

IECON 2017

Beijing, CHINA

The 43rd Annual Conference of the
IEEE Industrial Electronics Society

Date: October 29 - November 1, 2017

Venue: China National Convention Center, Beijing, China

Sponsored by: IEEE Industrial Electronics Society (IES)

Petri nets Approach for Designing the Migration Process Towards Industrial Cyber-Physical Production Systems

Ana Cachada*, Flávia Pires* José Barbosa*, Paulo Leitão*[‡],

* Polytechnic Institute of Bragança, Campus Sta Apolónia, 5300-253 Bragança, Portugal,

Email: {acachada, fpires, jbarbosa, pleitao}@ipb.pt

[‡] LIACC – Artificial Intelligence and Computer Science Laboratory, Rua Campo Alegre 1021, 4169-007 Porto, Portugal

Abstract—Presently, many industries are facing strong challenges related to the demand of customized and high-quality products. These pressures lead to internal company’s conflicts where current production systems have a rigid structure, forcing the company into a organization stall when a fast product change is required. Therefore, the need to smoothly migrate traditional systems into more feature-rich and cost-effective systems, namely Cyber-Physical Production Systems (CPPS), became a highly discussed topic. PERFoRM project focuses the conceptual transformation of existing production systems towards plug&produce ones to achieve flexible and reconfigurable manufacturing environments. In particular, the smooth migration process is considered crucial to effectively transpose existing production systems into truly CPPS. This paper describes the use of Petri nets to design the migration process under the PERFoRM perspective, taking advantage of its inherent capabilities to design, analyze, simulate and validate such complex processes.

Keyword: *smooth migration, cyber-physical systems, Petri nets.*

I. INTRODUCTION

The fourth industrial revolution, also known as Industry 4.0 [1], refers the modernization of traditional industries to improve their competitiveness, by promoting a new, distributed and intelligent business paradigm, supported by the Cyber-Physical Systems (CPS) [2] concept and enabled by the use of emergent technologies, such as Internet of Things (IoT), Big data, cloud computing and advanced data analytics. According to [3], it is expected that by 2025, 80% to 100% of manufacturing industry will be using IoT applications, implying that machines, sensors and other equipment mutually connects and communicates through the Internet, being able to perform data analytics to extract knowledge from the collected data to improve the system performance.

Aiming to implement this vision for the factories of the future, and since the objective should be the gradual digital upgrade of the existing facilities and not only the development of new facilities, a process is required to migrate from the traditional systems into the new Cyber-Physical Production Systems (CPPS). Broadly speaking the term “migration” refers to the switch-over of technology from older to newer systems or the change in the business structure, which will make the business more versatile, feature-rich and cost-effective [4]. According to [5], the decision to perform a system migration has different triggering sources, namely i) new business

opportunities become impossible to accomplish without a new system, ii) the system is no longer cost effective to support, iii) the system is inflexible and doesn’t respond to customer demands, iv) the system lacks visibility that could prevent equipment breakdown and disruption in the supply chain, and v) the system is impossible to be expanded. In the advent of Industry 4.0, the need to implement the new CPPS systems also constitutes an opportunity to trigger the migration process.

Some efforts have already been made in the migration of industrial automation systems, namely those performed during the IMC-AESOP project, where the change of current industrial process control systems into service-oriented automation systems is highlighted. In this work, the migration procedure comprehends four steps [6]: initiation, configuration, data processing and control execution. Another example is provided by the SOAMIG [7] project that also focuses the migration process for service-oriented systems. These approaches, besides to be focused on a single type of technology migration (i.e. service-orientation), are placed at high-level of abstraction, not being possible to establish a dynamic migration strategy as is required in the implementation of the Industry 4.0 vision. Additionally, some inspiration can be used from the migration of information systems, which defines 3 migration strategies, i.e. big bang, parallel and phased [8].

