Formal Support of Process Chain Networks using Model-driven Engineering and Petri nets

Elena Gómez-Martínez
Univ. Autónoma de Madrid (Spain)
mariaelena.gomez@uam.es

Francisco Pérez-Blanco
Univ. Rey Juan Carlos (Spain)
francisco.perez@urjc.es

Juan de Lara
Univ. Autónoma de Madrid (Spain)
juan.deLara@uam.es

Juan Manuel Vara
Univ. Rey Juan Carlos (Spain)
juanmanuel.vara@urjc.es

Esperanza Marcos
Univ. Rey Juan Carlos (Spain)
esperanza.marcos@urjc.es

ABSTRACT

Business process modelling is an essential activity for competitive enterprises, as it enables documenting, analysing, improving and automating their core processes. Several notations have been proposed, including BPMN, service blueprints or Process Chain Networks (PCNs). These are all graphical, intuitive notations, which are useful for discussion and documentation, but tool support and a formal semantics are required for process analysis. However, tool support is lacking and formal semantics is not available for some commonly used notations among business people, like PCNs.

To alleviate this situation, we present a modelling tool for PCNs, and a formal semantics based on Generalized Stochastic Petri nets, which permits analysis. We have realized our approach using Model-driven Engineering, and show its realization within *INNoVaServ*, a modeling environment for the design of business models and service process operations.

CCS CONCEPTS

• **Applied computing → Business process modeling**;

KEYWORDS

Business process model, Process chain network, Model-driven Engineering, Petri nets, Model transformations, ATL

ACM Reference Format:
Elena Gómez-Martínez, Francisco Pérez-Blanco, Juan de Lara, Juan Manuel Vara, and Esperanza Marcos. 2019. Formal Support of Process Chain Networks using Model-driven Engineering and Petri nets. In *The 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19), April 8–12, 2019, Limassol, Cyprus.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3297280.3299752

1 INTRODUCTION

Service design increases value by adding services to products.This process gives rise to *servitization*, which develops an organization's innovation capabilities so that, rather than merely offering products, it provides customers with complete product-service systems [13]. One of the main challenges for companies wishing to embraze servitization is to identify the changes required in their organization. In fact, in their process to servitization, companies need to redesign their business model [2].

Business models are conceptual tools helping companies to identify, understand, design, analyze, and change their business. They describe the bases upon which firms create, provide and capture value [9]. There are several notations to represent these models. Some of them are oriented to provide a quick, strategic overview of the organization, like the Business Canvas [9] or the e3Value models [5]. Others, like Service Blueprints [3], or Process Chain Networks (PCNs) [11] show the details of a given process.

Similar to BPMN, PCN is used by people from operations research, and allows visualizing the service delivery process. Nonetheless, while BPMN focuses on depicting communications between the different assets of the organization (departments, roles and systems), PCN focuses on customer-provider interaction [7]. However, there are currently no tools supporting modeling with PCNs, or enabling the verification and validation of PCN models, ensuring good performance of the designed process.

To fill this gap, we propose a semantics for PCNs by means of Petri nets [8]. Petri nets are backed by a theory supporting not only simulation, but also analysis of structural properties, invariants, deadlock, state safety or reachability, among others. More specifically, we propose using Petri nets extended with time and probabilities, so called *generalized stochastic Petri nets* (GSPNs) [1]. These support checking whether the service meets the contract, and are able to confirm whether it operates as expected. An analysis of the GSPN would be able to detect functional problems like conflicts, deadlocks, and performance problems related to waiting times, resource utilization or probabilities of specific state conditions [1]. For its practical realization, and its integration within modelling tools, we use Model-driven Engineering (MDE) [4].

Overall, this paper makes the following contributions: (i) a metamodel formally defining the syntax of PCNs, (ii) a formal semantics for PCNs based on GSPNs, and (iii) a PCN modelling tool integrated within *INNoVaServ*, a modelling environment supporting the combination of several process modelling notations, using MDE techniques.

