Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 100 (2016) 407 – 413

Procedia Computer Science

Conference on ENTERprise Information Systems / International Conference on Project MANagement / Conference on Health and Social Care Information Systems and Technologies, CENTERIS / ProjMAN / HCist 2016, October 5-7, 2016

Mathematical Model for Simulating an Application Integration Solution in the Academic Context of Unijuí University

Adriana R. Kraisig*, Franciéli C. Welter, Igor G. Haugg, Roberto Cargnin, Fabricia Roos-Frantz, Sandro Sawicki, Rafael Z. Frantz

Department of Exact Sciences and Engineering, Unijuí University. Rua do Comércio, 3000. Ijuí 98700-000, RS, Brazil.

Abstract

The enterprise applications consist of a set of applications that make up the software ecosystem. These can be developed in-house or acquired from third party companies. Generally these applications are developed in order to meet a demands/specific business need of the company without worrying of interacting with another existing system. The area of Enterprise Application Integration (EAI) is able to provide all techniques and tools in order to integrate heterogeneous applications software ecosystem of companies. The Guaraná DSL is a language that enables designing conceptual models of integration solutions using a concrete graphical and intuitive syntax. Integrating the applications is not a trivial task and solution development involves addition of costs (time and resources), risks such as the appearance of bugs that most often are observed only after the implementation of the conceptual model. In this paper we propose a simulation model using Petri Nets that enable the analysis of the behavior of an integration solution developed to a real-world integration problem at Unijuí University.

Peer-review under responsibility of the organizing committee of CENTERIS 2016

Keywords: Enterprise Application Integration; Domain-Specific Language; Conceptual Model; Simulation Model; Petri Nets.

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000 .
E-mail address: maryshelei@yahoo.com.br

Peer-review under responsibility of the organizing committee of CENTERIS 2016
doi:10.1016/j.procs.2016.09.176

1. Introduction

Software applications have become indispensable in the business environments. Enterprises usually buy third-party software packages or develop their own applications. In both cases it is common that there is no concern for their integration. Usually, this creates a problem and makes difficult software reuse whenever there is a new business process, or a current process that needs to be evolved and uses support of existing applications in the enterprise. There are software packages what can be purchased by companies, generally developed without having integration in mind.

It is common for the enterprise to create or buy, external applications to address their demands, when your software ecosystem becomes inefficient with respect to new data or functionality. A software ecosystem consists of a set of software solutions that support and automate activities and transactions of users[1]. A software ecosystem refers to a set of applications that provide a degree of relation[2].

To avoid unnecessary additional costs with the implementation of an integration solution it is important to analyze it to ensure the solution does not have any problem, chiefly regarding its performance[12]. An alternative would simulate the integration solution. The simulation is a way to analyze the behavior of a system, which is represented by a simulation model.

An integration solution is modeled using the Domain-Specific Language (DSL) of Guaraná platform. Guaraná DSL is composed of a set of applications and tasks that cooperate to integrate the different applications. In an integration solution messages are transmitted and contain information that is synchronized amongst applications[3]. Its representation is by means of a conceptual model, which describes the organization of the components in a high-level of abstraction. There are several platforms to implement integration solutions. Integrating applications is not a trivial task and solution development may involve additional costs, riscs, such as the appearance of bugs and performance bottlenecks that are usually observed after implementation.

Enterprise Application Integration (EAI) consists of a response to decades of building monolithic applications, which were not designed to work together with other applications. The EAI aims to solve a problem that has existed since the beginning of the first applications development: the difficulty of integrating applications that were created separately[4]. It is a combination of technologies that have the role of organizing business processes, offering integration between applications with no changes in its data structures. From a business point of view, is a unified information system with easy access to valuable and decisive information inside an enterprise. Proposes the use of integration solution, to provide communication between applications.

