



# Planification model-based process discovering

Sihem Mallek-Daclin<sup>1,\*</sup>, Layina Chaabaoui<sup>1</sup>, Nicolas Daclin<sup>1</sup>, Souad Rabah<sup>1</sup> and Gregory Zacharewicz<sup>1</sup>

<sup>1</sup>Laboratory of the Science of Risks, IMT Mines-Alès, 30100, Alès, France

\*Corresponding author. Email address: Sihem.Daclin-Mallek@mines-ales.fr

## Abstract

New enterprises modeling approaches are emerging to guide the development of new products to market. To this purpose, the Product Development Plan (PDP) is increasingly being developed. However, the PDP is a risky exercise, in which enterprises must invest time and precious resources without guarantee of success. In this way, it can be interesting to rely previous programs from other products to start developing a new PDP. In this work, the proposition is to guide PDP development with models build from experience of previous PDP using a standardized and recognized language: BPMN (Business Process Modelling Notation). Indeed, Process modeling is a significant challenge for organizations that aim to visualize in a simple and understandable way their business processes for descriptive, analysis or simulation purposes. However, previous programs to product development are mostly planned with Gantt charts. It is therefore necessary, for an easier reading and implementation of the processes, to allow a transition to Business process modelling from a planning model (ex. Gantt). As a result, it is proposed, in this paper, a process discovering methodology which aims to generate automatically a BPMN model from Gantt charts based on model mapping and transformation rules.

**Keywords:** Business process model, BPMN standard, planification model, Gantt charts, transformation models, process discovering.

## 1. Introduction

Product Development Plan (PDP) is a standard used by several companies to develop new products. PDP starts from market research to the product delivery with feedback regulation (Cooper *et al.*, 2001). PDP is defined with many stages where the output of each stage can be used as the input of another stage. These stages can be represented as activities and, as well, can form a business process (Van Kleef *et al.*, 2005).

In this way, understanding and modeling a process is becoming an important initial step in the construction of some actions and activities to follow for useful decision support. Product development is a complex process, and it is likely that many process models will be useful for making managerial decisions (Smith *et al.*, 1999). Business Process Management (BPM) is used for project development due to the

growing need to automate and facilitate the description of processes and workflows in an expressive and understandable language such as BPMN (Chinosi *et al.*, 2012) (BPMN, 2022) (Van Rosing *et al.*, 2014).

The objective here is the reuse of previous programs to integrate them into the new PDP. This paper presents a method of process discovering for exploiting previous programs to generate a first skeleton of the Product Process Model in BPMN. In our case, previous programs are formalized as planification models such as Gantt charts (Wilson, 2003). Therefore, it is proposed to generate a process model in BPMN from Gantt charts regarding some model mappings and transformation rules.

This work is part of the ONEWAY project which aims to ensure digital continuity throughout an aeronautical program. Especially our research work



has the objective to generate a PDP from old aeronautical programs.

The paper is structured as follows. After this brief introduction, Section 2 describes a state of the art about process discovering and especially on proposals of equivalence between Gantt planning models and BPMN process models. Section 3 introduces the proposed approach to generate business process model (BPMN) from planification model (Gantt charts). Then, the specified transformation rules are presented and detailed in section 4. To illustrate the proposed method, an application case is given in section 5. Then, section 6 presents discussion related to future works of this research work. Finally, section 7 presents a conclusion and some perspectives for this research work.

## 2. Process discovering: State of the art

Process discovering is one of the three categories of process mining. Process Mining is at the crossroads of Data Mining and Business Process Management: *it is an analytical approach that aims to build an exhaustive and objective view of processes based on factual data* (Van der Aalst, 2004). Process Mining relies on tools that use digital “traces” to reconstruct, visualize and analyze processes, thus providing full transparency and objectivity with respect to the real process. In several research works, Process Mining is based on techniques for discovering, evaluating, and extending process models. In this context, three major types of Process Mining techniques are identified (Van der Aalst, 2011):

1. **Discovery** (identification): identification of processes and building process models when no model exists a priori.
2. **Conformance** (conformity control): verification of the conformity of existing process models with analysis of deviations from a preliminary model.
3. **Enhancement** (extension): extension or improvement of an existing process model using information about the actual process recorded in some event log (Van der Aalst, 2016).

