PRODUCTIVE PETRI NET: A TECHNIQUE TO OBTAIN STANDARD INSTRUCTIONS

Caio Cesar Fattori
Gabriel Souza Pontes
Gustavo Lopes Duarte Oliveira
Maurício Fontoura Blos
Fabricio Junqueira
Diolino José dos Santos Filho
Paulo Eigi Miyagi
Escola Politécnica da Universidade de São Paulo
paicaio,gabriel.pontes,gustavo.lopes.oliveira,blosmauf,fabri,diolinos,pemiyagi@usp.br

Abstract. *The constant evolution of demands of society for the industrial production affects directly in the companies requirements. It is increasingly necessary for companies to reduce the use of non-recyclable resources and polluting emissions, beyond ensuring high quality, low cost and customization of products and services. This set of requirements is modifying the production structure in companies to ensure the participation of small and medium companies in the production. Furthermore, economical advantages offered in different countries or regions cause the geographical spread of companies and force them to search solutions to meet the needs of the world market. In this context, it is increasing, currently, the search for collaborative production among independent companies, focused on their capacities and adopting concepts like virtual enterprises, using the internet to exchange information. To assure the higher participation of companies, it is necessary that the needs of products or services are described in processes using standard languages and which may understood with same result by different companies with the same capacities. A tool that allows to describe processes as standard and which proposes to unify its understanding by different entities is the productive Petri net. In this tool, there are instructions, which are information of processes represented in high level and which vary according to the nature and features of the process required. This work presents a technique to define standard instructions applied in lathe machines and it shows an example of the production of a piece described in productive Petri net.*

Keywords: *Petri net, STEP, Virtual enterprise*

1. INTRODUCTION

It is understood that industrial production increases to meet the growth of costumer's demand for products and services. This demand imposes also requirement of lower prices, higher quality, lower delivery time and higher customization. Companies are searching for new business strategies for adapting to these requirements, as well as they are changing the structure of productive chain, collaborating each other in a production net, focusing their efforts in their main capacities or skills and outsourcing other productive activities that are out of their scope. Thus, industrial production migrates from a mass production system to a lean production system and a agile manufacturing (Fattori *et al.*, 2011; Vinodh and Kuttalingam, 2011).

A solution to collaboration among companies focused on their main capacities or skills is the concept of virtual enterprise (VE). VE integrates the business layers of companies, using the advances of information technology and of automation (Tao *et al.*, 2012; Vinodh and Kuttalingam, 2011). A VE is an agreement among companies developed to meet a specific business opportunity and, at the end of this, the VE dissolves. For this, it uses a computational environment with applications which aid in the fulfillment of stages of project and negotiation of companies. However, so that different companies, interested in participate of a VE, can offer their capacities or skills (as services) for a business opportunity, it is necessary for all companies, with their own management and control structures, communicating each other with the same understanding of messages (Leitão, 2009; Wang *et al.*, 2012).

A solution to ensure this unique understanding of messages is the description standardized of productive processes which meet the business opportunities (Bhandarkar and Nagi, 2000).

In heterogeneous work environment, such as the industrial production environment, the communication which assures the same understanding for all is a great challenge. A productive process to meet a business opportunity can have productive activities of different areas (as car manufacturing which includes machining, forming and assembling processes) so that involves, necessarily, companies which don't have the same knowledge of the process, beyond the specific activities assigned to them. The relation among productive activities of different areas which are in a same productive process, still don't have a lot attention in current researches.

A way for modeling the relation among productive activities of different areas is to interpret them as part of the class of discrete event systems, which has tools for modeling and controlling of heterogeneous systems, such as Petri net (PN).

This work proposes using an interpretation of PN for modeling the relations among productive activities of different areas, called productive Petri net (PPN). In section 2, it is presented the background of existing researches of productive processes description. In section 3, it is presented the PPN formalization and the modeling of productive activities with different process properties. In section 4, it is presented an example of using PPN in a productive process of turning process. In section 5, it is presented the conclusion of this work.

2. BACKGROUND

Collaboration among companies focused on their main competencies is fundamental within VE, which integrates the companies business layer. In VE, each company is seen as specific service providers (which can be manufacturing, transport, chemical or other processes) and a specific demand is seen as a business opportunity by a company which search, using internet, other companies able to meet the productive activities needed which that company cannot or not interested to meet (Fattori *et al.*, 2011; Tao *et al.*, 2012). However, it is necessary that all companies, with their own management and control structures, may communicate each other in the same way, with the same understanding of the messages exchanged (Wang *et al.*, 2012).

