

Marco Teórico para la Traducción de Procesos Workflow Extendido a Redes de Petri con Relojos

Gabriel Vilallonga

Universidad Nacional de San Luis - Departamento de Informática
Ejército de los Andes 950 - 5700 – San Luis – Argentina
E-mail: gvilallo@unsl.edu.ar

Germán Montejano

Universidad Nacional de San Luis - Departamento de Informática
Ejército de los Andes 950 - 5700 – San Luis – Argentina
E-mail: gmonte@unsl.edu.ar

R. Uzal

Universidad Nacional de San Luis - Departamento de Informática
Ejército de los Andes 950 - 5700 – San Luis – Argentina
E-mail: ruzal@unsl.edu.ar

Resumen

Workflow (Wf) es una de las tecnologías que habilita a la Reingeniería de Procesos de Negocios (BPR), la cual ha probado ser el medio óptimo para el mejoramiento de los proceso de negocios. La Interfase 1 de Wf es la que permite la Definición de Procesos de Wf (WPD). Esta definición no posee elementos para la validación, además de la falta de elementos temporales para la especificación de la variable tiempo.

Nuestro trabajo acerca el marco teórico para la validación de la WPD por medio de las Redes de Petri con Relojos (PNwC). Se brinda un marco para la implementación de una herramienta, que permita la validación de definiciones de procesos que poseen requerimientos temporales.

Este trabajo presenta, en forma breve razones de espacio, la correspondencia entre la WPD y una PNwC. De esta manera todos los tipos de análisis que se le realizan a las PNwC, obtenida por traducción de una WPD son practicados a la definición del proceso subyacente. La correspondencia es especificada por medio del Lenguaje de Especificación RAISE (RSL).

Palabras claves: Workflow, Reingeniería de Procesos de Negocios, Redes de Petri con Relojos, RAISE.

1 - Introducción

La Reingeniería de Procesos de Negocios (BPR) se la utiliza como una forma válida de alimentación a la ingeniería de software (SE), permitiendo el replanteo y rediseño del proceso de negocio para lograr un mejoramiento en las medidas de rendimientos [1, 2].

Workflow (Wf) es una de las tecnologías que permite la implementación BPR. Provee la automatización del proceso de negocio, en un todo o en parte. La WfMC presenta el Modelo de Referencia de Wf por la necesidad de definir interfases a los elementos de Wf [3]. La Interfase 1: Definición de Procesos (PD) [4] es la encargada de la transferencia de las definiciones de procesos desde las herramientas externas al motor de Wf donde éstas son normalizadas. La PD se define como la representación de un BP de forma que soporte manipulación automatizada, tal como el modelado, o la declaración por medio de un Sistema Gestor de Wf (WMS). La PD consiste de una red de actividades y sus relaciones la cual será ejecutada por uno o más motores Wf.

Se ha detectado que Wf presenta dos debilidades a la hora de modelar sistemas: la de no poseer elementos que permitan modelar la variable tiempo y no poseer herramientas para la validación de sus modelos [5, 6].

Por otro lado, las PNwC, extensión de las PN basada en Grafos Temporizados [7], poseen todas las virtudes de estas redes y permiten el modelado de la variable tiempo, por medio de la especificación de relojes, usando invariantes temporales en los lugares y condiciones temporales en las transiciones. Las PNwC poseen un método de análisis de las restricciones temporales en el espacio de estado que se genera a partir de la red definida [8, 9].

Existen trabajos como [10] que permiten predicción de tiempo total de un trabajo, ubicación de tiempo, y priorización de tareas, presentado un marco para la gestión de wf temporal. No hay una referencia puntual a la validación de dichos procesos. En [11] se propone el uso de la Lógica de Transacción Concurrente (CTR) como lenguaje de especificado, analizado y planificado de wfs. Con respecto a las restricciones temporales son tratados a nivel de eventos, algebra simple de restricciones, las cuales especifican que una tarea debe comenzar antes que otra, y que la ejecución de una tarea causa que otra se ejecute o no, entre otros casos. El tratamiento temporal por parte de los autores no es mayormente abordado.