We focus on process discovery relying on data stemming from a Gantt chart rather than from a data log events as it is traditionally done in the process mining. Thus, our approach remains in lines with the process mining approach: extract a process from a data source. There are some research works in the literature which establish some equivalence between Gantt charts and process model in BPMN. In this case, the first research work (Lazaropoulos, 2021) proposes heuristics where the most relevant decision is to select a Model Driven Engineering-based approach to extract processes from legacy databases. In this way, a Meta-model is proposed to extend the BPMN Meta-model with time-dimension. Then, a simplified business educational equivalence table is proposed

between process modeling standards: BPMN diagram and Gantt chart, which are extensively used for business process modeling in the enterprises.

In the same perspective, (Arevalo *et al.*, 2016) also proposes a metamodel to integrate business processes time perspective in BPMN due to the weakness of BPMN to express temporal dimensions. In this way, some transformation rules between Gantt Charts and BPMN are proposed to add time aspect to BPMN. So, more information to the BPMN metamodel are added to introduce flexible duration and flexible start and finish for an activity. Especially with the proposition of BPMN constructs for time dependencies represented by Gantt connections (Finish to Start, Start to Start, Finish to Finish and Finish to Start).

These works propose equivalence rules between Gantt and BPMN on which our research work relies regarding new constraints and including resource planning models so that the generated process model of the PDP meet the end-user expectations for their business use is as complete as possible. This will be presented next section with our proposed approach to discover process model from previous programs such as Gantt charts and Resource Plan.

## 3. Process discovering proposed approach

To discover and generate a PDP (Product Development Process) business process model from previous programs, we propose a model-based approach for process discovering based on planification models. Our approach relies the on Model Driven Engineering principles (Mens *et al.*, 2005) such as model mapping and transformation model. Model mapping allow to identify and formalize equivalence between model objects source (Gantt chart) and target (BPMN) for instance in terms of behavior or semantic. Transformation rules allow to implement and to generate a Business Process Model that represents a first version of the PDP. This approach uses mainly to source models:

- **Gantt Charts** that illustrate the time schedule and the dependencies among business processes with the description of Activities (Kumar, 2005) (Geraldi & Lechter, 2012). These Activities are described with their current schedule status by exploiting start and finish dates.
- **Resource Planning** that describes allocation of resources to Activities (Glass, 1998). Resource plan aims to determine the resources necessary for the smooth running of a project/process, whether they are human, material, digital or financial.

In addition, it is possible to consider other input sources that can be complementary to the two previously mentioned to generate the most complete PDP possible and those always according to end-user expectations and following the MDE principles (Whittle *et al.*, 2014).

This approach is implemented following methodology in Figure 1. This methodology aims to develop new PDP model with BPMN from previous programs such as Gantt charts, resources plan and other data (ex. Data objects, scheduling...).

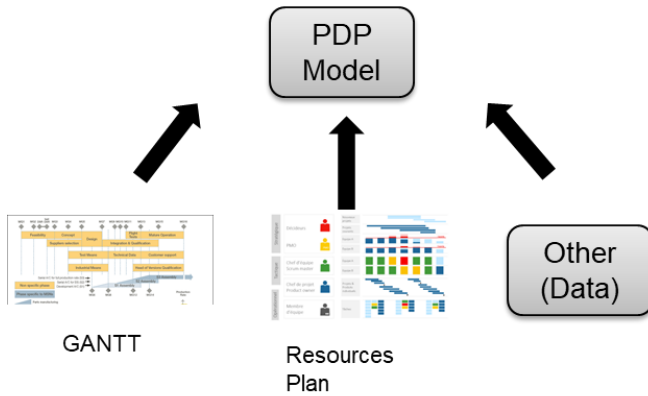


Figure 1. From previous programs to new PDP model

In a first stage, the proposed approach focuses on a the generation of PDP process model from Gantt Charts from defined mapping and transformation rules principle (Figure 2).

Next section describes and explains the established transformation rules for generating a process model in BPMN from Gantt Charts.