In each industrial production area (STEP-nc in milling processes (Bhandarkar and Nagi, 2000) or the STEP for virtual assembly in assembly processes (Bin *et al.*, 2009)), there are initiatives which are creating a standard of the understanding of productive activities information performed using a communication by high level languages and widely spread among companies which provide equipment or automation systems, as done by the STEP (Bhandarkar and Nagi, 2000).

Some work address this standardization of description of productive process as a feature extraction problem. In Rao *et al.* (2012), it is shown a technique to convert this features in NC part programs for turning machines.

It is observed, however, the lack of work which treats of standardization of the description of different productive processes, in special case of process which doesn't have the property of be sequential. The VE concept allows to integrate independent companies (each on which its solution to conduct a productive process) for meeting a demand of product or service. Thus, companies which work together in a VE can use their productive chain to attend the production in independent way and the process properties can be different from sequential (Leitão, 2009), which is the only property in current standard, such as STEP.

3. PRODUCTIVE PETRI NET

PPN is a 6-tuple $PPN = (P, T, F, W, M_0, B)$, which P is a set of passive elements called *places*, T is a set of active elements called *transitions*, with P and T non-empty ($P \cup T \neq \varnothing$) and disjoint ($P \cap T = \varnothing$), $F \subseteq (P \times T) \cup (T \times P)$ is a set of relations among *places* and *transitions* called *arcs*, $W : F \to \mathbb{N}^+$ is a set of weight associated to each *arc*, $M : P \to \mathbb{N}$ is a set of *marking* associated to each *place* and M_0 represents the initial state of net, also called *initial marking* (Yoo *et al.*, 2010), and B is a set of productive activities associated to each *transition* called *instruction boxes* (Fattori *et al.*, 2011).

A *instruction box* $b_i \in B$ is a sub-net of PN with *places*, *transitions*, *arcs* and *marking* and, for a *transition* $t \in T$ if $b_i \times t \neq \varnothing$, the *instruction box* b_i is associated to the *transition* t, and if $b_i \times t = \varnothing$ the *instruction box* b_i is not associated to the *transition* t.

Each *instruction box* $b \in B$ has some *instructions*, described in high level language in textual form inside of it, as stages of the productive process which must be performed in each instant. The high level language allow the standardization of the stages of the productive process, moving away from the specific operation and equipment configuration which are able to meet the described stage. To controllers, an *instruction* is a sequence of atomic commands of actuator movement. Basically, a command can be represented as a model of PN with signs and gates (Takahashi *et al.*, 1999), as shown in the figure 1.

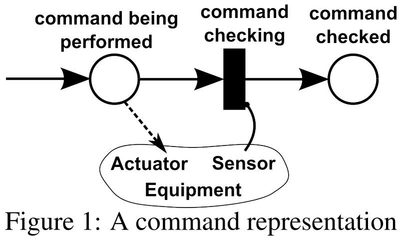


Figure 1: A command representation

The figure 1 represents a simple command, in which a signal is sent to an actuator of an equipment (for axis movement or other change) and a signal is received from a sensor of the same equipment to evaluate if the command was successful. The set of axis movements in equipments which produces a common characteristic of all similar equipments in the same resources, in which are applied the productive process, can be understood as an *instruction* which this equipment must perform. Thus, in PN, an *instruction* is represented as shown in figure 2.

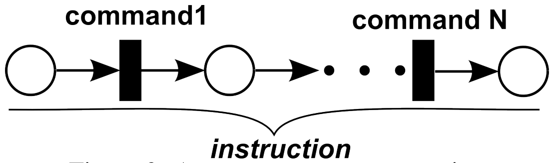


Figure 2: An *instruction* representation

Instructions $(i_1, i_2, \ldots, i_n)$ are models of PN represented in PPN as textual form by a high level language. For $i_k \subset b_l$, with $b_l \in B$, i_k is a productive activity which must be performed in a certain stage of the production in which the *transition* t_j is enabled for firing, with $t_j \in T$ and $b_l \times t_j \neq \varnothing$.

PPN is graphically represented by the elements in figure 3, in which the *place* is shown as a circumference, the *transition* is shown as a fulfilled rectangle, the *marking* (or token) is shown as a circle within the *place*, the *arc* is shown as an arrow with the arrowhead pointed to the second element of the relationship (in the case of figure, the *transition*), the *instruction box* is shown as a unfilled rectangle with a curve line linking its rectangle to the associated *transition*, and the *instructions* are shown as texts within the *instruction box*.