A method of Enterprise Application Integration should be functional and inexpensive[4]. Enterprises generally seek to analyze, still in the design phase, the integration solutions. All analysis done at this stage leads the possibility to observe the behavior of the solution, and identify potential performance bottlenecks. The identification of bottlenecks in the early stages of development can contribute to improve the quality of integration solutions.This paper proposes a formal simulation model, using Petri Nets, to analyze the behavior of the integration solution that synchronizes several services to students in the academic context of the Unijuí University.

This paper is structured as follows: Section 2 presents background information of Guaraná DSL and Petri Nets; Section 3, presents the case study; Section 4, comes up the simulation model using Petri Nets; and, Section 5 comes up the conclusion and future work.

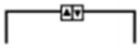
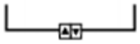
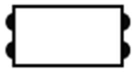
2. Background

This section provides a brief introduction to Guaraná DSL, which is used for conceptual modeling of the integration solutions, and Petri Nets, which are used for the development of the formal simulation model.

2.1. Guaraná DSL

A Domain Specific Language (DSL) is tailored to focus on a single and concrete domain and accurately describes its semantics[5]. Guaraná DSL is among the various technologies available to develop conceptual models of enterprise application integration solutions. The models designed with Guaraná DSL have a high-level of abstraction and are platform-independent[6]. This language has a graphical concrete syntax and it provides support for application integration patterns documented by Gregor Hohpe and Bobby Woolf[7].Guaraná DSL provides building blocks to represent several integration patterns, which include resources, processes, ports, tasks, and slots[8]. Table 1 shows the concrete sintax of these building blocks in Guaraná DSL language.

Table 1. Building blocks of Guaraná DSL (from [6]).

Notation	Concept	Notation	Concept
	Resource		Solicitor Port
	Integration Process		Responder Port
	Entry Port		Task
	Exit Port		Slot

A message is the abstraction of information that is exchanged and transformed by the integration solution, it consists of header and body. A task has inputs and outputs, through which flows messages and they communicate one each other by means of slots. A slot interconnects two tasks, allowing them to exchange messages asynchronously. Ports interact with applications by sending and receiving messages. The integration process contains the logic that performs processing, routing, modification and operations on the messages specified by using the different kinds of tasks of the language.

The routers are responsible for directing an inbound message to zero, one or more destinations, or yet replicate, correlate, or ordering of messages in the flow of an integration solution. The modifiers are responsible for adding or removing data from messages without changing their schema. Transformers, transform the contents of the message to another format by changing their schema. Timers, allow to control time within the integration flow. Guaraná DSL by Gregor Hohpe and Bobby Woolf[6] is based on Pipes and Filters architecture. In wich, a process can be divided into a number of independent service (filters), which are connected by channels (pipes). In Guaraná DSL, pipes are implemented by slots, and filters are implemented by tasks.

2.2. Petri Nets

Petri Nets (PNs) it was proposed by Carl Adam Petri, and according to Norian Marranghello[9] it is a general purpose mathematical tool. It is use, to model the behavior of discrete event systems, describing the relationship between conditions and events and allow analyzing properties such as parallelism, synchronization, and resource sharing[10]. Are formed by two components: an active, called the transition (rectangle) and other liabilities denominated place (circle). The places are equivalent to the state variables and transitions correspond to the actions performed by the system. These two components are connected by directed arcs. The token is represented graphically by a point inside a place. Petri Nets use different mathematical theories in its bases, such as bag theory, matrix algebra, and theory advocated by relations[11]. They are composed of Places, Transitions, and Arcs. A place equivalent to the state variables. A transition corresponds to the actions performed by the system. An Arch connecting places to transitions and can be classified into single or multiple. Figure 1, shows the notation of these elements.

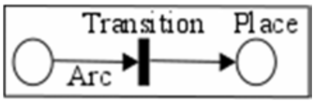


Fig. 1. Basic Elements of a Petri Nets (from [11]).