The EU H2020 PERFoRM (Production harmonizEd Reconfiguration of Flexible Robots and Machinery) project [9] is addressing the conceptual transformation of existing production systems towards CPPS to achieve flexible manufacturing environments based on rapid and seamless reconfiguration of machinery and robots as response to operational or business events. For this purpose, PERFoRM establishes a distributed and modular architecture [10], compliant with Industry 4.0 principles, where the production components and applications, covering the ISA-95 layers, interact by encapsulating their functionalities as services. In fact, an industrial distributed service-oriented middleware is used as a communication medium where the components can register their services, which later can be easily discovered by other components [9]. Furthermore, and besides the natural integration of new computational tools (e.g., for simulation and data analytics), PERFoRM also allows to integrate legacy systems through the use of technological adapters where legacy data formats are

converted into PERFoRM compliant data.

A particular attention in the project is devoted to the development of migration processes towards CPPS systems, and especially to those developed under the PERFoRM environment. This paper introduces a Petri nets approach to formal design the migration process towards the implementation of CPPS, taking advantage of its powerful mathematical background to model, analyze, simulate and validate complex processes. The use of Petri nets is valuable since the migration process can be easily modeled and validated during the design phase, which means that errors or misunderstandings can be easily detected and corrected, as well as alternatives for deployment can be simulated and optimized.

The rest of the paper is organized as follows: Section II describes the Petri nets formalism to design migration processes, and Section III presents the modeling of the PERFoRM migration process. Section IV describes the modeling of the one-shot, parallel and phased migration strategies, using stepwise refinement, and Section V presents the validation of the designed models, including the qualitative and quantitative analysis. The last section rounds up the paper with conclusions and outlines the next research steps.

II. PETRI NETS TO DESIGN MIGRATION PROCESSES

The migration process is rather complex and requires a formal methodology that synthesizes the process specifications and capture, understand and validate characteristics like concurrency, asynchronous operations, and deadlocks.

The representation of workflow of processes, as the migration process is, can be performed by using various techniques [11], [12]. ISAC (Information Systems Analysis/Change Analysis), DFD (Data Flow Diagram), SADT (Structured Analysis Design Technique), IDEF (Integrated DEFinition Methods) and BPML (Business Process Modeling Language) allow the representation of workflow processes but some do not define a formal model to express the process semantics, are more focused on execution processes than modeling and do not allow the formal analysis, simulation and validation of the processes in the design phase. The Petri nets formalism [13] is a suitable modeling, analysis and validation tool for the design of the migration process taking advantage of its well-founded mathematical theory to graphically and formally model and validate process specifications, exhibiting concurrency, parallelism, synchronization and resource sharing features.

For this purpose, a kind of Petri nets is used in this work, considering that places represent the state of the process and the transitions represent the logical aspects of the process behavior, which may be from two different types [14]:

- Immediate transition: fires in time zero and is used to model atomic activities, e.g., downloading a program.
- Timed transition: has associated the time that must elapse before the transition fires, and is used to represent time consuming activities, e.g., a machine repair.

In order to achieve a formal specification of the logic control structure, a top-down methodology is used, by refining step by step some timed transitions to include enough system

operation details for implementation purposes, i.e. replacing a timed transition by a more detailed and refined sub-Petri net so that a large Petri net can be obtained.

Besides the capability to synthesize the process specifications, another great advantage of using the Petri nets formalism is the capability to verify, simulate and validate the correctness of the system specifications during the design phase by using several formal analysis methods. These qualitative and quantitative analysis methods can be used to prove properties and to check the correctness of the model (e.g., safety properties, invariance properties and deadlocks) and to calculate performance measures (e.g., response times and occupation rates) and to evaluate alternative workflows.

Combining the modeling and analysis methods and using the mathematical foundation associated to the Petri nets formalism, a formal procedure can be applied to design migration processes from existing production systems towards CPPS.

III. MODELING THE PERFORM MIGRATION PROCESS

PERFoRM defines a new migration process that comprises five main stages stages [15], namely Preparation, Options of Investigation, Design, Transformation and Deployment, as represented in the Petri nets model illustrated in Figure 1.

