Published at:
17th International Conference on Enterprise Information Systems (ICEIS), Barcelona (Spain), pages 87-97, v1, SCITEPRESS, 2015.

Using Petri Nets to Enable the Simulation of Application Integration Solutions Conceptual Models

Fabricia Roos-Frantz[1], Manuel Binelo[1], Rafael Z. Frantz[1], Sandro Sawicki[1], Vitor Basto-Fernandes[2]

[1] *UNIJUÍ University, Department of Exact Sciences and Engineering, Ijuí, Brazil*

[2] *Polytechnic Institute of Leiria, School of Technology and Management, Leiria, Portugal*

{frfrantz, manuel.binelo, rzfrantz, sawicki}@unijui.edu.br, vitor.fernandes@ipleiria.pt

Keywords: Simulation, Stochastic Petri Net, Enterprise Application Integration, Integration Solution, Integration Patterns.

Abstract: Enterprise application integration concerns with the use of methodologies and tools to design and implement integration solutions to integrate a set of heterogeneous enterprise applications. Amongst the integration technologies to design and implement integration solutions is Guaraná. This technology provides a domain-specific language that enables the design of conceptual models. The quality of these models is essential to ensure proper integration. Discovering whether an integration solution can fail and in which conditions it is more likely to happen is a costly, risky, and time-consuming task, since current approaches require software engineers to construct the real solution. Generally, simulation is recommended when problems are impossible or expensive to be solved by actual experimentation. Guaraná conceptual models can be classified as stochastic, dynamic, and discrete, and thus it can be simulated taking the advantage of well-established techniques and tools for discrete-event simulation. Therefore, this paper proposes a simulation of Guaraná solutions by using Petri nets, in order to analyse such solutions based only on the conceptual models. It shows that an integration solution conceptual model designed with Guaraná can be translated into a formal model structured as a Stochastic Petri net. The equivalence of both models is verified by comparing the operation of the Guaraná runtime system with the behaviour of a Petri net execution process.

1 INTRODUCTION

Today's companies face the challenge of integrating their usually heterogeneous software ecosystem and reduce the high costs associated with this integration. Companies usually spend nearly 5 to 20 times more to integrate their applications than to develop a new functionality (Weiss, 2005). To cope with this problem, companies rely on methodologies and tools to design and implement integration solutions provided by the Enterprise Application Integration (EAI) community. An integration solution is able to keep a number of applications' data in synchrony or to develop new functionality on top of them, without changing or disturbing the applications being integrated (Hohpe and Woolf, 2003).

Currently, there are a number of integration technologies to design and implement integration solutions (Ibsen and Anstey, 2010; Fisher et al., 2010; Dossot and D'Emic, 2009; Frantz and Corchuelo, 2012). The EAI community is pushing the integration technologies to provide support for the realisation of the enterprise integration patterns documented by Hohpe and Woolf (2003). These conceptual patterns document a set of well-known services that are directly related to integration tasks and have been turned into a cookbook for software engineers to develop their solutions. The design of conceptual models is usually supported by a domain-specific language built on top of these patterns. In this paper we focus on Guaraná integration technology (Frantz and Corchuelo, 2012). Guaraná not only provides support for the design of conceptual models using such kind of domain-specific language, but it also proposes a model-centric development approach. According to the authors, Guaraná is based on the design of platform-independent models that can be refined, by means of model transformation techniques, to generate the corresponding source code for the integration solution.

The quality of conceptual models for integration solutions is essential to ensure proper integration. Discovering whether an integration solution can fail and in which conditions it is more likely to happen is a costly, risky, and time-consuming task. It incurs because the current approach practiced by software engineers requires the construction of the complete integration solution, its execution in a runtime system, and the collection of data from this execution, so that the integration solution can be analysed. An approach that enables the analysis of the integration solution without having to construct it, taking as input only the conceptual models devised at the design phase, would help to reduce cost, risk, and time spent.

A conceptual model is characterised by its high level of abstraction. A good solution would be to translate a conceptual model into another high level abstraction model that relies on automated analysis support, since the goal is not to generate source code. In the Model-Driven Engineering discipline (Schmidt, 2006), such transformation is classified as *horizontal* and *exogenous transformation*. "A horizontal transformation is a transformation in which the source and target models reside at the same abstraction level" and "exogenous transformations are transformations between models expressed using different languages" (Mens and Gorp, 2006).

