

Desafios para a Implantação de Soluções de Integração de Aplicações Empresariais em Provedores de Computação em Nuvem

Rafael Z. Frantz¹
rzfrantz@unijui.edu.br

Sandro Sawicki¹
sawicki@unijui.edu.br

Fabricia Roos-Frantz¹
frfrantz@unijui.edu.br

Rafael Corchuelo²
corchu@us.es

Vitor Basto-Fernandes³
vitor.fernandes@ipleiria.pt

Inma Hernández⁴
ichernandez@uautonoma.cl

Resumo

Nos últimos anos o campo de estudos conhecido como Integração de Aplicações Empresariais tem desempenhado um importante papel ao proporcionar metodologias, técnicas e ferramentas para que as empresas possam desenvolver soluções de integração, visando reutilizar suas aplicações e dar suporte às novas demandas que surgem com a evolução dos seus processos de negócio. A Computação em Nuvem é parte de uma nova realidade, na qual tanto pequenas como grandes empresas têm a sua disposição uma infraestrutura de TI de alta capacidade, a um baixo custo, na qual podem implantar e executar suas soluções de integração. O modelo de cobrança adotado pelos provedores de Computação em Nuvem se baseia na quantidade de recursos computacionais consumidos por uma solução de integração. Tais recursos podem ser conhecidos, basicamente, de duas formas distintas: a partir da execução real de uma solução de integração em um motor de orquestração, ou a partir da simulação do modelo conceitual que descreve a solução sem que a mesma tenha que ser previamente implementada. Ainda, é desejável que os provedores proporcionem modelos conceituais que descrevam detalhadamente a variabilidade de serviços e as restrições entre eles. A revisão da literatura técnica e científica evidencia que não existem metodologias, técnicas e ferramentas para estimar a demanda de recursos computacionais consumidos por soluções de integração, a partir de seus modelos conceituais. Além disso, os provedores de Computação em Nuvem não possuem ou disponibilizam os modelos conceituais dos serviços que possam ser contratados. Tais questões constituem a base para que se possa estabelecer um processo e desenvolver ferramentas de apoio a tomada de decisão para a implantação de soluções de integração de aplicações empresariais em provedores de Computação em Nuvem.

PALAVRAS-CHAVE: Integração de Aplicações, Computação em Nuvem, Simulação, Otimização.

¹ Professor Doutor da Universidade Regional do Noroeste do Estado do Rio Grande do Sul (Brasil)

² Professor Doutor da Universidade de Sevilha (Espanha)

³ Professor Doutor do Instituto Politécnico de Leiria (Portugal)

⁴ Professora Doutora da Universidade Autônoma do Chile (Chile)

Geralmente as empresas precisam utilizar os seus ecossistemas de software [1] para apoiar e aperfeiçoar os seus processos de negócio. Ditos ecossistemas são compostos de muitas aplicações, normalmente concebidas sem levar em conta sua possível integração. 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. Conforme a Figura 1, uma solução de integração está composta por processos que contém lógica de integração e portas de comunicação, que conectam processos ou aplicações do ecossistema à solução de integração.

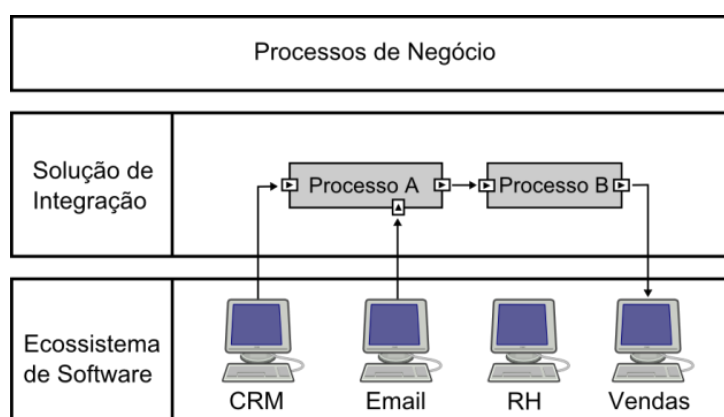


Figura 1: Integração de Aplicações Empresariais.

