# Use of Petri nets for Modelling Digital Twins at Inter-Company Level A Short Paper

STEFAN HAAG
Fachbereich Informatik
Hochschule Worms
Erenburgerstr. 19, 67549 Worms
GERMANY
haag@hs-worms.de http://www.hs-worms.de

*Abstract:* - Transformations inside a company can be modelled, simulated and, thus, analyzed by use of digital twins. These represent in a holistic approach machines, workpieces, data, and the processes linking those items together. Examples of transformations may include the conversion of a production line or the reorganization of a company's supply chain management. Because transformations have a cost associated with them, it is imperative to be able to somehow explore possible consequences of different alternatives. This can be achieved by use of models and simulation as provided by digital twins.

As is the case for both a single machine and a whole company, it is possible to model such a digital twin at an inter-company level with the goal to examine possible behavior of changes in integrated value chains.

By use of an appropriate modelling tool, different structures and used software products between the companies concerned can be alleviated.

*Key-Words:* - Horizontal Integration, IIoT, Cyber-physical System, Digital Twin, Simulation, Petri nets

## 1 Introduction

Digital twins are used to model and simulate real world problems at different scales, the main focus resting on smaller examples such as single machines or production lines due to the inherent complexity. Nonetheless, there are possibilities to model anything from a machine up to whole companies. [2]

However, it seems attainable to also model the integration between different companies along the value chain. Given the intricacy of large models, use of a small, easily understandable modelling language should prove advantageous. Also, as different companies (probably) use different modelling techniques, a common ground has to be found.

To this end, we propose the use of Petri nets to model inter-company integration by means of a web-based specification and simulation environment.

In the next part we explain the research method used, followed by an outline of reasons and advantages of using Petri nets. Then, we provide a small example and close with a conclusion and further research possibilities.

## 2 Research Method

According to Hevner et al. [3] there are seven guidelines for design science research. Their implementation is briefly explained as follows:

Design as an artifact: A web-based specification and simulation environment is used to examine the possibility of inter-company integration.

Problem relevance: Integration and optimization of value chains is a widely acknowledged task. Modelling and simulating given or planned connections between different companies exactly serves this purpose.

Design evaluation: Feedback from professionals as well as students using the presented tool and methods allows for improving the modelling capabilities of both tool and users.

Research contribution: An example is given on how to implement inter-company integration in a digital twin. This and further examples are used to teach both practitioners and students in the use of model-driven simulation.

Research strength: It is shown to be attainable for a new user – with minimal training time – to iteratively implement inter-company integration models of growing complexity.

Design as search process: The examples (and the tool) used for this short paper will be improved over time based on feedback by professionals and students. The improved materials will then be used in both instruction and solving real world applications.

Communication of research: Usage as well as research on usage are presented at conferences, lectures, and professional trainings.

## 2 Why Petri nets?

Petri nets have been researched and are semantically precisely defined for more than 50 years [7]. As they are computable by means of linear algebra, there exist many efficient algorithms to implement tools based on them.

A basic understanding of Petri nets can be achieved by use of only four different items (places, transitions, arcs, and tokens), while expansions allow for implementation of data and time concepts. [4] [5]

The most widespread languages for modelling business processes can easily be transferred into Petri nets representations and vice versa (cf. [6] for BPMN, cf. [8] for EPCs, cf. [10] for UML activity diagrams, cf. [1] for workflows). This means that whichever modelling language is used in the connected companies, it is possible to translate data and structures into a Petri net, solve a problem, and re-export the solution.

With regard to applied sciences, a modelling environment is needed to give practitioners the opportunity to actually use Petri nets. This has been a major drawback on the method as most of the tools are outdated or simply dysfunctional. In the meantime, however, there has been a new product in development, allowing the user to code Petri net models using a small, generic language. [9]

## 3 Connecting Companies

Two integration scenarios can be discussed, as shown in figure 1: While vertical integration means the connection of information systems from the top floor to the shop floor inside one and the same company, horizontal integration means the integration of business processes along the supply chain.


Fig.1 Integration of processes across company boundaries

This short paper establishes a generic process involving a company, as well as a supplier and a customer of said company as depicted in figure 2. Such a universal process could be anything along the supply chain. We assume the value chain already integrates all three companies.

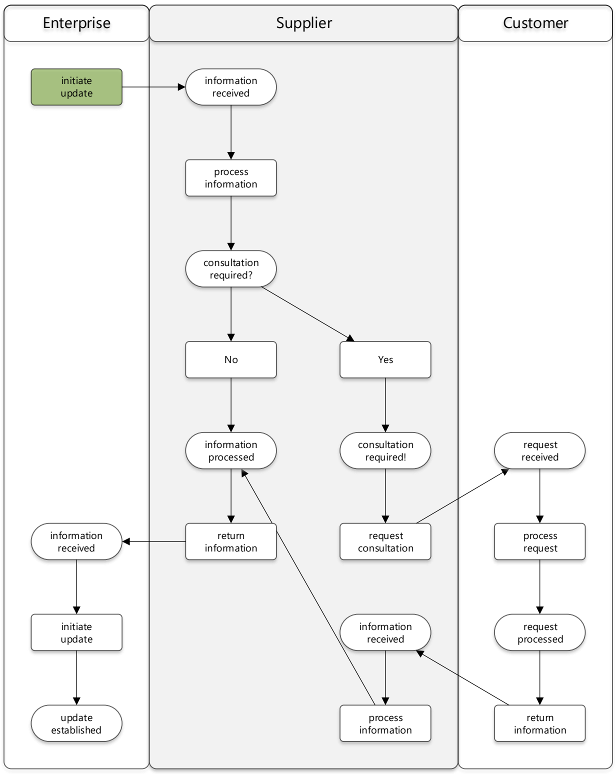


Fig 2 Generic inter-company process

The example process is supposed to be an order by the company called Enterprise to the supplier. The supplier examines whether further consultation is needed – if no, the corresponding information is passed directly to the enterprise. If, on the other hand, such consultation is needed, the integrated relationship chain allows for direct communication with the customer, whose answer would be processed and implemented by the supplier beforehand answering the enterprise's order.