Guaraná conceptual models can be classified as stochastic, dynamic, and discrete. As a discrete system, it can be simulated taking the advantage of well-established techniques and tools for discrete-event simulation. Simulation models are used to get answers about the operation of the system and may be able to analyse several aspects of it. Therefore, to be able to analyse Guaraná solutions based only on the conceptual models, we propose a simulation of such solutions by using Petri nets (Bause and Kritzinger, 1998). This mathematical modelling language can be used to model discrete event systems of any kind (David and Alla, 1994), and their graphical nature allows the visualisation of the system behaviour comprising concurrency, synchronisation and resource sharing. They are executable, have a mathematically defined semantics, and provide a mathematical theory for analysis and verification of certain properties of a process.

Guaraná follows the Pipes and Filters architectural style (Hohpe and Woolf, 2003). In this style, a larger process is divided into a number of smaller and independent services (Filters), which are usually unsynchronised by channels (Pipes). In an integration process, there are two main components: *i)* pipes, implemented by *slots*, and *ii)* filters, implemented by *tasks*. In an integration solution, messages flow through integration processes, going from slots to tasks or from tasks to slots, and are processed asynchronously. Similarly, Petri nets are structured as follows: *i)* they are bipartite graphs, since they have two types of nodes, *places* and *transitions*, *ii)* they are directed graphs, arcs can connect places to transitions and transitions to places only, and *iii)* they are multi-graphs, its possible to have multiple parallel arcs between a place and a transition. As in the case of an integration solution, in which the execution of a task depends on the availability of messages in all slots connected to its inputs, in Petri nets the execution of a given transition occurs only when a token is available in all input places connected to such transition.

The aim of this paper is to show that an integration solution conceptual model designed with Guaraná can be translated into a formal model structured as a Stochastic Petri net. We propose to verify the equivalence of both models by comparing the operation of the Guaraná runtime system with the behaviour of a Petri net execution process.

According to Rezai et al. (1995), Petri nets have proven themselves to be an excellent modelling and analysis tool for discrete event or asynchronous systems. In this work, the authors use a new extension of Petri nets, called Global Petri net that provides a means for combining differential and difference equations to represent time dependent or synchronous systems. The goal is to show how it can be used to model and simulate hybrid systems. An example of real time control systems was used to show the Global Petri net modelling and simulation capabilities.

Zhou (1998) presents a Petri net approach aiming at modelling, analysing, simulating, scheduling, and controlling for semiconductor manufacturing systems. This paper presents the fundamental concepts of Petri nets, its importance for qualitative and quantitative analysis of systems. The author also describes that this kind of system can be characterised as discrete-event systems, however, it can exhibit sequential, concurrent, and conflicting relations amongst the events and operations having its evolution dynamic over time.

Cavalieri (2000) proposes a methodology to improve the performance of a Flexible Manufacturing System (FMS), based on the use of Petri nets. The goal is to model the system to represent the general behaviour of the FMS, avoiding a detailed description of its operations. The Petri net model of the FMS is used to evaluate its performance through simulation.

Kuo (2004) proposes a simulation and control environment for the distributed event dynamic systems, based on the distributed agent-oriented Petri net, a type of high-level Petri nets. Davidrajuh (2008) presents a new Petri net tool for modelling and simulation of discrete-event systems. According to Alla and Ghomri (2012), Petri nets are widely used to model discrete event dynamic systems. The paper, the authors have presented the basic ideas describing the continuous and hybrid models. In addition, they describe that continuous and hybrid Petri nets allow modelling and analysis of continuous and hybrid systems on the same conceptual basis.

Narahari et al. (1989) report that the Stochastic Petri nets have emerged as a main performance-modelling tool for distributed systems and the discrete-event simulation is the preferred tool for its performance evaluation. The authors present several algorithms based on Stochastic Petri nets aiming at discrete-event simulations of distributed systems.

David and Alla (1994) present the basic concept of Petri nets and the classes of derived models that can be used for dynamic system modelling. The authors report that autonomous Petri nets enable a discrete event system of any kind to be modelled. Then, the relationship between system and its model based on Petri nets allows the qualitative validation of the functioning process.

Gold (2004) presents an integrated software engineering approach for the use of Petri nets in software development from analysis to testing of software. The author uses the classical software development process with simulation and mathematical analysis. The main advantage is that requirements can be validated earlier, thereby, the fault detection and correction is less expensive. In addition, the results can be simulated and analysed using Petri net tools.

Other authors also propose the use of Petri nets for conceptual modelling. van der Aalst et al. (2013) propose the use of Coloured Petri nets to model complex process, arguing that this extension of a classical Petri formalism is more expressive and thus it is not necessary to abstract from relevant aspects of such complex process. Russell et al. (2009) also apply Petri net formalism to get modelling benefits since they present commonalities with the domain of workflow systems.

From the analysis of the literature, it is clear that Petri nets have been widely used as support for system modelling and simulation, nevertheless it has not been explored yet in the context of EAI to simulate integration solutions taking as input their conceptual models.