Nos últimos anos os autores deste ensaio vêm desenvolvendo suas pesquisas no campo de Integração de Aplicações Empresariais, buscando proporcionar metodologias, técnicas e ferramentas para a concepção e a implementação de soluções de integração. Como principais resultados científicos já alcançados, destacam-se a criação de uma linguagem de domínio específico que permite construir modelos de soluções de integração de aplicações a um alto nível de abstração [44,49], uma ferramenta de software para dar suporte à implementação e execução de soluções de integração [45,47,48], um conjunto de algoritmos que compõem um monitor para detectar erros que possam ocorrer durante a execução de soluções de integração [46,50,51,52,53] e a formalização da linguagem de domínio específico visando uma melhor automatização deste monitor [54].

Tais contribuições deram origem e estão incorporadas à tecnologia para integração de aplicações empresarial denominada Guaraná¹.

Guaraná DSL, assim denominada a linguagem de domínio específico, permite modelar soluções de integração utilizando os padrões de integração documentados por Gregor Hohpe e Bobby Woolf [2]. 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. Tal característica 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. Guaraná SDK é uma implementação Java do Guaraná DSL, organizada em duas camadas. A primeira camada, denominada *framework*, implementa todos os conceitos básicos da linguagem de domínio específico. Já a segunda camada, denominada *toolkit*, proporciona uma implementação concreta do *framework*, assim como um motor de execução *multi-thread* assíncrono para executar as soluções de integração. Por natureza, as soluções de integração são distribuídas e, portanto, vulneráveis a erros que impactam diretamente no seu comportamento e resultado que devem produzir. Os erros acontecem em função de falhas, que podem ser permanentes (ex: defeitos no software) ou transientes (ex: indisponibilidade de um recurso necessário para a solução). Erros, quando não tratados resultam em falhas que são percebidas pelos usuários finais [55,56]. Guaraná FT é responsável pelo monitoramento e detecção de erros que possam ocorrer nas portas de comunicação presentes nos processos que compõe as soluções de integração.

A Computação em Nuvem [3] é outro campo de pesquisa que tem chamado à atenção da comunidade científica. Este campo vem transformando os atuais ecossistemas de software e revolucionando a forma com que as empresas proporcionam suporte informático aos seus processos de negócio. A Computação em Nuvem permite às empresas contratarem pacotes de serviços reduzindo enormemente seus custos em infraestrutura de TI, sem ter que sacrificar a qualidade do suporte informático prestado aos seus processos de negócio. Dentre os atrativos da Computação em Nuvem está o modelo de cobrança praticado pelos provedores de serviços e a disponibilidade de uma

¹ <http://www.guaranasolutions.com>

infraestrutura de TI de alta capacidade computacional a um baixo custo. Este novo modelo estabelece uma cobrança baseada na quantidade de recursos computacionais (memória, tempo de processador, transferência de dados, largura de banda de rede, etc.) consumidos pelas aplicações dos clientes.

Enterprise Service Bus (ESB) [4] é uma tecnologia central para muitas soluções de integração de aplicações empresariais. Seus elementos principais são um conjunto de adaptadores, uma linguagem e um motor de orquestração. Os adaptadores permitem aos engenheiros de software abstrair os detalhes relativos às distintas tecnologias para a comunicação com as aplicações do ecossistema de software. A linguagem de orquestração permite criar modelos que descrevam a um alto nível de abstração as soluções de integração. O motor de orquestração, também conhecido como motor de integração, proporciona todo o suporte necessário à execução de soluções de integração.

As novas demandas de migração de aplicações dos atuais ecossistemas de software das empresas para a Nuvem tornam também necessário o suporte à implantação e execução de soluções de integração de aplicações empresariais dentro da Nuvem. Para isto as Nuvens devem oferecer serviços de orquestração (OaaS). A orquestração como um serviço está intimamente relacionada ao modelo de negócio que busca proporcionar plataformas como um serviço (PaaS), no qual provedores de Computação em Nuvem proporcionam a seus clientes um ambiente de computação em que possam ser instaladas suas aplicações, incluindo suas orquestrações. Segundo Wlodarczyk et al. [5], a chave para avançar na área de integração de aplicações empresariais é proporcionar ESBs que executem dentro da Nuvem. Isto motiva a criação de motores de orquestração mais eficientes para as Nuvens, já que quanto mais eficiente for o motor de orquestração, menos recursos computacionais serão consumidos e conseqüentemente as empresas gastarão menos com a adoção da Computação em Nuvem. De forma complementar, quanto mais eficiente for o motor de orquestração, mais clientes poderão ser atendidos por um provedor de Computação em Nuvem.