Nuestro trabajo presenta un marco teórico que se centra en la obtención de la gramática extendida para la especificación de procesos Wf. A esta se le agregan elementos temporales, como son los relojes, para una posterior correspondencia con una PNwC, la cual posee igual comportamiento. De esta manera todos los tipos de análisis que se le realizan a las PNwC, obtenida por traducción, son practicados a la definición del proceso subyacente. De esta manera se puede validar el proceso definido en la Interfase 1 del Modelo de Referencia de Wf.

El establecimiento de la correspondencia de una WPD y una PNwC ha sido especificado en un lenguaje riguroso como lo es RSL, RAISE Specification Language, dotando a nuestro trabajo de un marco formal.

2 – Conceptos preliminares

2.1 RAISE

RAISE provee un método formal completo, junto con herramientas de soporte, para la aplicación de un acercamiento formal a la especificación, diseño e implementación de software [12].

El Lenguaje de Especificación RAISE, RSL, provee una notación basada en matemática, la cual es útil para especificar, diseñar y desarrollar formalmente software [13]. RSL permite especificar abstracciones, sistemas con especificaciones secuenciales como así también sistemas concurrentes, sistemas de gran tamaño a ser modularizados, y la separación de subsistemas que serán desarrollados por separado. RSL permite el diseño operacional de bajo nivel que será expresado a un nivel de detalle desde el cual se hace la extracción de código final. Permite la construcción desde la especificación al diseño usándose un único formalismo.

2.2 Redes de petri con Relojes

Una PNwC [8, 9] es una PN con Relojes, una PN extendida, basada en grafos temporizados [15, 16, 17, 18], con un conjunto finito de Relojes cuyos valores son incrementados uniformemente con tiempo. Las restricciones asociadas con el sistema son expresadas por medio de invariantes en los lugares y una condición de habilitación por cada transición. La reinicialización de un reloj puede ser especificada en cada transición. También, el disparo de una transición es una acción instantánea que no consume tiempo. El tiempo corre solo en los lugares, no más allá de lo establecido en el invariante del lugar.

Formalmente la estructura de una PNwC es una n-upla:

$$\text{PNwC} = \langle S, X, \text{Inv}, C, A \rangle \text{ donde:}$$

- **S**, estructura de una PN estándar,
- **X** conjunto finito de relojes, variables reales positivas, del sistema,
- **Inv: P → Ω**, asocia a cada lugar $p_i \in P$, un predicado restringido $\Omega \in \Omega_X$ llamado **invariante del lugar**.
- **C: T → Ψ**, asocia a cada transición $t \in T$, un predicado restringido $\Psi \in \Psi_X$ llamada **condición de la transición**.
- **A: T → w**, conjunto de relojes de la transición que son inicializados en cero $w \subseteq X$.

2.3 Workflow

Workflow es tecnología que permite la implementación de Reingeniería de Procesos de Negocios. Permite la automatización del proceso de negocio, durante el cual documentos, información, o tareas son pasados de un participante a otro, de acuerdo a un conjunto de reglas de procedimiento [3]. Wf normalmente comprende un cierto número de pasos lógicos, donde cada uno de estos es conocido como una actividad. Una actividad puede involucrar interacción manual con un usuario, o participante de Wf, o la actividad puede ser ejecutada usando como recursos a computadoras.

El WMS es un sistema que define, crea y maneja la ejecución de Wf a través del uso de software. El Modelo de Referencia de Wf, WMR, surge como la necesidad de definir la interfase a los elementos de Wf [4].

Todos los sistemas Wf son orientados a procesos. Una definición y creación de un proceso, o sea una representación de lo que debería ocurrir, comprende algunos subprocesos los cuales involucran actividades. Por lo tanto, Wf ejecuta las actividades automatizadas, mientras que la definición de procesos describe todas las actividades ya sean automatizables o no.