The rest of this paper is organised as follows: Section 2 gives an overview of Guaraná technology and Stochastic Petri nets; Section 3, presents our proposal, in which we show how to translate Guaraná conceptual models into Petri net formal models; Section 4 presents a case study; Section 5 discusses the analysis results obtained from the simulation of the formal model, and, finally, Section 6 concludes this paper.

2 BACKGROUND

In this section, we provide a brief overview of Guaraná technology and Stochastic Petri nets.

2.1 Guaraná

Guaraná technology is the result of a research effort to provide support for the design, implementation, and execution of enterprise application integration solutions. This technology is composed of a domain specific language, a software development kit, and a runtime system. The domain specific language is a graphical modelling language used to support software engineers in the design of integration solutions at a high level of abstraction (i.e., conceptual models) and based on the well-known integration patterns documented by Hohpe and Woolf (2003). The software development kit provides a Java application programming interface to implement conceptual models into executable code. The runtime system is used to execute the implemented integration solutions.

Guaraná language has a number of constructors, each of them representing a given application integration concept. *Messages* are an abstraction of a piece of information that are exchanged and transformed across an integration solution. They are composed of a header, a body, and one or more attachments. *Tasks* represent an atomic operation that can be executed on messages. They model how a set of inbound messages must be processed to produce a set of outbound messages, such as routing the inbound messages, modifying them, transforming them, performing time-related actions, and mapping them to/from objects. *Slots* are buffers connecting an input of a task to the output of another task, providing asynchronous message processing. *Ports* abstract processes away from the communication mechanism in an inter-process communication or in the communication of the integration solution with an application. *Processes* are the central processing units in an integration solution, they are composed of ports and tasks.

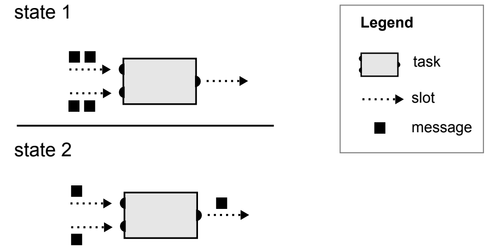

Figure 1: Sample of task execution.

In a conceptual model, an integration solution represents one or more integration processes trough which messages flow and are processed asynchronously. The workflow is implemented as a Pipes and Filters architecture, in which pipes are implemented by Slots and filters are implemented by Tasks. Tasks become ready to be executed every time there is a message available in every input. Becoming ready for execution implies that the runtime system creates a *work unit* to register the task into a queue of executable tasks, and latter can assign a thread to execute it; however, this does not entail that the task produces a set of outbound messages, since it depends on the inbound messages and the semantics implemented by the task. For instance, a filter is a task that reads a message from its input and may or may not remove this message from the workflow; such a task eventually may not produce outbound message. As slots enable asynchrony in an integration solution, messages are stored on them until the tasks are able to process these messages. When a task is executed, the integration solution changes its state, this concept is illustrated in Figure 1. A detailed discussion on the domain-specific language of Guaraná is provided by the authors in Frantz et al. (2011).

2.2 Stochastic Petri Net

In 1962 Carl Adam Petri invented the Petri nets as a formalism to describe the concurrency and synchronisation dynamics in distributed systems (Bause and Kritzinger, 1998). Petri nets are graphs composed of two kinds of nodes, places, and transitions. Places are represented by circles and Transitions by rectangles, these nodes are connected by arcs. Places can contain tokens, represented by dots inside the place, and the amount of tokens in a place is called a marking. An input arc connects a place to a transition and an output arc connects a transition to a place. When a transition is fired, the tokens in the places connected to input arcs are removed and tokens are added to the places connected to the output arcs. A transition is active and can be fired if the amount of tokens determined by the input arcs exists in the respective places. The amount of tokens generated by the output arcs are not necessarily the same as the removed by the input arcs. When a transition fires, the Petri net changes its state, this concept is illustrated in Figure 2.

Various variations of Petri nets have been used to model different classes of problems. Amongst them, the continuous-time stochastic Petri net, or simply stochastic Petri net (Molloy, 1981), formed by the tuple (P,T,I,O,M_0,Λ), where P is the set of places, T is the set of transitions, I is the set of input arcs, O is the set of output arcs, M_0 is the initial marking and Λ is

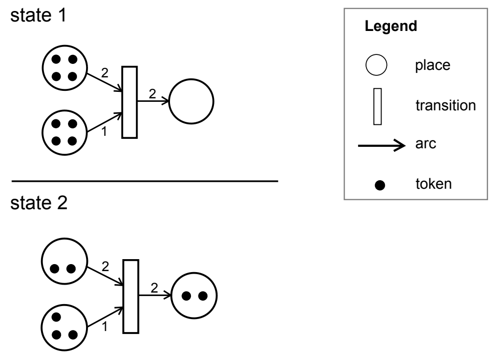

Figure 2: Sample of firing a transition.

the set of transition rates. The transition rate is a firing delay for the transition determined by a random variable X with a negative exponential probability density function $F_{X_i}(x) = 1 - e^{-\lambda_i x}$ where $-\lambda_i$ is the firing rate of transition t_i. The firing rate is marking dependent and the average firing delay of transition t_i in marking M_j is $[\lambda_i(M_j)]^{-1}$ (Marsan, 1990).

