma solução de integração utilizando Guaraná Cloud

Edinaldo G. da Silva

Universidade Regional do Noroeste do Estado do Rio Grande do Sul

Departamento de Ciências Exatas e Engenharias

Ijuí, RS, Brasil

edinaldo.silva@unijui.edu.br

Resumo—A integração de aplicações é um assunto cada vez mais abordado na área de Engenharia de Software. A criação de uma única aplicação que atenda a todas as necessidades de uma empresa é inviável, tornando necessário o estudo de ferramentas destinadas a integrar as diferentes aplicações presentes no ecossistema de software de uma empresa. Problemas de integração surgem de situações cotidianas e por vezes não se nota que rotinas de trabalho poderiam ser mais simples se algumas das aplicações que fazem parte de um ecossistema de software tivessem a capacidade de trocar informações ou até mesmo funcionalidades. A tarefa de atualização de currículos por parte de pesquisadores por vezes acarreta em retrabalho, já que é necessário que as informações sejam atualizadas em mais de uma aplicação, custando por consequência mais tempo para realização de uma única atividade. Tendo em vista tal problema, propõe-se uma solução de integração, que serve de suporte para pesquisadores, tornando necessário que os currículos sejam atualizados em apenas um local, não sendo necessário alterar as aplicações existentes.

Palavras-chave: Integração de Aplicações Empresariais; Guaraná Cloud

I. INTRODUÇÃO

Empresas tem se tornado cada vez mais dependentes de aplicações que automatizam as suas mais diversas necessidades. Com o crescimento dos ecossistemas de software das mesmas e a evolução de seus processos de negócio torna-se necessário que as mais diversas aplicações presentes nestes ecossistemas troquem dados ou até mesmo funcionalidades. Sabe-se que o desenvolvimento de uma única aplicação que atenda à todas as necessidades de uma empresa é financeiramente inviável, torna-se portanto necessário, o estudo de soluções que possibilitem a integração das aplicações que fazem parte de seus ecossistemas de software. Entretanto a tarefa de integrar soluções empresariais não é simples. É preciso observar que as aplicações a serem integradas podem ser proprietárias, terem sido desenvolvidas por diferentes empresas ou até mesmo em diferentes linguagens de programação. Além disso, é importante que as aplicações após serem integradas não criem dependências entre si, de modo que seja possível adicionar a elas novas funcionalidades, impedindo também que ecossistema seja prejudicado por eventuais falhas em uma das aplicações integradas [1].

Segundo Gregor Hohpe e Bobby Woolf uma solução de integração consiste basicamente em uma nova aplicação que deverá ser capaz de realizar a troca de dados ou funcionalida-

des 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].

A partir de um estudo sobre integração de aplicações empresariais, é possível encontrar uma variedade de tecnologias destinadas a tal finalidade, dentre elas destacam-se: Guaraná, Spring Integration, Mule e Apache Camel. Para modelagem e implementação do caso de estudo deste artigo, será utilizado Guaraná. Tal escolha justifica-se pelo fato de o mesmo possuir um DSL que prove a engenheiros de software diversas ferramentas capazes de conceber e implementar soluções de integração a um custo razoável [1]. É importante observar que o DSL nos prove uma representação gráfica do Guaraná, sendo que a implementação de soluções de integração é feita através da plataforma WEB chamada Guaraná Cloud.

II. CASO DE ESTUDO

A. O ecossistema

O problema de integração proposto neste artigo é real e propõe-se a gerar uma página HTML a partir de currículos de pesquisadores (professores e alunos), disponíveis na plataforma Lattes do CNPQ. Para o funcionamento de tal integração necessita-se inicialmente que seja realizado o download dos currículos no formato XML, disponível na plataforma Lattes. Tais currículos devem ser armazenados em um repositório de arquivos de onde o motor de integração do Guaraná Cloud os buscará, os converterá e posteriormente os enviará a outro repositório de arquivos no formato HTML.

B. O modelo conceitual

Para que um problema de integração seja resolvido é necessário que após o mesmo ser apresentado e analisado, seja feita uma modelagem conceitual, que no caso do Guaraná, apresentará todas as estruturas necessárias ao funcionamento correto da solução de integração. A Figura 1 apresenta o modelo conceitual da solução de integração proposta. O modelo foi feito utilizando o Guaraná DSL, disponível no Guaraná Cloud. Observa-se que existem na solução de integração 10 tarefas (T2 à T11), uma porta de entrada (P1) e 4 portas de saída (P2 à P5). Cada um dos currículos presentes em um diretório específico do DropBox entra na solução de integração através da porta P1, passando posteriormente a tarefa “T1” que é do

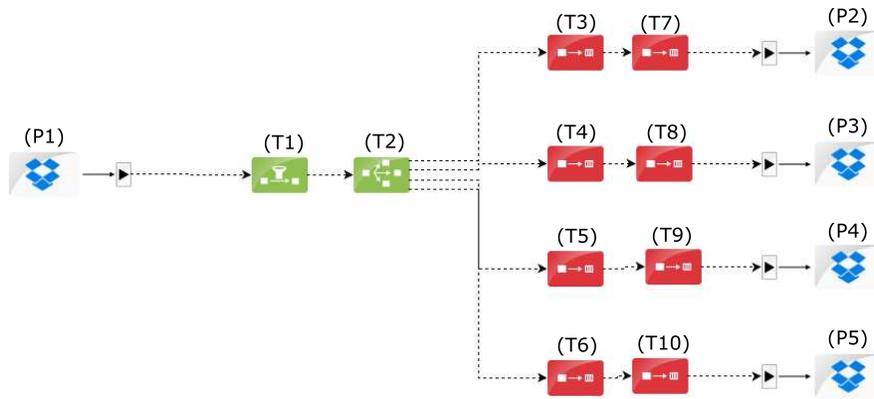


Figura 1. Modelo conceitual para a solução de integração.

tipo “Filter”. Tal tarefa deixará que somente arquivos com extensão XML sigam o fluxo da solução de integração. Caso a condição definida na tarefa “Filter” seja atendida, o arquivo será encaminhado a tarefa “Replicator”, esta tarefa fará quatro cópias do arquivo original e as encaminhará as tarefas “Translator” representadas na Figura 1 por “T3, T4, T5 e T6”. Cada uma das tarefas “Translator” é destinada a um diferente objetivo, por exemplo, a tarefa “T3” retirará do currículo do pesquisador somente as informações referentes a artigos publicados em periódicos, enquanto a tarefa “T4” retirará somente as informações referentes a capítulos escritos em livros.

Passadas as tarefas T3 à T6, os currículos que entraram no formato XML, são agora encaminhados no formato HTML as tarefas “T7, T8, T9 e T10”. Tais tarefas também são do tipo “Translator” porém atendem a uma finalidade diferente, elas recebem o arquivo gerado pelas tarefas anteriores e adicionará nele o caminho destino onde ele será armazenado. O último passo da solução está nas portas de saída “P2, P3, P4 e P5”, tais portas neste caso de estudo gravarão o arquivo resultante do processo de integração em um diretório Dropbox a ser definido pelo programador.

```

<xsl:for-each select="//CURRICULO-VITAE/PRODUCAO-BIBLIOGRAFICA/
TRABALHOS-EM-EVENTOS/ TRABALHO-EM-EVENTOS">
  <TR>
    <TD>Titulo:
    <xsl:value-of select="DADOS-BASICOS-DO-TRABALHO/@TITULO-DO-TRABALHO"/>
    <xsl:for-each select="//CURRICULO-VITAE/PRODUCAO-BIBLIOGRAFICA/
TRABALHOS-EM-EVENTOS/TRABALHO-EM-EVENTOS/AUTORES">
      <xsl:if test="@NOME-COMPLETO-DO-AUTOR!=''">
        <xsl:value-of select="@NOME-COMPLETO-DO-AUTOR"/>
      </xsl:if>
    </xsl:for-each>
    Local: <xsl:value-of select="DETALHAMENTO-DO-TRABALHO/
@TITULO-DOS-ANAIS-OU-PROCEEDINGS"/>
  </TD>
</TR>
</xsl:for-each>

```

Figura 2. Código XSLT.

C. Implementação

Passada a fase de modelagem da solução de integração utilizando o DSL disponibilizado, torna-se necessário que o

modelo seja implementado. Guaraná Cloud possui um DSL que além de oferecer diversas ferramentas que possibilitam conceber soluções de integração, torna também possível a implementação das mesmas a um custo razoável [1]. É importante observar que cada uma das tarefas apresentadas na Figura 1 possui características diferentes, tendo portanto implementações diferentes. Para que seja implementada a tarefa “T1”, por exemplo, é necessário que se tenha conhecimento da linguagem XPath. A Figura 2, apresenta uma parte de uma proposta de código XSLT, que tem por objetivo retirar dos currículos informações referentes a trabalhos publicados em eventos, buscando obter o nome do autor, o título do trabalho bem como o local de acontecimento do evento. Tal código é destinado a tarefas do tipo “Translator”, podendo ser usado nas tarefas “T3, T4, T5 e T6” apresentadas na Figura 1.

III. CONCLUSÃO

A partir do estudo e implementação de soluções de integração utilizando Guaraná Cloud, nota-se que o mesmo propicia uma vasta gama de ferramentas que possibilitam o desenvolvimento de soluções de integração tanto de baixo quanto de alto grau de complexidade. Utilizando-se de tal tecnologia foi possível modelar e implementar uma solução de integração que possibilita que currículos extraídos da plataforma Lattes do CNPq sejam analisados e transformados de acordo com os parâmetros escolhidos, tendo como saída uma página HTML que pode ser adaptada as necessidades de universidades ou grupos de pesquisa por exemplo, automatizando a atualização das informações referentes a publicações de seus pesquisadores em seus respectivos sites.

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.

Modelagem matemática e computacional para análise do comportamento de soluções de integração de aplicações através da criação de modelos de simulação em uma empresa de venda e aluguel de equipamentos

Leandro Fritzen Klem,
Universidade Regional do Noroeste do Estado do Rio Grande do Sul
Departamento de Ciências Exatas e Engenharias
Ijuí, RS, Brasil
leandro.klem@gmail.com

Resumo—Uma empresa precisa ter informações que possam ser consultadas rapidamente a fim de gerar relatórios precisos e tomar decisões rápidas sobre seu negócio. Para ter essas informações as empresas estão cada vez mais investindo, e dependendo, de tecnologias que ajudem na hora de projetar, armazenar, guardar seus produtos e gerar relatórios. As aplicações são um conjunto de softwares, que compõe o ecossistema de software das empresas. Geralmente, diferentes aplicações são desenvolvidas inicialmente sem a preocupação de alteração e da troca de informações com outras aplicações, ou seja, são aplicativos isolados. É a partir destas situações que surge a Integração de Aplicações Empresariais, cuja proposta é fazer com que dados de diferentes formatos, que utilizam diferentes protocolos de comunicação e formas de armazenamento funcionem juntos da forma mais eficiente possível. O Guaraná DSL é uma das tecnologias que possibilita projetar modelos conceituais de soluções de integração, utilizando uma sintaxe concreta gráfica e intuitiva. No entanto, a integração de aplicações não é uma tarefa trivial e o desenvolvimento da solução envolve além de custos (tempo e recursos), riscos como aparecimento de bugs que na maioria das vezes são observados somente após a implementação. Neste sentido, esta pesquisa propõe a análise do comportamento e a identificação de gargalos de performance ainda na fase de projeto, levando em consideração o modelo conceitual mediante o desenvolvimento de um modelo de simulação equivalente ao modelo conceitual, utilizando uma técnica matemática chamada Redes de Petri.

Palavras-chave: Simulação, Redes de Petri, Gargalos de Performance, Modelo Conceitual, Soluções de Integração, Venda Hospitalar, Enterprise Application Integration; Fault-tolerance.

I. INTRODUÇÃO

Uma empresa, seja ela grande, média ou pequena, precisa ter informações que possam ser consultadas rapidamente a fim de gerar relatórios precisos. Para ter essas informações as empresas estão cada vez mais investindo, e dependendo, de tecnologias que ajudem na hora de armazenar, guardar e gerar tais relatórios, resolvendo assim alguns de seus problemas.

Nosso estudo está focado em uma empresa especializada na venda, aluguel e suporte técnico de equipamentos hospitalares, onde são utilizados diversas aplicações para dar suporte aos

negócios. Tais aplicações foram adquiridas de diversos fornecedores e cada uma resolvia um problema imediato. Como resultado dessa prática, surgiu um ecossistema heterogêneo composto por muitas aplicações, a maioria dessas não foi projetada para interagir umas com as outras [7]. À medida que a empresa cresceu, sentiu-se a necessidade de um sistema mais ágil, fácil, e principalmente, integrado.

Uma alternativa para o problema da comunicação entre aplicações diferentes é o uso de uma solução de integração de aplicações. A integração de aplicações pode ser realizada de varias maneiras e utilizar diversos recursos.

Para este trabalho, é proposto estudar um modelo conceitual do projeto de solução de integração de aplicações dessa empresa, criado através do uso de uma linguagem de domínio específico (DSL), através do Guaraná DSL.

Depois verificar o comportamento do modelo conceitual utilizando simulação. Essa simulação, quando submetidas a cenários críticos de funcionamento, identifica possíveis gargalos de performance.

E a partir deste modelo conceitual desenvolver um modelo matemático, ou modelo formal, de simulação utilizando Redes de Petri.

A partir da verificação da simulação do modelo formal com Redes de Petri, apresentar soluções aos gargalos de performance identificados.

II. ENTERPRISE APPLICATION INTEGRATION

Enterprise Application Integration (EAI) ou solução de integração de aplicações é um conjunto de ferramentas e tecnologias cuja proposta é fazer com que dados de diferentes formatos, que utilizam diferentes protocolos de comunicação e formas de armazenamento funcionem juntos da forma mais eficiente possível [3]. Seu principal atrativo é a implementação de novas funcionalidades utilizando-se da base tecnológica dos sistemas legados já em funcionamento [8].

EAI é uma combinação de tecnologias com a função de organizar os processos de negócios, propondo integração entre as aplicações sem haver alterações nas suas estruturas de dados. A EAI tem como diferencial principal a integração de forma sistemática [8].

A análise de soluções de integração de aplicativos corporativos para prever seu comportamento e encontrar possível gargalos de desempenho é uma importante atividade que contribui para aumentar a qualidade das soluções entregues.

A EAI representa um sistema de informação unificado, com acesso fácil a informações valiosas e decisivas, possibilitando uma varredura constante de todos os sistemas integrados em busca da informação requerida. Identificamos diferentes aplicações integradas em torno de uma plataforma de integração que promove o livre intercâmbio de informações entre as aplicações e delas com os usuários do sistema.

Segundo [8] a maioria dos problemas com o desenvolvimento de software deriva de tentativas de integrar com sistemas existentes. A construção de soluções de integração prevê um alto investimento, porque exige um bom conhecimento sobre uma tecnologia de integração. Qualquer falha na solução de integração construída pode causar a falha na execução de todos os softwares.

Para evitar tais problemas, é possível realizar simulação utilizando um modelo executável dos requisitos em conjunto com técnicas matemáticas.

III. SIMULAÇÃO

Um sistema pode ter definições diferentes, dependendo da área em que é usado, sendo usado numa grande variedade de modos. Uma definição mais sucinta é a utilizada por [2] onde um sistema é um agrupamento de partes que operam juntas. Os próximos passos do trabalho são a modelagem do problema, para analisar o comportamento e a identificação de gargalos de performance ainda na fase de projeto a partir de um modelo matemático de simulação utilizando Redes de Petri.

