Temporal Validation of Extended Workflow Processes by means Petri Nets with Clocks

Daniel Riesco, Gabriel Vilallonga, Germán Montejano, Roberto Uzal, Welch Daniel
Universidad Nacional de San Luis - Departamento de Informática
Ejército de los Andes 950 - 5700 – San Luis – Argentina
e-mail: {driesco | gvilallo | gmonte | ruzal | wdaniel}@unsl.edu.ar
Sergio Gallina
Universidad Nacional de Catamarca – Departamento Electrónica
Maximio Victoria 55 – S.F. del Valle de Catamarca
e-mail: sgallina@tecno.unca.edu.ar

Abstract

At present the business processes have temporary requirements within their specifications. Logistics, e-commerce, are examples of these, among others. The time involved in business processes is important regarding the interaction of the actors. The sound of the technology involved in building these processes plays a key role in assessing the risk of implementation.

The possibility of having technologies with elements for such specifications is vital to accurately model of reality. Workflow (Wf) is the technology of wide acceptance and recognition that can improve business processes. The Wf architecture has the Interface 1, which lets you define process through its processes definition language (PDL), but has no elements to express temporary restrictions of this kind. Our works present a theoretical framework in which there is an extension of the grammar of the WPDL to allow the specification of time variable. It also establishes a correspondence between the PD and its underlying Petri Net with Clocks (PNwC) preserving its semantic. The correspondence is specified by means of the RAISE Specification Language (RSL).

The correlation between these formalisms lets you define business process with temporary restrictions validated.

Keywords: Business Process, Workflow, Wf Process Definition, Petri Nets, PN with Clocks, RAISE Specification Language.

1. Introduction

At present in commercial areas the premise is to provide quality services. Technological advances give support to this demand. The trade logistics, referred to a case, either monetary or merchandise operations require streamlined processes to ensure the satisfaction of all actors in the business.

The growing demand for services leads companies to adapt to these already having to make use of new technologies. This presents the need to redefine business processes. This involves applying process reengineering, which has given rise to different technologies that address this need.

The Business Process Reengineering (BPR) is used it as a valid form of feeding to the software engineering (SE), allowing to the reframing and redesign of the business process to obtain an improvement in the measures of yields [1, 2]. Workflow (Wf) is one of the technologies that allow BPR implementation. It provides the automatization of the business process, in a whole or partly. The WfMC presents the Model of Reference of Wf by the necessity to define interphase to the elements of Wf [3]. Interphase 1: The Process Definition (PD) [4] realizes the transference of the definitions of processes from the external tools to the Wf engine where these are enacted. The PD is defined as the representation of a BP that support automated manipulation, as the modeled one, or the declaration by means of a Wf Management System (WMS). The PD consists of a network of activities and its relations which will be executed by one or more Wf engines.

It has been detected that Wf presents two weaknesses in the modeling systems: it does not have elements that allow modeling the variable time and a tools for the validation of their models [5, 6].

The PNwC, a PN extension based on Timed Graphs [7], have all the potential of the PN and allow the modeled the variable time, by means of the specification of clocks, using place invariants and temporal conditions in the transitions. The PNwC has a method of analysis of the temporal restrictions in the state space that is generated from the defined net [8, 9]. Works like [10] allows prediction of total time of a work, location of time, and the task priority, where a frame is presented for the temporal Wf management. There is no a precise reference to the validation of these processes. In [11] sets out the use of the Concurrent Transitions Logic to specified, analyzed and planned of wfs. With respect to the temporal restrictions they are treated at level of events, simple algebra of restrictions, which specify that a task must begin before another one, and that the execution of a task cause that another one is executed or no, among other cases. The temporal treatment on the part of the authors is not mainly boarded.

Our work presents a theoretical frame that is centered in the extended grammar for the Wf specification processes. We add temporal elements, clocks, for a later correspondence with a PNwC, which has equal behavior. All types of analyses that are made to the PNwC, obtained by translation, are practiced to the definition of the underlying process. Thus the process defined in Interphase 1, of the Wf Reference Model, can be validated. The establishment of the correspondence of a WPD and a PNwC has been specified in a rigorous language as RSL, RAISE Specification Language, giving to as a formal frame.