Petri Nets can be classified according to their degree of abstraction, therefore, they can be separated into low and high-level. There are extensions that can cover low-level Petri Nets and high-level, which seek to include hierarchies and temporal aspects. Amongst them should be highlighted: the colored extensions, in which it is possible to have different types of tokens; hierarchical extensions, which aims to reduce the model size; timed extensions that add time to the models. Timed Petri Nets, can be divided into deterministic and stochastic. The deterministic, work with absolute time on implementation of the corresponding events, and the stochastic considers uncertainties in the moment of the execution of the events by associating probability functions to the system, which allows the determination of the run time[9].

3. Case Study

This section introduces the integration problem and its corresponding conceptual model designed using Guaraná DSL. The goal of this integration solution is to automate the generation of a list of all the subjects that a student did not attended yet and that will be offered in the next semester.

3.1. Software Ecosystem

The proposed solution involves five applications: Web Portal, Registration Subjects, Student Registry, Billing System and Mail Server. In this context, each application runs on different platforms. These applications were developed by the university. These applications are heterogeneous and they may cooperate to support the re-enrollments process. The mail server provides POP3 and SMTP interfaces. The Web Portal provides a variety of information for students, such as the disciplines already routed, the percentage already done the course. It is also where there are processes such as re-enrollments, so that every semester students must apply through the Portal disciplines who wish to attend the next semester. The Student Registry provides a database with information such, as name, RG student and enrolled course. Registration Subjects provides a database that contains information such, as the name of discipline, its course, semester, classes available and date. When a student is performing their registration or enrollment renewals through the Portal, it is possible through the information acquired in the application Student Registry, consult the application Registration Subjects, which subjects the student not yet realized and what disciplines will be offered next semester. Starting from this information to generate a list of possible subjects to be processed through the next academic semester. From the list of subjects, it is possible to check the price of the subjects in the Billing System, and generate the price of the subjects in the subjects list. This list may be sent to the student by e-mail, using the Mail Server.

3.2. Conceptual Model

The integration solution takes the requests from students from the web portal of Unijuí University through port P1 and produces messages which are stored into slot S1, see Fig. 1.Task T1 filters inbound messages, only accepting applications for enrollment renewals, which were held within the enrollment period. So, accepted requests are used to query the student registry application by means of port P2 for more information relating to the student who made the request. Task T2 is responsible to produce this query message. Task T3 replicates the response message from the Student Registry application. One copy is used to query the Registration Subjects, by means of port P3, for a list of disciplines present in the course in which the student is enrolled. The other copy (base message) waits in the inbound slot S6 of task T5, for its correlated response message so that both messages are made available to task T6, which enriches the base massage with the data from the response message. Task T7 discards messages that contain subjects that the student has already done and also subjects that are not offered in the next academic semester. After this, then Task T8 again replicates the message. One copy is used to query the Billing System, by means of port P4, for the cost of each subject. Task T11 enriches the base message with the cost of the subjects returned by the Billing System. Then, messages are transformed into an email message by task T13, containing the list of subjects and their costs, and forwarded to the Mail Server by means of port P5.