#### 4. Transformation rules

Transformation rules are made to provide all the necessary concepts that are used and deployed in Business process modeling especially in BPMN. In this way, it is mandatory to consider all the modeling entities which will be used in BPMN. Thus, each element in Gantt Charts is translated into one or more

elements in BPMN. Following this principle, each Gantt element can be extracted from the planification model to produce the corresponding BPMN element(s) representing Business Process Model at the end. Thus, these BPMN elements gather all the knowledge described in the Gantt model and represent the process model. The objective is to reveal the inner workflows and the exact input/output details of the connected Activities from Gantt Charts.

Some equivalence between Gantt Charts and BPMN can be found in the literature. For instance, (Lazaropoulos, 2021) proposes equivalence between some Gantt elements and BPMN elements such as phases and activity. For instance, a phase in Gantt is equivalent to a pool or a lane in BPMN according to the established hierarchy. However, in our case, regarding the constraints of the end-users, a phase is considered as a sub-process in BPMN and will be transformed into a Call Activity. This choice was established to allow the reuse of sub-processes if necessary. On the other hand, we kept the same equivalence for the activity which will be translated into a Task.

Furthermore, the first mappings established concern tasks that may not have any predecessors or successors. Thus, it is proposed to transform an activity with no predecessor in Gantt with a Start Event and a Sequence Flow preceding Task. In the same vein, an activity with no successor in Gantt is transformed in a Task with a sequence Flow and an End Event in BPMN.

Then, for each Gantt element, a mapping is defined and a transformation rules formalized end implemented to get the equivalent in BPMN. Table 1 explain these first proposed transformation rules.

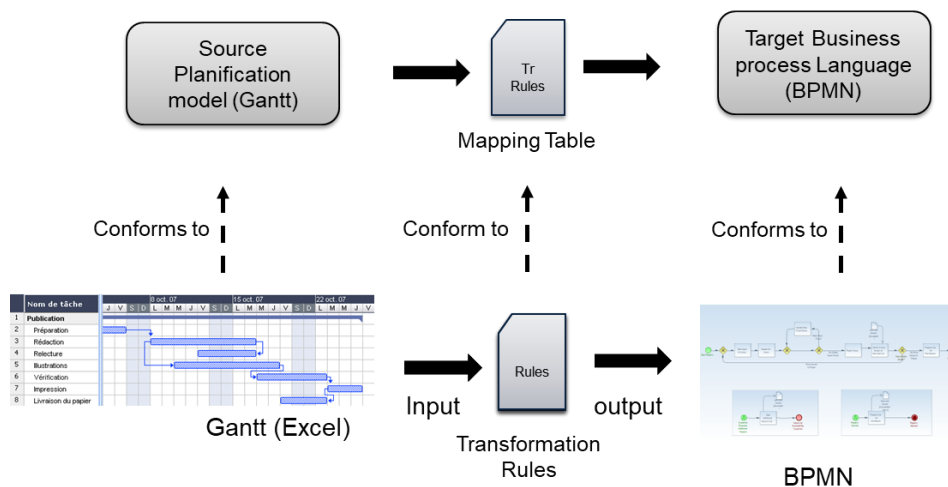


Figure 2. Transformation approach.

Table 1. Transformation rules of Basic Gantt Elements to BPMN Elements

Gantt Charts	Element Symbol	BPMN	Element Symbol(s)	Transformation description
Phase		Call Activity		A phase has several activities, it is therefore assimilated to a Call activity type sub process
Activity		Task		Activity is used to perform work in the process and is like Task in BPMN process model
No predecessor		Start Event		Activity that has no predecessor must be preceded by a Start Event
No successor		End Event		Activity that has no successor must be succeeded by an End Event

Gantt Charts is mainly used to represent dependencies between activities with predecessor and successor. In this way, there are four defined dependencies:

- Finish to Start (FS)** dependency. The activity cannot begin until the predecessor completes. However, it can start later. This is the most common type of dependency. This FS dependency is transformed as a Sequence Flow between Tasks.
- Start to Start (SS)** dependency. The activity cannot start before the predecessor activity begins. However, it can start later. This relationship is useful when considering an activity whose start date depends on the start date of another activity. In this case, we consider



that these activities can begin at the same time and work in parallel. This SS dependency is equivalent to a Divergent Parallel Gateway which precedes the two activities.