2. Preliminary concepts

2. 1 RAISE

RAISE provide a complete formal method, along with support tools, for the application of a formal approach to the specification, design and software implementation [12]. The RAISE Specification Language, RSL, provides a mathematical notation, which is useful to specify, design and develop software formally [13]. RSL allows specifying abstractions, systems with sequential specifications like concurrent systems, systems of great size being modularized, and the separation of subsystems that will be developed separately. RSL allows the operational design of low level that will be expressed at a detail level from which the extraction becomes of final code. It allows to the construction from the specification to the design being used an only formalism

2.2 Petri Nets with Clocks

A PNwC [8, 9] is a PN with temporal elements, Clocks, an extended PN, based in timed graphs [15, 16, 17, 18], with a finite set of Clocks whose values are increased uniformly with time. The restrictions associated with the system are expressed by means of invariants in the places and a condition of enabling by each transition. The reset of a clock can be specified in each transition. Also, the firing of a transition is an instantaneous action that does not consume time. The time runs single in the places, not beyond the established in the invariant of the place. Formally the structure of a PNwC is a t-upla:

PNwC = <S, X, Inv, C, A >

- **S**, a PN standard structure,
- **X,** finite set of clocks, real values of the system,
- **Inv:** $\mathbf{P \rightarrow \Omega}$, it associates to each place $\mathbf{p_i} \in \mathbf{P}$, a restricted predicate $\mathbf{\Omega} \in \mathbf{\Omega_X}$, **place invariant**.
- **C:** $\mathbf{T \rightarrow \Psi}$, it associates to each transition $\mathbf{t} \in \mathbf{T}$, a restricted predicate $\mathbf{\Psi} \in \mathbf{\Psi_X}$, transition condition.

A: $\mathbf{T \rightarrow w}$**,** transition clock set to reset, $\mathbf{w \subseteq X}$**.**

2.3 Workflow

Workflow is technology that allows the Business Process Reengineering implementation. It allows the automatization of the business process, during which documents, information, or tasks are passed from a participant to another one, according to a set of rules of procedure [3]. Wf normally includes a certain number of logical steps, where each one is known like an activity. An activity can involve manual interaction with a user, or participant of Wf, or the activity can be executed using computers like resources. The WMS is a system that defines, creates and handles the execution of Wf through the software use. The Wf Reference Model, WMR, arises like the necessity to define the interphase to the Wf elements [4].

All the Wf systems are oriented to processes. A process definition and creation, that is a representation of which would have to happen, include some subprocesses which involve activities. Therefore, Wf executes the automated activities, whereas the definition of processes describes all the activities automatizables or no.

2.3.1 Workflow Process Definition

The Wf Reference Model of is the model that describes the five interfaces that interacts with the Wf Engine. The Interphase 1 works in the handling of the PD transference from external tools to the Wf engine, where these are enacted. The WPD describes the process indeed. In the definitions of processes relations between the different activities settle down, transitions information and the implementation of these.

2.3.2 Workflow Process Activity

The WPA is used to define each elementary activity that conform the Wf process. The attributes can be defined to specify control information of an activity, alternatives of implementation, priority, data used specifically in BPR, and simulation situations. In general, the restrictions of transitions can be declared at level of the limit within the surrounding process while the specialized conditions of flows (subflows, loops, or internal parts of a route activity) operate internally to an activity. The following diagram shows to the generic structure of an activity and its variants:

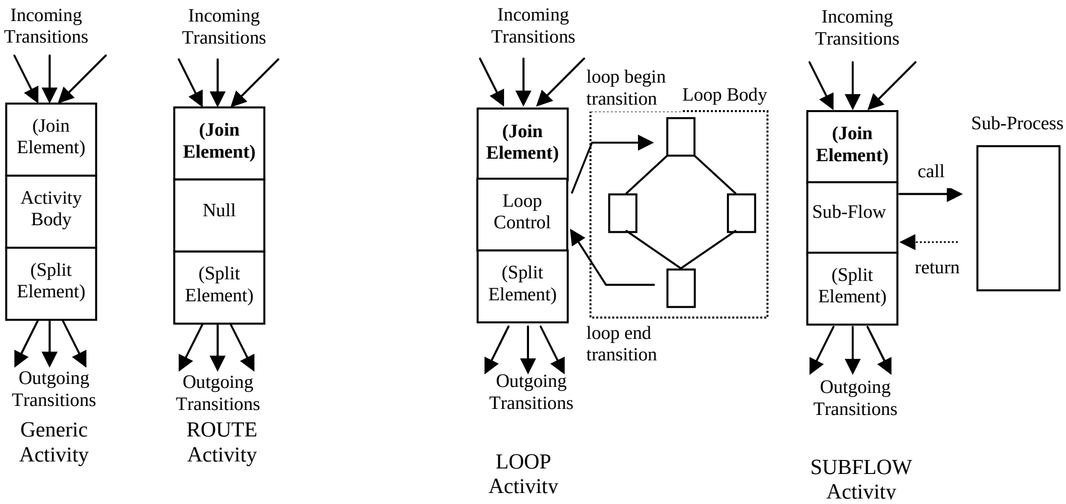

Fig. 1: Activity kinds

If exist multiple input or output for an activity, its definition provides the options to express restrictions of control flow and the conditions evaluation semantic.

2.3.3 Transition Information

The activities are related by means of control flow conditions (transition information). The Transition Information describes the possible transitions between activities, which are enabled and disenabled during the execution of the Wf, and the conditions in which these are made.

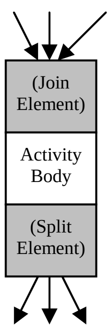

Fig. 2 – Atributes JOIN and SPLIT

The Transition Restriction has special attributes like descriptions of JOIN and SPLIT. Attribute JOIN describes the semantics of multiple input transitions. Attribute SPLIT describes the semantics of multiple output transitions for an activity. It is possible to express by means AND SPLIT and XOR SPLIT.

It specifies the attributes SPLIT and JOIN. For space reasons we just wide on the attribute SPLIT.

2.3.3.1 Atributo AND SPLIT

The AND SPLIT defines a number of concurrent threads represented by output transitions of a given activity. If transitions are conditional, the number of threads executed in parallel depends on the condition associated with each transition, which are evaluated in parallel.

There is a possibility that a transition contains a condition OTHERWISE. This leads to divide the evaluation.