Atualmente existem várias empresas provedoras de serviços para a Computação em Nuvem. Cada provedor oferece distintos planos que podem ser contratados de acordo com as necessidades de recursos computacionais das aplicações dos clientes. O custo dos recursos computacionais varia não apenas entre os distintos provedores, mas depende também do plano contratado em um mesmo provedor. A tomada de decisão referente ao

provedor/plano que mais se ajuste as necessidades do cliente pode representar uma grande economia de tempo e de recurso financeiro para as empresas com respeito a implantação de soluções de integração de aplicação em provedores de Computação em Nuvem.

A demanda de recursos computacionais que uma solução de integração necessita pode ser obtida, basicamente, de duas formas distintas: a partir da execução real de uma solução de integração em um motor de orquestração, ou a partir da simulação do modelo conceitual que descreve a solução. A primeira alternativa implica na implementação do modelo e em uma funcionalidade para a mensuração dos recursos computacionais que deve ser oferecida pelos ESBs. A segunda alternativa toma como base apenas o modelo da solução e simula sua execução com base nas características de cada elemento presente no modelo, sem que a solução de integração tenha que ser previamente implementada. Conhecer a demanda computacional é fundamental para o processo de tomada de decisão.

A seleção de um provedor de serviços de Computação em Nuvem passa pelos tipos de serviços que o mesmo oferece, a combinação destes serviços e seus custos. Para tanto é desejável que os provedores proporcionem modelos conceituais que descrevam detalhadamente a variabilidade de serviços e as restrições entre eles de forma que estes modelos possam servir de subsídio para o processo de tomada de decisão.

A revisão da literatura técnica e científica evidencia que não existem metodologias, técnicas e ferramentas para estimar a demanda de recursos computacionais consumidos por soluções de integração, a partir de seus modelos conceituais. Além disso, os provedores de Computação em Nuvem não possuem ou disponibilizam modelos conceituais dos serviços que possam ser contratados. Tais questões constituem a base para que se possa estabelecer um processo e desenvolver ferramentas de apoio a tomada de decisão, conforme a Figura 2.

Na literatura são utilizadas diversas ferramentas de software para simulação de modelos computacionais e matemáticos. Tais ferramentas podem servir de apoio no processo de estimação da demanda de recursos computacionais consumidos por soluções de integração, tais como: Ansys [6], MatLab/Simulink [7], Arena [8], AMESim [9] e AnyLogic [10]. Ansys é uma ferramenta amplamente utilizada na área das engenharias,

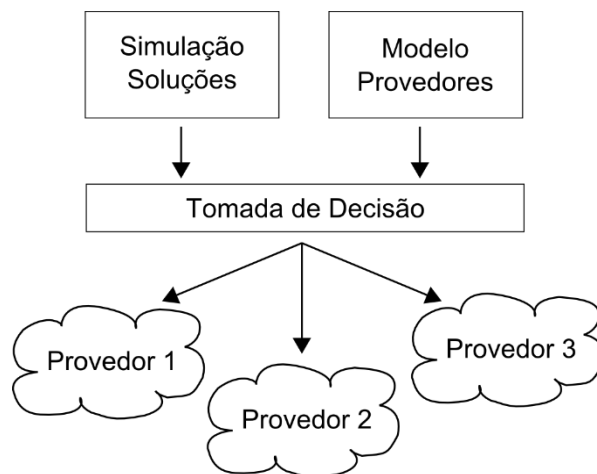


Figura 2: Processo para implantação de soluções de integração na Nuvem

possibilita a construção de plataformas de testes e simulação [11, 12]. MatLab/Simulink é uma ferramenta amplamente usada para simulação, modelagem e análise de sistemas dinâmicos, oferece ampla integração com o ambiente MatLab [13,14,15]. Arena é uma ferramenta aplicada na simulação discreta de diversos segmentos, entre eles a alocação de recursos e planejamento [16]. AMESim é um software de simulação direcionado para a modelagem e análise de sistemas em vários domínios de aplicação [17,18]. AnyLogic é utilizada para a modelagem e simulação de sistemas dinâmicos e eventos discretos, proporciona uma linguagem gráfica de modelagem e permite que o usuário estenda seus modelos de simulação para código Java [19,20,21].