Qualquer conjunto de partes, desde que unidas entre si e com objetivos comuns, pode ser considerado um sistema, ou ainda em outras palavras, um conjunto de elementos interdependentes que interagem com objetivos comuns formando um todo, e onde cada um dos elementos, somando seus comportamentos, resulte em um todo maior e unitário cujo resultado é maior do que as unidades poderiam ter se funcionassem independentes. [10].

A simulação tem como vantagens a diminuição dos custos e do tempo utilizando-se um modelo executável dos requisitos em conjunto com as técnicas matemáticas [6], no caso, as Redes de Petri. Além da diminuição dos custos, a experimentação de um sistema real envolve riscos, tanto materiais quanto humanos, o que não ocorre com um sistema simulado [2].

IV. MODELO

A experimentação com modelos consiste na criação de um modelo conceitual a fim de representar o funcionamento do sistema real. O modelo é abstrato, uma vez que tende a se aproximar do verdadeiro comportamento da realidade [10].

O comportamento do sistema é estudado pela construção de um modelo de simulação. Este modelo normalmente toma a forma de um conjunto de considerações relacionadas a operação do sistema [9]. Estas considerações são expressas através de relações matemáticas, lógicas e simbólicas entre as entidades, ou objetos de interesse, do sistema. Uma vez construído e validado, um modelo pode ser usado para investigar uma grande quantidade de questões do tipo “e se...” sobre o sistema do mundo real e possíveis gargalos [2].

V. ESTILOS DE INTEGRAÇÃO

A integração de aplicações pode ser realizada de varias maneiras e utilizar de diversos recursos. Não é possível determinar o melhor a ser considerado para uma organização, pois cada um deles possui seus prós e contras, então é preciso analisar os estilos e determinar qual deles mais acrescentará benefícios a aplicação. Dentre os estilos mais utilizados destaca-se arquivo, banco de dados, chamado de procedimento remoto e sistema de mensagens [7].

Há varias tecnologias disponíveis baseadas em mensagem seguindo os padrões de integração, documentados por Hohpe Woolf, que possibilitam projetar modelos conceituais de soluções de integração. Entre elas podemos citar o Camel, Mule, Spring e Guaraná. As tecnologias utilizadas para desenvolvimento de modelos de solução de integração usam uma linguagem de domínio específico (DSL). A DSL propõe uma linguagem própria para interpretação da solução de integração, dentro de um contexto, de modo a simplificar códigos complexos de modo a facilitar a compreensão da sua estrutura e funcionamento. A linguagem DSL tem excelência dentro do domínio que opera, porém perde eficiência se submetidas a situações fora do seu domínio [5].

Neste artigo, optamos pelo Guaraná DSL, que é uma tecnologia que possibilita projetar modelos conceituais de soluções de integração, utilizando uma sintaxe concreta, gráfica e intuitiva.

VI. REDES DE PETRI

Propostas por Carl Adam Petri, as redes de Petri, segundo [4], constituem-se em uma técnica de modelagem que permite a representação de sistemas, utilizando como alicerce uma forte base matemática. Essa técnica possui a particularidade de permitir modelar sistemas paralelos, concorrentes, assíncronos e não-determinísticos. Segundo [1] é também uma ferramenta gráfica de uso geral. Juntas, ferramenta gráfica e matemática, permitem modelar o comportamento de sistemas dinâmicos a eventos discretos, descrever as relações existentes entre condições e eventos e visualizar propriedades tais como paralelismo, sincronização e compartilhamento de recursos.

VII. CASO DE ESTUDO

Em determinada empresa especializada na venda, aluguel e suporte técnico de equipamentos hospitalares são utilizadas diversas aplicações para dar suporte aos negócios. Essas aplicações estão isoladas, ou seja, não conseguem trocar

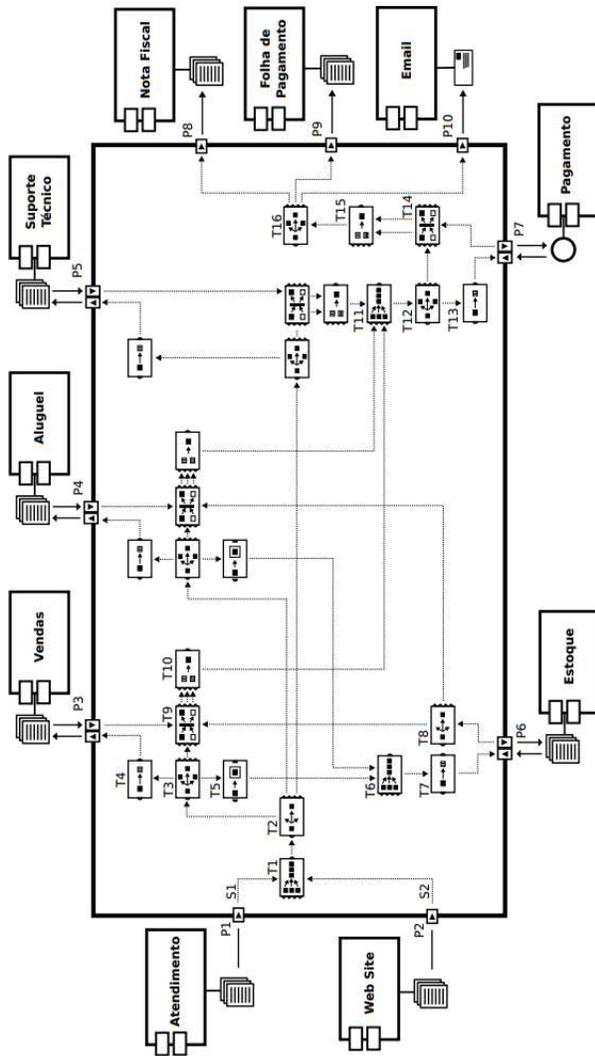


Figura 1. Solução de integração do problema de Venda Hospitalar.

informações ou trabalharem de forma orquestrada umas com as outras.

Tais aplicações foram adquiridas de diversos fornecedores e cada uma resolvia um problema imediato, mas à medida que a empresa cresceu e novos funcionários foram sendo contratados sentiu-se a necessidade de um sistema mais ágil, fácil, e principalmente, que trocasse informações de forma automatizada, já que estas estavam sendo realizadas de forma manual.

VIII. MODELO CONCEITUAL DE INTEGRAÇÃO

A solução de integração proposta foi modelada baseando-se na necessidade de automatização das tarefas, troca rápida e fácil de informações entre os aplicativos, como mostra a figura 1

IX. 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, para analisar o comportamento e a identificação de gargalos de performance ainda na fase de projeto a partir de um modelo matemático de simulação utilizando Redes de Petri.

REFERÊNCIAS

- [1] Natália Maria Cordeiro Barroso, José Marques Soares, Giovanni Cordeiro Barroso, João Cesar Moura Mota, and Hermínio Borges Neto. Modelagem de conceitos e processos matemáticos por redes de petri coloridas: o caso da integrabilidade de funções reais/modeling of concepts and mathematical processes by petri nets: the case of integrability of real functions. *Bolema*, 27(45):75, 2013.
- [2] Leonardo Chwif. *Redução de modelos de simulação de eventos discretos na sua concepção: uma abordagem causal*. PhD thesis, Universidade de São Paulo, 1999.
- [3] Shirleide Pereira da Silva Cruz. A construção da profissionalidade polivalente na docência nos anos iniciais do ensino fundamental: sentidos atribuídos às práticas por professoras da rede municipal de ensino do recife. 2012.
- [4] Carlos Renato Lisboa Francês. Introdução às redes de petri. *Laboratório de Computação Aplicada, Universidade Federal do Pará*, 2003.
- [5] 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. 2014.
- [6] Robert Gold. Petri nets in software engineering. 2004.
- [7] Gregor Hohpe and Bobby Woolf. *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] Sandro Sawicki, Rafael Z. Frantz, Vitor Manuel Basto Fernandes, Fabricia Roos-Frantz, Iryna Yevseyeva, and Rafael Corchuelo. Characterising enterprise application integration solutions as discreteevent system. *IGI Global*, 2015.
- [10] Ludwig Von Bertalanffy. *Teoria geral dos sistemas*. vozes, 1977.

Framework de Comparação da Evolução do grau de Manutenibilidade da Ferramenta de Integração de Aplicações Apache Camel

Matheus H. Rehbein

Universidade Regional do Noroeste do Estado do Rio Grande do Sul

Departamento de Ciências Exatas e Engenharias

Ijuí, RS, Brasil

matheus.rehbein@unijui.edu.br

Resumo—O desenvolvimento de uma aplicação que abranja todas as necessidades de uma empresa torna-se inviável, assim surge o conceito de integração entre aplicações empresariais. Atualmente existem uma série de ferramentas que dão suporte à integração de aplicações. Uma característica a ser observada entre elas é a respeito de sua manutenção. Quando uma ferramenta possui um alto grau de manutenibilidade em determinado parte do código, torna-se mais complexo mantê-la, fazendo com que a evolução fique mais difícil. Uma prática para definir o nível de manutenção da ferramenta é a partir da análise de métricas de manutenibilidade. Após obter esse nível de manutenção é possível fazer uma previsão que servirá como um indicador de como será a manutenção de uma ferramenta. Para este artigo, utilizou-se a ferramenta Apache Camel, onde apresentou o crescimento de seu grau e manutenibilidade durante seu ciclo de evolução.

Palavras-chave: Integração entre aplicações empresariais; Apache Camel; Manutenibilidade de software.

I. INTRODUÇÃO

Com a evolução de uma empresa, torna-se necessário a utilização de mais de uma aplicação em seu Ecossistema de Software [7]. A utilização de uma única Aplicação que contenha todas as funcionalidades que uma empresa necessita pode se tornar custoso e portanto inviável [2], assim surge a necessidade de uma integração entre as aplicações. Para Hohpe e Woolf [4], esta integração deve reutilizar as funcionalidades das aplicações, e quando surgir uma nova necessidade, implementá-la sem perder seu estado original.

Ferramentas de integração podem constantemente necessitar uma nova funcionalidade, correção de algum problema, ou melhora em sua performance. Essas etapas são conhecidas como Manutenção de Software [8]. A partir de uma extração de dados é possível a realização de uma medição para definir um grau de manutenibilidade de uma ferramenta. Quanto maior for o grau de manutenibilidade, mais complicado será adicionar, aprimorar ou manter uma funcionalidade do código.

II. APACHE CAMEL

Com a necessidade de integração entre aplicações, surgiu a necessidade de criar tecnologias para facilitar este processo. Ferramentas como Apache Camel, Spring Integration, Mule e Guaraná, destacam-se entre as demais [4].

Para este artigo, optou-se pela utilização da ferramenta Apache Camel, que possui código *open-source* e assim pode ser feita a extração de informações. Apache Camel possui um sistema de mensagens assíncronas, fazendo com que uma aplicação possa se comunicar com outra. Segundo Ibsen e Anstey [5] outra característica fundamental, é que o Camel não se restringe a um determinado tipo de dado para fazer os processos. Os autores também comentam que isso é um fator importante, pois poderá ser feita a integração entre as aplicações sem necessitar a conversão para um tipo padrão de dado.

III. CASO DE ESTUDO

A. Contextualização

Um contexto muito discutido na área de engenharia de software é a manutenibilidade de uma ferramenta. Existem técnicas para analisar e definir um grau de manutenção. Uma dessas técnicas é a análise de métricas que são extraídas a partir do código fonte da ferramenta.

Este artigo tem o objetivo de analisar a evolução do grau de manutenibilidade da ferramenta de integração Apache Camel a partir da versão 2.0 até 2.16. Dentre as métricas analisadas foram selecionadas “McCabe cyclomatic complexity” (MCC) e “Number of lines in methods” (MLC). Para fazer a extração das medidas, utilizou-se o software Metrics.

A medida MCC é utilizada para definir o quão complexo é um método, segundo McCabe [6] este valor não deve exceder dez, pois com aumento deste valor, a manutenção deste método se tornará mais difícil. Já a medida MLC define a quantidade de linhas que um método possui, Henderson-Sellers [3] afirma que este valor não pode ser maior que cinquenta. Quanto maior for este valor, mais difícil será de manter e entender o método [1].

B. Resultados

Como ressaltado anteriormente, o valor referente a MCC não deve ultrapassar dez. Na Figura 1 apresentado o valor médio e o valor máximo do MCC, o valor médio manteve-se com pequenas variações nas distintas versões,

Versões	2.0	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8	2.9	2.10	2.11	2.12	2.13	2.14	2.15	2.16
MCC - Mean	1,6	1,61	1,6	1,63	1,66	1,67	1,68	1,67	1,64	1,65	1,67	1,69	1,69	1,7	1,71	1,7	1,69
MCC - Max	42	42	43	46	46	46	46	46	46	46	46	48	48	48	50	50	64
MLC - Mean	4,16	4,14	4,2	4,3	4,41	4,46	4,54	4,52	4,57	4,62	4,71	4,73	4,71	4,75	4,79	4,79	4,73
MLC - Max	103	103	103	113	112	137	141	141	150	153	163	120	122	128	131	135	150

Figura 1. Framework de Comparação da Evolução.

não muito significativas, porém o valor máximo mostrou-se com crescimento. Pode-se observar que a primeira versão analisada do Camel possuiu método com o valor até 42, mais de quatro vezes maior que o recomendado. Este valor foi aumentando até 50 na versão 2.15, e então ocorreu uma variação, fazendo com que chegasse a 64 na versão 2.16. Essa variação ocorreu devido a reestruturação do método “getManagedObjectForProcessor” pertencente a classe “DefaultManagementObjectStrategy” presente no *package* “org.apache.camel.management”. Já o valor médio desta classe encontra-se com 6,83, mesmo possuindo o método com o maior valor de MCC. Sendo que esta classe possui 12 métodos, para que aconteça uma variação deste tipo, é necessário que o valor dos outros métodos contidos na classe sejam próximos a 1. Também foi verificado que todos métodos que possuíram o valor máximo na versão tiveram o maior nível de MLC de seu *package*, maior que o recomendável.

Também na Figura 1, apresenta-se os valores referentes a MLC, sendo eles o valor máximo e a média. Como citado anteriormente, é recomendado que um método não possua mais de cinquenta linhas de código. A média teve uma tendência de crescimento, porém nunca atingiu a quantidade de 5 linhas. Já o valor máximo teve crescimento até a versão 2.10, alcançando o valor de 163 linhas de códigos. O método que possuiu este valor não se encontra na versão seguinte, ele também era responsável pelo maior valor MCC presente. Na versão seguinte, o número máximo caiu para 120, mas mesmo assim continuou crescendo até chegar ao valor 150, na última versão. O valor máximo da última versão é também o método responsável pelo maior MCC, “getManagedObjectForProcessor”. Mesmo o método ficando com um valor elevado, o valor médio da métrica MLC na classe que o método pertence esteve abaixo do recomendado, possuindo 18,17. O *package* da classe que possui o método ficou com a média em 6,63, muito próximo da média geral.