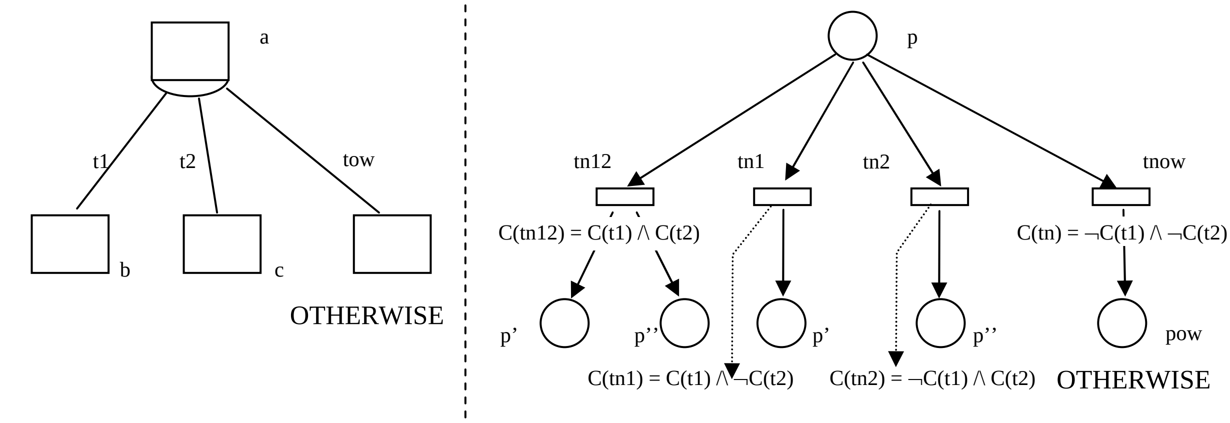

Fig. 3 – Atributes JOIN and SPLIT

The RSL definition explained above, by subtracting the defining attribute XOR, is given by the following specification:

```
Split: W.WfPD × W.Activity × N.NETwC × N.Place ~→ Bool
Split(WPD, act, NwC, ps) ≡
   (Split_And(act) ⇒
    let LTF: W.TRANSITION-list, SEC: E.Psi-set •
         ListTransFromW_O_OtherW(act)= LTF ∧
         SetExprComb(LTF) = SEC in
         ∀ expr: E.Psi • expr ∈ SEC ⇒
          (∃ ts: N.Trans • ts ∈ N.T(N.Nt(NwC))∧
            EqualCondTrExp(ts, expr) ∧ ps ∈ dom N.I(N.Nt(NwC))(ts) ∧
              SetPlaceOutTrans(WPD, NwC, expr, LTF) ⊆ dom N.O(N.Nt(NwC))(ts)
             ) ∧
          ExistOtherWise(act) ⇒
           (∃ tow:N.Trans, pow:N.Place •
              tow ∈ N.T(N.Nt(NwC)) ∧ pow ∈ N.P(N.Nt(NwC)) ∧
              EqualCondTrExp(tow, CondNeg(LTF)) ∧
              ps ∈ dom N.I(N.Nt(NwC))(tow) ∧ pow ∈ dom N.O(N.Nt(NwC))(tow)
              )
      end)
```

```
     ∨
      …… /* Split XOR */
pre(CorrespWf_PNwC(WPD, NwC) ∧ ps ∈ N.P(N.Nt(NwC)) ∧act ∈ W.ACT_LIST(WPD) ),
```

3. Extending Workflow with Clocks

The extension of Wf is carried out being based on concepts of the PNwC. This way, equips Wf with the ability to model systems where the time plays an important roll. With a tool like the PNwC, allows the validation of the model in the temporary aspect, giving to Wf the ability on which until now it does not count.

To the Wf abstract grammar the time concepts it is added and an extension of the concrete grammar is defined.

In order to obtain the extension of Wf, for the time handling, a grammar is included that allows to handle temporal expressions [14], that are those that will specify restrictions in Wf. In order to carry out this, modifications to the original grammar are made in those places in where the positioning of temporal restrictions is feasible, like in activities and transitions, so that it can handle this type of expressions. This is made having in account the propose grammar for the Wf Process Definition Language (WPDL) [4]. This extension allows the inclusion of expressions with clocks in conditions of activities and conditions of transitions. Also the inclusion of concepts of PNwC is made, the affectations.

In an activity is feasible to place restrictions at level of the limit of duration of this. In case of an LOOP activity this will have associate a temporal condition in the loop condition.

The next a graph shows the activity structure and its possible values for the most relevant fields. Those fields stand out in where it is possible to specify temporal restrictions.