- 3. Finish to Finish (FF)**. The activity cannot complete before the predecessor task completes. However, it can end later. However, like before, we consider that the two activities can finish at the same time. This dependency is equivalent to a Convergent Parallel Gateway in BPMN.
- 4. Start to Finish (SF)** dependency considers that the activity cannot complete before the predecessor task begins. However, it can end later. We consider here a sequential notion from the successor to the predecessor. This is represented with a Sequence Flow in BPMN between the successor and the predecessor.

Table 2. Transformation rules of Gantt connections to BPMN elements

Gantt Charts	Gantt Dependencies	BPMN	Element Symbol	Transformation description
Predecessor → Successor	Finish to Start	Sequence Flow		A finish to start dependency between activities represents sequencing in BPMN
	Start to Start	Divergent Parallel Gateway		Activities can begin together and work in parallel
	Finish to Finish	Convergent Parallel Gateway		Activities can finish together at the same time
	Start to Finish	Sequence Flow		A start to finish is a particular dependency where Activity must be finished when Activity predecessor starts.
Multiple Predecessors	Multiple Successors	Convergent Parallel Gateway		Activity that has multiple predecessors must be represented with a Task preceded by Convergent Parallel Gateway
		Divergent Parallel Gateway		Activity that has multiple Successors must be represented with a Task succeeded by Divergent Parallel Gateway
Predecessor from another Phase		Catch Signal Event		Activity that has Predecessor from another Phase must be represented with a Task preceded by a Catch Signal Event that will receipt a signal from the preceded Task
Successor from another phase		Throw Signal Event		Activity that has Successor from another Phase must be represented with a Task Succeeded by a Throw Signal Event that will send a signal to succeeded Task

Table 3. Transformation rules of special Gantt specifications to BPMN elements

Gantt Charts	Special Gantt specifications	BPMN	Element Symbol	Transformation description
Special Activities	Livable	Throw Signal Event	Throw 	Activity that has important key such as Livable is represented with Adding Throw Signal Event at the Task
	Delays (+xx days, -yy days)	Timer Event		Activity can have another activity as successor or predecessor with positive or negative lag. The Time Delay is represented by Timer Event in BPMN.

Furthermore, multiple predecessors and multiple successors are considered. In this case, Divergent and Convergent Parallel Gateways are respectively used to translate these multiple dependencies in BPMN. Moreover, predecessor and successor can be from another phase. In this way, as presented previously, phases are transformed onto Call Activities in BPMN. Consequently, dependencies between activities from different phases must be translated into Intermediate Signal Event. When an activity has predecessor from another Phase, this one must be represented with a Task preceded by a Catch Signal Event that will receipt a signal from the preceded Task. Then, when an activity that has successor from another Phase, it must be represented with a Task Succeeded by a Throw Signal Event that will send a signal to the succeeded Task. Table 2 details all these transformation rules.

Lastly, there are some special specifications in Gantt Charts that must be considered and translated into BPMN. The first one is related to activities that are more important with the generation of deliverable. These activities can be represented in BPMN with the generation of Throw Signal Event to show the importance of these tasks. The second one is related to Delay which is represented as a positive or negative lag between activities in Gantt Charts. In this case, the lag (delay) is transformed into a Timer Event in BPMN. Table 3 explains these transformation rules.

To illustrate this approach and the transformation rules established in this research work, a use case is presented in the following section.

## 5. Application case study

Figure 3 presents a project which details the steps for preparing a publication. This example illustrates all

types of dependencies and is modeled into BPMN with our approach (Figure 4).

First, as shown in Figure 3, the Preparing activity has no predecessor. In this way, the transformation in BPMN is done with the addition of a Start Event which precedes the Preparing Task. After, the Redaction activity begins as soon as the Preparing activity ends. This is a Finish to Start Dependency that is transformed into Sequence Flow. Next, Reviewing begins after Redaction begins, and runs parallel to it. The writer and reviewer work as a team until the text is finalized. If the Redaction task completes later than expected, the Reviewing task is also delayed so that both tasks complete at the same time. This is a Finish-to-Finish dependency that can be transformed in BPMN with a Convergent Parallel Gateway. However, in BPMN a Task must have predecessor and in this way the Reviewing Task must have a Start Event as predecessor. The creation of illustrations (Illustration Activity) is linked to the Redaction Activity. A time lag has been introduced to give the writer a bit of a lead over the illustrators. Therefore, if the writing starts later than expected, the task of creating the illustrations is also postponed. However, this lag is not considered in our approach to facilitate the obtained BPMN model, and we assume that both tasks work at the same time. This is a Start-to-Start dependency which is translated into a Divergent Parallel Gateway. The final Verification Task is linked to the creation of the illustrations but can start before the end of this task. A two-day overlap was therefore defined. If artwork creation is delayed or takes longer than expected, final verification will also be delayed. This is again a Finish to Start dependency which is translated into BPMN with a Sequence Flow between the two Tasks.