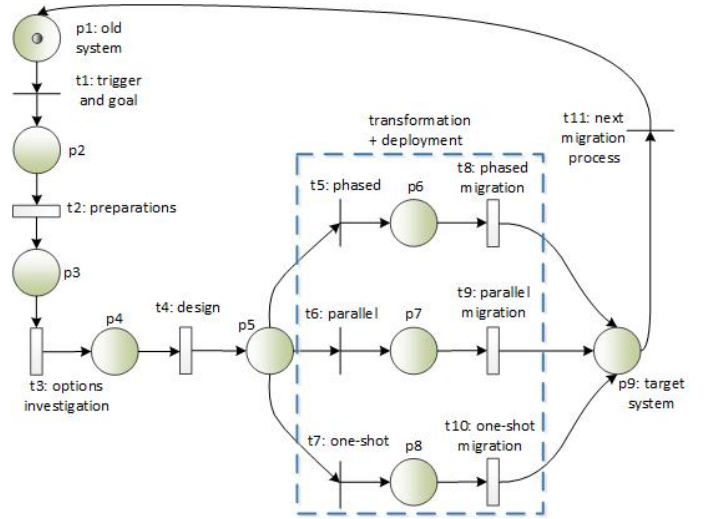


Figure 1. Petri Net for the general migration process.

The migration process starts with the *Preparation* phase, where the old system is analyzed and the general structure of the target system is defined, considering the process main goal. The next phase is known as *Options of Investigation*, where several technological design options are explored and critical interdependencies that can affect the implementation of the target system are identified. After this exploratory step, the optimal migration strategy is selected, considering the advantages and drawbacks for the factory. Afterwards, the *Design* phase is executed, where the planning of the selected migration strategy is performed (as well as defining the number of adaptors and new tools). After planning the migration project, the feasibility tests are carried out, followed

by the *Transformation* phase. In this phase, the established migration plan is implemented and the target system is verified before the *Deployment* phase where the installed system is commissioned and validated. Aligned with the literature, three different migration strategies are considered to implement the transformation and deployment phases, i.e. one-shot, parallel and phased strategies (represented by transitions $t8$, $t9$ or $t10$).

Once the migration plan is successfully completed, the commissioning of the target system has been achieved (place $p9$). This process can be cyclical, meaning that a new migration process can be initiated if necessary and the environment that was defined as "target system" is now the "legacy system".

IV. MODELING THE MIGRATION STRATEGIES

This section details each migration strategy defined in the migration process previously described (note that the preparation, options investigation and design stages are similar to the three migration strategies and out of scope in this work).

A. Modeling the One-Shot Strategy

The One-Shot strategy was inspired in the Big Bang strategy, where all the changes to be executed happen in a single period of time, that comprises the time to uninstall the old system and the time to install and validate the target system. The application of this strategy requires that the target system has to be completely defined and validated off-line. With this system ready, the old system is switched off and the target system is deployed as an integrated solution, being commissioned only if successfully validated. This strategy, broadly used e.g., in automotive industry, represents a high risk for the company since the old system is shut down which makes almost impossible to rollback.

Analyzing the Petri nets model for the migration process (see Figure 1), the One-Shot migration strategy is performed when the transition $t10$ is fired, which can be exploded into a sub-Petri nets model represented in Figure 2. This migration strategy comprises the execution of a sequence of steps that starts with the development of the necessary system components based on new technologies or paradigms. After this stage, the system is ready to be deployed in the factory and the original system can be switched off (transition $t10.t2$).

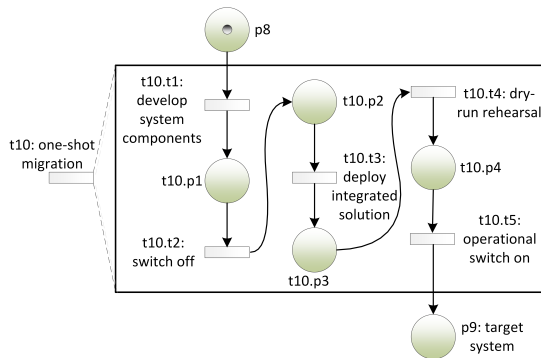


Figure 2. Petri nets model for the One-Shot Migration Strategy.