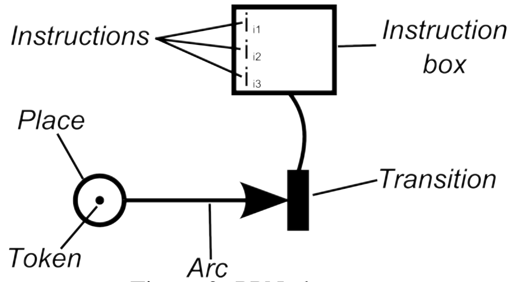


Figure 3: PPN elements

In figure 3, each *instruction* is a model of PN, as shown in figure 2, and which has a textual representation in the *instruction box*. The sub-net of *instruction boxes* can be detailed as a macro activity (Takahashi *et al.*, 1999), as shown in figure 4.

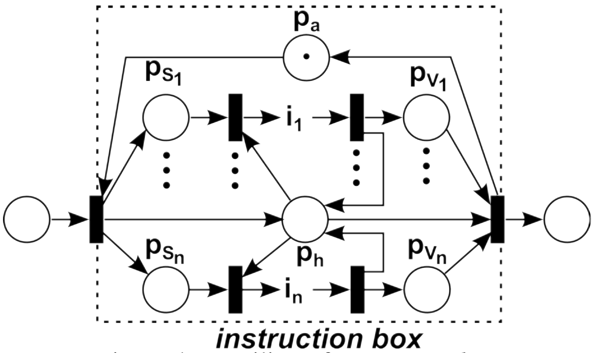


Figure 4: Detailing of *instruction box*

In the figure 4, *instructions* are models of PN and the *place* p_h is an enabler which chooses each stage of the productive process will be performed in each instant. After conclusion of a stage, the *place* p_h chooses other stage to be performed. To indicate which stages were concluded, the *places* p_Vs are used. When all *places* p_Vs and the *place* p_h have a token, all stages of the *instruction box* are concluded.

The PPN dynamic behavior is similar to the PN dynamic behavior, in which there is a preset $^\bullet t$ of a *transition* $t \in T$, such $^\bullet t = \{p \in P | (p,t) \in F\} \cup \{p_V \in b \cup p_h \in b | b \times t \neq \varnothing | b \in B\}$. When $\forall p \in P | p \in {}^\bullet t$, $M(p) \geq W(p,t)$, and, inside the *instruction box*, when $\forall p_V \in {}^\bullet t$, $M(p_V) = 1$, and $p_h \in {}^\bullet t$, $M(p_h) = 1$, we say that the *transition* t is enabled for firing. It is also defined a post set $t^\bullet$ of a *transition* $t \in T$, such $t^\bullet = \{p \in P | (t,p) \in F\}$. A *transition* t enabled for firing can fires, changing the state of the net from i to state $i+1$, and this implies:

- $\forall p \in P | p \in {}^\bullet t$, $M_{i+1}(p) = M_i(p) - W(p,t)$;
- $\forall p \in P | p \in t^\bullet$, $M_{i+1}(p) = M_i(p) + W(t,p)$;
- $\forall p_V \in {}^\bullet t$, $M_{i+1}(p_V) = 0$;

- $p_h \in {}^{\bullet}t$, $M_{i+1}(p_h) = 0$; and
- $p_a \in b|b \times t \neq \varnothing|b \in B$, $M_{i+1}(p_a) = 1$.

When, for a *transition* $t \in T$, with $\forall p \in P|p \in {}^{\bullet}t\ M(p) \geq W(p,t)$, and $p_a \in b|b \times t \neq \varnothing|b \in B$, $M(p_a) = 1$, we say that t is enabled for performing. A *transition* enabled for preforming can performs the *instructions* of the its associated *instruction box*. In this case, there is a changing only on the internal model of the *instruction box*, with this changes:

- $p_a \in b|b \times t \neq \varnothing|b \in B$, $M(p_a) = 0$;
- $\forall p_S \in b|b \times t \neq e\varnothing|b \in B$, $M(p_S) = 1$; and
- $p_h \in b|b \times t \neq \varnothing|b \in B$, $M(p_h) = 1$.

The internal PPN model of the *instruction box* is a $PN = (P_B, T_B, F_B, M_B)$, in which P_B is a set of *places*, T_B is a set of *transitions*, with P_B and T_B non-empty and disjoint, $F_B \subseteq (P_B \times T_B) \cup (T_B \times P_B)$ is a set of *arcs* and M_B a set of *marking* in the *places* of the *instruction box* model. It is defined for $t \in T_B$ two subsets ${}^{\bullet}t_B = \{p \in P_B|(p,t) \in F_B\}$ and $t_B^{\bullet} = \{p \in P_B|(t,p) \in F_B\}$. When $\forall p \in P_B|p \in {}^{\bullet}t_B$, $M_B(p) = 1$ and $\forall p \in P_B|p \in t_B^{\bullet}$, $M_B(p) = 0$, we say that the *transition* of the internal model of the *instruction box* is enabled for firing, and after fires, $\forall p \in P_B|p \in {}^{\bullet}t_B$, $M_B(p) = 0$ and $\forall p \in P_B|p \in t_B^{\bullet}$, $M_B(p) = 1$.