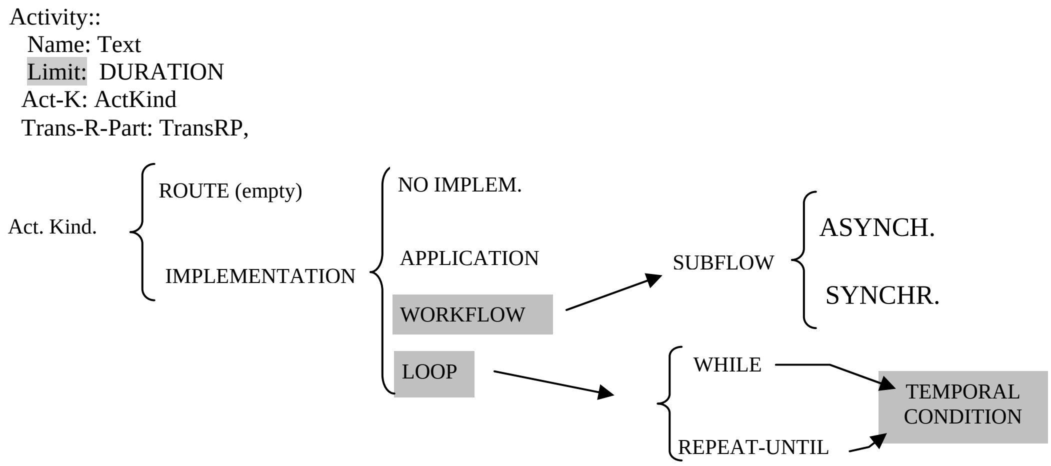

Fig. 4: Activity Structure

The following Wf grammar defines this activity:

```
<Activity Kind Information> ::= ROUTE | IMPLEMENTATION <implementation>
      <implementation> ::= NO
            | APPLICATION <genereric tool list> | WORKFLOW <subflow reference>
            | LOOP <loop kind>
                    CONDITION <loop condition>
```

The Wf extension, by means of the inclusion of a temporal expression, is made along with the expressions of Wf, where the possibility of coexistence of some of the types of expressions or both exists. The original condition of Wf is:

```
<loop condition> ::= <condition>
```

The grammar is extended, in abstract form, to express temporary restrictions:

```
<loop condition> ::= <condition> |  <RestrTempExp> |
        <condition> <ANDOp> <RestrTempExp>
```

Is necessary to enable to put temporal predicates and the affectations on the transition structure like thus also in the conditional transitions. Thus the abstract grammar of Wf is extended with time concepts and the extension of the concrete grammar is defined.

The Wf grammar what express the Transition Information is:

```
<Transition Information List> ::=
    TRANSITION <transition id> …..
          <transition kind description>…..
      END_TRANSITION
  [<Transition Information List>]
```

The inclusion of temporal elements is necessary to express restrictions. In base the definition of previous transition, includes the ability of reset for a set of clocks of the system by means of the inclusion of the concept of Affectation. The extension turns out to be the following one:

```
<Transition Information List> ::=
    TRANSITION <transition id>
      ..<transition kind description> … <Afectation>
      END_TRANSITION
```

where <Afectation> ::= <CLOCK List>

With respect to the extension made to Wf in conditions of transitions, given its original grammar:

```
<transition condition> ::= <condition> |
                              OTHERWISE
```

is extended to support the temporal restrictions expressed:

```
<transition condition> ::= <condition> | <RestrTempExp> | <condition> <ANDOp> <RestrTempExp> |
                              OTHERWISE
```

In [14] includes a grammar that allows to handle temporal expressions, and modifications to the grammar have been made originates of Wf, in those places in where the positioning of temporal restrictions is feasible, like in activities and transitions.

4. Semantic of Workflow and PNwC elements

The extension of Wf is made to allow the specification of temporal expressions for the validation and analysis of its models, without having to arrive at the phase of simulation.

The extensions are made at level of process definition, where a set of clocks is added and the possibility to express temporal restrictions. At level of activities expressions that allow checking the maximum time of permanence and in the loop activities add expressions for the handling of the condition of this activity. With respect to the transitions the affectation concept is added, set of clocks to be reset, to put to zero, along with a temporal expression to denote a restriction, which is due to fulfill, to enable to fire.

These extensions in Wf allow making the correspondence with the PNwC in direct form. The PNwC that is obtained can be analyzed for the verification of the temporary specifications modeled by means of Wf.

The correspondence between Wf and PNwC becomes formal being based on schemes RSL of each one of these. A new scheme RSL is obtained that formalizes the possibility that rigorously from any extended Wf a PNwC with equal semantics to the obtained from the definition of the Wf process.