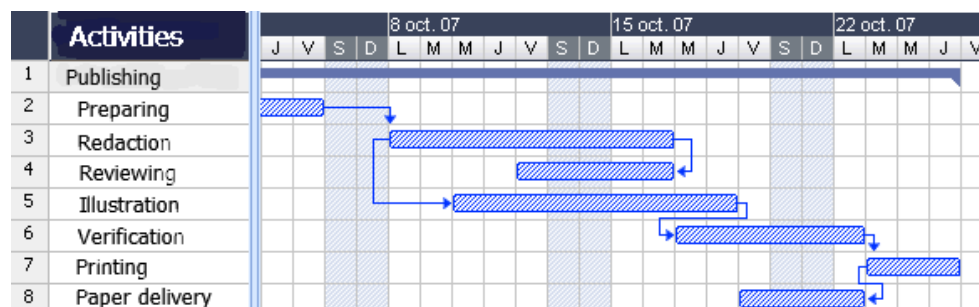


Figure 3. Example of Gantt Charts: Publishing activities

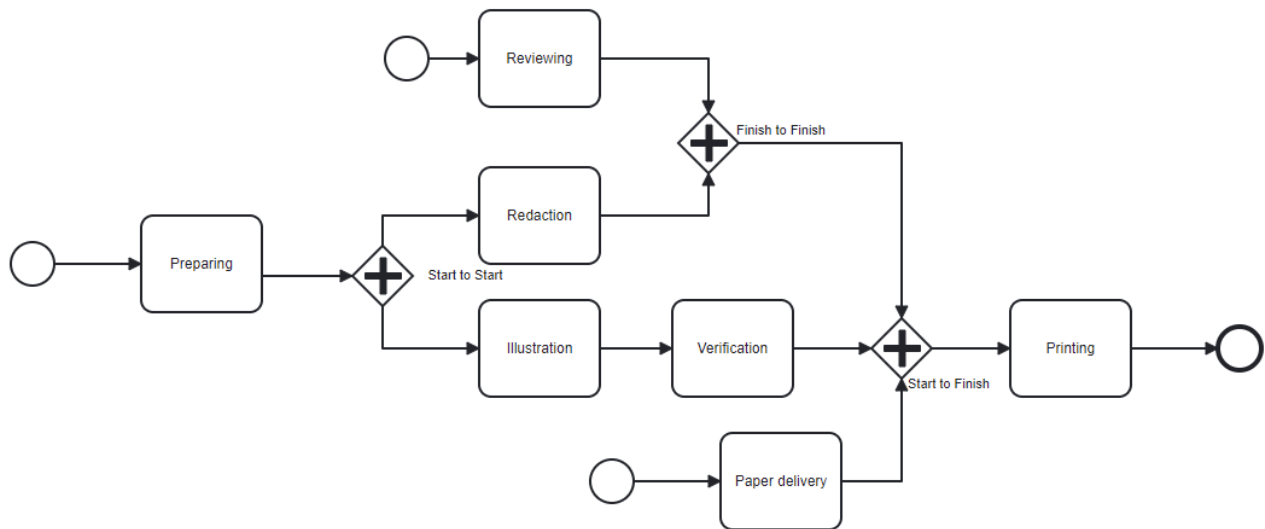


Figure 4. Business Process Model from Gantt Charts.