2 DEFINING THE SYNTAX OF PCNs

PCN is a service modelling technique proposed by Scott E. Sampson [10, 11], which allows visualizing the service delivery process of a given service with certain level of detail. The main abstractions

SAC '19, April 8–12, 2019, Limassol, Cyprus

ACM ISBN 978-1-4503-5933-7/19/04.
https://doi.org/10.1145/3297280.3299752

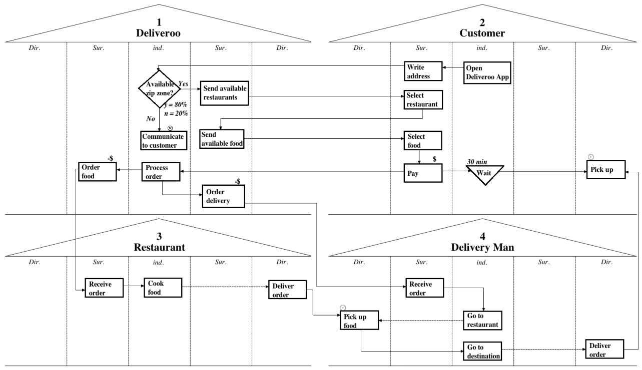


Figure 1: Deliveroo delivery process

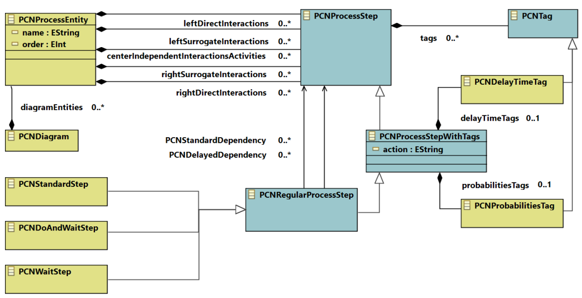


Figure 2: PCN meta-model (excerpt)

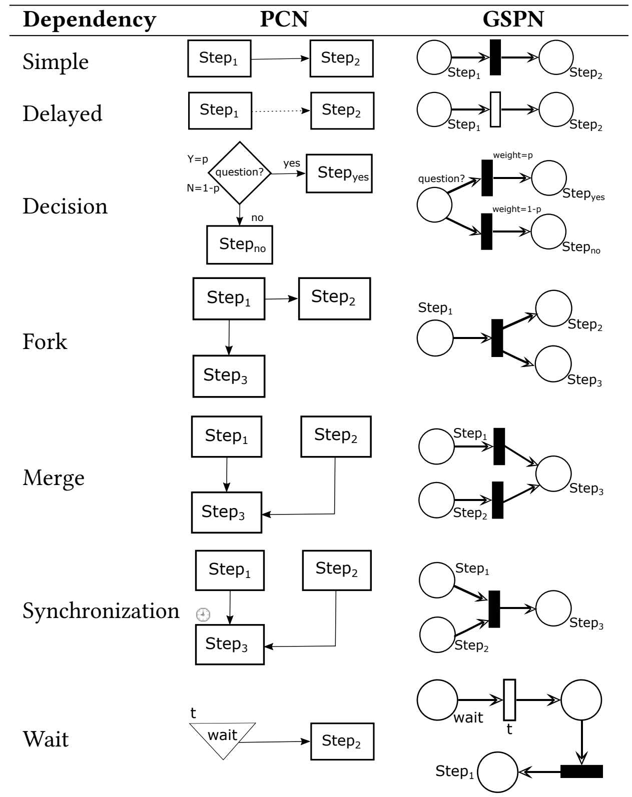
of the language are: (i) *Processes*, defined as sequences of steps performed by entities acting on resources; (ii) *Process chains*, which are sequences of process steps with a clear objective, typically to improve the value or the state of some entities; (iii) *Entities*, which participate in a process and can make decisions about the initiation or progress of a process chain; and, (iv) *Domains*, encapsulating the set of steps that are initiated and/or controlled by an entity. Fig. 1 shows a PCN diagram for the Deliveroo food delivery process, in which Deliveroo, Customer, Delivery Man and Restaurant are the entities involved. Three regions are distinguished in every entity depending on the degree of interaction between the entities: *direct interaction* region (Dir.), *surrogate interaction* region (Sur.) and *independent process* region (Ind.).