2.3.1 Definición de Procesos Wf

El Modelo de Referencia de Wf es el modelo que describe las cinco interfaces que representa la Interoperabilidad del Motor de Wf.

La definición de la Interfase 1 trabaja en el manejo de la transferencia de PD desde herramientas externas al motor de Wf donde estos son normalizados.

La WPD describe el proceso en si mismo. En las definiciones de procesos se establecen relaciones entre las diferentes actividades, información de transiciones y la implementación de éstas.

2.3.2 Actividad de Proceso Wf

La Definición de Actividad de Wf se usa para definir cada actividad elemental que conforma el proceso de Wf. Los atributos pueden ser definidos para especificar información de control de una actividad, alternativas de implementación, prioridad, y datos usados específicamente en BPR y situaciones de simulación.

En general, las restricciones de transiciones pueden ser declaradas a nivel del límite dentro del proceso circundante mientras las condiciones de flujos especializadas (subflujos, loops, o partes internas de una actividad de ruteo) operan internamente a una actividad. El siguiente diagrama muestra la estructura genérica de una actividad y sus variantes:

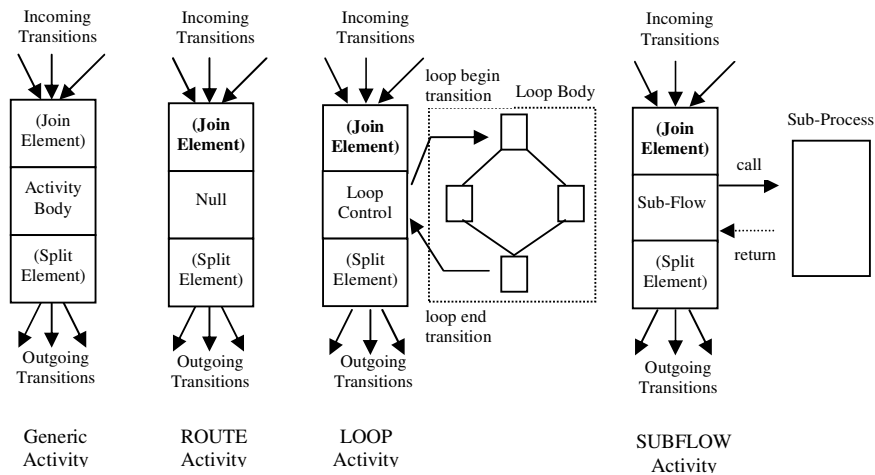


Figura 1: Tipos de Actividades

En caso de existir múltiples transiciones de entrada o salida para una actividad, las opciones para expresar restricciones de flujo de control y semánticas de evaluación de condiciones son provistas en la definición de las actividades.

La Restricción de Transición tiene atributos especiales como descripciones de JOIN y SPLIT. El atributo JOIN describe la semántica de múltiples transiciones de entrada.

El atributo SPLIT describe la semántica de múltiples transiciones de salida para una actividad. Tiene dos posibles maneras de expresarse: AND SPLIT y XOR SPLIT.

2.3.3 Información de Transición

Las actividades están relacionadas unas con otras por medio de condiciones de control de flujo (información de transición). La Información de Transición describe las posibles transiciones entre actividades, las cuales se habilitan y deshabilitan durante la ejecución del Wf, y las condiciones en que éstas se realizan.

3. Extensión de Workflow con Relojes

La extensión de Wf se lleva a cabo basándose en conceptos de las PNwC. De esta manera, se dota a Wf de la habilidad de modelar sistemas donde el tiempo juega un rol importante. La posibilidad de contar con una herramienta como las PNwC, permite la validación del modelo en el aspecto temporal, dándole a Wf una habilidad con la que hasta ahora no cuenta.

A la gramática abstracta de Wf se le incorporan conceptos de tiempo y se define una ampliación de la gramática concreta.