Once the old system is shut down, the integrated solution is deployed and a dry-run rehearsal is performed to certify that the target system is ready to run (transition $t10.t4$). When the successfully completed, the system is switched on ($t10.t5$) and the migration project is commissioned.

Some timed transitions of the Petri net model can be also exploded to introduce more control details. As example, Figure 3 illustrates the sub-Petri nets model for the transition $t10.t1$ that represents the development of system components, introducing particularities related to the PERFoRM environment.

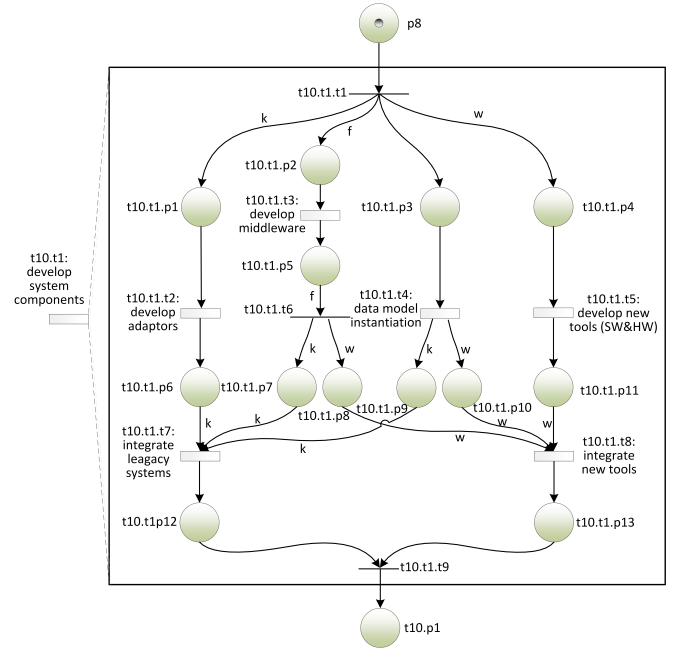


Figure 3. Petri nets model for the "develop system components" transition.

Initially, several actions are performed in parallel, namely the development of k adaptors (transition $t10.t1.t2$), installation of f middlewares (transition $t10.t1.t3$) and development of w new monitoring and analytics tools (transition $t10.t1.t5$) and instantiation of the data model, which are key components in the PERFoRM system. Note that the k , f and w values are defined during the design phase and are mapped into tokens that populate the places that represents the four referred parallel activities.

Once the entire set of adaptors are developed, the data model instantiated and the middlewares installed, the legacy systems can be integrated in the PERFoRM ecosystem (transition $t10.t1.t7$). On the other hand, the new tools are integrated in the PERFoRM ecosystem (transition $t10.t1.t8$) once all new tools are developed, the middleware is installed and the data model is instantiated. When all these software and hardware components are integrated within the PERFoRM ecosystem, the next tasks of the One-Shot strategy can be performed, as previously described.

As previously referred, the implementation of this strategy implies the shut down of the production site for a period of time. This down time is strongly dependent on the scope and

magnitude of the migration: if the migration only comprises software systems, the down time is smaller, but if the migration also considers hardware devices, the down time is higher since the complexity to uninstall components and program and install new components is higher.

B. Modeling the Parallel Strategy

The parallel strategy is based on the implementation of the target system, side by side, with the old system. This configuration must be kept until the target system has proven its viability. At this point, the target system is switched as Master, and the old system can be switched off or kept as Slave. Since both systems are running together, the occurrence of problems in the target system (running as slave) is mitigated by the use of the old system and provides a safer period of time to correct its behavior.

Figure 4 depicts the Petri nets model for the parallel strategy.