We use MDE to formally define the syntax of PCNs. In MDE the syntax of modelling languages is defined through a meta-model. This is typically a class diagram describing the primitives of the language, their properties and relations. Fig. 2 shows an excerpt of the meta-model we have built for PCN. A diagram (class PCNDiagram) is made of entities (PCNProcessEntity), which organize their process steps into 5 regions, represented as the 5 composition references stemming from PCNProcessEntity. Some process steps may contain tags (class PCNProcessStepWithTags), to encode delays and probabilities. Finally, several types of process steps are defined, but we only show three of them (PCNStandardStep, PCNWaitStep and PCNDoAndWaitStep).

3 A GSPN SEMANTICS FOR PCNs

Our approach for analyzing PCN diagrams relies on Petri nets. These are bipartite graphs with two types of nodes: places and

Table 1: Transformation of PCN dependencies into GSPNs



transitions, visualized as circles and rectangles respectively. Tokens, represented by black dots inside a place, represent a concrete system state. A transition is enabled when every input place has at least one token. The firing of enabled transitions represents a change in the system state. When a transition fires, a token is removed from each input place, and a token is added to each output place.

As Petri nets lack a temporal interpretation needed for performance evaluation, we use GSPNs. These distinguish three kind of transitions: immediate; with probabilities; and with exponentially distributed random firings. Immediate transitions fire at zero time. Transitions with probabilities are here used to represent the system routing rates, that is, decisions. Exponential transitions, drawn as white boxes, account for the time that takes an activity to complete. We chose GSPNs for two reasons: (i) GSPNs provide a formal notation which avoids ambiguities while representing the stochastic behaviour of systems, such as their capacity to represent routing rates, timing, parallel executions and forks and joins; (ii) several tools have been developed for analysis, such as GreatSPN,[1] TimeNet[2], WoPeD[3] and ProM[4], among others.

In our translation (summarized in Table 1), process steps are transformed into places, and dependencies to transitions, either

[1] http://www.di.unito.it/~amparore/mc4cslta/editor.html
[2] https://timenet.tu-ilmenau.de/
[3] https://woped.dhbw-karlsruhe.de/
[4] http://www.promtools.org/doku.php

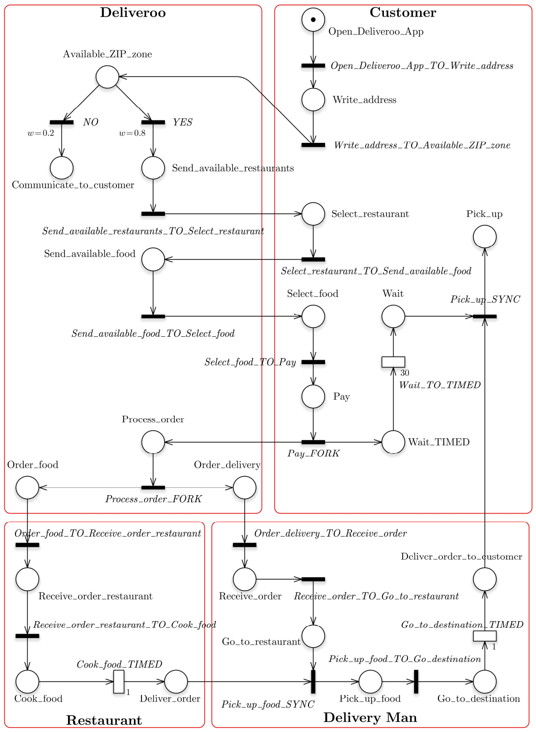


Figure 3: GSPN for the Deliveroo delivery service.

immediate (for simple dependencies), timed (for delayed dependencies), or with probabilities (for decisions). For instance, a decision step with a probabilistic tag is translated into a place with two output transitions, each one with weight p and $1 - p$, respectively, corresponding to the probability tag of each decision. Note that some elements and tags, such as regions, non/monetary costs and benefits, have no translation, since they are not relevant for GSPN analysis. Wait steps are translated into a timed transition between two places, representing the start and end of the wait step. Fig. 3 shows the result from transforming the PCN in Fig. 1.

4 TOOL SUPPORT

We have provided tool support for modelling of PCNs, and its analysis using GSPNs in the *INNoVaServ*[5] tool. The tool facilitates the management of business models expressed with different languages. It currently supports five modelling notations: Canvas, e3value, Service Blueprint, BPMN and the newly added PCN.