The code for this process is shown in figure 3. There is some intrinsic information in this almost minimal model:

The code states three organizational units (which – on a smaller scale – also could be departments inside one and the same company), where places P and transitions T are allocated to the units by use of the orga-tag. The used tool automatically layouts the needed swim lanes, while the parameter design='m' changes the standardized forms of places (circles) and transitions (squares) in such a way that the labels can be printed inside.

The path of information follows the arcs written in shorthand. The notation allows for fast writing of connections spanning several nodes.

```
O Orga {
        U u1 (label='Enterprise');
        U u2 (label='Supplier');
        U u3 (label='Customer');
}

N Companies (design='m') {
        T tea (orga=u1,
                label='initiate update');
        P pea (orga=u1,
                label='information received');
        T teb (orga=u1,
                label='update information');
        P peb (orga=u1,
                label='update established');

        P psa (orga=u2,
                label='information received');
        T tsa (orga=u2,
                label='process information');
        P psb (orga=u2,
                label='consultation required?');
        T tsb (orga=u2,
                label='No');
        P psc (orga=u2,
                label='information processed');
        T tsc (orga=u2,
                label='return information');
        T tsd (orga=u2,
                label='Yes');
        P psd (orga=u2,
                label='consultation necessary');
        T tse (orga=u2,
                label='request consultation');
        P pse (orga=u2,
                label='information received');
        T tsf (orga=u2,
                label='process information');

        P pca (orga=u3,
                label='request received');
        T tca (orga=u3,
                label='process request');
        P pcb (orga=u3,
                label='request processed');
        T tcb (orga=u3,
                label='return information');

        A (tea, psa, tsa, psb, tsb, psc, tsc, pea,
                teb, peb);
        A (psb, tsd, psd, tse, pca, tca, pcb, tcb,
                pse, tsf, psc);
}
```

Fig 3 Full code example for the generic process

Information flow through this process is depicted using tokens in Petri nets. As higher nets are used, these tokens can represent data objects, a feature omitted here. Also omitted is the possibility to import and export data from and to databases, for example by use of .csv-files.

Higher net tokens may represent arbitrary data objects, be it a work order or a workpiece on the shop floor. An example for how such a struct-like record can be constructed is given in figure 4.

```
R RecordSet {
        i  :  int,
        f  :  float,
        c  :  char,
        t  :  time,
        d  :  date,
        dt :  datetime
};
```

Fig 4 Code example for a record using all available data types

In case the model gets too large, there is also the possibility to link to sub-processes which, after execution, return information to the calling process.

As the used tool allows for inclusion of external data as well as computation and selection orders on arcs and nodes, the models can be simulated. Thus, it is possible to examine both small and large alterations in any part of the depicted processes.

## 5 Conclusions

Using a generic process model we have shown the possibility to establish and simulate digital twins which span over multiple organizations. The system shown is being used in both professional and educational settings.

The easily computable and mathematically precise form of Petri nets allow for efficient realization of the models, while the relatively simple yet mighty semantics facilitate a fast understanding and implementation.

Overall, usage of appropriate tools as shown here make it clear that establishing digital twins for integrated companies is possible and can be achieved by using Petri nets.

*References:*

[1] van der Aalst W. M. P., Petri-net-based Workflow Management Software, *Proceedings of the NFS Workshop on Workflow and Process Automation in Information Systems*, 1996, pp. 114-118.

[2] Haag S., Simon C., Simulation of Horizontal and Vertical Integration in Digital Twins, *ECMS 2019 : 33rd International ECMS Conference on Modelling and Simulation*, 2019, pp. 284-289.

[3] Hevner A. R., March S. T., Park J., Ram S., Design Science in Information Systems Research, *MIS Quarterly*, 28, 1, 2004, pp. 75-105.

[4] Jensen K, Kristensen L. M., *Coloured Petri Nets*, Springer, Berlin, Heidelberg, Germany, 2009.
[5] Popova-Zeugmann L., *Time and Petri Nets*, Springer, Berlin, Heidelberg, Germany, 2013.
[6] Raedts I. G. J., Petkovic M., Usenko Y. S., van der Werf J. M. E. M., Groote J. F., Somers L. J. A. M., Transformation of BPMN models for behaviour analysis, *Proceedings of the 5th International Workshop in conjunction with the 9th International Conference on Enterprise Information Systems*, 2007, pp. 126-137.
[7] Reisig W., *Petrinetze – Modellierungstechnik, Analysemethoden, Fallstudien*, Springer-Vieweg, Wiesbaden, Germany, 2010.
[8] Rittgen P., Business process diagrams: an UML extension, *2003 Information Resources Management Association International Conference*, 2003, pp. 560-563.
[9] Simon C., Eine Petri-Netz-Programmiersprache und Anwendungen in der Produktion, *AKWI Tagungsband*, 2017, pp. 61-70.
[10] Staines T. S., Intuitive Mapping of UML 2 Activity Diagrams into Fundamental Modeling Concept Petri Net Diagrams and Colored Petri Nets, *15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems*, 2008, pp. 191-200.