De acordo com Quinton et al. [22], durante a implantação de uma aplicação na nuvem, as empresas se deparam com um grande número de recursos em diferentes níveis de funcionalidade disponibilizados por diversos provedores de serviços na nuvem. Esta variabilidade de soluções de nuvem aumenta a complexidade da decisão, a qual geralmente é feita de maneira *ah hoc*. Esses autores argumentam que a seleção de provedores e serviços pode ser sistematizada e parcialmente automatizada usando técnicas de Linha de Produtos de Software (do inglês, *Software Product Lines*) [23,24].

Para representar a variabilidade dos provedores e serviços, presente nesse domínio, Quinton et al. [22] propõem o uso da técnica oriunda de Linha de Produtos, denominada Modelos de Características (do inglês *Feature Models*) [25]. Em [26], os autores propõem a combinação de Modelos de Características e Ontologias para representar a variabilidade do ambiente da nuvem e automatizar o processo de seleção

das características desejadas nos modelos de características. Além destes, outros autores também sugerem o uso de técnicas de Linhas de Produtos para representar modelos de provedores [27,28,29].

A aplicação de modelos matemáticos e técnicas de otimização à área de Engenharia de Software é um campo de grande interesse acadêmico, conhecido pelo nome de Search-based Software Engineering (SBSE) [42], e vem sendo explorado por distintos grupos de pesquisa tanto a nível nacional como internacional. Em um recente artigo, Freitas et al. [43] fazem uma revisão da literatura e apresentam alguns resultados obtidos com a aplicação de modelos matemáticos e técnicas de otimização para a resolução de problemas da Engenharia de Software, no que se refere ao apoio a tomada de decisão nos campos de Engenharia de Requisitos, Teste de Software e Estimativa de Software. Harman et al. [32] publicaram recentemente um artigo no qual fazem uma revisão e classificação da literatura em SBSE referente aos últimos dez anos, constatando um crescimento expressivo da quantidade de trabalhos que aplicam técnicas de otimização para resolver os mais variados problemas em distintos campos da Engenharia de Software.

De acordo com Silva Neto e Oliveira [30] a programação matemática pode ser um instrumento de auxílio à tomada de decisão. A programação linear é uma área do conhecimento dentro da programação matemática que envolve uma função objetivo a ser otimizada e um conjunto de restrições sobre esta função. Segundo Rafikov [31], a programação linear estuda métodos de maximização e minimização de uma função objetivo linear de várias variáveis sob a condição de que as variáveis satisfaçam as restrições expressas na forma de desigualdades lineares. Sob essa ótica, a programação matemática pode se constituir em ferramenta de apoio a tomada de decisão e implantação de soluções de integração na Nuvem.

Métodos Heurísticos também podem auxiliar na tomada de decisões. Segundo Harman et al. [32], inúmeras técnicas de otimização podem convergir para boas soluções sem ficarem presos em mínimos locais, tais como: Simulated Annealing [33], Algoritmos Genéticos [34], Programação Genética [35], Hill Climbing [36], Busca Tabu [37] e GRASP [38]. A Metaheurística Simulated Annealing é um fenômeno natural baseado na cristalização de metais utilizado na resolução de problemas de otimização. Seu processo de avaliação é aleatório e tem relação direta com a temperatura. Sua convergência tende

a ficar gulosa (Greedy) [39] à medida que a temperatura é reduzida. Esta característica evita que suas soluções fiquem presas em mínimos locais. Algoritmos Genéticos são aplicados em problemas complexos de otimização, problemas com diversos parâmetros ou características que precisam ser combinadas em busca da melhor solução. GRASP (Greedy Randomized Adaptive Search Procedure) é uma metaheurística aplicada comumente a problemas de otimização. Trata de um processo iterativo cujo objetivo é obter soluções aproximadas, eventualmente ótimas, para problemas de otimização combinatória. Busca Tabu é uma estratégia para resolver problemas de otimização combinatória, cuja habilidade adaptativa visa utilizar outros métodos para evitar soluções com mínimos locais.