3 PROPOSAL

In this work, we propose the translation of the Guaraná model into a stochastic Petri net model. This translation brings several advantages. The first benefit is the fact that stochastic Petri nets are a very well established mathematical model, they can be translated into other formal models and provide various statistical analysis of the model behaviour. Another benefit is that Stochastic Petri have simple components (places, transitions, and arcs), are very generalisable, and there are numerous tools available for Petri nets simulation and analysis.

In order to translate the Guaraná model into a Petri net, it is necessary to use a table, our Rosetta Stone, to translate each Guaraná component into an equivalent Petri net. The translation of a specific Guaraná solution into a stochastic Petri net is carried out by connecting these Petri net components in the same fashion of the components in the Guaraná solution. In this paper we have considered the following kinds of tasks: communicator, mapper, modifier, router, stream dealer, and timer. *Communicator* tasks enable to send or receive messages through ports to applications being integrated. *Mapper* tasks allow to map the body of a message from XML to Stream or vice-versa. Modifier tasks add or remove data from the body and the head of inbound messages. *Router* tasks get one or more messages from its input slots and transfer or copy them to one or more output slots. *Stream dealer* tasks allow to operate on messages whose body contains raw data. *Timer* tasks allow to execute timer-based operation on messages. Figures 3, 4, 5, 6, 7, 8 and 9 show the translation of each type of task from Guaraná into its Petri net representation.

Petri net	Guaraná	Task name
		In Communicator
		Out Communicator
		In-Out Communicator
		Out-In Communicator

Figure 3: Communicator tasks.

Petri net	Guaraná	Task name
		XML2Stream
		Stream2XML

Figure 4: Mapper tasks.

Petri net	Guaraná	Task name
		Slimmer
		Context-based Slimmer
		Content Enricher
		Context-based Content Enricher
		Header Enricher
		Context-based Header Enricher
		Header Promoter
		Header Demoter
		Set Correlation ID
		Set Return Address

Figure 5: Modifier tasks.

The translation from Guaraná to Petri net is a one way process. This happens because, even though all information necessary to the time and concurrency simulation can be translated to the Petri net,

Petri net	Guaraná	Task name
		Correlator
		Merger
		Resequencer
		Filter
		Idempotent Transfer
		Dispatcher
		Distributor
		Replicator
		Semantic Validator
		Threader

Figure 6: Router tasks.

Petri net	Guaraná	Task name
		Zipper
		Unzipper
		Encrypter
		Decrypter
		Encoder
		Decoder

Figure 7: Stream dealer tasks.

the domain-specific language task components have semantic information that cannot be translated. The result is that various semantically different Guaraná components are mapped to identical Petri net. Take for example the components *zipper* and *content enricher*, both have different meaning in the domain-specific language of Guaraná, but share the same abstract functionality, that take a message, do an operation on the message and output the message. Both are represented by the Petri net composed of two places and one transition, as can be seen in Figures 5 and 7.

When we translate a solution, every message in Guaraná becomes a token, and every *work unit* in Guaraná becomes a transition firing in Petri net, so that the execution of the stochastic Petri net is equiv-

Petri net	Guaraná	Task name
		Delayer
		Ticker
		Expire Checker

Figure 8: Timer tasks.

Petri net	Guaraná	Task name
		Translator
		Splitter
		Aggregator
		Chopper
		Assembler
		Cross Builder

Figure 9: Transformer tasks.

alent to the execution of the single-threaded Guaraná runtime system.

The firing probability, or firing rate of each transition of the Petri net, may have two different meanings in the translated model. First, it can be the actual firing rate, i.e., how many times an operation was performed. One example is the filter in Figure 10, with transition T_1 and T_2 with the respective firing rates a and b. Transition T_1 lets messages (tokens) pass to the next stages, while transition T_2 removes messages from the system. If rate $a = 0.9$ and rate $b = 0.1$, then 10% of messages are filtered out from the system. Second, it can be the time a task takes to execute. This interpretation can be made because stochastic Petri nets (also called temporal Petri nets), can be analysed from a temporal perspective. The probability associated with the transition determines its frequency of firing, and since time is the inverse of frequency, the time consumed for each task can be computed by $t = 1/f$, in work unities. Using a *slimmer* task as example, if the transition has a fire rate of 0.5, which means that the task has an average processing time of two work unities.