Para lograr la extensión de Wf, para el manejo de tiempo, se le incluye una gramática que permite manejar expresiones temporales [14], que son las que especificarán restricciones en Wf. Para llevar a cabo esto, se realizan modificaciones a la gramática original de Wf en aquellos lugares en donde es factible la colocación de restricciones temporales, como en actividades y transiciones, para que pueda manejar este tipo de expresiones. Esto se realiza teniendo en cuenta la gramática propuesta para el Lenguaje de Definición de Procesos de Wf, (WPDL, de su sigla en inglés) [4]. Dicha extensión permite la inclusión de expresiones con relojes en condiciones de actividades y de condiciones de transiciones. También se realiza la inclusión de conceptos de PNwC, como lo es el de afectaciones.

En una actividad es factible colocar restricciones a nivel del límite de duración de esta. En caso de ser una actividad de tipo LOOP esta tendrá asociada una condición temporal en la condición del loop.

A continuación se presenta un gráfico en donde se muestra la estructura de una actividad y los posibles valores para los campos más relevantes. Se destacan aquellos campos en donde es posible especificar restricciones temporales.

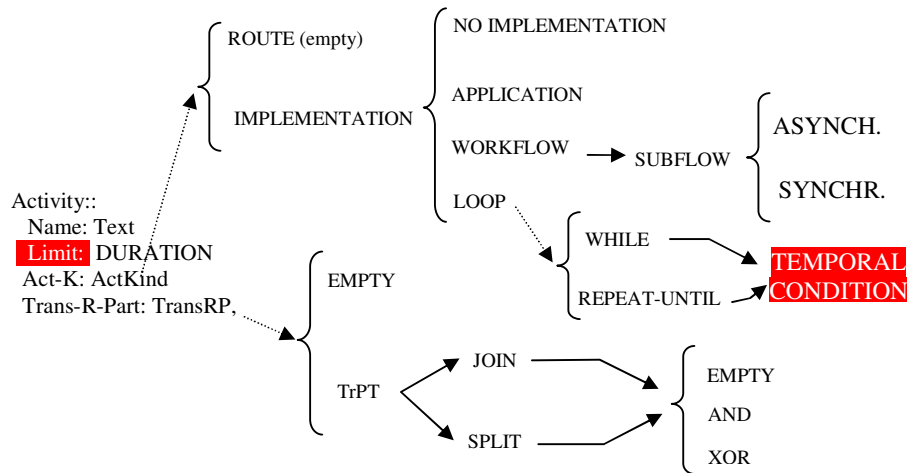


Figura 2: Estructura de una Actividad

La gramática de Wf que la define es:

```

<Activity List> ::=
    ACTIVITY <activity id>
        .....
        [LIMIT <duration>]
        <Activity Kind Information> .....[<Transition Restriction Part>].....
    END_ACTIVITY
    [<Activity List>]
    .....
<Activity Kind Information> ::= ROUTE
  
```

```

| IMPLEMENTATION <implementation>
<implementation> ::= NO
| APPLICATION <generic tool list> | WORKFLOW <subflow reference>
| LOOP <loop kind>
| CONDITION <loop condition>
.....
<subflow reference> ::= <execution> <process id>
<execution> ::= ASYNCHR | SYNCHR
<loop kind> ::= WHILE | REPEAT_UNTIL
<loop condition> ::= <condition> .....

```

Se realiza la extensión de Wf por medio de la inclusión de una expresión temporal junto con las expresiones de Wf, donde exista la posibilidad de coexistencia de alguno de los tipos de expresiones o ambos. La condición original de Wf es:

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

se extiende la gramática, en forma abstracta, para que puedan expresarse restricciones temporales:

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

Sobre la estructura de una transición se habilita a la colocación de predicados temporales a nivel de afectaciones de esta, como así también en aquellas que son transiciones condicionales. Así a la gramática abstracta de Wf se le incorporaron conceptos de tiempo y se ha definido una ampliación de la gramática concreta.