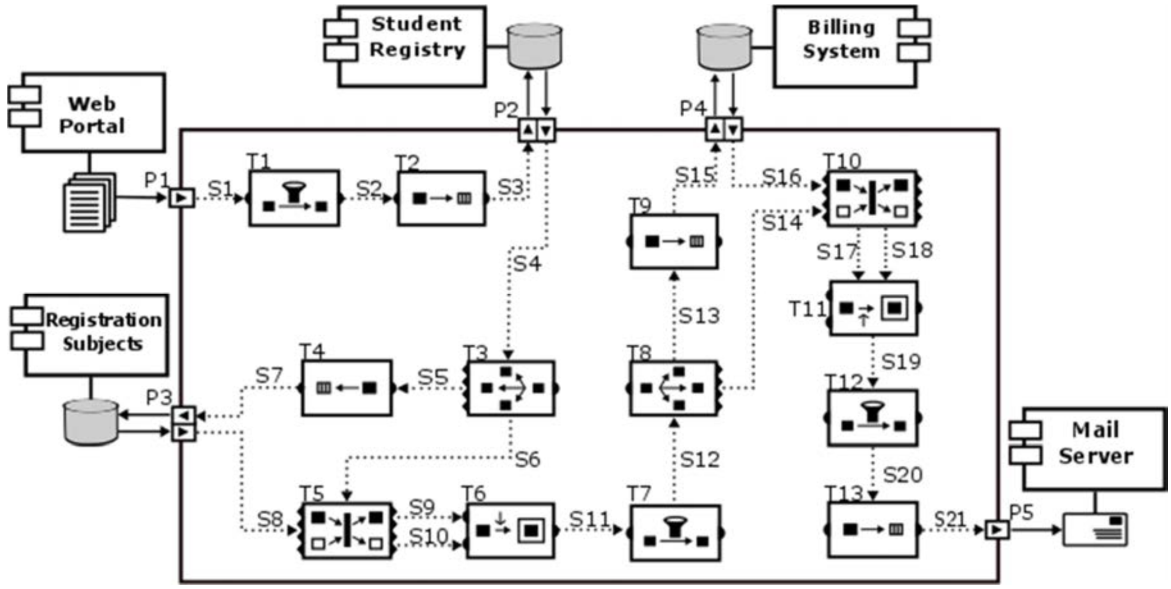
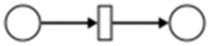
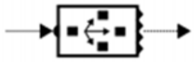
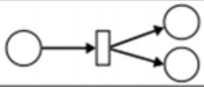
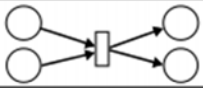
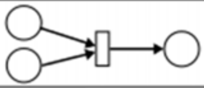
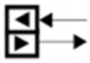
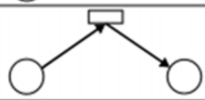
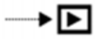
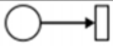
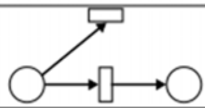


Fig. 2. The conceptual model for the integration problem at Unijuí University.

4. Simulation Model

The development of a simulation model comprises the three big steps explained by Ray Paul and David Balmer[13]: formulation the model, the model implementation, and the analysis of results. The behavior of a simulation system is based on a simulation model. After being built the model, it is possible for it to be validated by simulation. From the validation of the model it becomes possible to identify performance bottlenecks looking for to reduce costs of implementation, risk and time. Table 2 presents the mapping of the building blocks from Guaraná DSL, as to its corresponding notation in Petri Nets. Fig. 2 shows the corresponding simulation model using Petri Nets proposed in this paper to enable the simulation of the conceptual model of the case study.

Table 2. Translating elements from Guaraná DSL into Petri Nets.

Concept	Description	Notation in Guaraná	Notation in Petri Nets
Entry Port	Post messages		
Translator	Transform a message in other		
Replicator	Replicate a message to all outpouts		
Correlator	Organize messages that have the same identity		
Context-based Slimmer	Organize and complete a message		
Solicitor Port	Request a message from an external application		
Exit Port	Outpout messages		
Filter	Filter messages		

5. Conclusion

A simulation model could provide many advantages, including: reducing costs, performance bottlenecks identification and optimization of the time spent by software engineers in the implementation of the model itself. This paper presents a proposal for a formal simulation model based on Petri Nets built from the conceptual model of an integration solution developed with Guaraná DSL. As future work, it is proposed to carry out simulation model using a simulation software, with the objective to identify potential performance bottlenecks in the integration solution.