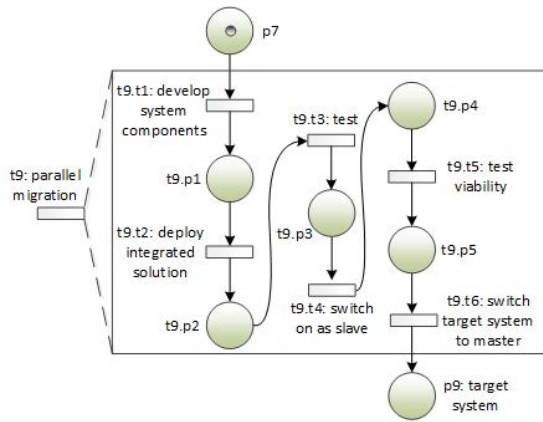


Figure 4. Petri nets model for the Parallel migration strategy.

As in the one-shot strategy, the first step is related to develop the system components. After all components have been developed, the integrated solution is deployed (transition $t9.t2$), and posteriorly its functionality tested (improving the system if any problem arises). When the new solution is successfully tested and is fully improved, the next step is related to switch on the target system as slave system and maintain the old system as master. After concluding successfully the viability tests, the target system is switched as master system, finalizing the migration process (transition $t9.t6$).

C. Modeling the Phased Strategy

The phased strategy is applied by deploying the new system through sequential phases, which requires a well planned implementation that carefully considers the interdependencies and the priorities of the involved processes. An important characteristic of this strategy is its recursive nature, meaning that one of the migration strategies can be selected for each phase. As an example, if a phased strategy is applied to migrate the entire factory, the migration of each production line can adopt the one-shot, parallel or phased strategy, and if this last one is selected, then again one of the migration strategies can be selected for each workstation.

Figure 5 illustrates the Petri nets model for the phased strategy. Once the strategy is selected during the design stage, one important note that needs to be taken in consideration is the number of phases and the associated strategy for each one. This information is associated to different variables used to regulate the flow of tokens along the Petri nets model: b represents the number of phases using the one-shot strategy and p represents those using the parallel strategy. The number of phased phases is calculated by $L-(p+b)$, where L is the total number of phases.

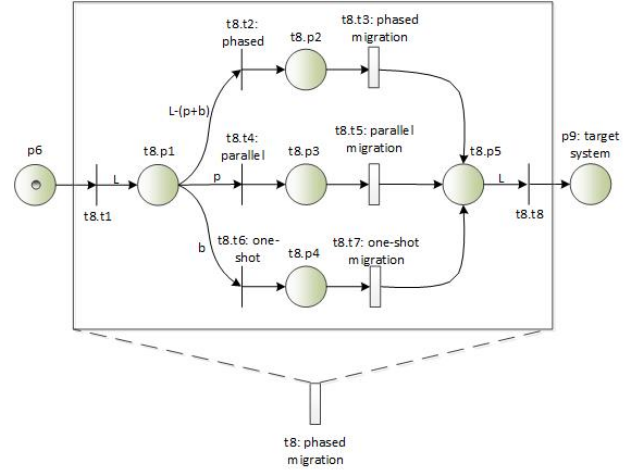


Figure 5. Petri nets model for the Phased migration Strategy.

After selecting this migration strategy, each one of the migration phases are properly executed, considering the defined strategy for each one. A migration phase using the phased strategy will trigger the recursive application of the same Petri nets model, and migration phases using the one-shot and parallel strategies will invoke, respectively, the Petri nets models illustrated in Figure 2 and Figure 4.

The migration process is concluded when the defined phases are all successfully implemented.

V. VALIDATION OF THE PETRI NETS MODELS

The designed Petri nets models for the implementation of the different migration strategies, for the transformation of traditional production systems into CPPS, were edited, analyzed and validated by using the Petri nets Development toolKit (PnDK) [16]. In this paper, the validation is illustrated by performing a qualitative and quantitative analysis to the general migration process (see Figure 1).

A. Qualitative Analysis

The qualitative analysis is related to the structural and behavioral validation of the designed Petri nets models, and particularly the verification of the structural and behavioral characteristics of the model, obtaining information related to the existence of deadlocks, bounded capacity of resources, and conflicts within the system [17]. The analysis of the behavioral properties for the Petri nets model representing the general migration process is illustrated in Figure 6.