A programação com restrições [41] é outro instrumento que pode contribuir para a tomada de decisão, dado que ela oferece métodos específicos de domínios para resolver problemas lineares. A programação com restrições consiste em modelar e solucionar um problema de satisfação de restrições (CSP). Um CSP consiste em um conjunto de restrições sobre um conjunto finito de variáveis com seus respectivos domínios. A solução de um CSP se dá pela busca dos valores que satisfazem todas as suas restrições. Dentro desta área ainda podemos fazer uso das ferramentas para resolver um Problema de Otimização com Restrições (COP), o qual adiciona ao CSP uma função de otimização.

É interessante ressaltar que os problemas de otimização consistem em encontrar o valor mínimo e máximo de uma função objetivo, tendo em vista um conjunto de restrições. Problemas inversos, formulados como problemas de otimização, também podem ser úteis na tomada de decisões [40].

Tomando como base a discussão apresentada neste ensaio, conclui-se que é possível utilizar modelos e ferramentas matemáticas para estimar a demanda de recursos computacionais necessários à execução de soluções de integração, a partir de seus modelos conceituais, sem a necessidade de implementá-las. Da mesma forma, conclui-se que é possível inferir os modelos que descrevem os serviços dos provedores de Computação em Nuvem, possibilitando o desenvolvimento de processos e ferramentas de apoio à tomada de decisão, especialmente quanto à contratação de infraestruturas de TI de alta capacidade computacional para instalação e execução de soluções de integração na Nuvem.

Bibliografia

- [1] Messerschmitt, D.; Szyperski, C. A. Software Ecosystem: Understanding an Indispensable Technology and Industry. MIT Press, 2003.
- [2] Hohpe, G.; Woolf, B. Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley, 2003.
- [3] Mell, P.; Grance, T. Draft NIST working definition of cloud computing.
<http://csrc.nist.gov/groups/SNS/cloud-computing>, 2011.
- [4] Chappell, D. Enterprise Service Bus: Theory in Practice. O'Reilly Media, 2004.
- [5] Wlodarczyk, T. W.; Rong, C.; Thorsen, K. A. H. Industrial cloud: Toward inter-enterprise integration. In CloudCom, pages 460–471, 2009.
- [6] Wang, J.; Based on the finite element software ANSYS/ls-dyna metal plate covering parts forming process simulation and optimization research. Advanced Materials Research. Vols 912-914(2014), pp. 589-592, Trans Tech Publications. 2014.
- [7] Soni, J.S.; Pareek, S.; Agrawal, H.P., Modeling and Performance Analysis of Single Stage Power Factor Corrected Resonant Converter Using MATLAB/Simulink. 5th International Conference on Computational Intelligence and Communication Networks (CICN), pp.639,643. 2013.
- [8] Mohd Azrin Bin Mohd Said; Napsiah Binti Ismail. Improvement of Production Line Layout Using Arena Simulation Software. (Volumes 446 - 447), pp. 1340-1346, Trans Tech Publications Applied Mechanics and Materials. 2013.
- [9] Dong-xu, Z.; Xiao-hua, Z.; Peng-yu, W.; Qing-nian, W., Co-simulation with AMESim and MATLAB for differential dynamic coupling of Hybrid Electric Vehicle. Intelligent Vehicles Symposium, IEEE. 2009.
- [10] Brailsford, S.; Churilov, L.; Dangerfield, B., Discrete-Event Simulation and System Dynamics for Management Decision Making, Published by John Wiley & Sons. 2014.
- [11] Li, B.; Yao, X.; Li, Y.; Tan, W.; Lou, H; Ge, D., Simulation & Optimization for the Gear System of a 6-DOF Manipulator Using Flexible Dynamic of ANSYS, The Open Mechanical Engineering Journal, 8, 69-76, 2014.
- [12] Helal, H.; Rong, W.; Wang, L.; Helal, W., Structure Optimization of the 2D Angular Rotation of a Micro/Nano Table Using the Interface between Matlab and ANSYS. Materials Testing; Vol. 56, No. 5, pp. 408-413. 2014.
- [13] Altas, I.H.; Sharaf, A.M., A Photovoltaic Array Simulation Model for Matlab-Simulink GUI Environment. International Conference on Clean Electrical Power., pp.341, 345. 2007.
- [14] Soares, F.; Costa Branco, P.J., Simulation of a 6/4 switched reluctance motor based on Matlab/Simulink environment. IEEE Transactions on Aerospace and Electronic Systems, vol.37, no.3, pp. 989,1009. 2001.
- [15] Brigati, S.; Francesconi, F.; Malcovati, P.; Tonietto, D.; Baschiroto, A.; Maloberti, F., Modeling sigma-delta modulator non-idealities in SIMULINK(R). Proceedings of the 1999 IEEE International Symposium on Circuits and Systems, vol.2, no., pp. 384,387, vol.2. 1999.
- [16] Takus, D.A.; Profozich, D.M., Arena Software Tutorial. Winter Simulation Conference (WSC'97) pp.541-544. 1997.
- [17] Sun, G.; Wei, M.; Shao, J.; Pei, M., Automotive Powertrain Modeling and Simulation Based on AMESim. SAE Asia Pacific Automotive Engineering Conference. 2007.
- [18] Jun, G.; Yafeng, W.U.; Nisheng, CHU.; Application of AMESim in aircraft hydraulic system. Computer Aided Engineering. 2006.