The Printing Activity cannot start until the booklet is finalized after Verification Activity. This is still a Finish to Start Dependency. Furthermore, it is imperative that the delivery of the paper to the printers be completed by the date scheduled for the start of printing. However, this activity is independent of the previous activities. In addition, printers have limited storage space, and do not want to receive the paper before they need it. The date of delivery of the paper is therefore governed by the date scheduled for the start of printing. If the latter is postponed for any reason, the date by which the paper must be available is also postponed. This is a Start to Finish dependency. In this case, it is considered in BPMN that the Printing Task will begin after Paper Delivery, Verification and Redaction/Reviewing. This is represented with a Convergent Parallel Gateway from these tasks. Finally, Printing is the end of the process and has no successor, thus, this one must be succeeded by an End Event in BPMN.

To finish, the Phase of Publishing is translated in BPMN with a Call Activity is preceded by a Start Event and Succeeded by an End Event. This Call Activity will be able to call a sub-process presented in Figure 4.

This example illustrates a simple scenario, where each task has only one predecessor. An activity can however have several predecessors or even predecessors of another phase, in which case it will be necessary to take into consideration the other transformation rules established in this paper.

## 6. Discussion and further works

The presented research work aims to develop a methodology that generate Business Process Model based on previous programs, especially the planification models that are represented with Gantt Charts. So, this paper initiated a big picture and a discussion to allow to identify new perspectives for the

industry 4.0. it is a part of the development of a Digital Twin for service (DT4S) of an aircraft PDP whose long-term objective is to enable real time monitoring of business processes to enhance the service aspect in Digital Twin in the frame of administrative or service workflows that require to be “digitally twined” (Rabah *et al.*, 2022).

In this context, new challenging activities should be developed to detect some deviations and readjust them in real world to prove the overall DT4S Concept. In this case, future work will focus, in the first step, on the development of Digital Twin for Business Process based on DT4S concept. In a second step, this work will focus on the detection of possible deviations of the established process compared to the real process that takes place in real life. These two future aspects are presented in the following sections.

### 6.1. Digital Twin for Business Processes

These works at the crossroads of Business Process, Digital Twin, and service industry. The targeted Digital Twin as define by (Rabah *et al.*, 2022) will allow real-time monitoring of business processes, PDP in our case. The potential of the development of Digital Twin for Business Process is to allow stakeholders to show and to flow in real time what happen in the real world, make real time decision and predict various situation (Eramo *et al.* 2021) To go further, the real impact factor of Digital Twin is to detect some possible deviations that can occur during runtime and proposes some solutions to readjust the Business Process regarding Key Performance Indicators (KPI) (Velimirovića *et al.*, 2011) such as Lead Time for example.

In this context, it is necessary to get a PDP model as complete as possible. Therefore, considering for resources is crucial step. It is more important to consider the aspect of resources that will be involved and allocated to the tasks of the future PDP. For this purpose, modelling of resources is an important step

towards a BPMN simulation through BPSIM formalism (BPSim, 2013) in order to develop a DT4S. As a perspective, resources can be defined by attributes that describes their roles, behaviors, and competencies. The work of (Yilmaz *et al.*, 2009) provides an interesting insight to do this.

The Digital Twin will have to retrieve data from the real world in real time to verify what is happening in real life has been foreseen in the generated PDP. Otherwise, if deviations are observed, the Digital Twin will propose solutions to the users for decision support. This decision will consider the most effective PDP possible with the most optimized KPI.

The BPMN is shown here as a pivot language between a non-formal language such as Gantt Charts - which is oriented towards graphical modeling - and a more formal language dedicated to simulation. This first idea makes it possible to link the world of modeling and that of simulation. (El Kassis *et al.*, 2022)'s work is part of this connection. In this way, with the connection between modeling and simulation, some solutions can be proposed to possible deviations which may appear during execution. of the process (PDP in our case) in real time.

## 6.2. Detection of deviations and readjustment

As presented previously, the main goal of the development of the Digital Twin for Business Process is the detection of deviations that can occurred during runtime. In fact, the process activities are essentially carried out by humans thus making them prone to many mistakes. The detection of deviations is a crucial step in the enactment and execution of business processes. As well as their handling and correction as the viability and smooth running of the entire process are at stake. Doing this automatically, continuously and in real time will provide users the tools to better understand and handle these unexpected situations. It is also fundamental to be able to determine the seriousness of the deviations. Indeed, all the deviations don't have the same consequences on the process. In this case, a classification can be proposed considering two categories: minor and major.