Durante a evolução das versões o valor máximo da medida MCC nunca esteve dentro do recomendado, possuindo versão com até seis vezes maior, que irá resultar em uma manutenção mais complexa. Também identificou-se que os valores referentes ao número máximo de linhas de códigos nos métodos esteve maior que o recomendável, tornando assim mais complicado a manutenção e entendimento deste método. Mesmo tendo uma queda na versão 2.11, ele continuou crescendo e atingiu um valor três vezes maior que o recomendado na última versão. Os valores que representam

as médias, tanto do MCC como do MLC, estiveram bastante estáveis e baixos, isso ocorre porque grande parte dos métodos possuem valores abaixo do recomendado. Portanto, pode-se afirmar que não são todos os métodos que estão com um grau de manutenibilidade alto. Porém, com os valores máximos obtidos a partir do MCC e MLC, provou-se que algumas partes do Camel possuem um alto grau de manutenção, dificultando sua evolução.

IV. CONCLUSÃO

Após análise de distintas versões, ficou visível que a evolução do Apache Camel teve um impacto em seu grau de manutenibilidade referente a complexidade de alguns métodos, tornando a manutenção mais complexa. Tornou-se visível que os métodos que possuem o valor máximo da métrica MCC alto, também possuem um nível de MLC maior que o recomendado, podendo-se afirmar que um método com MLC alto, poderá influenciar na medida MCC do mesmo método.

A partir deste artigo, é possível afirmar que o grau de manutenibilidade da ferramenta Camel vem crescendo e tornando-se cada vez mais complexo. Para diminuir esses efeitos, deve-se reestruturar diversas partes do código. O processo de reestruturação pode se tornar demorado ou até mesmo inviável, podendo assim acarretar em uma ferramenta com um tempo de manutenção possivelmente alto.

REFERÊNCIAS

- [1] Rafael Z. Frantz, Rafael Corchuelo, and Fabricia Roos-Frantz. A methodology to evaluate the maintainability of enterprise application integration frameworks. *International Journal of Web Engineering and Technology*, 10(4):334–354, 2015.
- [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] Brian Henderson-Sellers. *Object-oriented metrics: measures of complexity*. Prentice-Hall, Inc., 1995.
- [4] Gregor Hohpe and Bobby Woolf. *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional, 2004.
- [5] Claus Ibsen and Jonathan Anstey. *Camel in action*. Manning Publications Co., 2010.
- [6] Thomas J. McCabe. A complexity measure. *Software Engineering, IEEE Transactions on*, (4):308–320, 1976.
- [7] David G. Messerschmitt, Clemens Szyperski, et al. Software ecosystem: understanding an indispensable technology and industry. *MIT Press Books*, 2005.
- [8] Jane Radatz, Anne Geraci, and Freny Katki. Ieee standard glossary of software engineering terminology. *IEEE Std*, 610121990(121990):3, 1990.

Modelagem de uma Solução de Integração no Contexto da Administração Pública Municipal

Ivan E. M. Kühne

Universidade Regional do Noroeste do Estado do Rio Grande do Sul
Departamento de Ciências Exatas e Engenharias
Ijuí, RS, Brasil
oitentaetres@gmail.com

Resumo—Pode ser observada uma tendência crescente no uso de sistemas informatizados em organizações de diversas naturezas. Entretanto, a simples adição de novas aplicações nem sempre atinge às necessidades dessas organizações de forma satisfatória. Pode ser necessário recorrer às metodologias, técnicas e ferramentas do campo da Integração de Soluções Empresariais. Entre essas, podemos citar a linguagem Guaraná DSL, que foi criada com o objetivo de possibilitar a modelagem de soluções de integração em alto nível de abstração. Nesse artigo é demonstrado o uso da Guaraná DSL para a modelagem de uma solução de integração para um caso de estudo identificado no contexto da administração pública da cidade de Ijuí - RS: o ecossistema de *software* envolvido na geração de boletos de impostos municipais.

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

I. INTRODUÇÃO

Empiricamente, pode ser observado que a parcela de organizações, seja no setor público ou privado, que dependem de sistemas informatizados para a execução dos seus processos de negócio é cada vez maior. Ao conjunto dessas aplicações dentro de cada organização é dado o nome de ecossistema de *software*. Ainda de forma empírica, pode ser observado que, dentro de cada organização, a parcela de processos de negócio que depende do seu ecossistema de *software* tem mostrado uma tendência crescente.

Entretanto, a simples adição de novas aplicações ao ecossistema de *software* nem sempre cumpre satisfatoriamente com os objetivos da empresa. De acordo com Hohpe and Woolf [3], novas demandas relativas aos processos de negócio vão surgindo ao longo da existência das organizações. Assim, pode tornar-se necessária não só a adição de novas aplicações, mas também a criação de condições para que elas possam trabalhar conjuntamente, através do compartilhamento de informações ou funcionalidades. Essa não é uma tarefa fácil, tendo em vista que as aplicações podem ter sido desenvolvidas em linguagens de programação diferentes, usarem diferentes formatos de dados e arquivos e terem sido desenvolvidas em épocas diferentes e por pessoas ou empresas diferentes. Além disso, as aplicações normalmente não são desenvolvidas tendo em vista a sua futura integração.

Para que a integração seja possível, desenvolveu-se ao longo dos anos, dentro da área da Engenharia de *Software*, o campo da Integração de Soluções Empresariais (EAI), que fornece

metodologias, técnicas e ferramentas para construção de soluções de integração. Segundo Frantz et al. [1], a EAI deve preservar e sincronizar os dados e aplicações já existentes, além de criar novas funcionalidades a partir deles.

II. GUARANÁ DSL

Entre as ferramentas desenvolvidas no campo da EAI está a linguagem de domínio específico (DSL) Guaraná DSL. Conforme Ghosh [2], uma DSL é uma linguagem de programação criada para resolver um tipo específico de problema. Cada DSL é limitada à resolução de problemas dentro do domínio para o qual foi criada, diferentemente das linguagens de propósito geral (GPL), que são capazes de modelar soluções para uma ampla gama de problemas. Entretanto, dentro desse domínio limitado, a DSL é extremamente expressiva, de forma que pode ser compreendida até por pessoas que não possuem familiaridade com linguagens de programação. Outro fator que facilita essa compreensão é que a sintaxe e semântica das DSL modelam conceitos no nível de abstração do domínio do problema, não no nível de abstração do domínio da solução.

A linguagem Guaraná DSL oferece suporte aos padrões de integração documentados por Hohpe and Woolf [3] e se destina à criação de soluções de integração em alto nível de abstração. Como é característico das DSL, seu alto nível de abstração permite que as soluções concebidas e modeladas sejam compreensíveis por profissionais sem familiaridade com a área de Engenharia de *Software*, como gerentes e administradores.

III. CASO DE ESTUDO

O caso de estudo escolhido para este artigo foi identificado no contexto da administração pública municipal da cidade de Ijuí. Ele consiste no ecossistema de *software* envolvido no processo de geração de boletos do Imposto Predial e Territorial Urbano (IPTU) e do Imposto sobre Serviços de Qualquer Natureza (ISSQN), ambos de periodicidade anual, para os contribuintes.

A. *Ecossistema de Software*

O IPTU e o ISSQN são calculados pela aplicação ARcetil. Esse cálculo pode ser ativado de três maneiras. A principal maneira é a ativação automática que é feita anualmente para todos os cidadãos cadastrados, em data pré-determinada. Entretanto,

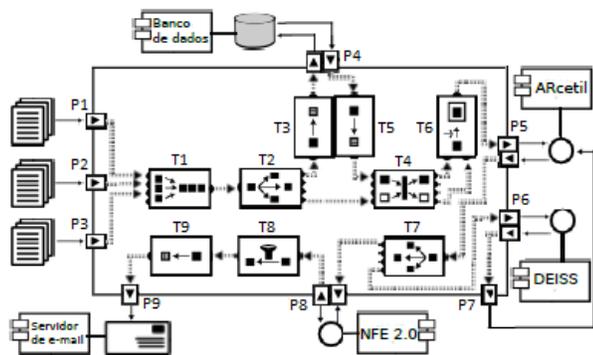


Figura 1. Modelo conceitual da solução de integração proposta.

aqueles contribuintes que possuem vencimentos em aberto podem solicitar uma via atualizada através da *internet* ou no balcão de atendimento. Os dados necessários para o cálculo do IPTU e do ISSQN são obtidos a partir de uma consulta a um banco de dados que é administrado pela Secretaria Municipal da Fazenda.

Usando os dados obtidos, o ARcetil calcula o imposto devido. A partir do valor desse imposto, a aplicação NFE 2.0 gera a nota fiscal eletrônica, que é enviada para o cidadão no formato XML através de um servidor de *e-mail*. Também a partir do imposto calculado, a aplicação DEISS gera a declaração de entrega, que é arquivada pelo ARcetil. Atualmente a integração entre essas aplicações é feita de forma manual, por um operador responsável por cada uma delas. Essa metodologia de trabalho é menos ágil e mais sujeita a erros em relação a um sistema integrado de forma informatizada.

B. Solução de Integração

O modelo conceitual proposto está representado na Figura 1 e busca integrar as aplicações do ecossistema identificado utilizando o estilo de integração baseado em mensagens [3]. O processo conta com três portas de entrada (P1, P2 e P3), que recebem as requisições para o cálculo de imposto de acordo com o método que elas foram feitas (automático, via *internet* ou via balcão de atendimento). Essas três portas de entrada encaminham as solicitações para a mesma tarefa *Merger* T1, que é responsável por juntar esses três fluxos de mensagens em um único. Cada mensagem é encaminhada para a tarefa *Replicator* T2, que faz com que ela seja duplicada. Uma das cópias é encaminhada para a tarefa *Translator* T3 e em seguida para a consulta ao banco de dados, através da porta de solicitação P4. Essa passagem pela tarefa *Translator* T3 é necessária para a conversão da mensagem para um formato compreensível pelo banco de dados.

A outra cópia é encaminhada para a tarefa *Correlator* T4. O resultado da consulta retorna à solução de integração através da porta P4 e é encaminhado para a tarefa *Translator* T5, responsável por traduzi-lo para o formato adequado para o processo. A mensagem traduzida é encaminhada para a tarefa *Correlator*

T4, responsável por reunir as mensagens que são relativas a um mesmo contribuinte. As mensagens correlacionadas são enviadas para a tarefa *Context-Based Content Enricher* T6, que enriquece a mensagem original com as informações retornadas pelo banco de dados.

A mensagem enriquecida é enviada à aplicação ARcetil através da porta de solicitação P5. Essa aplicação é responsável pelo cálculo do imposto devido. O resultado desse cálculo retorna ao processo através da porta P5 e é encaminhado para a tarefa *Replicator* T7, responsável por duplicar a mensagem. Uma das cópias é encaminhada para a aplicação DEISS através da porta de solicitação P6. Essa aplicação cria a declaração de entrega, que retorna ao processo através da porta P6. Em seguida, a mensagem relativa a essa declaração é encaminhada, através da porta de saída P7, à aplicação ARcetil, onde é processada para que os seus dados sejam arquivados.

A outra cópia criada pela tarefa *Replicator* T7 é encaminhada, através da porta de solicitação P8, à aplicação NFE 2.0, que faz a geração da nota fiscal eletrônica correspondente. O resultado dessa operação retorna ao processo através da porta P8 para ser enviado ao contribuinte. Para que isso ocorra, primeiro a mensagem passa pela tarefa *Filter* T8, que elimina as mensagens que não contém um endereço de *e-mail* válido. As mensagens restantes são encaminhadas à tarefa *Translator* T9, que faz a tradução para o formato aceito pelo servidor de *e-mail*. Em seguida, as mensagens transformadas são enviadas para esse servidor através da porta de saída P9. Finalmente, a nota fiscal eletrônica é enviada ao contribuinte.

IV. CONCLUSÃO

Através do uso da Guaraná DSL, foi possível modelar uma solução de integração no contexto da administração pública. No presente caso de estudo não foi identificada nenhuma diferença em relação à modelagem de soluções de integração para organizações de administração privada.

Em caso de implementação real da solução de integração proposta, poderia-se fazer as cinco aplicações trabalharem de maneira integrada, através da troca de mensagens, sem a necessidade de intervenção manual. A eliminação dessa necessidade diminui, se não suprime, a possibilidade de erros que foi identificada durante a análise do caso de estudo. Assim pode-se perceber mais uma vantagem que a EAI pode agregar aos processos de trabalho das organizações em que ela é implementada, que é a diminuição ou supressão das ocorrências de erro. No futuro podem ser encontrados mais casos de estudo que possuem essa característica e pode ser feita a implementação da solução de integração proposta.

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] Debasish Ghosh. *DSLs in action*. Manning Publications Co., 2010.
- [3] Gregor Hohpe and Bobby Woolf. *Enterprise Integration Patterns - Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley, 2003.

Execução de Soluções de Integração de Aplicações Empresariais na Nuvem: Perspectivas e Desafios

Daniela F. Sellaro

Universidade Regional do Noroeste do Estado do Rio Grande do Sul

Departamento de Ciências Exatas e Engenharias

Ijuí, RS, Brasil

dsellaro@gmail.com

Resumo—As empresas buscam alternativas tecnológicas que proporcionem competitividade para seus processos de negócios. Suas aplicações devem funcionar de forma integrada para oferecer acesso às informações e funcionalidades de forma rápida e segura. Uma das alternativas adotadas é o uso dos recursos computacionais oferecidos pela computação em nuvem. Tais recursos são fornecidos por demanda, tendo seu pagamento vinculado ao consumo. A revisão da literatura técnica e científica mostra que as ferramentas utilizadas para integrar aplicações apresentam vulnerabilidades que podem impactar a execução das mesmas na infraestrutura computacional da nuvem. Um dos desafios da execução de soluções de integração na nuvem é evitar o consumo inapropriado dos recursos computacionais para que não resulte num aumento nos custos empresariais. O presente artigo tem como objetivo introduzir o contexto da pesquisa, apresentar o problema a ser tratado e discutir ideias para sua solução, com vistas ao desenvolvimento de um modelo teórico para um motor que possa executar soluções de integração de forma eficiente na nuvem.

Palavras-chave: Integração de Aplicações Empresariais; Computação da Nuvem; Motor de execução; iPaaS; Simulação.

I. CONTEXTUALIZAÇÃO

A tecnologia da informação (TI) constitui um dos pilares de sustentação de qualquer empresa. Seu ecossistema de software [21] é formado por aplicações que podem diferir quanto ao fabricante, a versão, a linguagem de programação, a plataforma, etc. Porém, as empresas necessitam integrar esses sistemas, a fim de obter mais agilidade no acesso e processamento das informações.

A Integração de Aplicações Empresariais (EAI) é o campo de estudo que proporciona metodologias, técnicas e ferramentas para a modelagem e implementação de soluções de integração. EAI se propõe a encontrar soluções para que as diversas aplicações de uma empresa possam trabalhar de forma integrada, mantendo o sincronismo dos dados e permitindo que novas funcionalidades sejam incorporadas, com o mínimo de impacto sobre as antigas [13].

Uma solução de integração é um programa computacional, que faz o papel de ponte entre aplicações da empresa. Ela permite conectar diferentes aplicações, compreendendo suas funcionalidades e dados, por meio da utilização de artefatos executáveis, que funcionam como adaptadores [16].

As soluções de integração podem ser desenvolvidas a partir de plataformas de integração. As plataformas geralmente fornecem uma linguagem de domínio específico, um kit de

ferramentas de desenvolvimento, um motor de execução e uma ferramenta de monitoramento. Desses, é o motor que proporciona todo o suporte necessário à execução das soluções de integração [13].