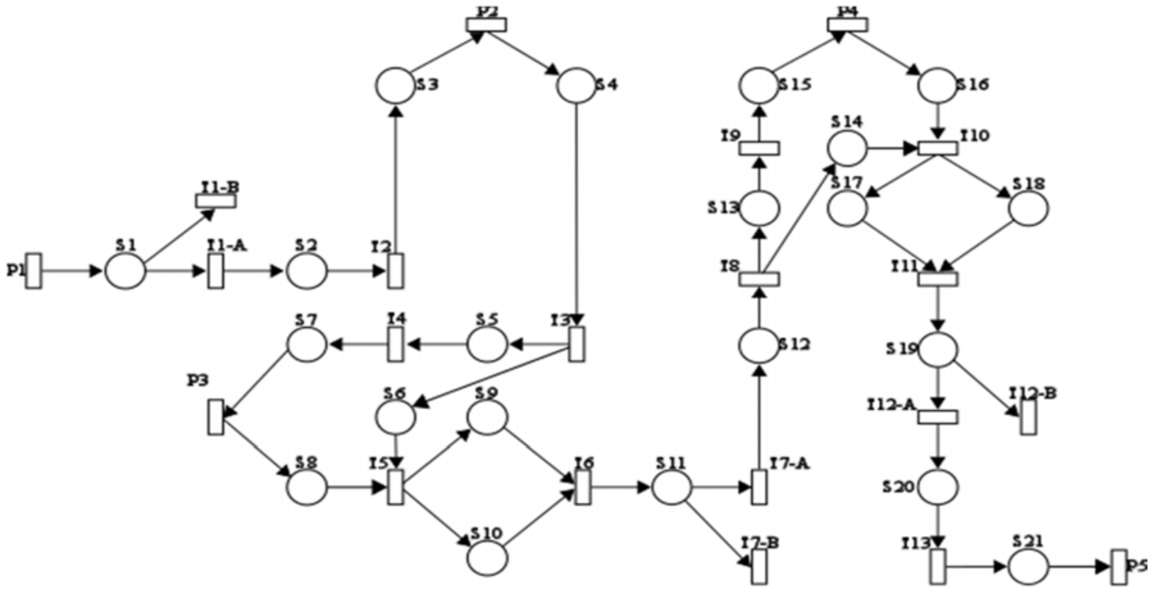


Fig. 3. The simulation model proposed.

Acknowledgements

The research work on which we report in this paper is supported by CNPq, CAPES, FAPERGS, and the internal Research Programme at UNIJUI University.

References

1. Bosch J. From software product lines to software ecosystems. In: Carnegie Mellon University. *Proceedings of the 13th International Software Product Line Conference*. p. 111-119; 2009.
2. Yu L, Ramaswamy S, Bush J. Symbiosis and software evolvability. IT *Professional*, IEEE, v. 10, n. 4, p. 56-62; 2008.
3. 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*, Elsevier, v. 111, p. 89-104; 2016.
4. Linthicum DS. *Enterprise application integration.* [S.l.]: Adisson-Wesley Professional; 2000.
5. Fowler M. *Domain-specific languages*. Pearson Education; 2010.
6. Frantz R.Z., Quintero AMR, Corchuelo R. A Domain-Specific Language to design enterprise application integration solutions. *International Journal of Cooperative Information Systems,* World Scientific, v. 20, n. 02, p. 143-176; 2011.
7. Hohpe G, Woolf B. *Enterprise integration patterns: Designing, building, and deploying messaging solutions.* Addison-Wesley Professional; 2004.
8. Frantz R.Z., CorchueloR, González J. Advances in a DSL for application integration. *Actas de lostalleres de las Jornadas de Ingenieríadel Software y Bases de Datos*, v. 2, n. 2; 2008.
9. Marranghello N. Redes de petri: Conceitos e aplicações. *São Paulo: DCCE/IBILCE/UNESP*; 2005.
10. Barroso NMC, et al. Modelagem de conceitos e processos matemáticos por redes de petri coloridas: o caso da integrabilidade de funções reais. *Boletim de Educação Matemática*, SciELO Brasil, v. 27, n. 45, p. 75-95; 2013.
11. Francês, CRL. Introdução às redes de petri. *Laboratório de Computação Aplicada,Universidade Federal do Pará*; 2003.
12. Sawicki S, Frantz R.Z., Fernandes VMB, Roos-Frantz F, Yevseyeva I, Corchuelo R. *Characterising enterprise application integration solutions as discreteevent system*. IGI Global; 2015.
13. Paul RJ, Balmer DW. *Simulation modeling*.Chartwell-Bratt; 1993.