Furthermore, deviations can be characterized by different criteria such as the type, the cause, or the time of occurrence (Smatti *et al.*, 2015). There are different approaches to deal with deviations. Most of them act upon two points: how to detect them and how to handle them (Smatti *et al.*, 2015).

Today there are two main methods used to detect deviations (Bendraou, 2012). In the first one, the detection is done at predefined interval (milestone, ex. at the end of each activity). The drawback is that the deviation can be detected lately leading so that corrective actions cannot compensate the drift. The second one consists in representing a model that includes all the modifications than can be implemented during the process execution (Bendraou,

2012). This method avoids having a long period of time between the moment when the deviation happens and the moment when it is detected. However, it requires to have much more detailed process models which can be laborious.

For the detection of deviations, PSEEs (Process-Centered Software Engineering Environments) are developed to support software development (Gruhn, 2002). They define which software activities should be carried out as well as when, by whom and using which tools. They remind users of the activities that are to be done and those done automatically (Gruhn, 2002). More recently, another PSEE named PRAXIS was developed (Stirna, 2012). It keeps tracks of different types of information such as the actions that are carried out by the user, the activity that is currently executed (Bendraou, 2012). It uses PraxisRules, a rule-based language, to detect deviations when the process is executed. It is a Domain Specific Language whose main goal is to specify whether an action performed during the execution of an activity is acceptable or not. This PSEE is based on a set of rules automatically extracted from the process model. Then, the process modeler must associate a risk level, based on his knowledge, to each rule. When the process is enacted, Praxis monitors what happens and generates a deviation report that lists all the deviations that were detected as well as their risk level.

The next step is to generate Correction Plans to readjust the detected deviations. These corrections can be represented in the form of a sequence of actions with the proposition of correction plans according to the most critical deviations.

## 7. Conclusion

In the context of the development of new products to market, the Product Development Plan (PDP) is becoming increasingly important. In this research work, the new PDP is generated from some data such as planification model or resource plan. . In this way, the proposed research work addresses the question of the generation of a new PDP with BPMN notation from planification model such as Gantt charts.

First, it is necessary to translate planification model -Gantt Charts- to Business Process Model -BPMN. Therefore, some mapping and transformation rules are established giving equivalences between Gantt Charts and BPMN. However, this first generated PDP is not complete will be extended with resource aspect. It is proposed herein to embrace resource plan with BPSim which is an extension of BPMN considering resource and time aspect.

It should be remembered that the objective of these first research works is to allow to follow PDP set up in real life with the PDP generated to detect possible deviations. In this way, a Digital Twin is developed to this purpose. In fact, Digital Twin will allow to retrieve data from real world and compare and compare what happens in real life with the model. If a

deviation is detected, then some solutions can be proposed to remove it and improve the new PDP.

It is to note that the transformation principle is general and can be applied to any process. On the other hand, one of the limits of this research work is related to the input format. Indeed, the format used is specific to the ONEWAY project and if we want to be more general for a wider use, it would be interesting to move towards standardized Gantt tools that can be used by a greater number of users.

Future works are related also to the qualification of the deviations that can occurred during the monitoring. In fact, these deviations can be of different types. They can be of temporal type or even of topological type.

### Acknowledgements

This research was partially supported by DGAC for ONEWAY project. We thank our colleagues from AIRBUS and CIMPA who provided insight and expertise that greatly assisted the research.