Entre as mais recentes plataformas de integração empresarial para projetar e implementar soluções de integração destacam-se Guaraná [11], Camel [17], Spring Integration [10] e Mule [8]. Essas plataformas permitem construir soluções de integração baseadas no padrão arquitetural *Pipes&Filters* [14].

Uma das inovações tecnológicas que as empresas estão aderindo é a computação em nuvem [6]. Sua finalidade é fornecer serviços de TI por demanda, sendo o pagamento vinculado ao consumo [2]. A computação em nuvem oferece flexibilidade e escalabilidade na infraestrutura [4]. Tipicamente, a computação em nuvem é dividida em três modelos de serviços: *Software as a Service (SaaS)*, *Platform as a Service (PaaS)* e *Infrastructure as a Service (IaaS)*.

No modelo SaaS, a aplicação é hospedada em um servidor



Figura 1. Modelos da computação em nuvem.

remoto e acessada pela Internet, sem a necessidade de instalação no computador do cliente. PaaS caracteriza-se pela entrega de uma plataforma para desenvolvimento e teste com a finalidade de facilitar a implantação de aplicações sem os custos e complexidade de gerenciamento do hardware. O objetivo principal do IaaS é tornar mais fácil e acessível o fornecimento de recursos, tais como servidores, rede, armazenamento e outros recursos de computação, fundamentais para construir um ambiente sob demanda. Com a tendência de migração das plataformas de integração para a nuvem, surge uma subclassificação para PaaS: *Integration Platform as a Service (iPaaS)*. iPaaS proporciona uma plataforma para desenvolvimento e teste com a finalidade de facilitar a implementação de soluções de integração de aplicações, podendo essas aplicações serem locais (*on-premisse*), ou da nuvem [22].

Muitas empresas já migraram seus sistemas periféricos (e.g. armazenamento de dados, serviço de e-mail) para nuvem, porém resistem em migrar as aplicações e serviços que compõem o núcleo estratégico do seu negócio. Apesar dos benefícios significativos de computação em nuvem, temem os riscos relacionados com a sua segurança, desempenho e disponibilidade [1].

II. PROBLEMA DE PESQUISA

A diversidade de tecnologias envolvidas na integração de aplicações aumenta o desafio de fornecer a interoperabilidade entre as aplicações. As aplicações possuem requisitos de qualidade específicos, quando uma solução de integração é executada na nuvem [23].

Potočnik and Juric [22] apontam que as plataformas de integração em nuvem (iPaaS) existentes não oferecem mesmo o nível de recursos que a integração *on-premise*. Na transferência de dados, por exemplo, a latência (tempo de ida e volta de um pacote de dados) pode se tornar um problema no ambiente da nuvem. Alguns dos principais requisitos de uma iPaaS enumerados pelos autores dizem respeito a mecanismos de sincronização dos dados, orquestração de serviços e monitoramento do desempenho:

- Integridade dos dados: dados de diferentes fontes devem ser sempre completos e consistentes. Significando que iPaaS deve oferecer mecanismos de sincronização para fornecer a transformação e migração automática de dados.
- Orquestração: iPaaS deve permitir a orquestração de serviços em processos de negócios compostos por diversos serviços e aplicações na nuvem.
- Monitoramento: iPaaS deve fornecer recursos de monitoramento de desempenho. Ele deve ser capaz de detectar violações de acordos de qualidade de serviços *QoS*.

Pesquisa conduzida pela empresa *Dynamic Markets* [20] revelou que um dos maiores obstáculos para a adoção da computação em nuvem está em conectar, sincronizar e relacionar dados, aplicativos e processos entre aplicações *on-premise* e as da nuvem.

Relatório do Gartner [9] aponta que tanto mundialmente como no Brasil, a capacidade da PaaS foi indicada como requisito principal. Temem que na nuvem não haja a mesma capacidade de armazenamento e processamento que existe localmente para as aplicações que fazem parte do núcleo estratégico do negócio. Há suspeitas de que a capacidade da PaaS está diretamente relacionada com a capacidade do motor de execução que compõe essas plataformas [13].

Os motores das atuais plataformas para integração adotam dois modelos: *process-based execution model* e o *task-based execution model*. No *process-based execution model*, o motor controla a instância do processo inteira, ou seja, não há meios de interagir com as tarefas internas do processo. A implicação é que esse modelo requer um mecanismo para reunir e correlacionar as mensagens e decidir quando uma nova instância de processo pode ser iniciada. Em seguida, cada instância de processo necessitará de uma *thread* exclusiva

para ser alocada por processo. Uma desvantagem do *process-based execution model* é que a execução poderá ficar suspensa, no caso em que uma instância de processo esteja aguardando por um longo tempo pela resposta de uma requisição. Embora existam mecanismos para continuar a execução, eles requerem consumo adicional de recursos computacionais. Já o *task-based execution model*, trata com as tarefas dentro do processo instanciado. Nesse ponto, o motor pode alocar *threads* mais eficientemente. Em outras palavras, nenhuma *thread* ficará ociosa, desde que exista uma tarefa apta a ser executada, independente do processo ao qual pertença. A correlação das mensagens é tratada internamente. As tarefas prontas para serem executadas vão para uma fila de execução, aguardar que uma *thread* livre que a execute. Um ponto de fragilidade desse modelo é que, se a taxa de entrada das mensagens for alta numa determinada porta de entrada, a fila poderá conter uma sequência grande de uma mesma tarefa [12].

Uma das finalidades da computação em nuvem é oferecer recursos como: armazenamento, processamento e largura de banda de rede. Tais recursos são fornecidos e cobrados por demanda. A computação em nuvem é comercializada por provedores especializados, sendo o custo contabilizado por unidade utilizada. Os provedores se propõem a oferecer larga capacidade de armazenamento e alta disponibilidade [2].

III. JUSTIFICATIVA

A computação em nuvem oferece uma infraestrutura ágil, escalável e dinâmica. Essa capacidade de adaptar-se a demanda de recursos computacionais do cliente é chamada de elasticidade computacional [7]. Porém, embora essa elasticidade computacional seja uma vantagem do ponto de vista técnico, há indícios de que ela pode elevar o custo cobrado pelo provedor, se o aumento do consumo for proveniente de uma ineficiência na execução da aplicação [13].

O controle do custo cobrado pelo consumo de recursos é um dos desafios da computação em nuvem. Esta preocupação é verificada pelo crescente número de trabalhos científicos, que buscam mensurar esse custo e otimizar o consumo. Khajeh-Hosseini et al. [18] propuseram uma ferramenta de modelagem de custos que permite os arquitetos de software modelarem o custo de aplicações, dados e requisitos de infraestrutura, além de seus padrões de uso de recursos computacionais. Chaisiri et al. [5] desenvolveram um algoritmo para estimar o custo do provisionamento de recursos de nuvem (*OCR- optimal cloud resource provisioning*) pela formulação de um modelo de programação estocástica.

Khatua et al. [19] afirmam que o gerenciamento da utilização dos recursos da infraestrutura computacional é um dos pontos mais importantes a ser observado para o sucesso na adoção da computação em nuvem. O autor afirma ainda, que uma má gestão desses recursos pode levar a custos superiores ao valor da implementação de seu projeto físico.

Não foram encontrados trabalhos científicos que abordem o melhoramento da eficiência do motor, e que conseqüentemente reduzam os custos associados à execução de soluções de integração, que utilizam a infraestrutura da nuvem.

IV. PERSPECTIVAS PARA SOLUÇÃO

Frantz et al. [13] apontam que deve-se focar em pesquisas para o desenvolvimento de motores mais eficientes para execução de soluções de integração na plataformas de integração. Pois, quanto melhor for o motor de execução, menor será o consumo dos recursos computacionais e por consequência, menor será o custo para executar soluções de integração na nuvem. Com base nesse argumento, a proposta de pesquisa aqui apresentada, pretende, por meio da modelagem matemática, estudar os dois modelos de execução: *process-based execution model* e o *task-based execution model*, verificando os pontos fortes e os fracos desses modelos, na execução de um processo de negócio utilizando-se da infraestrutura da nuvem. Sendo a simulação o campo de estudo que permite prever informações a respeito do comportamento e desempenho de sistemas por meio de experimentos e utilizando os modelos teóricos desses sistemas, propõe-se que o comportamento de um motor de execução de uma plataforma de integração possa ser estudado por meio de técnicas de simulação. Para tanto, pretende-se encontrar na literatura, o modelo matemático que melhor representem esse tipo sistema. Trabalhos recentes de modelagem matemática com teorias como: Teoria das Filas [24], Processos de Markov [15] e Redes de Petri [3] para caracterizar sistemas estocásticos, dinâmicos e discretos. Tais estudos servirão como referências para a investigação do modelo pretendido. Na literatura são utilizadas diversas ferramentas de software para simulação de modelos computacionais e matemáticos. São exemplos de tais ferramentas: *Ansys*, *MatLab/Simulink*, *Arena*, *AMESim* e *AnyLogic*. Por meio da simulação, pretende-se construir modelos que permitam simular os motores, a fim de identificar pontos de vulnerabilidades que afetem os requisitos de qualidade (latência, quantidade de mensagens processadas em determinado intervalo de tempo, dentre outros). O motor proposto deve ser capaz de anular ou minimizar esses pontos de vulnerabilidades. Acredita-se que os resultados obtidos conduzirão ao modelo de um motor mais eficiente para executar soluções de integração baseadas no padrão *Pipes&Filters* [14] na nuvem.

V. AGRADECIMENTOS

Agradeço aos professores e aos colegas do GCA pelo apoio e a CAPES pela bolsa de estudos.

REFERÊNCIAS

- [1] Thomas Boillat and Christine Legner. Why do companies migrate towards cloud enterprise systems? A post-implementation perspective. *Proceedings - 16th IEEE Conference on Business Informatics, CBI 2014*, 1:102–109, 2014. ISSN 2378-1963. doi: 10.1109/CBI.2014.46.
- [2] Matthias Brantner, Daniela Florescu, David A. Graf, Donald Kossmann, and Tim Kraska. Building a database on s3. In *SIGMOD Conference*, pages 251–264, 2008.
- [3] Roberto S. Cargnin, Fabricia Roos-Frantz, Rafael Z. Frantz, and Sandro Sawicki. Simulação de uma solução de integração com redes de petri estocásticas para o problema da central telefônica na unijui. *Salão do Conhecimento*, 1(1), 2015.
- [4] Eddy Caron, Luis Rodero-Merino, Frédéric Desprez, and Adrian Muresan. Auto-scaling, load balancing and monitoring in commercial and open-source clouds. 2012.

- [5] Sivadon Chaisiri, Bu-Sung Lee, and Dusit Niyato. Optimization of resource provisioning cost in cloud computing. *Services Computing, IEEE Transactions on*, 5(2):164–177, 2012.
- [6] Victor Chang, Robert John Walters, and Gary Brian Wills. Organisational sustainability modelling—an emerging service and analytics model for evaluating cloud computing adoption with two case studies. *International Journal of Information Management*, 36(1):167–179, 2016.
- [7] Ariel da Silva Dias. *Computação em nuvem elástica auxiliada por agentes computacionais baseada em histórico para web services*. PhD thesis, Universidade de São Paulo, 2015.
- [8] David Dossot, John D’Emic, and Victor Romero. *Mule in action*. Manning, 2014.
- [9] Key Findings. Survey Analysis : Brazil Ready for SaaS? (October), 2012.
- [10] Mark Fisher, Jonas Partner, Marius Bogoevici, and Iwein Fuld. *Spring integration in action*. Manning Publications Co., 2012.
- [11] 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.
- [12] Rafael Z. Frantz, Rafael Corchuelo, and Carlos Molina-Jiménez. A proposal to detect errors in enterprise application integration solutions. *Journal of Systems and Software*, 85(3):480–497, 2012.
- [13] 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. *XIX Jornada de Pesquisa*, pages 1–11, 2014.
- [14] Gregor Hohpe and Bobby Woolf. *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional, 2004.
- [15] Márcia Maria Horn, Guilherme Henrique Schiefelbein Arruda, Sandro Sawicki, Rafael Z. Frantz, and Fabricia Roos-Frantz. Modelagem matemática e computacional de uma solução de integração utilizando cadeias de markov. *Salão do Conhecimento*, 1(1), 2015.
- [16] Ying Huang and Jen-Yao Chung. A web services-based framework for business integration solutions. *Electronic Commerce Research and Applications*, 2(1):15–26, 2003.
- [17] Claus Ibsen and Jonathan Anstey. *Camel in action*. Manning Publications Co., 2010.
- [18] Ali Khajeh-Hosseini, Ian Sommerville, Jurgen Bogaerts, and Pradeep Teregowda. Decision support tools for cloud migration in the enterprise. In *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, pages 541–548. IEEE, 2011.
- [19] Sunirmal Khatua, Anirban Ghosh, and Nandini Mukherjee. Optimizing the utilization of virtual resources in cloud environment. In *VECIMS*, pages 82–87, 2010.
- [20] Dynamic Markets. Cloud for business managers: the good, the bad, and the ugly.
- [21] David G. Messerschmitt, Clemens Szyperski, et al. Software ecosystem: understanding an indispensable technology and industry. *MIT Press Books*, 1, 2005.
- [22] Martin Potočník and Matjaz B Juric. Integration of saas using ipaas. *Assisted ServiceS (CLASS)*, page 35, 2012.
- [23] Flávio R.C. Sousa, Leonardo O Moreira, and Javam C Machado. Computação em nuvem: Conceitos, tecnologias, aplicações e desafios. *II Escola Regional de Computação Ceará, Maranhão e Piauí (ERCEMAPI)*, pages 150–175, 2009.
- [24] Arléte Kelm Wiesner, Gabriel Freytag, Rafael Z. Frantz, Sandro Sawicki, and Fabricia Roos-Frantz. Teoria das filas como técnica matemática para representar o comportamento de uma solução de integração empresarial. *Salão do Conhecimento*, 1(1), 2015.

Modelo de Simulação de uma Solução de Integração Teórica utilizando a Ferramenta MATLAB/Simulink

Amanda Preissler

Universidade Regional do Noroeste do Estado do Rio Grande do Sul

Departamento de Ciências Exatas e Engenharias

Santa Rosa, RS, Brasil

amandapreissler@gmail.com

Resumo—Este trabalho realiza um estudo sobre a Teoria das Filas e sua relação com a área de Integração de Aplicações. Para isso, desenvolve um modelo de simulação teórico utilizando a ferramenta *MatLab/Simulink* através do *toolkit* denominado *SimEvent* a partir de um modelo conceitual desenvolvido por meio da tecnologia Guaraná.

Palavras-chave: *Enterprise Application Integration*; Simulação, Tecnologia Guaraná, *MATLAB/Simulink*, Teoria das Filas.

I. INTRODUÇÃO

Atualmente a maioria das empresas possui um ou mais sistemas de informação para apoiar e aperfeiçoar seus processos de negócios. Neste contexto, acabam adquirindo ou desenvolvendo aplicações para apoiar a tomada de decisões. Estas aplicações compõe o ecossistema de *software* da empresa, que geralmente é heterogêneo e não foi projetado para ser reutilizado, gerando assim redundância de dados [5]. A utilização de integração dessas aplicações, acaba com redundância de dados, e com a ausência dos mesmos, pois as informações são solicitadas nas demais aplicações do ecossistema de *software*.