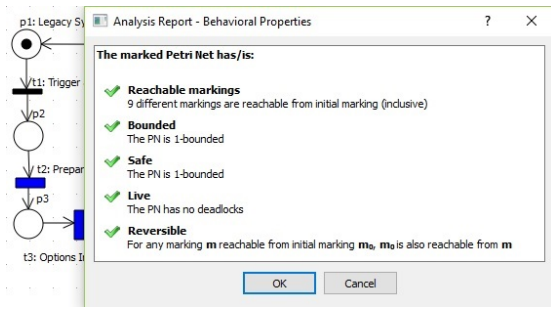


Figure 6. Behavioral analysis of the Petri nets model.

This analysis allows to extract the following conclusions:

- *Safe and 1-Bounded*: the maximum number of tokens that can be in a place is one, which means that only one migration strategy can be selected for the overall migration process.
- *Reversible*: the initial marking is reachable from all reachable markings, which means that after concluding a migration process, a new one can be started if necessary.
- *Absence of deadlocks*: for each reachable marking there is at least one transition that can be triggered to reach another marking, which means that the migration process doesn't stop in any particular step.

Additional characteristics can be extracted through the analysis of the P- and T- invariants, as illustrated in Figure 7.

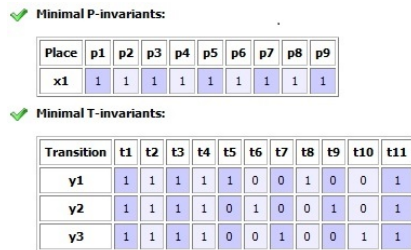


Figure 7. P- and T-invariants of the Petri nets model.

The analysis of the P-invariants allows the verification of mutual exclusion relationships among places, functions and resources involved in the structure and behavior of the model. For the Petri nets model of the general migration process there are only one P- invariant, $x1 = \{p1, p2, p3, p4, p5, p6, p7, p8, p9\}$ and, by its analysis, it is possible to confirm that only one place can be marked at any time, meaning the mutual exclusion among the several phases of the migration process.

The T-invariants represent the several sequences of operation, i.e. the work cycles, exhibited by the behavior model. From the analysis of the T-invariants, it is possible to confirm the existence of 3 invariants and its physical meaning can be translated as follows: 1) $y1 = \{t1, t2, t3, t4, t5, t8, t11\}$ represents the execution of phased strategy, 2) $y2 = \{t1, t2, t3, t4, t6, t9, t11\}$ represents the execution of the parallel strategy, and 3) $y3 = \{t1, t2, t3, t4, t7, t10, t11\}$ represents the execution of the one-shot strategy.

Since the model representing the general migration process comprises several timed transitions that are refined and exploded (see Figure 1), the complete analysis of this large model requires the analysis of all sub-Petri nets and the application of the theorems established by [18] and generalized by [19] about the preservation of boundedness and liveness properties in Petri nets obtained using the stepwise refinement. The Vallette theorem [18] states that all properties of a large Petri net can be deduced from the behavioral analysis of the initial Petri net and each one of the sub Petri nets, performed independently.

For this purpose, all timed transitions from the large Petri net, and also the timed transitions included in the exploded sub Petri nets, were analyzed using the same procedure as previously described. As an example, the validation of the sub-Petri nets model "develop system components" was performed, as illustrated in Figure 8, considering $k=6, f=2$ and $w=4$.

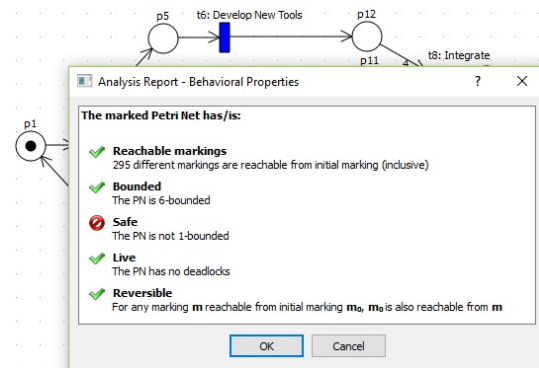


Figure 8. Behavioral analysis of the "Develop system components" model.