An important analysis in production which can be done in the PPN model is called sequential representation of the productive process. The sequential representation of the productive process (τ) is a sequence of *instruction* which can be performed if we attend the PPN dynamics among initial and end states which can be observed in this. The amount of sequential representations of PPN varies according to process properties found in the model, because the presence of some characteristics increases the combination of possible sequences. The sequential representation of PPN transforms a productive process model with various process properties in a productive process model with a single property, sequence, which can be interpreted by current tools, as STEP-nc (Bhandarkar and Nagi, 2000).

To cover a larger amount of productive process representations, PPN offers the possibility of modeling productive processes both those use only one equipment and those use more than one equipment with same or different functions. Besides, PPN allows modeling process properties of productive process stages which are not necessarily sequential. The sequence of stages of the productive process is the most common property found in production and is represented in PPN as shown in figure 5.

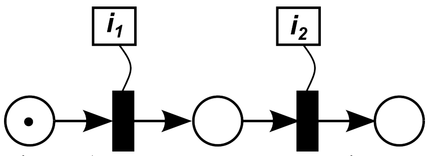


Figure 5: *Instructions* sequence in PPN

In this property, a stage has to be concluded as condition to start another stage. In the figure 5, the *instruction* i_1 precedes the execution of the *instruction* i_2, i.e., to start the execution of i_2, it is necessary to conclude the execution of i_1. The sequential representation of this productive process is $\tau_1 = i_1, i_2$.

Other common property of execution of stages of productive processes is the parallelism. Parallelism is widely used in productions with independent stages and different functions, as chemicals processes in the derivatives after the petroleum fractionation. The figure 6 represents the modeling of the property parallelism of stages of the productive process.

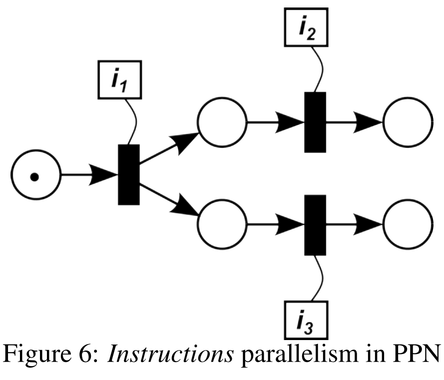


Figure 6: *Instructions* parallelism in PPN

In the figure 6, the *instructions* i_2 and i_3 are executed independently each other and can be done at the same time or not, because they not share resources. To start these stages, the stage i_1 must be completed. The sequential representations of this productive process are $\tau_1 = i_1, i_2, i_3$ or $\tau_2 = i_1, i_3, i_2$.

A property related to parallelism is the synchronization of stages. The synchronization is a common property of execution of productive processes with two or more stages which occur independently, but somehow become a single stage at some point, like the productive processes of assembling cars. The figure 7 represents a model of synchronization property in stages of productive process.

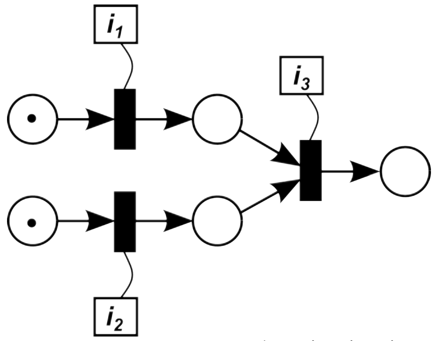


Figure 7: *Instructions* synchronization in PPN

In the figure 7, the *instructions* i_1 and i_2 are performed independently each other and only when both are concluded the *instruction* i_3 has its performing started, which can be an assembly of a product obtained by i_1 with the product obtained by i_2. The sequential representations of this productive process are $\tau_1 = i_1, i_2, i_3$ or $\tau_2 = i_2, i_1, i_3$.

In productive processes, some stages of production can be chosen instead of other stages to obtain the same result. This property of execution of stages of productive processes is called conflict, associated to a choice of each stage must be performed, as the production of a engine which can be done using machining or lamination techniques. The figure 8 represents a model of the conflict property between stages of productive process.

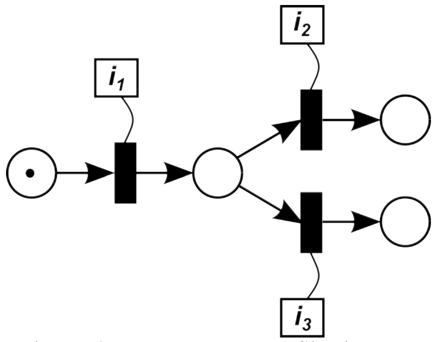


Figure 8: *Instructions* conflict in PPN

In the figure 8, the *instructions* i_2 and i_3 are in conflict and each stage must be performed depends of the choice after the conclusion of the *instruction* i_1, however only one of them must be executed. The sequential representations of this productive process are $\tau_1 = i_1, i_2$ or $\tau_2 = i_1, i_3$.