II. REFERENCIAL TEÓRICO

A. Integração de Aplicações Empresariais

Contempla as ferramentas, os métodos e o planejamento que possibilitam à organização obter vantagem com a integração de todas as aplicações em um sistema empresarial unificado, capaz de compartilhar as informações e suportar os processos de fluxos de negócios. A abordagem da Integração de Aplicações Empresariais (EAI) busca o compartilhamento de dados e processos de negócios, reduzindo a complexidade da integração. Algumas empresas de tecnologia têm provido os seus produtos com novas características e recursos para atender a essas necessidades [1].

B. Tecnologia Guaraná DSL

Permite modelar soluções de integração utilizando os padrões de integração documentados por *Gregor Hohpe* e *Bobby Woolf* [5]. Os modelos desenvolvidos com esta linguagem são independentes das tecnologias de integração voltadas à implementação e podem ser transformados automaticamente a código de uma ou outra tecnologia. Permite que engenheiros de *software* centrem seus esforços na criação de modelos para a solução do problema, reduzindo os custos envolvidos no processo de aprendizagem e uso das distintas, e muitas vezes complexas, tecnologias voltadas à implementação [3].

C. Simulação

Há inúmeras razões para justificar o uso da simulação, tais como a dificuldade em observar um processo real, a complexidade do sistema que o torna impossível de ser representado por um conjunto de equações matemáticas, de solução analítica viável. Além disso, uma vez que o modelo está definido, o mesmo pode ser usado várias vezes, para identificar possíveis eventualidades no sistema e também dar apoio às decisões importantes sobre futuras mudanças no cenário de atendimento [4]. De acordo com o tipo de problema o modelo de simulação pode ser:

1) *Previsão*: usada para prever ocorrências futuras, baseadas nas suposições de seu comportamento atual e ao longo do tempo.

2) *Investigação*: usada para buscar informações sobre o comportamento do sistema.

3) *Comparação*: usada para avaliar os efeitos de mudanças nas variáveis de controle.

D. Ferramenta Matlab/Simulink

O *Simulink* fornece um pacote de um *software* para modelar, simular, e analisar sistemas dinâmicos. Também suporta sistemas lineares e não lineares, podendo ser modelados em tempo contínuo e/ou tempo discreto [4]. Utiliza-se de uma interface gráfica com o usuário para construção dos modelos a partir de diagramas em blocos, através de operações de clique e arraste do mouse.

E. Teoria das Filas

A Teoria das Filas é um método analítico que estuda a formação de filas por meio de fórmulas matemáticas e utiliza de modelos de filas para representar os diversos tipos de sistemas de filas que surgem na prática. Um sistema de filas pode ser representado por diferentes modelos, tendo elementos característicos comuns a todos que fazem parte do processo básico, ela 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 baseada em padrões de integração [6]

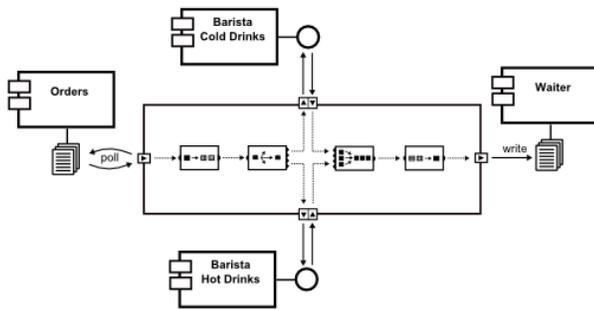


Figura 1. Modelo Conceitual de Integração.

III. CASO DE ESTUDO

O modelo de simulação proposto é baseado no cenário *Coffee Shop*, introduzido por *Gregor Hohpe*, e descreve como os pedidos dos clientes são processados em uma cafeteria [2]. O fluxo de trabalho do sistema pode ser descrito de uma maneira bem simples: temos uma solicitação do cliente ao caixa, onde esse faz o registro no sistema e adiciona-o em uma fila de pedidos. Os pedidos podem ser bebidas quentes ou bebidas frias, e são destinados aos baristas. Os baristas, por sua vez, não tem ideia do número total de bebidas em um pedido ou a qual pedido a bebida está relacionada, eles recebem os pedidos de maneira individual. Ao terminar a bebida, ela é encaminhada junto com as demais à fila do garçom. Cada bebida carrega a informação de qual pedido pertence. Nesse modelo existem duas filas, a de novos pedidos e a de pedidos concluídos, que são entregues pelos garçons.

A Figura 1, mostra a solução de integração gerada na Tecnologia Guaraná.

IV. CONCLUSÃO

A integração de aplicações permite reutilizar aplicações e integrar aplicações heterogêneas, pode-se, através de a simulação analisar o comportamento do sistema. Prevendo quais caminhos serão tomados e qual será o resultado final, se o resultado apresentar problemas de otimização, ou não for o esperado, com alterações nos modelos e novos testes, para encontrar o melhor resultado e inserir o novo modelo na empresa, integra-se todas as aplicações necessárias, sem gastar tempo e recursos. Utilizando o *Simulink* criou-se um modelo de simulação e foram realizados testes com o mesmo, para analisar sua equivalência com o Guaraná. Os próximos passos para o trabalho são a simulação e a identificação de possíveis otimizações da solução.

REFERÊNCIAS

- [1] Leonardo Grandinetti Chaves, Rodrigo Baroni, and Marta Ferreira. Análise da integração de aplicações (eai) no contexto de portais corporativos de médias e grandes empresas brasileiras. *FACE-Universidade FUMEC-Belo Horizonte/MG-SMSI II*, 2005.
- [2] Rafael Z. Frantz. *Cafe-solution*, 2016.
- [3] 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. *XIX Jornada de Pesquisa*, pages 1–11, 2014.
- [4] Dirceu Scaldelai. Desenvolvimento de uma ferramenta para simulação de call centers. 2011.
- [5] Arléte Kelm Wiesner. Modelagem matemática e computacional para análise do comportamento de soluções de integração de aplicações através da criação de modelos de simulação com a teoria das filas. *UNIJUI - Universidade Regional do Noroeste do Estado do Rio Grande do Sul-Ijuí/RS*, 2015.
- [6] Arléte Kelm Wiesner and Rafael Frantz. Modelagem matemática e computacional para identificação de gargalos de desempenho em soluções de integração. *III Seminário de Formação Científica e Tecnológica*, 2015.

Proposta de uma Solução para Avanços Científicos e Tecnológicos no Cultivo de Aveia: Uso de modelagem matemática e inteligência artificial na categorização das cultivares de aveia quanto a resistência genética e uso de fungicida

Eldair Fabricio Dornelles
Universidade Regional do Noroeste do Estado do Rio Grande do Sul
Departamento de Ciências Exatas e Engenharias
Ijuí, RS, Brasil
eldair.dornelles@gmail.com

Resumo—A aveia branca é uma espécie cultivada em estação fria, o seu cultivo possui múltiplos propósitos, desde o uso para a sucessão e rotação de culturas, quebrando ciclo de pragas, até a produção de alimentos para consumo animal e humano, proporcionando produtos saudáveis, ricos em fibras e proteínas. O cultivo deste cereal vem crescendo ano após anos, impulsionado pelo aumento de consumidores cada vez mais exigentes, que procuram alimentos saudáveis em busca de melhor qualidade de vida. Com o aumento das áreas cultivadas com este cereal, passou a existir uma atenção especial em relação às pragas que atacam este cultivo, pois aumentou também os risco de surgimentos de epidemias, que podem causar perdas significativas em uma plantação. Dentre as doenças que atacam esta planta, as que mais se destacam são as doenças foliares, ferrugem-da-folha e mancha-amarela. Essas doenças danificam a área foliar da aveia, diminuindo assim a área responsável pela fotossíntese, em consequência, afetam na quantidade e a qualidade da produção. No Brasil existem diversas cultivares deste produto, sendo 23 recomendadas para cultivo. Neste trabalho é apresentado uma proposta para proporcionar avanços científicos e tecnológicos na ecofisiologia e manejo de aveia, para isso propõe-se a utilização de redes neurais artificiais e algoritmos genéticos, associados a modelagem matemática para realizar a categorização das 23 cultivares de aveia recomendadas para cultivo no Brasil, quanto ao nível de resistência às doenças foliares e uso de fungicida na elaboração de grãos mais saudáveis e com menores impactos ambientais

Palavras-chave: Avena Sativa; Puccinia Coronata; Drechslera Avenae; Otimização de Cultivo; Qualidade do Grão.

I. INTRODUÇÃO

A aveia é uma gramínea anual pertencente família Poaceae, cultivada e distribuída em seis continentes [13]. Dentre os demais cereais, ocupa o sétimo lugar em área de cultivo e produção mundial. É uma espécie que suporta muito bem as baixas temperaturas, porém apresenta fragilidade à ambientes húmidos, são melhores adaptadas nas áreas entre 35° e 50° de latitude no hemisfério norte e 20° a 40° de latitude no hemisfério sul [9]. O maior produtor mundial de aveia é a Rússia, já na América do Sul, os principais produtores são

Chile, Argentina e Brasil [3]. Na Figura 1 as faixas horizontais indicam a melhor região para cultura da aveia.

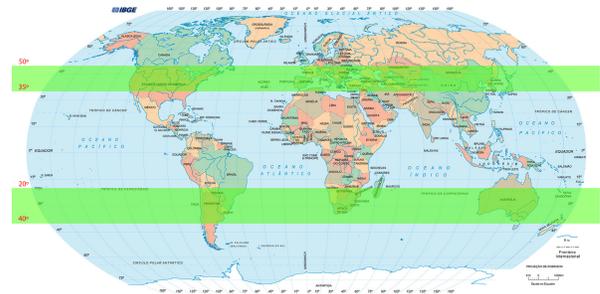


Figura 1. Áreas de melhor adaptação para cultivo da aveia

Atualmente no Brasil não se possui informações precisas sobre a área total de cultivo da aveia. Porém constata-se que a produção de aveia está concentrada na parte sul. Através da análise de dados históricos, pode-se notar que o estado do Rio Grande do Sul vem se destacando na produção deste cereal, tendo sido responsável por 61,6% da produção nacional no período de 2007 a 2011. Outro estado que se destaca é o Paraná, responsável por 36,9% da produção durante o mesmo período [3].

Com base no histórico de produção de aveia no Rio Grande do Sul, no período de 2006 a 2010, percebe-se que a microrregião geográfica de Ijuí era a maior produtora, sendo responsável por 23,3% da produção do estado, seguido pela microrregião de Cruz Alta com 19,6% e Passo Fundo, responsável por 13,5% da produção estadual.

No Rio Grande do Sul a aveia branca é uma excelente alternativa para cultivo de inverno e para o sistema de rotação de culturas, pois pode ser cultivada conforme a necessidade dos produtores, tendo em vista que é um cultivo de múltiplos propósitos. Oferece diversos produtos, tanto para alimentação

animal quanto para a alimentação humana. Na alimentação animal, destaca-se na produção de ração, feno e silagem. Já na alimentação humana fornece alimentos saudáveis, ricos em fibra e proteínas, dentre as quais se destaca a fibra beta-glucana, a qual auxilia na redução do colesterol LDL [5], e reduz os riscos de doenças cardíacas.

Tendo por base as propriedades nutricionais encontradas na aveia e os benefícios proporcionados na alimentação humana, impulsionado pela busca constante por alimentos mais saudáveis que possam proporcionar uma melhor qualidade de vida, constata-se que vem ocorrendo uma expansão gradual da área de cultivo deste cereal. Em paralelo surgiu também algumas preocupações referentes ao seu cultivo, as quais serão abordadas na próxima seção. O restante deste documento encontra-se organizado da seguinte forma: na seção II é abordado o problema. Na seção III é descrito a justificativa, e por fim na seção IV é apresentado a solução proposta.

II. PROBLEMA

Com o aumento das áreas cultivadas e a busca pela maior produtividade de grãos, surgiram também alguns riscos, tais como, epidemias de doenças foliares, principalmente devido a uniformidade genética da aveia em relação aos fungos causadores das doenças, e a contaminação do grão em consequência ao uso excessivo de fungicidas. Dentre as doenças foliares destacam-se a ferrugem (*Puccinia Coronata*), e a mancha amarela (*Drechslera avenae*), essas tem recebido uma atenção especial [12], tendo em vista que esses dois patógenos podem comprometer até 100% da produção em uma plantação. Dentre as doenças que atacam as cultura da aveia, a ferrugem-da-folha, é a mais destrutiva, esta doença é causada pelo fungo *Puccinia Coronata*, e a sua disseminação ocorre em todos os locais do mundo onde há o cultivo desta planta [1]. Este patógeno forma pústulas pequenas e ovais, e expõem uma massa alaranjada. Quando há severidade da doença, as plantas apresentam aspectos de folha queimada. Tal doença se manifesta com maior intensidade em ambientes com muita humidade e temperaturas médias entre 18° e 22° [4]. Na Figura 2 é apresentado a foto de uma folha com ferrugem.



Figura 2. Ferrugem da Folha (*Puccinia coronata*)

A segunda doença que mais afeta a cultura da aveia é a mancha amarela, esta doença é causada pelo fungo *Drechslera Avenae*, e passou a receber uma atenção especial devido ao

cultivo da aveia com foco na produção de grãos. A mesma ataca a folha da aveia, diminuindo a área de fotossíntese da planta, afetando diretamente na quantidade e na qualidade da produção de grãos. [2]. Na Figura 3 é apresentado a foto de uma folha com a mancha amarela.



Figura 3. Mancha Amarela (*Drechslera avenae*).

III. JUSTIFICATIVA

No Brasil existe uma grande variedade de cultivares de aveia, dentre elas, algumas possuem maior resistência aos patógenos que outras, isto ocorre em consequência da variação genética encontrada nessas cultivares. Porém tal resistência não é permanente, devido a constante evolução dos fungos causadores das doenças foliares. Sendo assim a forma mais eficiente de combater tais doenças e proporcionar melhores rendimentos de grãos, passa a ser o uso de fungicidas.

No entanto o uso de fungicidas não pode ser considerado o único meio de conter as doenças e melhorar a produção, tendo em vista que a aveia é um produto consumido *in natura*, e o uso indiscriminado de fungicida na produção da aveia pode acarretar em um produto com alto teor de toxinas, as quais posteriormente podem gerar sérios problemas de saúde aos consumidores [11]. Tais problemas de saúde podem surgir em questão de meses, anos ou após décadas. Dentre esses problemas pode-se citar o surgimento de cânceres, malformação congênita, distúrbios endócrinos e neurológicos [10].

No meio ambiente, o uso exagerado de agrotóxicos possui grande potencial de atingir o solo e as águas, principalmente devido aos ventos que podem espalhar o produto por quilômetros, e as chuvas que lavam as folhas tratadas, carregam o fungicida pelo solo, podendo chegar até os lençóis freáticos ou rios, provocando sérios danos a flora e a fauna [8].

IV. SOLUÇÃO PROPOSTA

A solução proposta consiste em realizar a caracterização das cultivares de aveia branca, classificando-as em grupos de resistência às doenças foliares, considerando as condições meteorológicas e diferentes números de aplicações de fungicidas.

Será acompanhado todo o processo de cultivo das 23 cultivares, desde a separação das sementes para o plantio, aplicação de fungicida, surgimento dos fungos causadores das doenças foliares, até a colheita e verificação da quantidade e qualidade da produção.