### References

- Bendraou Reda. (2012). PraxisRules: An Approach for Detecting and Handling Process Deviations 2.1 Motivation.
- BPMN. (2022) Business Process Modeling Notation, V2.0. (<http://www.bpmn.org/>)
- BPSim. (2016) Business Process Simulation Specification. (<http://www.bpsim.org/>)
- Chinosi, M. and Trombetta, A. (2012) BPMN: An Introduction to the Standard. *Computer Standards & Interfaces*, 34, 124-134.
- Cooper, R. G., Edgett, S. J. and Kleinschmidt E. (2001) Portfolio management for new products. 2 editions, *Basic Books*.
- El Kassis, M., Zacharewicz, G., Troussset, F., Daclin, N. (2022) Business Process Web-based platform for multimodeling and simulation. Journ e francophone de la mod elisation et de la simulation, JFMS 2022, 28 mars – 2 avril, Carg ese, France.
- Eramo, R., Bordeleau, F., Combemale, B., van Den Brand, M., Wimmer, M., & Wortmann, A. (2021). Conceptualizing digital twins. *IEEE Software*, 39(2), 39-46.
- Geraldi, J., & Lechter, T. (2012). Gantt charts revisited: A critical analysis of its roots and implications to the management of projects today. *International Journal of Managing Projects in Business*, 5(4), 578-594.
- Glass, R.L. (1998) "Enterprise Resource Planning—Breakthrough and Vor Term Problem?," *Data Base* 29:2, 14-15.
- Gruhn, V. (2002). Process-centered software engineering environments a brief history and future challenges. *Annals of Software Engineering*, 14(1-4), 363-382. doi: 10.1023/A:1020522111961
- Kumar, P. P. (2005). Effective use of Gantt chart for managing large scale projects. *Cost engineering*, 47(7), 13-21.
- Lazaropoulos, A. G. (2021) Business Education and Training During the Enterprises' Business Education and Training During the Enterprises' Digital Transformation: Notation Alignment and Equivalence Rules Among the Enterprises' Business Process Models. *Regional Economic Development Research*.
- Mens, T., Van Gorp, P. (2006) A Taxonomy of Model Transformation. *Proc. Int'l Workshop on Graph and Model Transformation*
- Rabah, S., Zacharewicz, G., and Chapurlat, V. (2022). Digital Twin for services (DT4S): Conceptual strategy. *MIM, 10<sup>th</sup> IFAC Conference on Manufacturing Modelling, Management and Control*, Nantes, France
- Smatti, M., Ahmed-Nacer, M., & Nacer, M. A. (2014). Dealing with deviations on software process enactment: Comparison framework Dealing with Deviations on Software Process Enactment: Comparison Framework. Retrieved from <https://www.researchgate.net/publication/277476621>
- Smatti, M., Oussalah, M., & Nacer, M. A. (2015). A review of detecting and correcting deviations on software processes. ICISOFT-EA 2015 - 10th International Conference on Software Engineering and Applications, Proceedings; Part of 10th International Joint Conference on Software Technologies, ICISOFT 2015, 15-25. doi: 10.5220/0005517500150025
- Smith, R. P. and Morrow, J. A. (1999). Product development process modeling. *Design Studies*, Volume 3, Issue 3, Pages 237-261.
- Stirna, J. (n.d.). Marite Kirikova. In CEUR-WS.org (Vol. 855).
- Van der Aalst, W. (2011) Process Mining: Discovery, Conformance and Enhancement of Business Processes. *Springer*. ISBN: 978-3-642-19344-6.
- Van der Aalst, W. (2016) Process Mining Data Science in Action. *Berlin: Springer Berlin*. doi:10.1007/978-3-662-49851-4.
- Van Kleef, E., Van Trijp, H. C.M. and Luning, P. (2005) Consumer research in the early stages of new product development: a critical review of methods and techniques. *Food Quality and Preference*, Volume 16, Issue 3, Pages 181-201.
- Van der Aalst, W., and Weijters, A. (2004). Process mining: A research agenda. *Computers in Industry*, 53(3), 231-244. doi:10.1016/j.compind.



- Velimirovića, D., Velimirovićb, M. and Stankovića, R. (2011) ROLE AND IMPORTANCE OF KEY PERFORMANCE INDICATORS MEASUREMENT. *Serbian Journal of Management* 6 (1) 63 - 72
- Von Rosing, M., von Scheel, H. and Scheer, A. (2014) The Complete Business Process Handbook, Body of Knowledge from Process Modeling to BPM, Volume I, 1st Edition, Morgan Kaufmann.
- Whittle, J., Hutchinson, J., Rouncefield, M. (2014). The State of Practice in Model-Driven Engineering. *IEEE Software* 31(3), 79–85
- Wilson, J. M. (2003) Gantt charts: A centenary appreciation. *European Journal of Operational Research*, Volume 149, Issue 2, Pages 430–437.
- Yilmaz, L., & Ören, T. (2009). Agent-Directed Simulation. *Agent-Directed Simulation and Systems Engineering*, 111–143.