A not so common property, but found in processes which use the same resources and equipments for more than one stage of the productive process, is the arbitrary sequence of stages. The property arbitrary sequence of stages of productive process is that in which two or more stages of process must be performed, cannot be performed in parallel (because they share resources), but there is no correct order of performing to obtain the same result in the end. This property is the motive of the representation of *instruction box* with more than one *instruction* inside in the PPN model. The figure 9 shows the property of arbitrary sequence of stages of productive process.

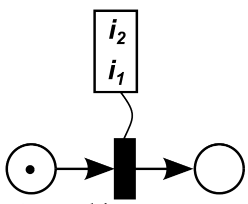
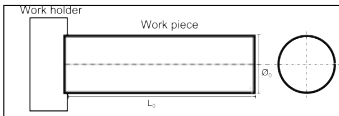
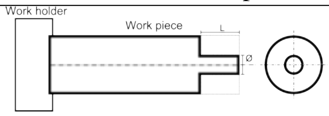
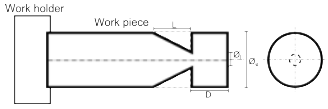
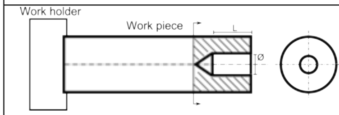
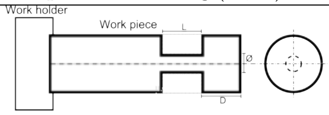
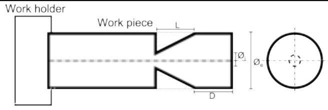


Figure 9: *Instructions* arbitrary sequence representation

In the figure 9, *instructions* i_1 and i_2 must be performed in sequence each other, but the first *instruction* to be performed can be any of them, like turning processes of grooving and boring in the same work piece. The sequential representations of this productive process are $\tau_1 = i_1, i_2$ or $\tau_2 = i_2, i_1$.

Table 1: Information used to develop a manual delivered to lab technician.

	 $plain\_turning(Ø, L)$	 $right\_hand\_taper\_turning(D, Ø_i, Ø_e, L)$
 $boring(Ø, L)$	 $groove\_turning(D, Ø, L)$	 $left\_hand\_taper\_turning(D, Ø_i, Ø_e, L)$

4. EXAMPLE

This example of PPN was done with lab technician of the Laboratório de Máquinas de Operação da Escola Politécnica da Universidade de São Paulo.

Initially, lab technician of turning machines were consulted above the most common characteristics in the daily operations, using these equipment. A list was developed with those characteristics and their descriptions in *instructions* form. Table 1 shows the information of this list and the information is used in a manual delivered to lab technician.

In the table 1, the cell in first line and first column at left has the representation of the cylindrical part used in machining process, called work piece. In this cell, L_0 means the initial length and $Ø_0$ means the initial diameter of work piece. In the next cells, the symbol D means the distance between the start of machining process and the counterpoint, Ø is the final diameter or the hole diameter, $Ø_e$ is the external diameter, $Ø_i$ is the internal diameter, and L is the length of the process or hole depth. The cell in second line and first column at left represents an operation of boring in the center of the work piece, modeled by function $boring(Ø, L)$. The cell in first line and second column represents an operation of plain turning, modeled as function $plain\_turning(Ø, L)$. The cell in second line and second column represents an operation of grooving, modeled as function $groove\_turning(D, Ø, L)$. The cell in first and third column represents an operation of taper turning by right hand, modeled as function $right\_hand\_taper\_turning(D, Ø_i, Ø_e, L)$. The cell in second line and third column represents an operation of taper turning by left hand, modeled as function $left\_hand\_taper\_turning(D, Ø_i, Ø_e, L)$.

In the figure 10, it is represented the design of final piece used in the example.