This analysis allows to conclude that this model is reversible, absent of deadlocks and 6-bounded (a maximum of 6 tokens may be hosted in one place, representing the actions to develop 6 adaptors for the identified legacy systems).

Since all sub-Petri nets were validated, concluding that they are bounded and absent of deadlocks, it is possible to conclude that, according to the Vallette theorem [18], the large Petri nets model for the general migration process is also bounded and absent of deadlocks.

B. Quantitative Analysis

The quantitative analysis is related to the simulation of the temporized Petri nets models by performing the token-game, which requires the association of the time parameter to the transitions. For this purpose, and considering the general Petri nets model representing the migration process, deterministic distribution times will be used as follows: transitions representing the logical conditions, i.e. $t1, t5, t6, t7$ and $t11$ have 1 time unit (t.u.), while the transitions related to preparations, options investigation and design phases, i.e. $t2, t3$ and $t4$ have 2 t.u. Additionally, the transitions representing the transformation and deployment phase, i.e. $t8, t9$ and $t10$ have 10 t.u.

The information of the time evolution in this Petri nets model can be summarized with a Gantt diagram. Figure 9

refers the temporal sequence of the migration process dynamics when the parallel migration strategy is selected. The analysis of the results allows to verify important characteristics, such as cyclic evolution and mutual exclusion activities.

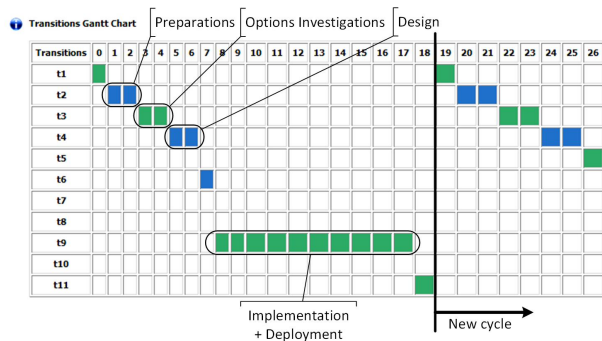


Figure 9. Gantt diagram for the performance analysis.

The previous qualitative and quantitative analysis allowed to validate the correctness of the Petri nets model representing the designed migration process towards CPPS to be used within the PERFoRM ecosystem, as well as to understand and synthesize the process specifications. Based on the structural and performance analysis, optimized strategies, re-tuning of some parameters and also re-design of the migration process can be implemented and tested.

VI. CONCLUSION

The smooth migration, from existing production systems towards the new and more effective CPPS, is a critical issue for the success of the so-called fourth industrial revolution. This topic is being addressed by the PERFoRM project that uses of the Petri nets formalism to design, verify, simulate and validate the migration process, taking advantage of the graphical and mathematical foundation.

This paper described the application of Petri nets to design the migration process under the scope of the PERFoRM ecosystem, which comprises five main phases: preparation, options investigation, design, transformation and deployment. The modeling of this process was refined by successively explode the timed transitions to include more details, which in this paper was illustrated by modeling the three migration strategies, namely One-Shot, Parallel and Phased, that can be used to implement the transformation and deployment phases.

The designed Petri nets models were analyzed, simulated and validated by conducting a qualitative and quantitative analysis. From this analysis it is possible to state that the proposed migration process is structurally and behaviorally validated, as well as properly simulated.

Future work will be devoted to the further specification of the migration process using Petri nets, particularly the preparation, options investigation and design phases, while the transformation and deployment phases will be further matured. Finally, the designed migration process will be applied to specific real industrial use cases in the context of migrating current production systems into CPPS.

ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 680435.