In the specifications below demonstrates that for all extended WPD corresponds a PNwC. This is carried out making the formal specification of the correspondence between different concepts to respect the semantics of Wf. From the WPD specifies the structure of a PNwC, this is:

```
WORKFLOW_PROCESS_DEF ::
  WORKFLOW-ID: Text
  ACTIVITIES: ACTIVITY-set
  TRANSITONS: TRANSITION-set
  CLOCKS: Clock-set,
```

```
NETwC ::
  Nt : Net
  X : T.Clock-set
  Inv : Place → InvValue
  C : Trans → E.Psi
  A : Trans ~→
     (T.Clock m→ T.RealPos)↔ re_A,
```

```
Net::
  P: Place-set
  T: Trans-set
  I: Trans m→ (Place m→ Nat)
                 ↔ re_I
  O: Trans m→ (Place m→ Nat)
                 ↔ re_O,
```

Fig. 5: Structures Correspondence

The correspondence between set of activities, transitions, and clocks of Wf with places, transitions and PNwC clocks, become by means of the name of each one of the elements of these. This way, one makes sure that all activity, transition and clock of Wf also are present in PNwC.

4.1 Activity Kinds

Two types of activities exist, the type of Activity ROUTE and type IMPLEMENTATION. Activity ROUTE is a "dummy" activity that allows the expressions of cascade transitions conditions. The correspondence of this type of Wf activity and its PNwC, this single routes activity is represented by a place. Once established the invariant that corresponds to this activity, according to the temporal restriction that has been assigned, goes immediately to the treatment of the output transition restrictions of the activity, JOIN and SPLIT. The type of activity IMPLEMENTATION, is classified as well in: NO IMPLEMENTATION, APPLICATION, SUBFLOW and LOOP. The axiom that specifies the correspondence between these types of activity, NO IMPLEMENTATION, APPLICATION, and their PNwC is:

```
∀ WPD: W.WfPD, a : W.Activity •
   a ∈ W.ACT_LIST(WPD) ∧
   (RouteAct(a) ∨ NoImplement(a) ∨ Application(a)) ⇒
        (∃ NwC : N.NETwC, p: N.Place •
             CorrespWf_PNwC(WPD, NwC) ∧ p ∈ N.P(N.Nt(NwC)) ∧
             Corresp(WPD, a, NwC, p)  ∧ Split(WPD, a, NwC, p)
         ),
```

4.1.1 LOOP Implementation Activity

Type LOOP allows expressions of repetitions or cycles in the network, of two possible forms, supporting the structures of programming "WHILE.. DO.." and "REPEAT... UNTIL ". The body of Loop is connected with the Loop Control Activity by means of the corresponding Loop connecting Transition. This connection of loops allow cycles in the network. They connect the loop body with the loop activity that is implemented by this body. The loop condition is expressed in the loop activity, and not like transition condition.

4.1.1.1 UNTIL.REPEAT Loop

In an implemented activity as loop REPEAT-UNTIL the evaluation of the associate condition is made when finalizing the first cycle. If the condition is fulfilled it leaves the cycle, otherwise it will be continued in the cycle until the condition becomes true. In the following graph is to the representation of an activity loop, type REPEAT-UNTIL and its corresponding PNwC.