Fazendo uso dos dados coletados, pretende-se identificar padrões de informações, visando gerar modelos de previsibilidade do progresso da doença pelas condições climáticas e o nível de dano sobre a produtividade.

Para tal, pretende-se realizar uma modelagem matemática através dos dados reais obtidos, utilizando métodos estatísticos de regressão linear, regressão linear múltipla, regressão polinomial, e regressão de adaptabilidade e estabilidade. Tais métodos de regressão tem sido muito utilizados em diversos trabalhos aplicados a agricultura. Dentre os diversos trabalhos que fazem a utilização destes métodos pode-se citar aqui o trabalho de Mantai et al. (2015), neste trabalho foi utilizado regressões lineares para estimar a produtividade de biomassa dia^{-1} em aveia pelo uso de doses de N-fertilizante [7].

Para a geração do modelo de previsibilidade será também utilizado duas técnicas de inteligência artificial, redes neurais artificiais, e algoritmos genéticos. Ambas as técnicas podem ser utilizadas na busca de soluções em diversos problemas de otimização e categorização de variáveis, conforme pode se perceber através do trabalho desenvolvido por Leal et al. (2015), os quais utilizaram redes neurais artificiais para prever a produtividade de grãos de milho a partir de atributos do solo. O algoritmo de treinamentos utilizado foi o *resilient propagation*, e conforme afirmaram os autores, a adoção de RNAs permite melhor predição da produtividade de grãos. [6].

Tendo conhecimento dos recursos oferecidos pelos métodos matemáticos e técnicas de inteligência artificial apresentados nesta seção, pretende-se utilizar estes para a categorização das 23 cultivares de aveia branca, e gerar informações que auxiliem os produtores nas tomadas de decisões no manejo desta cultura buscando limitar ao máximo o uso de fungicidas. Obtendo assim, produtos de melhor qualidade, mais saudáveis e reduzindo os índices de agressão ao meio ambiente.

REFERÊNCIAS

- [1] Renata Pereira da Cruz, Luiz Carlos Federizzi, and Sandra Cristina Kothe Milach. Genética da resistência à ferrugem-da-folha em aveia. *Pesquisa Agropecuária Brasileira*, 36:1127 – 1132, 09 2001.
- [2] Cinara Sabadin Dagneze, Nathalia Sabino Ribas, and Vanessa Becker. Indicações técnicas para a cultura da aveia. *XXXIV Reunião da comissão brasileira de pesquisa em aveia.*, 2014.
- [3] Claudia De Mori, Renato Serena Fontaneli, and Henrique Pereira Dos Santos. Aspectos econômicos e conjunturais da cultura da aveia. page 26, 2012.
- [4] C. A. Forcelini, E. M. Reis, and E. L. Floss. Indicações técnicas para a cultura da aveia (grãos e forrageira). *Palestras. In: Reunião da Comissão Brasileira de Pesquisa de Aveia.*, 2008.
- [5] Maraisa Crestani Hawerth, Fernando Irajá Félix de Carvalho, Antonio Costa de Oliveira, José Antonio Gonzalez da Silva, Luiz Carlos Gutkoski, João Francisco Sartori, Leomar Guilherme Woyann, Rosa Lía Barbieri, and Fernando José Hawerth. Adaptability and stability of white oat cultivars in relation to chemical composition of the caryopsis. *Pesquisa Agropecuária Brasileira*, 48:42 – 50, 01 2013.
- [6] Aginaldo José Freitas Leal, Eder Pereira Miguel, Fabio Henrique Rojo Baio, Danilo de Carvalho Neves, and Ulcilea Alves Severino Leal. Redes neurais artificiais na predição da produtividade de milho e definição de sítios de manejo

- diferenciado por meio de atributos do solo. *Bragantia*, 74:436 – 444, 12 2015.
- [7] Rubia D. Mantai, José A. G. da Silva, Airam T. Z. R. Sausen, Juliane S. P. Costa, Sandra B. V. Fernandes, and Cassiane Ubessi. A eficiência na produção de biomassa e grãos de aveia pelo uso do nitrogênio. *Revista Brasileira de Engenharia Agrícola e Ambiental*, 19:343 – 349, 04 2015.
- [8] Josino C. Moreira, Silvana C. Jacob, Frederico Peres, Jaime S. Lima, Armando Meyer, Jefferson J. Oliveira-Silva, Paula N. Sarcinelli, Darcilio F. Batista, Mariana Egler, Mauro V. Castro Faria, Alberto José de Araújo, Alexandre H. Kubota, Mônica de O. Soares, Sergio R. Alves, Cláudia M. Moura, and Rosane Curi. Avaliação integrada do impacto do uso de agrotóxicos sobre a saúde humana em uma comunidade agrícola de Nova Friburgo, RJ. *Ciência e Saúde Coletiva*, 7:299 – 311, 00 2002.
- [9] J. Paul Murphy and L. A. Hoffman. The origin, history, and production of oat. in: Marsahal, h. g. sorrells. m. e (ed.) oat science and technology. pages 1–28, 1992.
- [10] Wanderlei Pignati, Noemi Pereira Oliveira, and Ageo Mário Cândido da Silva. Vigilância aos agrotóxicos: quantificação do uso e previsão de impactos na saúde-trabalho-ambiente para os municípios brasileiros. *Ciência e Saúde Coletiva*, 19:4669 – 4678, 12 2014.
- [11] Priscila Pauly Ribas and Aida Teresinha Santos. Matsumura. A química dos agrotóxicos: impacto sobre a saúde e meio ambiente. *Revista Liberato, Novo Hamburgo*, v. 10, n. 14, p. 149-158, jul./dez. 2009, 2009.
- [12] Eduardo Alano Vieira, Fernando Irajá Félix de Carvalho, Márcia Soares Chaves, Antonio Costa de Oliveira, Ivandro Bertan, Andreza Figueirola Martins, Irineu Hartwig, Giovanni Benin, Igor Pirez Valério, and Daniel Andrei Robe Fonseca. Padrão de resistência de genótipos de aveia à ferrugem-da-folha na definição de hibridções. *Pesquisa Agropecuária Brasileira*, 41:607 – 614, 04 2006.
- [13] Fernanda Hart Weber, Luiz Carlos Gutkoski, and Moacir Cardoso Elias. Caracterização química de cariopses de aveia (*Avena sativa* L) da cultivar UPF 18. *Food Science and Technology (Campinas)*, 22:39 – 44, 01 2002.

Análise do Comportamento de uma Solução de Integração na Área de Reservas de Viagens utilizando Redes de Petri

Francisco da Silveira

Universidade Regional do Noroeste do Estado do Rio Grande do Sul

Departamento de Ciências Exatas e Engenharias

Ijuí, RS, Brasil

franciscosilveira25@gmail.com

Resumo—A integração de aplicações é um mercado em crescimento na comunidade empresarial. Os ecossistemas de *softwares* das corporações estão crescendo e a medida que se modernizam vão adquirindo novas aplicações de diferentes funções e de linguagens variadas. A área de Integração de Aplicações Empresariais (EAI), fornecem ferramentas e métodos que fazem a interoperabilidade entre esses sistemas [4]. Existem ferramentas que possibilitam a criação de soluções de integração, tais como Guaraná DSL, Mule Soft, Apache Camel e Spring Integration. Entretanto, em fase de projeto, não é possível analisar o comportamento da solução de integração para encontrar possíveis gargalos de desempenho. Este trabalho propõe desenvolver um modelo de simulação equivalente ao modelo conceitual gerado na etapa de projeto com o objetivo de analisar o comportamento de uma solução de integração ainda na fase de projeto. Utiliza-se, como caso de estudo, uma solução de integração da área de reserva de viagens, projetada por meio da tecnologia Guaraná. Para desenvolver o modelo formal de simulação propõe-se a utilização de redes de Petri como ferramenta matemática de representação dos componentes da solução de integração.

Palavras-chaves: Solução de Integração, Modelagem Matemática e Computacional, rede de Petri.

I. INTRODUÇÃO

Os negócios empresariais estão amplamente ligados às tecnologias de *softwares* que realizam atividades fundamentais nos processos de negócios das corporações. As aplicações que fazem a conexão entre os processos reais e os conceituais estão em ascensão e a utilização de tecnologias de integração se destacam entre os diversos investimentos das corporações. Neste cenário há uma grande variedade de *softwares*, aplicações que atuam para atender a demanda de setores específicos das Empresas e que são construídos com linguagens variadas e para inúmeras funções [3]. Esta situação é consequência do crescimento acentuado das empresas que a medida que seus negócios foram aumentando foram também sendo adicionados estes novos *softwares* de diferentes arquiteturas [4]. No momento em que é criada uma integração entre esses aplicativos diz-se que foi desenvolvida uma solução de integração entre aplicações e que há interoperabilidade entre os diversos sistemas. Nesta área de soluções de integração destacam-se as plataformas de integração que fornecem um ambiente de criação de uma Linguagem de Domínio Específico DSL

(*Domain-Specific Language*). Neste sentido observa-se que há um ponto importante a ser melhorado na construção de uma solução de integração que é o tempo de trabalho despendido no desenvolvimento, na implementação, nos testes necessários na integração das aplicações de um empresa e que de forma alguma é trivial. Esta pesquisa propõe o desenvolvimento de um modelo de simulação em rede de Petri de uma solução de integração na plataforma Guaraná DSL para uma agência de viagens, um problema proposto em [3].

II. REFERENCIAL TEÓRICO

As referencias teóricas desta pesquisa são constituídas pelos conceitos de ecossistema de *software*, integração de aplicações, simulação de modelos de sistemas e redes de Petri.

A. Redes de Petri

Redes de Petri são grafos compostos por dois tipos de nós, lugares e transições. Os lugares são representados por círculos e as transições por retângulos, esses nós são conectados por arcos. Os lugares podem conter fichas ou *tokens*, representados por pontos contidos no círculo e a quantidade de *tokens* em um lugar é chamada de marcação. Um arco de entrada conecta um lugar para uma transição e um arco de saída conecta uma transição para um lugar. Quando uma transição é acionada, os *tokens* nos lugares ligados pelos arcos de entrada são removidos e então são adicionados aos lugares ligados pelos arcos de saída. Uma transição está ativa e pode ser disparada se a quantidade de *tokens* nos lugares satisfazem o peso determinado pelos os arcos de entrada. O peso dos arcos de saída não são necessariamente o mesmo peso que os arcos de entrada. Quando uma transição dispara, a rede de Petri muda seu estado, este conceito é denominado conjunto de marcações. As marcações representam a quantidade de *tokens* após uma sequência finita de disparos das transições [2].

B. Formalismo Algébrico das Redes de Petri

Definição 1: Uma Rede de Petri é uma Sextupla:

$$RP = (P, T, A_r, K, W, M_0) \quad (1)$$

em que:

- $P = \{p_1, p_2, \dots, p_m\}$ é um conjunto finito de lugares;
- $T = \{t_1, t_2, \dots, t_n\}$ é um conjunto finito de transições;
- $A_r \subseteq (P \times T) \cup (T \times P)$ é um conjunto arcos;
- $K : P \rightarrow N \cup \{\infty\}$ é a função capacidade;
- $W : A_r \rightarrow N^+$ é a função Ponderação (Peso);
- $M_0 : P \rightarrow N$ é a função da marcação inicial, que satisfaz:

$$\forall p \in P : M_0(p) \leq K(p). \quad (2)$$

C. Simulação de Sistemas

A definição de sistema é bem ampla, mas em geral um sistema é um conjunto de elementos independentes que trabalham para um objetivo comum. Os sistemas estabelecem a funcionalidade dos objetos de produção e também a organização das próprias entidades de produção. A diversidade dos sistemas existentes faz com haja muitas maneiras distintas de se representar o seu comportamento e conforme ilustra a Figura 1 há algumas maneiras de se estudar um sistema:

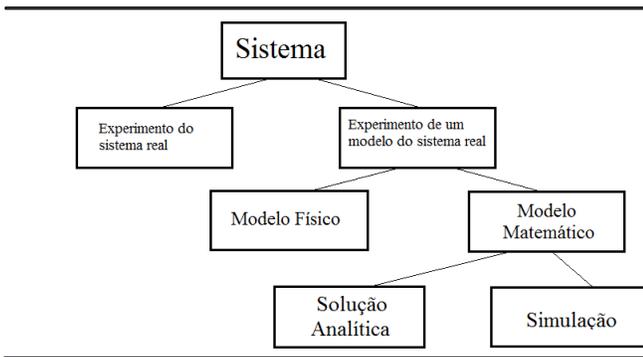


Figura 1. Formas de estudo de um sistema. [5]

O modelo de um sistema sugere uma interpretação dos seus componentes numa linguagem matemática para que se consiga realizar a análise quantitativa do desempenho do sistema real. Um modelo representa as funcionalidades e os aspectos mais importantes do sistema. O modelo pode ser físico ou matemático. O modelo físico representa os componentes de forma real, não-computacional, enquanto que o modelo matemático fornece um formato analítico de análise, com representações de equações ou então a opção de simulação computacional. A simulação computacional é um processo que descreve um modelo computacional para o sistema real [5]. Um modelo pode ser classificado como determinístico ou estocástico.

Se um modelo de simulação não contém variáveis probabilísticas, é chamado determinista. Em um modelo determinista, para um conjunto conhecido de dados de entrada terá como resultado um conjunto único de saída. No modelo estocástico ou modelo probabilístico utiliza-se uma ou mais variáveis aleatórias como entrada que resultam em resultados aleatórios.

Os resultados da simulação estocástica são considerados como estimativas estatísticas das características reais de um sistema. Modelos estáticos representam um sistema em um determinado momento. Eles não mudam ao longo do tempo

e são utilizados em outros campos, tais como em métodos numéricos, problemas de otimização, teste de algoritmos e modelos de Monte Carlo. Os modelos dinâmicos representam sistemas em que há mudança de estado ao longo do tempo. A maioria das simulações usam modelos dinâmicos, porque a maioria dos sistemas mudam com o tempo. Nos modelos de simulação discreta as variáveis de estado mudam em etapas únicas no tempo, enquanto que nos modelos de simulação contínua as variáveis de estado mudam continuamente em função do tempo [5] conforme ilustra a Figura 2:

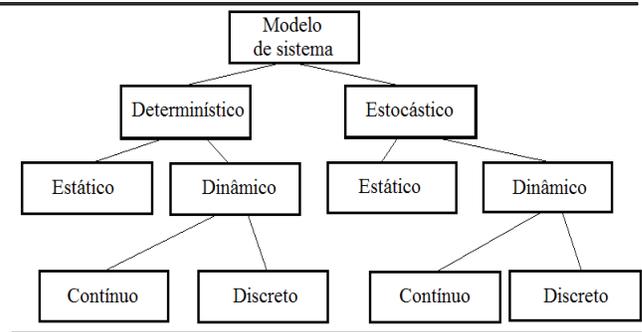


Figura 2. Características do modelo de sistema [5]

A simulação de eventos discretos é usada para modelos de sistemas que mudam de estado, em momentos discretos no tempo a partir da ocorrência de eventos. Um modelo de simulação de eventos discretos visa reproduzir as atividades centrais das entidades envolvidas no sistema para contemplar seus principais aspectos de comportamento. Assim, é necessário definir os estados de um sistema e os eventos que alteram o sistema na mudança de um estado para outro. O estado de um sistema pode ser definido como um conjunto de variáveis usadas para descrever o comportamento do sistema num tempo particular. Este conjunto é chamado de variáveis de estado [5].