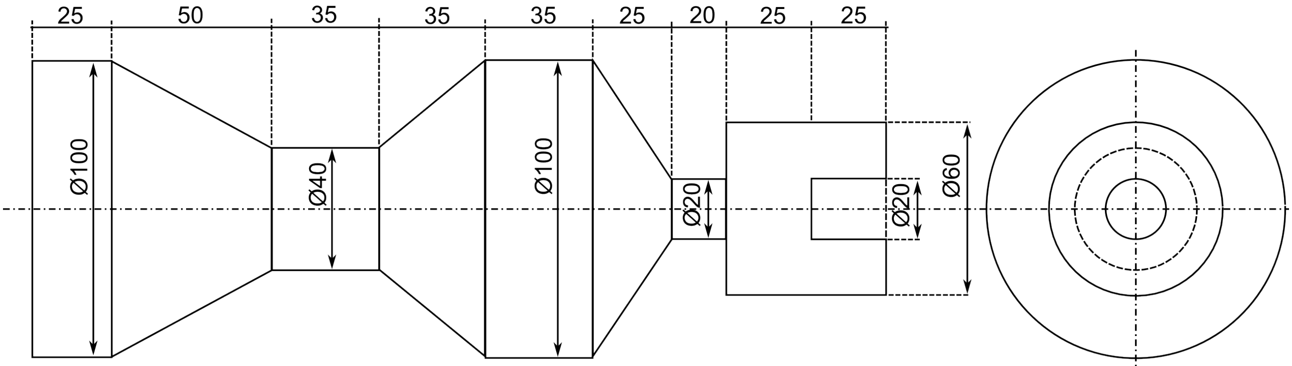


Figure 10: Final piece

The figure 11 presents the PPN, which models the productive process of the example. In this example, the initial piece has $L_0 = 275mm$ and $Ø_0 = 12mm$. Used *instructions* are compatible with the characteristic representation of STEP.

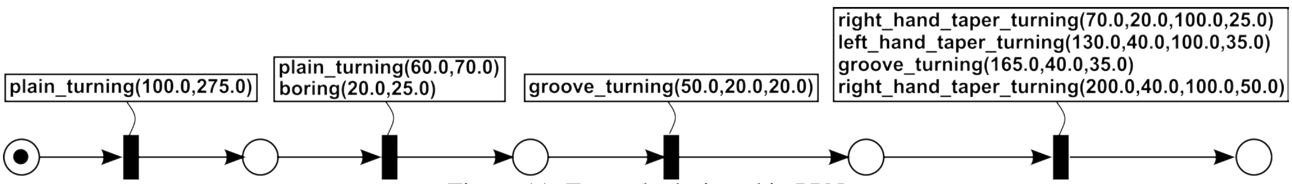


Figure 11: Example designed in PPN

Two possible sequential representations of the productive process modeled in the figure 11 are:

$$\begin{aligned}\tau_1 = \; & plain\_turning(100.0, 275.0), plain\_turning(60.0, 70.0), boring(20.0, 25.0),\\ & groove\_turning(50.0, 20.0, 20.0), right\_hand\_taper\_turning(70.0, 20.0, 100.0, 25.0),\\ & left\_hand\_taper\_turning(130.0, 40.0, 100.0, 35.0), groove\_turning(165.0, 40.0, 35.0),\\ & right\_hand\_taper\_turning(200.0, 40.0, 100.0, 50.0)\end{aligned}$$

$$\begin{aligned}\tau_2 = \; & plain\_turning(100.0, 275.0), boring(20.0, 25.0), plain\_turning(60.0, 70.0),\\ & groove\_turning(50.0, 20.0, 20.0), left\_hand\_taper\_turning(130.0, 40.0, 100.0, 35.0),\\ & groove\_turning(165.0, 40.0, 35.0), right\_hand\_taper\_turning(200.0, 40.0, 100.0, 50.0),\\ & right\_hand\_taper\_turning(70.0, 20.0, 100.0, 25.0)\end{aligned}$$

There are other 22 possible sequential representations of this productive process. In this example, those two sequential representations above are used for testing.

To automate a production, *instructions* must be turned into code which can be interpreted by equipment, such as G-code.

In the table 2, the codes represent the feature extraction, or *instruction*s, with a simplification of only one step, i.e., with a little material removal. If it is necessary to make more steps of material removal, the code must be implemented with a cycle of steps. In this example, only the tool-path was described. The code should consider the differences among G-codes in different manufacturers.

Information of operation (like rotation or speed) and material specification are not described in this example, and they should be inserted previously in real production.