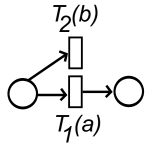

Figure 10: Petri net representation for a filter task.

4 CASE STUDY

This case study consists of a real-world integration problem that builds on a project to automate the registration of new users into a unique repository of the Huelva's County Council. This repository contains information about users that comes from both a local application and a web portal. It is expected that every new user is notified and provided with his/her digital certificate by secure e-mail.

4.1 The software ecosystem

The integration solution involves six applications, namely: Local Users, Portal Users, LDAP, Human Resources System, Digital Certificate Platform, and Mail Server. Each application runs on a different platform, and, except for the LDAP, the Digital Certificate Platform, and the Mail Server, they were not designed with integration concerns in mind.

The Local Users is the first application developed in house; it aims to manage the county council information systems' users. Note that, this is a standalone application and does not provide an authentication service. The Portal Users is an off-the-shelf application that the web portal uses to manage its users. In addition, a unique repository for users has been set up using an LDAP-based application, so that it can provide authentication access control for several other applications inside the software ecosystem. The Human Resources System is a legacy system developed in house to provide personal information about the employees. It is a part of the integration solution since we require information like name and e-mail to compose notification e-mails. Another application developed in house is the Digital Certificate Platform, which aims to manage digital certificates; it was designed with integration concerns in mind. Amongst other services, this application can be queried to get a URL that temporarily points to a digital certificate that users can download after authenticating. Finally, the Mail Server runs the Council's e-mail service, which is used exclusively for notification purposes.

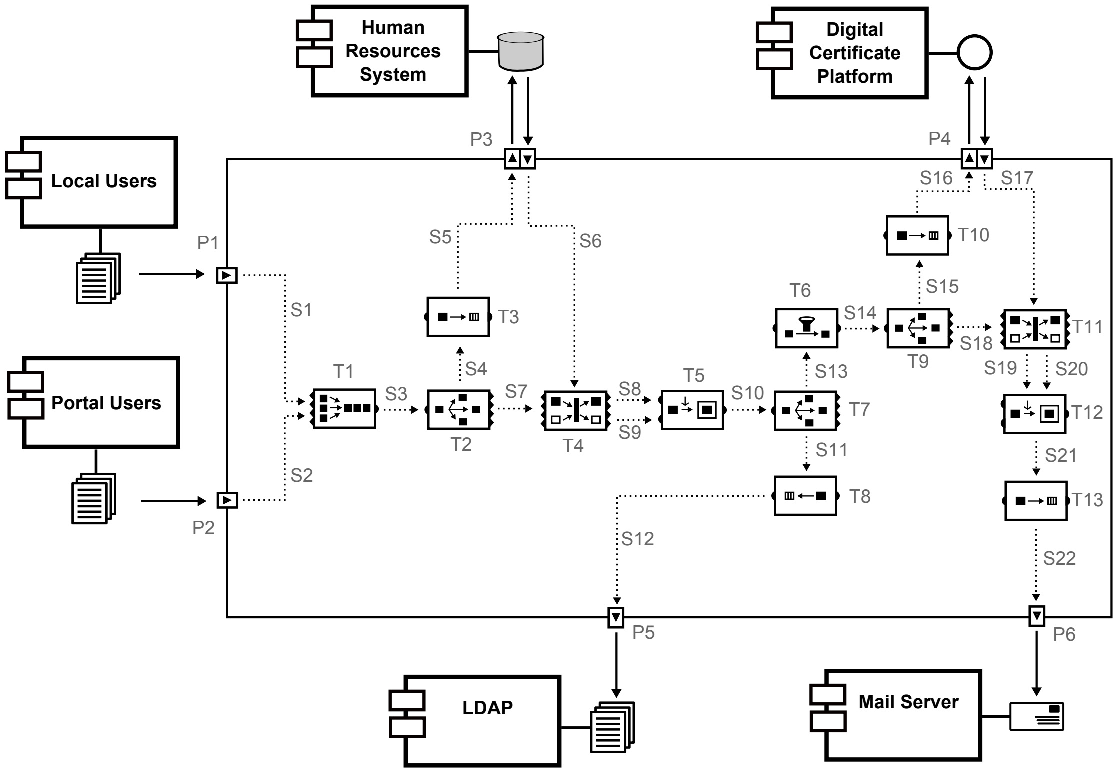

Figure 11: The Huelva's County Council conceptual model for the integration problem.