As redes de Petri usadas para esta pesquisa modelam sistemas dinâmicos de variáveis discretas no tempo e seus grafos podem ser simulados computacionalmente por *softwares* específicos que suportam essa técnica matemática, um exemplo é a ferramenta *PIPE2* [1].

III. PROBLEMA DE INTEGRAÇÃO

Considere uma agência de viagens que necessite de uma solução de integração que facilite o processo de busca de voos e hotéis de forma automática. Assim, o objetivo é conceber uma solução de integração que solicite um pedido de reservas de viagens para a companhia de voos, para o hotel especificado, para a central de fatura e também para o servidor de email, conforme ilustra a Figura 3.

A. O ecossistema de software

A solução de integração envolve cinco aplicações: sistema de viagens, sistema de fatura, o servidor de e-mail, as companhias de Voos e os Hotéis. O sistema de viagens é um sistema de *software* da própria agência de viagens utilizado

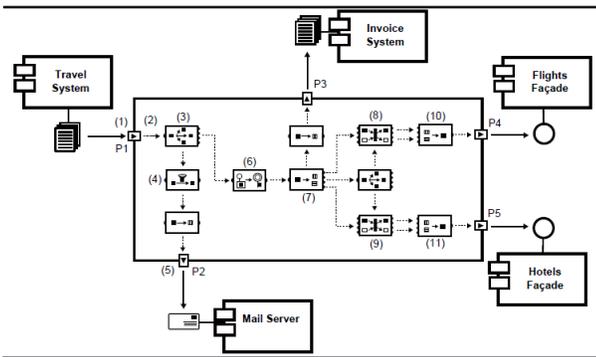


Figura 3. Solução de Integração desenvolvida pela tecnologia Guaraná. [3]

para registrar informações sobre seus clientes e pedidos de reserva. O serviço de fatura é executado no sistema de fatura, que é um sistema de *software* separado o qual permite que os clientes paguem suas viagens utilizando os seus cartões de crédito. O servidor de e-mail executa o serviço de e-mail e é usado para fornecer aos clientes informações sobre suas reservas. As aplicações de voo e hotéis representam interfaces que permitem reservar voos e hotéis. Ambos, além do servidor de correio, representam aplicativos que foram projetados com a preocupação de desenvolver integração entre os sistemas. Ao contrário, o sistema de viagens e o sistema de fatura são sistemas de *software* que foram concebidos sem levar em conta a integração, assim a solução de integração deve interagir com eles por meio de sua camada de dados. A única suposição que fazemos é que cada reserva registrada no sistema de viagens contém todas as informações necessárias sobre o pagamento, o voo, o hotel e um localizador de registro que identifica exclusivamente a reserva. A solução de integração deve periodicamente pesquisar o sistema de viagens para novas reservas de viagens para que o voo e o hotel possam ser reservados o mais breve possível [3].

IV. CONCLUSÃO

A obtenção e a análise dos resultados da simulação do modelo em redes de Petri de uma solução de integração são valiosos na percepção de possíveis gargalos de desempenho da solução ainda na fase de projeto com isso oportuniza melhorias em relação aos gastos e as previsões de orçamentos na implementação de uma solução de integração no ecossistema de *software* da corporação. Este trabalho procurou apresentar a proposta para compreender o comportamento de uma solução de integração voltada para a reserva de viagens por meio da utilização de Redes de Petri e simulação.

REFERÊNCIAS

- [1] Plataforma de simulação pipe2.
- [2] Ilka Cristina Fernandes de Souza Telles. Um modelo em rede de petri para o sistema automático de injeção de uma máquina injetora de plástico. Master's thesis, Universidade Federal do Rio de Janeiro COPE/UFRI, 2007.
- [3] Rafael Z. Frantz. *Enterprise Application Integration - An Easy-to-Maintain Model-Driven Engineering Approach*. PhD thesis, Universidad de Sevilla, 2012.

- [4] Fabricia Roos-Frantz, Vitor Basto-Fernandes, Rafael Z. Frantz, Manuel Binelo, and Sandro Sawicki. Using petri nets to enable the simulation of application integration solutions conceptual models. page 87. 17th International Conference on Enterprise Information Systems, Barcelona. Proceedings of the 17th International Conference on Enterprise Information Systems, 2015.
- [5] Sandro Sawicki, Rafael Z. Frantz, Vitor Manuel Basto Fernandes, Fabricia Roos-Frantz, Iryna Yevseyeva, and Rafael Corchuelo. Characterising enterprise application integration solutions as discrete-event systems. In *Francisco Miranda; Carlos Abreu. (Org.). Handbook of Research on Computational Simulation and Modeling in Engineering*, volume 1, pages 261–288. Pennsylvania: IGI Global, 2015.

Comparação da Evolução do Grau de Manutenibilidade da Ferramenta Spring Integration

Thainan H. Feistel

Universidade Regional do Noroeste do Estado do Rio Grande do Sul

Departamento de Ciências Exatas e Engenharias

Ijuí, RS, Brasil

qthainan@gmail.com

Resumo—Com o grande crescimento da necessidade das organizações de integrar seus sistemas, faz-se necessário a utilização de ferramentas que proporcionem esta integração. Geralmente as ferramentas estão em constante evolução, seja para ampliar a área de abrangência ou apenas para estarem sempre otimizadas. A partir da ideia de evolução que se opta por utilizar métricas, as quais permitem identificar como está indo a evolução de uma ferramenta, e com isso possibilitar com que se mantenha um controle na implementação de novas versões. Neste artigo é apresentado um estudo e análise de distintas versões da ferramenta de integração Spring Integration, baseando-se na utilização de métricas.

Palavras-chave: Integração de Aplicações Empresariais; Spring Integration; Manutenibilidade de Softwares.

I. INTRODUÇÃO

Quando uma organização cresce em tamanho, são criados diferentes departamentos com áreas de foco distintas. O particionamento dos funcionários em diferentes setores dentro de organização normalmente é necessário. Mesmo dividida em diversos subgrupos, uma organização necessita da troca de informações entre seus setores, o compartilhamento de processos e dados se faz necessário para a execução de tarefas que envolvam dois ou mais setores. Mas como cada parte da organização tem uma aplicação diferente executando os seus processos, normalmente não se tem como fazer a troca de processos e dados entre esses setores, pois são diversos os fatores que fazem com que as aplicações sejam incompatíveis.

É a partir de um vasto ecossistema de softwares de uma empresa que a integração de aplicações empresariais se faz necessária, o surgimento do mesmo deve-se a compra de softwares de diferentes empresas, o que torna possível com que uma organização tenha produtos de áreas específicas. Uma das principais vantagens de se ter um grande ecossistema de softwares é o custo reduzido, pois o desenvolvimento de uma aplicação que ira abranger a todos os requisitos de uma organização é significativamente maior [3].

Todo software está sempre em processo de evolução, evolução esta que pode ser vista como a manutenção do mesmo. Se não controlada, esta evolução pode ficar cada vez mais complexa de se fazer. É a partir desta necessidade de se manter um controle na evolução que se opta pela utilização de métricas, as quais tem o objetivo extrair medidas de seu código fonte e identificar em quais pontos o software está ficando mais difícil de se manter. A partir do conhecimento destas medidas,

torna-se possível realizar uma análise das mesmas para decidir o que irá se fazer para diminuir o nível de complexidade da manutenibilidade do software. Este processo permite com que se tenha uma evolução no software sem que nível de complexidade do código fonte chegue a um ponto inviável de se manter.

A integração de aplicações pode ser desenvolvida a partir de diferentes ferramentas, dentre elas destacam-se Apache Camel, Spring Integration, Mule e Guaraná. Sendo que Spring Integration foi a ferramenta escolhida para estudo e análise neste artigo.

II. SPRING INTEGRATION

A ferramenta de integração Spring Integration providencia uma extensão do modelo de programação utilizado por outras aplicações da plataforma Spring para dar suporte aos clássicos padrões de integração propostos por Gregor Hohpe e Bobby Woolf [4]. É introduzido um sistema baseado em mensagens que possibilita a integração com sistemas externos por meio da utilização de adaptadores, tais adaptadores proporcionam um alto nível de abstração, o que deixa menor a complexidade das implementações. Segundo Mark Fisher [1], o principal objetivo da ferramenta de integração Spring Integration é providenciar um modelo simples para a implementação de soluções de integração empresariais complexas.

III. CASO DE ESTUDO

A. O problema

A evolução de um software é sempre necessária a partir do início de seu ciclo de vida. Evolução esta que pode ser vista como a implementação de novas funcionalidades ou apenas como a realização de simples manutenções. Consequentemente, a evolução resulta em distintas versões de um software, sendo que cada versão subsequente pode apresentar uma complexidade maior. É a partir desta evolução que se faz necessária a utilização de métricas para identificar o grau de manutenibilidade de cada uma das versões, sendo que cada métrica existente foi desenvolvida para abordar um aspecto em específico do software [2].

B. Metodologia

Realizou-se um levantamento de todas as versões existentes da ferramenta Spring Integration com a finalidade de identificar em quais versões as mudanças foram mais significativas,

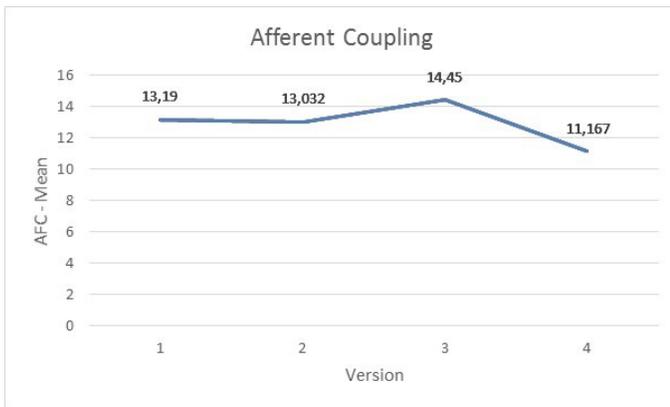


Figura 1. Gráfico da Métrica Afferent Coupling.

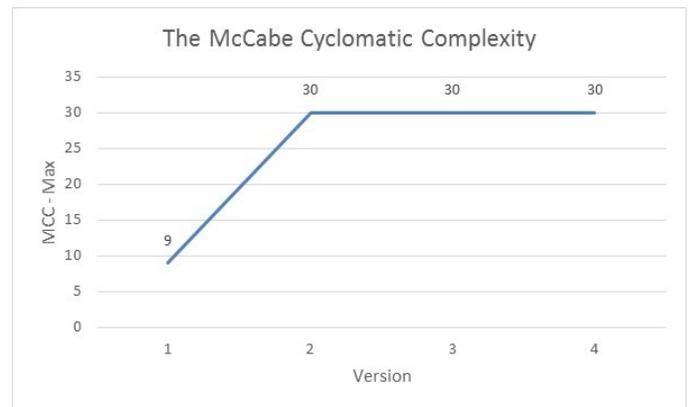


Figura 2. Gráfico da Métrica McCabe Cyclomatic Complexity.

o que resultou na escolha de quatro versões distintas. Após a seleção das versões, se fez necessária a utilização do software Metrics, que permite a extração das medidas necessárias para estudo a partir do código fonte do Spring Integration. O resultado desta extração foi uma tabela contendo os números de cada versão com base nas métricas utilizadas, onde é possível identificar se o software está crescendo controladamente, ou se, a cada versão, ele está ficando mais complexo de implementar.

IV. ANÁLISE DOS RESULTADOS

A partir dos dados extraídos das distintas versões da ferramenta Spring Integration, realizou-se uma análise com o objetivo de identificar possíveis problemas que poderiam vir a existir com o crescimento da mesma. Como já ressaltado anteriormente, cada uma das métricas existentes aborda um aspecto em específico da ferramenta [2], a partir disso, torna-se possível a realização de uma análise detalhada de cada parte da ferramenta com base na métrica utilizada.

A utilização das métricas tem como objetivo indicar se a ferramenta está aumentando seu nível de complexidade em versões subsequentes, ou se está aprimorando sua implementação para que o nível de complexidade baixe. Com a análise realizada é possível notar que em um ponto a complexidade está aumentando, e que em outro a ela está diminuindo. O que pode ser visualizado nos gráficos 1 e 2.

A métrica *Afferent Coupling* pode ser definida como o número de classes fora de um pacote que dependem de uma ou mais classes de dentro deste mesmo pacote [2]. Quanto maior o valor desta medida, mais complexo será de se realizar a manutenção da ferramenta, devido ao fato de existir um maior número de dependências entre as classes [5].

Observando o gráfico 1 pode-se notar que houve uma queda da versão 3 para a versão 4, isso acontece devido a divisão realizada em um pacote que estava com um número grande de classes, fazendo com que a dependência que existia das outras classes quanto a este pacote ficasse menor.

A métrica *McCabe Cyclomatic Complexity* pode ser utilizada como um indicador de quão complexo um método

está [2]. De acordo com McCabe [6], esse valor não deve ser maior do que dez.

Observando o gráfico 2 pode-se notar que ocorreu um crescimento somente da versão um para a dois, e que nas outras versões não ocorreram mudanças. Como o número desta medida não deve ser maior que dez, pode-se dizer que existem métodos na ferramenta que estão com um alto nível de complexidade, e que ainda não foi realizado nada para tentar baixar este nível.

V. CONCLUSÃO

As ferramentas de integração surgiram a partir da necessidade de se integrar aplicações. Estas ferramentas estão em constante evolução, com a finalidade de estarem sempre aptas a atender distintos tipos de problemas. O que acaba resultando na existência de distintas versões de uma ferramenta, sendo que a cada versão subsequente a ferramenta terá uma eficiência maior. Esta evolução pode fazer com que a complexidade de implementação da ferramenta chegue a um nível muito alto, fazendo com que se torne impossível realizar até uma simples manutenção. A utilização de métricas que determinam o nível de complexidade da evolução de um software permitiu averiguar o grau de complexidade atingido pela ferramenta Spring Integration, a qual está em constante evolução.

REFERÊNCIAS

- [1] Mark Fisher, Jonas Partner, Marius Bogoevici, and Iwein Fuld. *Spring integration in action*. Manning Publications Co., 2012.
- [2] Rafael Z. Frantz, Rafael Corchuelo, and Fabricia Roos-Frantz. A methodology to evaluate the maintainability of enterprise application integration frameworks. *International Journal of Web Engineering and Technology*, 10(4):334–354, 2015.
- [3] 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.
- [4] Gregor Hohpe and Bobby Woolf. *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional, 2004.
- [5] Robert Cecil Martin. *Agile software development: principles, patterns, and practices*. Prentice Hall PTR, 2003.
- [6] Thomas J. McCabe. A complexity measure. *Software Engineering, IEEE Transactions on*, (4):308–320, 1976.