REFERENCES

- [1] H. Kagermann, W. Wahlster, and J. Helbig, "Securing the future of German manufacturing industry: Recommendations for implementing the strategic initiative INDUSTRIE 4.0," German National Academy of Science and Engineering (ACATECH), Tech. Rep., 2013.
- [2] E. A. Lee, "Cyber-physical systems-are computing foundations adequate," in *Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap*, vol. 2, 2006.
- [3] D. Verzijl, K. Dervojeda, S.-K.-F. Jorn, F. Nagtegaal, L. Probst, and L. Frideres, "Smart Factories Capacity Optimisation," European Union, Tech. Rep., 2014.
- [4] Transvive, *Migration Strategies & Methodologies [White Paper]*, (2011). [Online]. Available: <http://www.platformmodernization.org/transvive/Lists/ResearchPapers>
- [5] *Siemens Process Automation System Migration and Modernization Strategies*. ARC Advisory Group, 2007.
- [6] J. Delsing, F. Rosenqvist, O. Carlsson, A. W. Colombo, and T. Bange mann, "Migration of industrial process control systems into service oriented architecture," in *Proc. of the 38th Annual Conference on IEEE Industrial Electronics Society (IECON'12)*, 2012, pp. 5786–5792.
- [7] C. Zillmann, A. Winter, A. Herget, W. Teppe, M. Theurer, A. Fuhr, T. Horn, V. Riediger, U. Erdmenger, U. Kaiser, D. Uhlig, and Y. Zimmermann, "The SOAMIG Process Model in Industrial Applications," in *Proceedings of the 15th European Conference on Software Maintenance and Reengineering*, 2011, pp. 339–342.
- [8] P. Madkan, "Empirical Study of ERP Implementation Strategies-Filling Gaps between the Success and Failure of ERP Implementation Process," *Intl Journal of Information & Computation Technology*, vol. 4, no. 6, pp. 633–642, 2014.
- [9] PERFoRM-Production harmonizEd Reconfiguration of Flexible Robots and Machinery. [Online]. Available: <http://www.horizon2020-perform.eu>
- [10] P. Leitão, J. Barbosa, A. Pereira, J. Barata, and A. Colombo, "Specification of the PERFoRM architecture for seamless production system reconfiguration," in *Proceedings of the 42nd Annual Conference of the IEEE Industrial Electronics Society (IECON'16)*, 2016, pp. 5729–5734.
- [11] W. M. van der Aalst, "Making work flow: On the application of petri nets to business process management," in *International Conference on Application and Theory of Petri Nets*. Springer, 2002, pp. 1–22.
- [12] H. Mili, G. Tremblay, G. B. Jaoude, É. Lefebvre, L. Elabed, and G. E. Boussaidi, "Business process modeling languages," *ACM Computing Surveys*, vol. 43, no. 1, pp. 1–56, 2010.
- [13] T. Murata, "Petri Nets: Properties, Analysis and Applications," *IEEE*, vol. 77, no. 4, pp. 541–580, 1989.
- [14] A. Colombo, R. Carelli, and B. Kuchen, "A Temporized Petri Net Approach for Designing, Modelling and Analysis of Flexible Production Systems," *International Journal of Advanced Manufacturing Technology*, vol. 13, no. 3, pp. 214–226, 1997.
- [15] A. Calà, A. Lüder, A. Cachada, F. Pires, J. Barbosa, P. Leitão, and M. Gepp, "Migration from Traditional towards Cyber-Physical Production Systems," in *15th IEEE International Conference on Industrial Informatics (INDIN'17)*, 2017.
- [16] J. M. Mendes, A. Bepperling, J. Pinto, P. Leitão, F. Restivo, and A. W. Colombo, "Software Methodologies for the Engineering of Service-Oriented Industrial Automation : The Continuum Project," in *33rd Annual IEEE International Computer Software and Applications Conference Software*, 2009, pp. 452–459.
- [17] K. Feldmann, C. Schnur, and W. Colombo, "Modularised, distributed real-time control of flexible production cells, using petri nets," *Control Engineering Practice*, vol. 4, no. 8, pp. 1067–1078, 1996.
- [18] R. Valette, "Analysis of Petri Nets by Stepwise Refinements," *Journal of Computer and System Sciences*, vol. 18, pp. 35–46, 1979.
- [19] I. Suzuki and T. Murata, "A method for stepwise refinement and abstraction of Petri nets," *Journal of Computer and System Sciences*, vol. 27, no. 1, pp. 51–76, 1983.