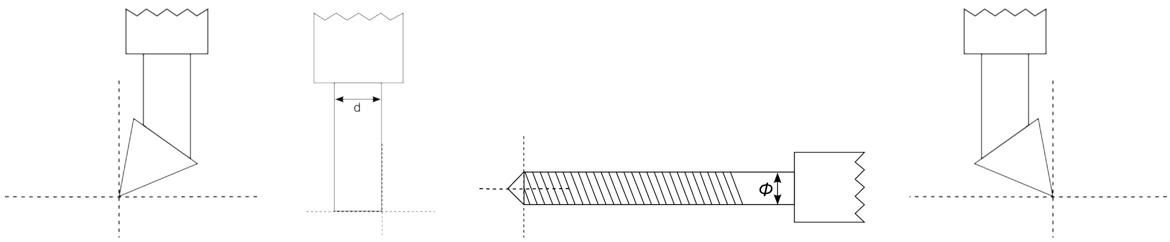
Table 2: G code generation

Instruction	G-code	Observations
$boring(Ø, L)$	N10 M06 T03 N20 G90 G00 X0 Z(s) N30 G01 Z($-L$) N40 G01 Z($-L+r$) N50 G01 Z(s)	Some manufacturers offer the function G83, which is the peck drilling cycle
$plain\_turning(Ø, L)$	N10 M06 T01 N20 G90 G00 XØ Zs N30 G01 Z($-L$) N40 G00 X($Ø + s$)	Some manufacturers offer the function G71, which is the fixed cycle, multiple repetitive cycle
$groove\_turning(D, Ø, L)$	N10 M06 T02 N20 G90 G00 X($Ø_0 + s$) Z($-D$) N30 G01 XØ N40 G01 Z($d - D - L$) N50 G00 X($Ø_0 + s$)	None
$right\_hand\_taper\_turning(D, Ø_i, Ø_e, L)$	N10 M06 T04. N20 G90 G00 X($Ø_e + s$) Z($-D - L$) N30 G01 X($Ø_e$) N40 G01 X($Ø_i$) Z($-D$) N50 G01 X($Ø_0$) N60 G01 X($Ø_0 + s$)	Some manufacturers offer the function G77, which is the tapping cycle
$left\_hand\_taper\_turning(D, Ø_i, Ø_e, L)$	N10 M06 T01 N20 G90 G00 X($Ø_e + s$) Z($-D$) N30 G01 X($Ø_e$) N40 G01 X($Ø_i$) Z($-D - L$) N50 G01 X($Ø_0$) N60 G01 X($Ø_0 + s$)	Some manufacturers offer the function G77, which is the tapping cycle

The cutting tools T01, T02, T03 and T04, in table 2, are presented in figure 12.

With the table 2 data, we can make the G-code sequence for machining of the example, in each sequential representation, τ_1 in the figure 13 and τ_2 in the figure 14.

The model, in figure 11, was delivered to lab technicians of turning machines with a manual, extracted from table 1. They were asked to make a sketch of the final piece. The sketch is presented in figure 15.



(a) T01 (b) T02 (c) T03 (d) T04

Figure 12: Cutting tools

N10 M06 T01	N130 G90 G00 X(62) Z(−50)	N250 G01 X40 Z(−165)
N20 G00 X100 Z02	N140 G01 X20	N260 G01 X102
N30 G01 Z(−275)	N150 G01 Z(−68)	N270 M06 T02
N40 G00 Z02	N160 G00 X(62)	N280 G00 X102 Z(−165)
N50 G01 X60	N170 M06 T04	N290 G01 X40
N60 G01 Z(−70)	N180 G00 X(102) Z(−95)	N300 G01 Z(−198)
N70 G00 X62 Z02	N190 G01 X100	N310 G00 X102
N80 M06 T03	N200 G01 X20 Z(−70)	N320 M06 T04
N90 G00 X0 Z02	N210 G00 X102	N330 G00 X102 Z(−250)
N100 G01 Z(−25)	N220 M06 T01	N340 G01 X100
N110 G01 Z02	N230 G00 X102 Z(−130)	N350 G01 X40 Z(−200)
N120 M06 T02	N240 G01 X100	N360 G01 X102

Figure 13: G code for τ_1

N10 M06 T01	N130 M06 T02	N250 G01 X40
N20 G90 G00 X100 Z02	N140 G00 X62 Z(−50)	N260 G01 Z(−198)
N30 G01 Z(−275)	N150 G01 X20	N270 G00 X102
N40 G00 X(102)	N160 G01 Z(−68)	N280 M06 T04
N50 M06 T03	N170 G00 X62	N290 G00 X102 Z(−150)
N60 G00 X0 Z02	N180 M06 T01	N300 G01 X100
N70 G01 Z(−25)	N190 G00 X102 Z(−130)	N310 G01 X40 Z(−200)
N80 G01 Z02	N200 G01 X100	N320 G01 X102
N90 M06 T01	N210 G01 X40 Z(−165)	N330 G00 Z(−95)
N100 G00 X60 Z02	N220 G01 X102	N340 G01 X100
N110 G01 Z(−70)	N230 M06 T02	N350 G01 X40 Z(−70)
N120 G00 X62	N240 G00 X102 Z(−165)	N360 G01 X102