Análise comparativa entre Redes de Petri, Cadeias de Markov e Teoria das Filas

Franciéli C. Welter, Adriana R. Kraissig
Universidade Regional do Noroeste do Estado do Rio Grande do Sul
Departamento de Ciências Exatas e Engenharias
Ijuí, RS, Brasil
fran-c-w@hotmail.com, akraissig@yahoo.com.br

Resumo—Por meio da simulação, é possível analisar o comportamento e o desempenho de sistemas reais, encontrando gargalos de desempenho, que possam surgir. O aumento do número de pesquisas envolvendo simulação, gera a necessidade de analisar mais profundamente, as diferentes técnicas matemáticas, existentes nesta área do conhecimento. A Integração de Aplicações Empresariais (EAI) consiste em integrar aplicações monolíticas, que não foram pensadas para trabalhar em conjunto umas com as outras. Com isso, este artigo, busca fazer uma comparação das diferentes características das técnicas matemáticas Redes de Petri, Cadeias de Markov e Teoria das Filas, visando analisar através de um *framework* de comparação, os principais aspectos, desde a perspectiva da área de integração de aplicações.

Palavras-chave: Simulação; Técnicas Matemáticas.

I. INTRODUÇÃO

As técnicas matemáticas, encontram aplicabilidade em diferentes áreas da computação, dentre elas merece destaque o trabalho desenvolvido com as soluções de integrações, no que tange a simulação. A simulação possui várias possibilidades, porém neste estudo, surge com o intuito de encontrar possíveis gargalos de desempenho. A Integração de Aplicações Empresariais (EAI), consiste em uma resposta para décadas de criação de aplicações monolíticas, que não foram pensadas para trabalhar em conjunto umas com as outras. A EAI proporciona metodologias, técnicas e ferramentas, para projetar e implementar soluções de integração [2].

Pode-se representar uma solução de integração através de um modelo conceitual, que demonstra como estão organizados os componentes do modelo, em um alto nível de abstração. Quando uma solução de integração é implementada, gera-se custos, além do aparecimento de falhas. As falhas podem ser evitadas, mas para isso, é preciso identificar possíveis gargalos de desempenho, nos modelos conceituais, ainda na fase de projeto, gerando uma diminuição de custos, riscos e tempo.

Neste contexto, surge a simulação de soluções de integração, no campo da EAI. A simulação é realizada, com o intuito de encontrar possíveis gargalos de desempenho, ainda na fase de projeto, com o auxílio de uma técnica matemática. Portanto, a simulação pode ser útil neste contexto, mas a escolha de uma técnica formal para criar modelos de simulação, não é uma tarefa trivial.

O objetivo deste artigo, é apresentar um *framework* de comparação, que permita, desde uma perspectiva da EAI, comparar técnicas matemáticas. Desta forma, aplica-se este

framework, para comparar as técnicas matemáticas Redes de Petri, Cadeias de Markov e Teoria das Filas, visando compreender suas propriedades, de forma que a análise permita comparar características destas técnicas matemáticas, voltadas para a simulação.

A partir de comparações entre as diferentes técnicas matemáticas, será possível compreender melhor, o comportamento das propriedades de cada uma delas, diante da simulação, no campo da EAI. O restante do artigo, está organizado da seguinte forma: a Seção II apresenta o *framework* de comparação; a Seção III trata da discussão dos dados e por fim as conclusões estão na Seção IV.

II. FRAMEWORK DE COMPARAÇÃO

O *framework* de comparação elencou um conjunto de cinco propriedades, que permitem analisar as técnicas matemáticas, sobre a perspectiva da Integração de Aplicações Empresariais (EAI). Abaixo encontram-se relacionadas as propriedades observadas.

Padrão Arquitetônico: Algumas soluções de integração são desenvolvidas com o padrão arquitetônico proposto pela tecnologia Guaraná. O Guaraná é composto por *slots* (S) e tarefas (T). As técnicas matemáticas estudadas, também são desenvolvidas com diferentes padrões arquitetônicos. Desta maneira, pode-se encontrar e comparar equivalências de padrão arquitetônico, entre o Guaraná e as técnicas matemáticas. Os *slots* (S), equivalem à lugar (L), estados (E) e filas (F). Enquanto que as tarefas (T), correspondem à transição (TR), estados (E) e *server* (SE).

Característica Elementar: Cada técnica matemática, possui suas características elementares, a partir das quais, é possível compreender, de que modo ocorre o funcionamento da técnica. Neste sentido, as técnicas matemáticas podem ser consideradas processo estocástico (PE), que se caracteriza, por uma família de variáveis aleatórias, que representam a evolução de um sistema de valores, de acordo com o tempo. Distribuição de probabilidade (DP), que descreve a chance que uma variável tem, de assumir um valor, ao longo de um espaço de tempo. Podendo ser subdivididas em: hierárquica (H), que visa reduzir o tamanho do modelo; colorida (C), na qual podem ser inseridas marcas, ou cores; temporizada (T), onde pode-se especificar o tempo; generalizada (G), na qual associa-se o tempo somente para alguns eventos; contínua (CO), onde

o disparo ocorre com fluxo contínuo; e sincronizada (SI), na qual há interferência de eventos externos.

Disciplina de Disparo: Para que as técnicas matemáticas tenham seu disparo iniciado, são levados em consideração, alguns aspectos, que visam comparar as disciplinas de disparo, utilizadas por cada técnica. Para que o disparo seja iniciado, observa-se a distribuição de probabilidade (DP), que descreve a chance que uma variável tem, de assumir um valor, ao longo de um espaço de tempo. Retardo de disparo (RD), que é o tempo que cada disparo leva para ocorrer. Expressões de guarda (EG), que tornam a análise mais fácil. Vetor de probabilidade inicial (VP), através do qual o disparo pode ser iniciado. Além das filas FCFS: primeiro a chegar, primeiro a ser atendido; FIFO: primeiro a entrar, primeiro a sair; LLCFS: último a chegar, primeiro a ser atendido; LIFO: último a chegar, primeiro a sair; e fila de prioridade (FP), onde o cliente com maior prioridade, tem preferência no atendimento.

Modelagem de Sistemas: Cada uma das diferentes técnicas matemáticas é capaz de modelar tipos diversos de sistemas. Esta propriedade se caracteriza por comparar qual sistema cada técnica abrange. Dentre estes, existe o sistema distribuído (SD), que é um conjunto de computadores independentes entre si, que se apresenta a seus usuários como um sistema único. Sequências de variáveis aleatórias discretas (SV), onde o conjunto de todos os possíveis valores que ela pode assumir, é finito ou contavelmente infinito. Sistemas que oferecem serviços cuja demanda cresce aleatoriamente (SA), ou seja, onde não há uma lógica de crescimento.

Ambiente de Simulação: As técnicas matemáticas podem ser implementadas em diferentes ambientes de simulação. Esta propriedade analisa em quais ambientes, cada técnica pode ser implementada. Dentre os ambientes está o Java (J), que proporciona um ambiente necessário para desenvolver e executar aplicativos em Java. Aspen (AP), que permite uma maior agilidade na modelagem, análise e compreensão dos resultados. PIPE2 (P2), que possui uma *interface* acessível, sendo possível, desenhar e executar, de maneira, rápida e eficaz. Oris2 (O2), que possui uma *interface* de trabalho complexa, sendo difícil manusear as funções proporcionadas pelo mesmo. Weblab (W), cujo objetivo principal é possibilitar a realização e controle em tempo real de experimentos. SIG (SG), que é um sistema de informações geográficas. Maxima (MX), que compreende dois tipos distintos de recurso, recurso do sistema e recurso aplicado a expressões matemáticas. Monte Carlo (MC), que possibilita levar em conta, o risco em análises quantitativas e tomadas de decisão. Arena (AR), que simula eventos discretos. ProModel (PM), que permite reproduzir uma empresa, em modelo computacional. Automod (AU), que possibilita simular com precisão as instalações industriais, com qualquer nível de complexidade. Simgrip (SP), que oferece uma série de recursos de interação, controle e visualização com o intuito de facilitar e potencializar a simulação.

III. DISCUSSÃO

Na pesquisa, estudou-se três diferentes técnicas matemáticas: Redes de Petri, Cadeias de Markov e Teoria das Filas. Estas solucionam problemas envolvendo solução de integração de aplicação empresarial (EAI). Nesta pesquisa, as diferentes técnicas foram comparadas, levando em consideração as propriedades: Padrão Arquitetônico, Característica Elementar, Disciplina de Disparo, Modelagem de Sistemas e Ambiente de Simulação. Este *framework* de comparação, deu origem à Figura 1:

Propriedades	Redes de Petri	Cadeias de Markov	Teoria das Filas
Padrão Arquitetônico Slot - Tarefa	L - TR	E - E	F - SE
Característica Elementar	PE H C T G CO SI	PE	DP
Disciplina de Disparo	DP RD EG	VP	FCFS FIFO LLCFS LIFO FP
Modelagem de Sistemas	SD	SV	SA
Ambiente de Simulação	J AP P2 O2 W	SG MX MC	AR PM AU SP

Figura 1. Comparação entre Propriedades

A técnica matemática Rede de Petri, de acordo com [1] é uma técnica de modelagem que permite a representação de sistemas, utilizando como alicerce uma forte base matemática. Possui a particularidade de permitir modelar sistemas paralelos, concorrentes, assíncronos e não-determinísticos.

As Redes de Petri são consideradas processos estocásticos, pois, se modificam através de eventos que ocorrem em decorrência do tempo. As Redes de Petri são classificadas em alto nível e baixo nível [3]. Existem extensões que podem abranger as Redes de Petri, as quais buscam incluir hierarquias e aspectos temporais. Dentre elas merecem destaque: as extensões temporizadas que congregam aspectos temporais determinísticos aos modelos; as extensões estocásticas que aliam-lhes aspectos temporais não determinísticos e as extensões hierárquicas que visam representar modelos de sistemas complexos de forma mais compreensível.

A técnica matemática Cadeias de Markov, é um tipo especial de processo estocástico, onde as distribuições de probabilidade para os passos futuros, dependem somente do estado presente, desconsiderando como chega-se a tal estado. Se o espaço de estados é discreto (enumerável), então o modelo de Markov é denominado Cadeia de Markov.

As Cadeias de Markov podem ser classificadas em estados, dentre estes, merecem destaque: os estados alcançáveis e co-

municantes, os estados recorrentes e transientes e propriedades de periodicidade. De acordo com [6] os estados são alcançáveis, se um estado j é alcançável, portanto, é alcançável, se o sistema conseguir entrar no estado j eventualmente quando este começa no estado i . Para o mesmo autor, um estado j é comunicante, se o estado j é alcançável, a partir do estado i , e o estado i é alcançável, a partir do estado j . Este autor também trata, dos estados recorrentes e transientes. Para ele um estado é transiente, se entrando neste estado, o processo pode nunca retornar novamente para ele e um estado é recorrente, se entrando no mesmo, o processo definitivamente irá retornar para este estado.

A técnica matemática Teoria das Filas, de acordo com [5], é um corpo de conhecimentos matemáticos, aplicado ao fenômeno das filas. Sendo assim, para [4] os sistemas de filas se descrevem, genericamente, por um processo de chegada de clientes a um sistema de atendimento. Esse sistema recebe um ou mais serviços, executados por certa quantidade de servidores. Nesse sentido, as formações de filas ocorrem, porque a procura pelo serviço, é maior do que a capacidade do sistema de atender a esta demanda.

Para aferir o comportamento dos sistemas de filas, pode-se associar medidas de desempenho, tais como: tempo médio de espera dos clientes na fila; tempo médio de chegada de clientes e probabilidade de encontrar o sistema lotado. Consideram-se elementos básicos de filas os seguintes: o cliente (que procura por atendimento) e o canal de atendimento (que realiza o processo de atendimento do cliente). Quanto ao tipo de fila, classificam-se em cinco estruturas básicas: canal único-fase única; canal único-fases múltiplas; canais múltiplos-fase única; canais múltiplos-fases múltiplas e misto.

Mediante a pesquisa, observou-se que ambas técnicas matemáticas, tem importância nas simulações, porém, possuem peculiaridades. Observando-se a figura 1, pode-se analisar cada uma das propriedades consideradas no *framework* de comparação. A propriedade Padrão Arquitetônico, compara a notação gráfica do Guaraná, com a notação gráfica das técnicas. Observa-se que nas Redes de Petri, slot (S) equivale à lugar (L), e tarefa (T) equivale à transição (TR). Nas Cadeias de Markov, slot (S) e tarefa (T) equivalem à estados (E). Na Teoria das Filas, slot (S) corresponde à fila (F) e tarefa (T) corresponde à server (SE). A propriedade Característica Elementar, compara a composição básica de cada técnica. Percebe-se que Redes de Petri, são um processo estocástico (PE), sendo subdivididas em: hierárquica (H), colorida (C), temporizada (T), generalizada (G), contínua (CO) e sincronizada (SI). As Cadeias de Markov são um processo estocástico (PE), enquanto a Teoria das Filas se caracteriza como distribuição de probabilidade (DP). A propriedade Disciplina de Disparo, visa comparar as disciplinas de disparo, que cada técnica matemática utiliza. Redes de Petri tem seu disparo iniciado, observando-se a distribuição de probabilidade (DP), retardo de disparo (RD) e expressões de guarda (EG). Cadeias de Markov tem seu disparo iniciado através de um vetor de probabilidade inicial (VP). Teoria das Filas dispara através das filas: FCFS, FIFO, LLCFS, LIFO, e também com fila

de prioridade (FP). A propriedade Modelagem de Sistemas, procura comparar quais sistema cada técnica abrange. Redes de Petri, modela sistemas distribuídos (SD). Cadeias de Markov, modela sequências de variáveis aleatórias discretas (SV). Teoria das Filas, modela sistemas que oferecem serviços com demanda aleatória (SA). A propriedade Ambiente de Simulação, analisa em quais ambientes de simulação, cada técnica pode ser implementada. Redes de Petri pode ser implementada em: Java (J), Aspen (AP), PIPE2 (P2), Oris2 (O2), Weblab (W). Cadeias de Markov, tem sua implementação em: SIG (SG), em Maxima (MX), e também em Monte Carlo (MC). Teoria das Filas, pode ser implementada em: Arena (AR), ProModel (PM), Automod (AU) e Simscript (SP).

IV. CONCLUSÃO

Concluiu-se através da pesquisa realizada entre as técnicas matemáticas Redes de Petri, Cadeias de Markov e Teoria das Filas, que Redes de Petri é melhor quando busca-se criar modelos de simulação de soluções de integração. Isto se deve, ao fato de que esta técnica, abrange mais possibilidades de análise, pois sua equivalência, quanto à notação gráfica, envolvendo o Guaraná, ocorre tranquilamente. Além disso, possui um disparo simples, sendo que, a disciplina de disparo, é de fácil entendimento. Outro ponto positivo é que existem várias possibilidades de implementação, em diversos ambientes, como, por exemplo Java. Além disso, pode-se observar que através das técnicas matemáticas, é possível analisar a solução de integração, para desta forma encontrar gargalos de desempenho.

REFERÊNCIAS

- [1] Carlos Renato Lisboa Francês. Introdução às redes de petri. *Laboratório de Computação Aplicada, Universidade Federal do Pará*, 2003.
- [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] Norian Marranghello. Redes de petri: Conceitos e aplicações. *São Paulo: DCCE/IBILCE/UNESP*, 2005.
- [4] Flávio Gomes Moraes, Gecirlei Francisco Silva, and Tacyanne Assis Rezende. Introdução à teoria das filas. *Universidade Federal do Mato Grosso*, 2011.
- [5] Daniel Augusto Moreira. *Pesquisa operacional-curso introdutório*. Cengage Learning Edições Ltda., 2010.
- [6] Fernando Nogueira. Modelagem e simulação-cadeias de markov. *Notas de Aula UFJF Juiz de Fora-2008*, 2008.