La gramática de Wf que expresa la Información de Transición es:

```

<Transition Information List> ::=
TRANSITION <transition id>...<transition kind description>... END_TRANSITION
[<Transition Information List>]
<transition kind description> ::=
FROM <activity id> TO <activity id>
[CONDITION <transition condition>] | FROM LOOP <activity id> TO <activity id>
| FROM <activity id> TO LOOP <activity id>
<transition condition> ::= <condition> | OTHERWISE

```

Se realiza la extensión de Wf por medio de la inclusión de un elemento importante en el manejo de la variable tiempo.

Tomando como base la definición de transición anterior, se le realiza la inclusión de la habilidad de reseteado para un conjunto de relojes del sistema por medio de la inclusión del concepto de Afectación.

La extensión resulta ser la siguiente:

```

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

```

donde

```
<Afectation> ::= <CLOCK List>
```

Con respecto a la extensión realizada a Wf en condiciones de transiciones, dada su gramática original:

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

se extiende para que puedan expresarse restricciones temporales:

```

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

```

En [14] se incluye una gramática que permite manejar expresiones temporales, y se han realizado modificaciones a la gramática original de Wf, en aquellos lugares en donde es factible la colocación de restricciones temporales, como en actividades y transiciones.

4 - Semántica de Elementos Workflow y PNwC

La extensión de Wf se realiza para que permita la especificación de expresiones temporales para la validación y análisis de sus modelos, sin tener que llegar a la fase de simulación.

Las extensiones son realizadas a nivel de definición de proceso, donde se agrega un conjunto de relojes y la posibilidad de contener expresiones temporales. A nivel de actividades se agregan expresiones que permitan chequear el tiempo máximo de permanencia, y dentro de las actividades tipo loop, expresiones para el manejo de la condición de dicha actividad. Con respecto a las transiciones se agrega el concepto de afectación, conjunto de relojes a ser reinicializados en cero, junto con una expresión temporal para denotar una restricción, la cual se debe cumplir, para habilitar a dicha transición a disparar.

Estas extensiones en Wf permiten realizar la correspondencia con las PNwC en forma directa. La PNwC que se obtiene puede ser analizada para la verificación de las especificaciones temporales modeladas por medio de Wf.

La correspondencia entre Wf y PNwC se formaliza basándose en los esquemas RSL de cada uno de estos. De esta manera se obtiene un nuevo esquema RSL que formaliza rigurosamente la posibilidad de que a partir de cualquier Wf extendido se obtiene una PNwC con igual semántica que la definición de proceso de Wf, teniendo la posibilidad de analizar y validar las restricciones temporales.

En las siguientes especificaciones se demuestra que para toda WPD extendido le corresponde una PNwC. Esto se lleva a cabo realizando la especificación formal de la correspondencia entre distintos conceptos para respetar la semántica de Wf.

A partir de la WPD se especifica la estructura de una PNwC, esto es:

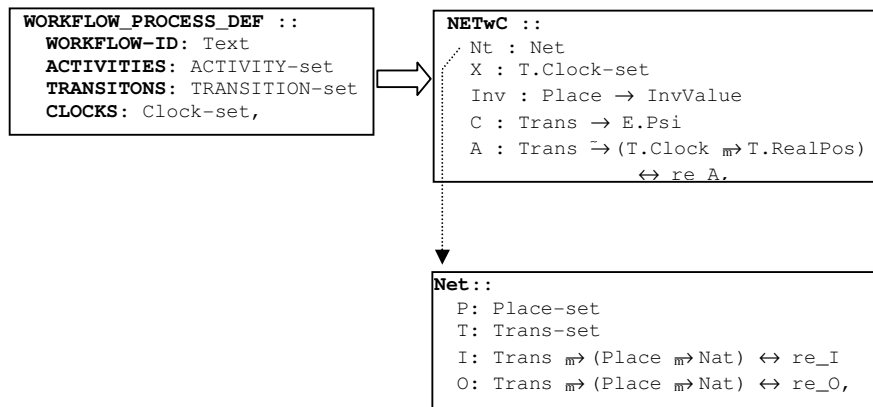


Figura 4: Correspondencia de Estructuras

La correspondencia entre conjunto de actividades, transiciones, y relojes de Wf con lugares, transiciones y relojes PNwC, se lo hace por medio del nombre de cada uno de los elementos de estos. De esta manera, se asegura que toda actividad, transición y reloj de Wf también está presente en PNwC.

4.1 - Tipos de Actividades

Existen dos tipos de actividades, el tipo de Actividad ROUTE y el tipo IMPLEMENTATION.

La actividad ROUTE es una actividad “dummy” que permite expresiones de condiciones de transiciones en cascada. Con respecto a la correspondencia de este tipo de actividad Wf y su correspondiente PNwC, al ser esta actividad solo de ruteo es representada por un lugar. Una vez establecido el invariante que le corresponde a dicha actividad, según la restricción temporal que se le ha asignado, se pasa inmediatamente al tratamiento de las restricciones de transiciones de salida a la actividad, JOIN y SPLIT.

El tipo de actividad IMPLEMENTATION, a su vez se clasifica en: NO IMPLEMENTATION, APPLICATION, SUBFLOW y LOOP.

El axioma que especifica la correspondencia entre estos tipos de actividad, NO IMPLEMENTATION, APPLICATION, y su PNwC es el siguiente:

$$\begin{aligned}
 & \forall \text{ WPD: W.WfPD, a : W.Activity } \bullet \\
 & \quad a \in \text{W.ACT_LIST(WPD)} \wedge \\
 & \quad (\text{RouteAct}(a) \vee \\
 & \quad \text{NoImplement}(a) \vee \\
 & \quad \text{Application}(a)) \Rightarrow \\
 & \quad (\exists \text{ NwC : N.NETwC, p: N.Place } \bullet \\
 & \quad \quad \text{CorrespWf_PNwC(WPD, NwC)} \wedge p \in \text{N.P(N.Nt(NwC))} \wedge \\
 & \quad \quad \text{Corresp(WPD, a, NwC, p)} \\
 & \quad \quad \wedge \text{Split(WPD, a, NwC, p)} \\
 & \quad) ,
 \end{aligned}$$

4.1.1 - Actividad de Tipo Implementación LOOP

El tipo LOOP permite expresiones de repeticiones o ciclos en la red, de dos posibles formas, semejantes a las estructuras de programación “WHILE .. DO..” y “REPEAT ... UNTIL”.

El cuerpo de Loop está conectado con la Actividad de Control de Loop por medio de la correspondiente Transición de Conexión de Loop.

Las transiciones de conexión de loops permiten ciclos en la red. Conectan el cuerpo del loop con la actividad de loop que está implementada por este cuerpo. La condición de loop es expresada en la actividad de loop, y no como condición de la transición.

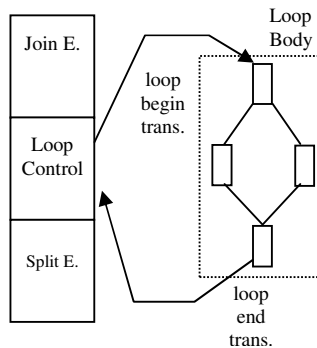


Figura 5: Actividad Loop

4.1.1.1- Loop REPEAT-UNTIL

En una actividad implementada como un loop REPEAT-UNTIL la evaluación de la condición asociada se realiza al finalizar el primer ciclo. Si la condición se cumple se abandona el ciclo, de lo contrario se seguirá ciclando hasta que la condición sea verdadera.

En el siguiente gráfico se muestra la representación de una actividad loop, tipo REPEAT-UNTIL y su correspondiente PNwC.