4.2 Conceptual Model

The integration solution we have devised using Guaraná is composed of one orchestration process that exogenously co-ordinates the applications involved in the integration solution. Figure 11 shows the conceptual model of this integration solution. Some ports use text files to communicate with Local Users, Portal Users, and LDAP; the Human Resources System is queried by means of its database management system; and, the communication with the Digital Certificate Platform and the Mail Server is performed by means of APIs. Translator tasks were used to translate messages from canonical schemas into the schemas with which the integrated applications work.

The workflow begins at entry ports P1 and P2, which periodically poll the Local Users and Portal Users logs to find new users. Inbound messages wrapping the data that has been polled are written by these ports to slots S1 and S2. Inside the process, task T1 gets messages coming from both ports and adds them to slot S3. Task T1 is a merger task, which does not change the message content. The next task in the flow is a replicator that creates two copies of every message it gets from slot S3, so that one copy can be used to query application Human Resources System by means of solicitor port P3 for information about the employee who owns a user record, and the other copy (called base message) is written to slot S7 so latter it can be correlated by task T4 with the response from the Human Resources System. Task T3 is a translator that change the schema of messages to create the query message. Next, task T5 enriches the base message with the information in the response message and then task T7 replicates this enriched message with copies to the LDAP and the Digital Certificate Platform. The new user record is written to the LDAP by means of exit port P5. Before querying the Digital Certificate Platform, task T6 filters out messages that do not include an e-mail address. Messages that go through task T12, which enriches them with the corresponding certificate. Finally, exit port P6 communicates with the Mail Server application to send the certificate and notify the employee about his/her inclusion in the LDAP.

4.3 Formal Model

Figure 12 shows the stochastic Petri net that models the case study presented in the previous section. The net was built following the translation from Guaraná tasks to the respective Petri nets, as presented in Section 3. These nets were connected to

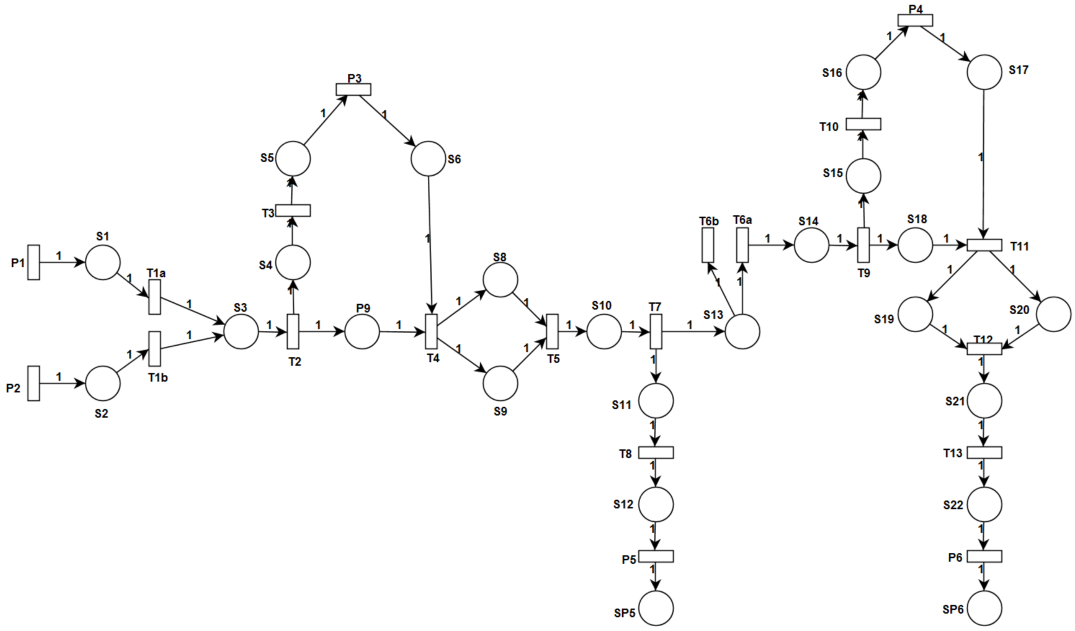

Figure 12: Corresponding Stochastic Petri net for the integration problem.

each other in the same fashion of the tasks interconnections shown in Figure 11. In the resulting Petri net, transitions were named according to their corresponding tasks, and places were named according to their respective slots.

All information needed to create the stochastic Petri net was available in the original Guaraná conceptual model, with the exception of the firing rates of the transitions. In this work, in order to evaluate the impact of the message processing across the network, all firing probabilities were set to 1 with the exception of the filter task, represented by the transitions $T6a$ and $T6b$, that were set to 1.0 and 0.1 respectively, simulating a filtering of 10% of messages.