We extended the tool with the transformation described in Sec. 3, implemented using ATL [6]. This way, both the input PCN model and the output GSPN model are persisted as XMI files, representing instances of their corresponding meta-models. Nevertheless, most Petri net tools cannot read XMI, but use PNML as input format. PNML is an XML-based standard format for storing Petri nets. Hence, we export the GSPN models into PNML using a M2T transformation. Even though PNML is a standard, most net tools use variations of it. Thus, we created variants of the code generator for GreatSPN, TimeNet and WoPeD.

[5]http://www.kybele.es/innovaserv/

5 RELATED WORK

There are several studies comparing the PCN notation with other process modeling notations [7]. However, as it is a notation mainly used in the operations research community, no tool has been found supporting this notation. The only way of defining PCN diagrams to date was using generic diagramming apps.

There are many tools for analysis of business models using Petri nets, but no initiative targeting PCN models. Instead, most works focus on workflow languages or BPMN. Most notably, van der Aalst uses PNs to give semantics to workflow languages [12], proposing a subclass of PNs, called Workflow nets (WF-net). In a WF-net, transitions represent tasks that comprise a business process and places represent the conditions preceding and following the tasks. A WF-net has a distinguished source and sink place, and requires all nodes to lie on some path from this source place to the sink place. The computational complexity of determining soundness of WF-nets is high. WF-nets are supported by the WoPeD tool.

Altogether, we can conclude that both tool support for PCN, and its semantics using GSPNs are novel contributions of this work.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we have presented tool support for modelling with PCNs and a semantics based on GSPN. This semantics allows different structural, functional and performance GSPN analyses (whose details we had to omit by space constraints).

In the future, we plan to support presenting the GSPN analysis results in terms of the PCN, facilitating their comprehension. We will also extend the analysis techniques to other notations supported by *INNoVaServ*, like e3value.

ACKNOWLEDGMENTS

Work supported by Spanish MINECO (TIN2014-52129-R) and the region of Madrid (S2013/ICE-3006 and P2018/TCS-4314)

REFERENCES

[1] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. 1995. *Modelling with Generalized Stochastic Petri Nets.* John Wiley and Sons.
[2] T. S Baines, H.W. Lightfoot, O. Benedettini, and J.M. Kay. 2009. The servitization of manufacturing: A review of literature and reflection on future challenges. *Journal of manufacturing technology management* 20, 5 (2009), 547–567.
[3] M.J. Bitner, A.L. Ostrom, and F.N. Morgan. 2008. Service blueprinting: a practical technique for service innovation. *California manag. rev.* 50, 3 (2008), 66–94.
[4] M. Brambilla, J. Cabot, and M. Wimmer. 2017. *Model-Driven Software Engineering in Practice, Second Edition.* Morgan & Claypool Publishers.
[5] J. Gordijn. 2004. E-business value modelling using the e3-value ontology. In *Value creation from e-business models.* Elsevier, 98–127.
[6] F. Jouault, F. Allilaire, J. Bézivin, and I. Kurtev. 2008. ATL: A model transformation tool. *Sci. Comput. Program.* 72, 1-2 (2008), 31–39.
[7] Y. Kazemzadeh, Simon K. Milton, and L.W. Johnson. 2015. Process chain Network (PCN) and business process modeling notation (BPMN): a comparison of concepts. *J. of Management and Strategy* 6, 1 (2015), 88.
[8] T. Murata. 1989. Petri Nets: Properties, Analysis and Applications. *Proc. IEEE* 77, 4 (1989), 541–580.
[9] A. Osterwalder and Y. Pigneur. 2010. *Business model generation: a handbook for visionaries, game changers, and challengers.* John Wiley & Sons.
[10] S. Sampson. 2012. Visualizing service operations. *J. Serv. Research* 15, 2 (2012), 182–198.
[11] S. Sampson and M. Passey. 2011. Introduction to Process-Chain-Network Analysis.
[12] W.M.P. van der Aalst and K.M. van Hee. 2002. *Workflow Management: Models, Methods, and Systems.* MIT Press.
[13] I. Visnjic and B. Van Looy. 2012. Servitization: Disentangling the impact of service business model innovation on the performance of manufacturing firms. (2012).