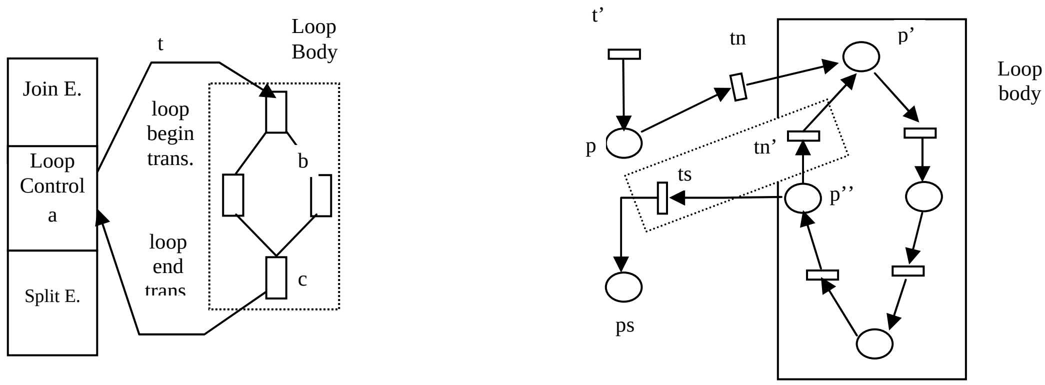

Fig. 6: A Loop Activity and a PNwC

The function RSL that specifies the correspondence, respects the identifiers contained in the figure 6. The formalization in RSL is the:

```
Act_LoopRU_P: W.WfPD × W.Activity × N.NETwC × N.Place ~→ Bool
Act_LoopRU_P(WPD, a, NwC, p) ≡
  (∃ p', p'', ps: N.Place, tn, tn', ts: N.Trans,
        t, t': W.TRANSITION, b, c: W.Activity •
        p ∈ N.P(N.Nt(NwC)) ∧ Corresp(WPD, a, NwC, p)
       ∧
        t ∈ W.TRANS_LIST(WPD) ∧ b ∈ W.ACT_LIST(WPD) ∧ … etc.
       ∧
        CorrespTT(WPD, t, NwC, tn) ∧ FromLoopTo(t) ∧
        a = FromLoop(t) ∧ b = To(t) ∧ Corresp(WPD, b, NwC, p') ∧
        p ∈ dom N.I(N.Nt(NwC))(tn) ∧ p' ∈ dom N.O(N.Nt(NwC))(tn)
       ∧
        CorrespTT(WPD, t', NwC, tn') ∧ FromToLoop(t') ∧
        c ∈ W.ACT_LIST(WPD) ∧ c = From(t') ∧ a = ToLoop(t') ∧
        Corresp(WPD, c, NwC, p'') ∧
        EqualCondTrExp(tn', CondLoop(a)) ∧
        EqualCondTrExp(ts, Neg(CondLoop(a)))
       ∧
        p'' ∈ dom N.I(N.Nt(NwC))(tn') ∧ p'' ∈ dom N.I(N.Nt(NwC))(ts) ∧
        ps ∈ dom N.O(N.Nt(NwC))(ts) ∧ p' ∈ dom N.O(N.Nt(NwC))(tn')
       ∧
        Split(WPD, a, NwC, ps)
  )
  pre ( p ∈ N.P(N.Nt(NwC)) ∧ a ∈ W.ACT_LIST(WPD) ),
```

Similar way one of the different types from activities and the information of transition are dealed with each. They are not included in this publication for space reasons. All these formalizations are found in [14].

5 - Correspondence Formal Specification

The extension of Wf with temporal elements allows making the correspondence with the PNwC in direct form. The PNwC obtained can be analyzed for the verification of the modeled temporal specifications by means of Wf. In this section is the function RSL that specifies the correspondence of Wf, elements and properties of the WPD, with PNwC elements. This formal correspondence is based on RSL. A new scheme RSL is obtained that formalizes the possibility to extend a WPD to PNwC. The semantics obtained is the same that the WPD, having the possibility of analyzing and of validating the temporary restrictions.

```
CorrespWf_PNwC: W.WfPD × N.NETwC → Bool
CorrespWf_PNwC(WPD, NwC) ≡
   (∀ a: W.Activity, tw: W.TRANSITION, xw: T.Clock •
        a ∈ W.ACT_LIST(WPD) ∧ tw ∈ W.TRANS_LIST(WPD) ∧ xw ∈ W.CLOCKS(WPD) ⇒
        (∃ p: N.Place, tn: N.Trans,
              xn: T.Clock •
                  p ∈ N.P(N.Nt(NwC)) ∧ tn ∈ N.T(N.Nt(NwC)) ∧
                  xn ∈ N.X(NwC) ∧ Corresp(WPD, a , NwC, p) ∧
                  CorrespTT(WPD, tw , NwC, tn) ∧  CorrespClk(WPD, xw, NwC, xn)
            ∧
              (Join(WPD, a, NwC, p) ∧ ((RouteAct(a) ∨ NoImplement(a) ∨ Application(a))  ⇒
                  Split(WPD, a, NwC, p)) ∨ RepeatUntil(a) ⇒
                       Act_LoopRU_P(WPD, a, NwC, p) ∨ While(a) ⇒
                            Act_LoopWD_P(WPD, a, NwC, p) ∨ SubflowAsy(a) ⇒
                                 Act_SFAsy(WPD, a, NwC, p) ∨ SubflowSynchr(a) ⇒
                                      Act_SFSynchr(WPD, a, NwC, p)
      )
   ),
```