5 ANALYSIS

In order to analyse the behaviour of the Petri net system, simulations were done using the PIPE software (Pere et al., 2007). Twenty independent simulations were carried out and the mean results are presented in this section. Table 1 presents the total number of tokens remaining in the Petri net places after $20,000$ transition firings. The tokens in the Petri net are analogous to the messages in the modelled integration solution, and the places are analogous to slots, that act as buffers, retaining unprocessed messages. We can see that place $S3$ retained a significant number of tokens. This place models the slot right after the merger task that receives input messages in parallel. This result is consistent to the notion that a critic point in a system is processing of incoming messages. In the simulated system, the sum of the probability ratios of the parallel input transitions for the merger task represented by transitions $T1a$ and $T1b$, is two times

Table 1: Number of tokens in places after $20,000$ firings.

Place	Average total tokens
S1	63.36184
S2	41.46832
S3	1005.9868
S4	22.0765
S5	64.1405
S6	22.48608
S7	108.70306
S8	28.79736
S9	28.79736
S10	27.72302
S11	21.73742
S12	28.34858
S13	6.96606
S14	9.29324
S15	9.8257
S16	6.96136
S17	8.99286
S18	25.77992
S19	6.77466
S20	6.77466
S21	7.33414
S22	18.0256

Table 2: Number of tokens in places after $20,000$ firings.

Transition	Firings	Expected firings	Firing rate	Global probability
T1a	1059.09	995.024	106.44%	0.0497
T1b	1114.86	995.024	112.04%	0.0497
T2	1167.97	995.024	117.38%	0.0497
T3	1145.89	995.024	115.16%	0.0497
T4	1059.26	995.024	106.46%	0.0497
T5	1030.47	995.024	103.56%	0.0497
T6a	911.06	995.024	91.56%	0.0497
T6b	84.71	99.500	85.14%	0.00497
T7	1002.74	995.024	100.78%	0.0497
T8	981.01	995.024	98.59%	0.0497
T9	901.77	995.024	90.63%	0.0497
T10	891.95	995.024	89.64%	0.0497
T11	875.99	995.024	88.04%	0.0497
T12	869.22	995.024	87.36%	0.0497
T13	861.88	995.024	86.62%	0.0497
P1	1178.22	995.024	118.41%	0.0497
P2	1100.56	995.024	110.61%	0.0497
P3	1081.75	995.024	108.72%	0.0497
P4	884.98	995.024	88.94%	0.0497
P5	952.66	995.024	95.74%	0.0497
P6	843.86	995.024	84.81%	0.0497
Total	20,000	20,000	–	–

bigger then the probability of the transition $T2$, right after place $S3$, as can be seen in Table 2.

Table 2 shows the transitions and their respective firings. Different from all other transitions, transition $T6b$ has a global probability of 0.00497, which corresponds to filtering 10% of messages. The table presents the total number of firings for each transition along with the expected firings number, according to the transition global probability. The *Firing rate* column is the rate of the *Total firings* in relation to *Expected firings* column. We can see that the transitions with the higher ratios are those located at the beginning of the chain of events, while the transitions with lower ratios are those at the end of the chain of events.

6 CONCLUSIONS

Guaraná technology offers a set of tools and methodologies to support the design and implementation of enterprise application integration solutions. Currently, the simulation approach adopted by software engineers to analyse an integration solution depends on the activities related with the construction, execution, and collection of data from such execution. It is recognised that there is a high cost, risk and development time spent associated with these activities. The simulation approach proposed in this paper aims at coping with this problem, by offering a way to analyse integration solution characteristics in an early phase of the development cycle.

In this paper, we proposed a simulation approach using Petri nets, in order to be able to analyse Guaraná solutions based only on the conceptual models. Petri nets are frequently used to model discrete event systems of any kind, which is the case of integration solutions. This work shows that Petri nets are an useful tool to simulate integration solutions in software engineering. It was demonstrated that the Guaraná conceptual model can be translated into a stochastic Petri net and simulated, enabling to identify its characteristics. Important aspects such as message staking in buffer slots, concurrency, tasks over-demand and number of processed messages can be observed using Petri net simulation tools, such as PIPE. The simulation of integration solutions reduces the costs by making possible to identify possible design flaws ahead of software design and prototype.

The correct identification of the problem leads to precise definition of the objectives, and the type of simulation model that must be developed. In our approach, we used simulation to understand the possible problems related to integration solutions. We consider that the simulation approach presented in this paper can improve the quality of enterprise application integration solutions developed using Guaraná.

We noticed that the area of Enterprise Application Integration in combination with the area of simulation is poorly investigated. Therefore, this work allowed the opening of promising research lines within the scope of application integration solutions using discrete-event simulation.

ACKNOWLEDGEMENT

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

REFERENCES

Alla, H. and Ghomri, L. (2012). Modeling and Simulation by Hybrid Petri Nets. In *Proceedings of the Winter Simulation Conference (WSC)*, pages 1–8.