Figure 14: G code for τ_2

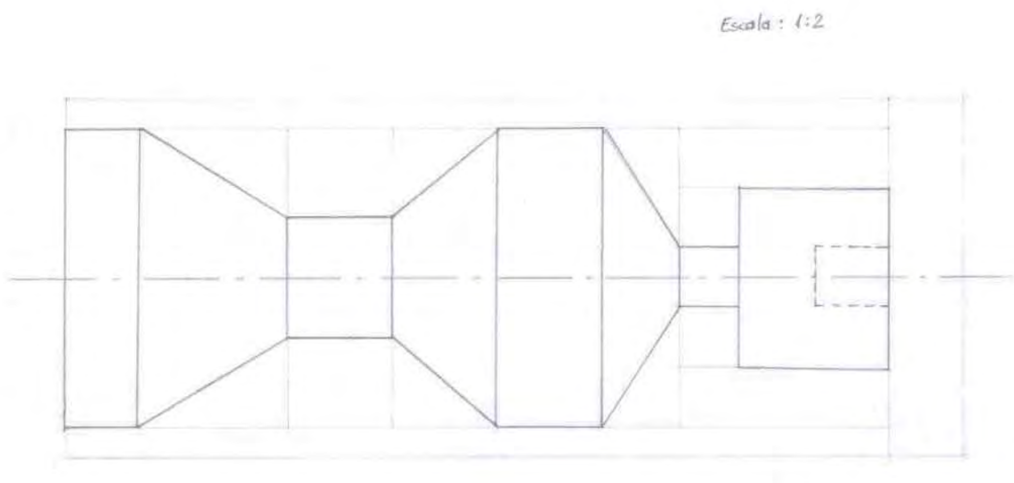


Figure 15: Sketch of final piece

5. CONCLUSIONS

Beyond the presented example, other study cases are been made and they confirm that PPN can describe productive processes with sequential, parallelism, synchronization, conflict and arbitrary sequence process properties. With this description based on PPN, it expands the capacity of collaboration among companies, because that shows the independent processes which can be performed at same time and, eventually, the multiple companies able to meet the same stages of the productive process.

With the tests that are been done, it was possible to create sketches for different machining equipments from PPN by lab technician. As the used language is compatible with STEP-nc, the PPN is able to adapt the standard of STEP-nc with the representation of process properties of arbitrary sequence of processes, which was not originally provided.

6. ACKNOWLEDGMENTS

The authors would like to thanks CNPq, FAPESP and CAPES for the financial support to the present project.

7. REFERENCES

Bhandarkar, M.P. and Nagi, R., 2000. "STEP-based feature extraction from STEP geometry for agile manufacturing". *Computers in Industry*, Vol. 41, pp. 3–24.

Bin, Y., Wei, X. and Zu-wen, W., 2009. "STEP-based research and realization on CAD data transformation for virtual assembly". In *World Congress on Computer Science and Information Engineering*. Vol. 1, pp. 707–710.

Fattori, C., Dobrianskyj, G., Junqueira, F., Santos Filho, D. and Miyagi, P., 2011. "Description of productive processes in a collaborative environment". In *Proceedings of Industrial Electronics Conference*. Melbourne, 37, pp. 379–384.

Leitão, P., 2009. "Agent-based distributed manufacturing control: A state-of-art survey". *Engineering Applications of Artificial Intelligence*, Vol. 22, pp. 979–991.

Rao, S.S., Sastyanarayana, B. and Sarcar, M.M.M., 2012. "Automated generation of NC part programs for turned parts based on 2-D drawing image files". *International Journal of Production Research*, Vol. 50, pp. 3470–3485.

Takahashi, K., Hasegawa, K. and Miyagi, P., 1999. "Hierarchical programming of sequential control systems by mark flow graph". In *Proceedings of IEEE International Conference on Emerging Technologies and Factory Automation*. Vol. 2, pp. 1297–1305. doi:10.1109/ETFA.1999.813138.

Tao, F., Qiao, K., Zhang, L., Li, Z. and Nee, A.Y.C., 2012. "GA-BHTR: an improved genetic algorithm for partner selection in virtual manufacturing". *International Journal of Production Research*, Vol. 50, No. 8, pp. 2079–2100.

Vinodh, S. and Kuttalingam, D., 2011. "Computer-aided design and engineering as enablers of agile manufacturing: A case of study in an Indian manufacturing organization". *Journal of Manufacturing Technology Management*, Vol. 22, No. 3, pp. 405–418.

Wang, X., Wong, T.N. and Wang, G., 2012. "Service-oriented architecture for ontologies supporting multi-agent system negotiations in virtual enterprise". *Journal of Intelligent Manufacturing*, Vol. 23, No. 4, pp. 1331–1349.

Yoo, T., Jeong, B. and Cho, H., 2010. "A Petri nets based functional validation for services composition". *Expert Systems with Applications*, Vol. 37, No. 5, pp. 3768–3776.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.