6 - Conclusions

Our work presents a theoretical frame to validate the WPD by means of PNwC. The proposal is based on the extension of the WPDL grammar to offer the possibility of specifies clocks and restrictions on these in activities and transitions. The Wf extensionS enable to model and to validate the variable time, being avoided the phase of simulation. The correspondence has been formalized by means of RSL, assuring the coherence between the involved concepts.

The PNwC obtained from the WPD allows applying the validation algorithm that makes the control of the temporal restrictions; this is inconsistency in places invariants and transitions conditions. The deadlock checking is realized. Therefore the presented frame enables to the future development of tools that allows check the process where the time plays a fundamental roll, allowing qualitative analysis by means PNwC.

References

[1] Hammer, M. and Champy, J. "Reengineering the Corporation: A Manifesto for Business Revolution", Harper Collins Publishing, Inc., 1993

[2] Hammer, M, "Beyond Reengineering: How the process-centered organization is changing out work and our lives", Harper Collins, 1996.

[3] Hollingsworth, D. Workflow Management Coalition. The Workflow Reference Model. Document Number TC00-1003. Issue 1.1. Jan-95. http://www.wfmc.org

[4] Workflow Management Coalition. "Interface 1: Process Definition Interchange. Process Model". Document Number WfMC TC-1016-P Version 1.1. Oct-99. http://www.wfmc.org

[5] W. Goebel, K. Messner, B. Schwarzer. Experience in introducing Workflow Management in a Large Insurance Group. 34th Hawaii ICSS. 2001.

[6] M. Oba, S. Onada, N. Komoda. Evaluating the Quantitative Effects of Workflow System Based on Real Cases. 33th Hawaii ICSS. 2000.

[7] R. Alur and D. L. Dill, A theory of timed automata. Theoretical Computer Science 126 (1994) 183:235.

[8] G. Montejano, D. Riesco, G. Vilallonga, "An Analysis Algorithm for Timed Petri Nets" Software Engineering (SE'98). International Association of Science and Technology for Development. Las Vegas, USA. 1998.

[9] D. Riesco, G. Montejano, G. Vilallonga, A. Dasso, R. Uzal. "Underlying Formalism for a Timed Petri Net Algorithm", IASTED. Octubre de 1999, Scottsdale, Arizona, USA.

[10] J. Leon Zhao, Edward A. Stohr. "Temporal workflow management in a claim handling system". ACM SIGSOFT Software Engineering Notes. Proc. of Int. Joint Conference on Work Activities Coor. and Colab. WACC ´99. Vol. 24 I. 2.

[11] Hasan Davulcu, Michael Kifer, C. R. Ramakrishnan, I. V. Ramakrishnan. "Logic Based Modeling and Analysis of Workflow". Proc. 7th. ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. May 1998.

[12] The RAISE Method Group, "The RAISE Development Method", Prentice Hall, 1995.

[13] The RAISE Language Group, "The RAISE Specification Language", Prentice Hall, 1992.

[14] Vilallonga, Gabriel D. "Definición y Validación de Procesos Workflow Temporizados Basados en Redes de Petri con Relojes". Tesis de Maestría. UNSL. San Luís, Argentina. 2.004.