Bause, F. and Kritzinger, P. S. (1998). Stochastic Petri Nets: An Introduction to the Theory. *SIGMETRICS Perform. Eval. Rev.*, 26(2):2–3.

Cavalieri, S. (2000). Improving Performance of a Flexible Manufacturing System by Petri Net Based Modelling and Simulation. In *Industrial Electronics Society, 2000. IECON 2000. 26th Annual Confjerence of the IEEE*, volume 2, pages 1298–1303 vol.2. IEEE.

David, R. and Alla, H. (1994). Petri Nets for Modeling of Dynamic Systems: a Survey. *Automatica*, 30(2):175–202.

Davidrajuh, R. (2008). Developing a New Petri Net Tool for Simulation of Discrete Event Systems. In *Modeling Simulation, 2008. AICMS 08. Second Asia International Conference on*, pages 861–866. IEEE.

Dossot, D. and D'Emic, J. (2009). *Mule in Action*. Manning.

Fisher, M., Partner, J., Bogoevici, M., and Fuld, I. (2010). *Spring Integration in Action*. Manning.

Frantz, R. Z. and Corchuelo, R. (2012). A Software Development Kit to Implement Integration Solutions. In *27th Symposium On Applied Computing*, pages 1647–1652.

Frantz, R. Z., Reina-Quintero, A. M., and Corchuelo, R. (2011). A Domain-Specific Language to Design Enterprise Application Integration Solutions. *International Journal of Cooperative Information Systems*, 20(2):143–176.

Gold, R. (2004). Petri Nets in Software Engineering. *Working Papers*.

Hohpe, G. and Woolf, B. (2003). *Enterprise Integration Patterns - Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley.

Ibsen, C. and Anstey, J. (2010). *Camel in Action*. Manning.

Kuo, C.-H. (2004). Development of Distributed Agent-oriented Petri Net Simulation and Control Environment for Discrete Event Dynamic Systems. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 5, pages 5001–5006 vol.5. IEEE.

Marsan, M. A. (1990). Stochastic petri nets: An elementary introduction. In Rozenberg, G., editor, *Advances in Petri Nets 1989*, pages 1–29. Springer-Verlag New York, Inc., New York, NY, USA.

Mens, T. and Gorp, P. V. (2006). A Taxonomy of Model Transformation. *Electron. Notes Theor. Comput. Sci.*, 152:125–142.

Molloy, M. K. (1981). *On the Integration of Delay and Throughput Measures in Distributed Processing Models*. PhD thesis, University of California, Los Angeles.

Narahari, Y., Suryanarayanan, K., and Reddy, N. (1989). Discrete Event Simulation of Distributed Systems Using Stochastic Petri Nets. In *TENCON '89. Fourth IEEE Region 10 International Conference*, pages 622–625. IEEE.

Pere, B., Lladó, C. M., Puijaner, R., and Knottenbelt, W. J. (2007). PIPE v2. 5: A Petri Net Tool for Performance Modelling. In *Proc. 23rd Latin American Conference on Informatics (CLEI 2007)*.

Rezai, M., Ito, M., and Lawrence, P. (1995). Modeling and simulation of hybrid control systems by Global Petri Nets. In *Circuits and Systems, 1995. ISCAS'95., 1995 IEEE International Symposium on*, volume 2, pages 908–911 vol.2. IEEE.

Russell, N., van der Aalst, W., and ter Hofstede, A. (2009). Designing a workflow system using coloured petri nets. In Jensen, K., Billington, J., and Koutny, M., editors, *Transactions on Petri Nets and Other Models of Concurrency III*, volume 5800 of *Lecture Notes in Computer Science*, pages 1–24. Springer Berlin Heidelberg.

Schmidt, D. C. (2006). Guest editor's introduction: Model-driven engineering. *IEEE Computer*, 39(2):25–31.

van der Aalst, W., Stahl, C., and Westergaard, M. (2013). Strategies for modeling complex processes using colored petri nets. In Jensen, K., van der Aalst, W., Balbo, G., Koutny, M., and Wolf, K., editors, *Transactions on Petri Nets and Other Models of Concurrency VII*, volume 7480 of *Lecture Notes in Computer Science*, pages 6–55. Springer Berlin Heidelberg.

Weiss, J. (2005). Aligning Relationships: Optimizing the Value of Strategic Outsourcing. Technical report, IBM.

Zhou, M. (1998). Modeling, Analysis, Simulation, Scheduling, and Control of Semiconductor Manufacturing Systems: A Petri Net Approach. *Semiconductor Manufacturing, IEEE Transactions on*, 11(3):333–357.