- [19] Law, A.M., Simulation Modeling and Analysis with Expertfit Software. McGraw-Hill Science, ISBN 978-0-07-329441-4. 2006.
- [20] Ivanov, D.A.; Sokolov, B.; Kaeschel, J., A multi-structural framework for adaptive supply chain planning and operations control with structure dynamics considerations, European Journal of Operational Research. 2009.
- [21] Yuri, G.; Karpov, R.; Ivanovski I; Voropai, N; Popov, D.B., Hierarchical Modeling of Electric Power System Expansion by AnyLogic Simulation Software. IEEE St. Petersburg PowerTech. St. Petersburg, Russia. 2005.
- [22] Quinton, C.; Duchien, L.; Heymans, P.; Mouton, S., e Charlier, E. Using feature modelling and automations to select among cloud solutions. In 2012 Third International Workshop on Product Line Approaches in Software Engineering (PLEASE). IEEE, Jun. 2012, pp. 17–20.
- [23] Clements, P., e Northrop, L. M. Software Product Lines: Practices and Patterns, 2001.
- [24] Pohl, K.; Böckle, G., e Linden, F. Software Product Line Engineering: Foundations, Principles and Techniques, 2005.
- [25] Kang, K.C.; Cohen, S.G.; Hess, J.A.; Novak, W.E.; Peterson, A. S., Feature-Oriented Domain Analysis FODA Feasibility Study. Carnegie-Mellon University Software Engineering Institute. 1990.
- [26] Quinton, C.; Romero, D., e Laurence, D. Automated Selection and Configuration of Cloud Environments Using Software Product Lines Principles. In Proceedings of the 7th IEEE International Conference on Cloud Computing, CLOUD'14. Anchorage, Alaska (USA), 27-02 2014.
- [27] Schmid, K. e Rummler, A. Cloud-based software product lines. In Proceedings of the 16th International Software Product Line Conference - Volume 2 (SPLC '12), Vol. 2. ACM, New York, NY, USA, 2012.
- [28] Mietzner, R.; Metzger, A.; Leymann, F., e Pohl, K. Variability modeling to support customization and deployment of multi-tenant-aware Software as a Service applications. In Proceedings of the ICSE Workshop on Principles of Engineering Service Oriented Systems (PESOS'09). 18-25, 2009.
- [29] García-Galán, J.; Rana, O. F.; Trinidad, P.; Ruiz-Cortés, A. Migrating to the Cloud: a Software Product Line based analysis. In 3rd International Conference on Cloud Computing and Services Science (CLOSER), 2013, pp. 416–426.
- [30] Silva-Neto, B.; Oliveira, A. A programação matemática na análise de sistemas produção agropecuária. Parte I. UNIJUI, 2007.
- [31] Rafikov, M. Métodos de programação linear e não linear. UNIJUI, 2002.
- [32] Harman, M.; Mansouri, S. A.; Zhang, Y. Search-based software engineering: Trends, techniques and applications. ACM Computing Survey, 45, 1, Artigo 11, 61p. 2012.
- [33] Kirkpatrick, S.; Gelatt, Jr. C. D.; Vecchi, M. P., Optimization by Simulated Annealing. Science, n. 220, 1983, p. 671 – 680.
- [34] Goldberg, D. E., Genetic Algorithms in Search, Optimization, and Machine Learning. EUA: Addison-Wesley, 1989.
- [35] Banzhaf, W.; Nordin, P; Keller, R; Francone, F., Genetic Programming: An Introduction: On the Automatic Evolution of Computer Programs and Its Applications (The Morgan Kaufmann Series in Artificial Intelligence), 1998.
- [36] Greiner, R., Probabilistic Hill-Climbing: Theory and Applications; In: CSCSI-92, May 1992.
- [37] Glover, F. "Tabu Search — Part I", ORSA Journal on Computing, 1989, 1: 3, 190-206.
- [38] Feo, A.; Resende, M.G.C., A probabilistic heuristic for a computationally difficult set covering problem. Operations Research Letters, 8:67–71, 1989.

- [39] Bendall, G.; Margot, F., Greedy Type Resistance of Combinatorial Problems, *Discrete Optimization* 3 (2006), 288–298.
- [40] Tarantola, A., *Inverse Problem Theory*, Elsevier, 1987.
- [41] Apt, K. *Principles of Constraint Programming*. Cambridge University Press, 2003.
- [42] Harman, M. e Jones, B.F. Search-based software engineering. *Information and Software Technology*, 2001, pp. 833-839.
- [43] Freitas, F.G.; Maia, C.L.B.; Coutinho, D.P.; Campos, G.A.L.; Souza, J.T., *Aplicação de Metaheurísticas em Problemas da Engenharia de Software: Revisão de Literatura*. II Congresso Tecnológico Infobrasil, 2009.
- [44] Frantz, R.Z.; Reina-Quintero, A.M.; Corchuelo, R., A Domain-Specific Language to Design Enterprise Application Integration Solutions. *International Journal of Cooperative Information Systems*, 20(2):143-176. 2011.
- [45] Frantz, R.Z.; Corchuelo, R., A software development kit to implement integration solutions. 27th Symposium On Applied Computing, 1647-1652. 2012.
- [46] Frantz, R.Z.; Corchuelo, R.; Molina-Jiménez, C., A proposal to detect errors in Enterprise Application Integration solutions. *Journal of Systems and Software*, 85(3):480-497. 2012.
- [47] Frantz, R.Z.; Corchuelo, R.; Arjona, J.L., An Efficient Orchestration Engine for the Cloud. *IEEE International Conference on Cloud Computing Technology and Science*, 711-716. 2011.
- [48] Frantz, R.Z.; Corchuelo, R.; Roos-Frantz, F. On the Design of a Maintainable Software Development Kit to Implement Integration Solutions. *Journal of Systems and Software*, 2013. (Enviado e aguardando revisão – extensão do artigo [12]).
- [49] Frantz, R.Z. A DSL for enterprise application integration. *International Journal of Computer Applications in Technology*, 33(4):257-263, 2008.
- [50] Frantz, R.Z.; Corchuelo, R.; Molina-Jiménez, C., Towards a Fault-Tolerant Architecture for Enterprise Application Integration Solutions. *OTM Workshops*, 294-303, 2009.
- [51] Frantz, R. Z.; Corchuelo, R.; Osuna, C. R.; Molina-Jiménez, C., Monitoring errors in integration workflows. *International Conference on Software Engineering Research and Practice*, 598-604, 2011.
- [52] Frantz, R.Z.; Corchuelo, R.; Molina-Jiménez, C., Error-Detection in Enterprise Application Integration Solutions. *Springer*, vol. 220, ch. 18, 170-179, 2011.
- [53] Frantz, R.Z.; Corchuelo, R.; Molina-Jiménez, C., Una Arquitectura para el Diseño de Soluciones de Integración de Aplicaciones Empresariales con Soporte para Tolerancia a Fallos. *VI Jornadas Científico-Técnicas en Servicios Web y SOA (JSWEB)*, vol. 1, 51-62, 2010.
- [54] Klein, M.J.; Sawicki, S.; Roos-Frantz, F.; Frantz, R.Z, On the Formalisation of an Application Integration Language Using Z Notation. In: *16th International Conference on Enterprise Information Systems*, 2014, 1:314-319, Lisbon.
- [55] Campbell, R.H.; Randell, B., Error recovery in asynchronous systems. *IEEE Transactions on Software Engineering*, 12 (8), 811–826, 1986.
- [56] Avizienis, A.; Laprie, J.-C.; Randell, B.; Landwehr, C., Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 1 (1), 11–33, 2004.