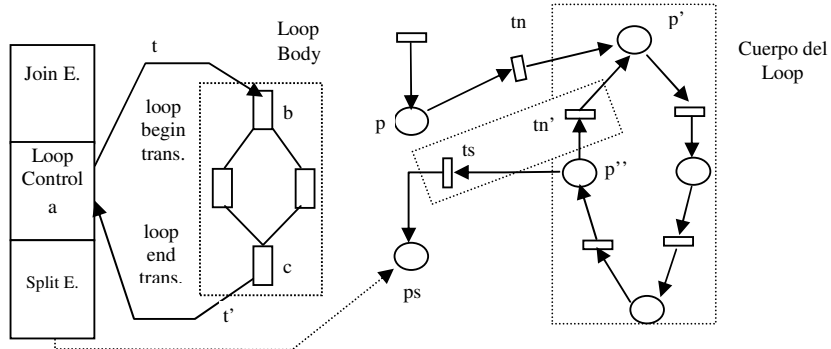


Figura 6: Correspondencia de una Actividad Loop y una PNwC

La función RSL que especifica la correspondencia, respeta los identificadores contenidos en la figura. La formalización en RSL es la siguiente:

$Act_LoopRU_P: W.WfPD \times W.Activity \times N.NETwC \times N.Place \rightarrow Bool$

$Act_LoopRU_P(WPD, a, NwC, p) \equiv$

```

(∃ 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) ),

```

De manera similar es tratado cada uno de los distintos tipos de actividades y la información de transición. No son abarcados en esta publicación por razones de espacio. Todas estas formalizaciones son encontradas en [14].

5 - Especificación Formal de la Correspondencia

La extensión de Wf con elementos temporales permiten realizar la correspondencia con las PNwC en forma directa, salvo en casos puntuales. La PNwC que se obtiene puede ser analizada para la verificación de las especificaciones temporales modeladas por medio de Wf.

En esta sección se muestra la función RSL que especifica la correspondencia de Wf, elementos y propiedades de la WPD, con elementos de las PNwC.

Dicha correspondencia se formaliza basándose en los esquemas RSL de cada uno de estos. De esta manera se obtiene un nuevo esquema RSL que formaliza rigurosamente la posibilidad de que a partir de cualquier Wf extendido se obtiene una PNwC con igual semántica que la definición de proceso de Wf, teniendo la posibilidad de analizar y validar las restricciones temporales.

```

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 - Conclusiones

Nuestro trabajo presenta el marco teórico para la validación de la WPD por medio de PNwC. La propuesta se basa en la extensión de la gramática de WPDL para brindar la posibilidad de especificar relojes y restricciones sobre estos en actividades y transiciones

La extensión realizada a Wf lo habilita para modelar y validar la variable tiempo, evitando la fase de simulación.

Cada una de las correspondencias realizadas ha sido formalizada por medio de RSL, asegurando la solidez de los conceptos incorporados.

La obtención de una PNwC partir de una WPD permite que a dicho modelo se le aplique un algoritmo que realiza la validación y chequeo de las restricciones temporales, esto es inconsistencia en invariantes de lugares y condiciones de transiciones, detección de bloqueos temporales y estados temporalmente inalcanzables.

Por lo tanto el marco presentado habilita al desarrollo futuro de una herramienta que permita el modelado de procesos donde el tiempo juega un rol fundamental, permitiendo de esta manera el análisis cualitativo de estos por medio PNwC.

Referencias

- [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, A. Dasso, L. Favre "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 International Conference Software Engineering and Applications, 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 International Joint Conference on Work Activities Coordination and Colaboration 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 Relojos. ". Tesis de Maestría. UNSL. San Luis, Argentina. 2.004.

- [15] Yovine, S. Méthodes et outils pour la vérification symbolique de systèmes temporisés, Phd thesis, Institut National Polytechnique de Grenoble, France, May 1993.
- [16] Henzinger, Nicollin, Sifakis and Yovine. “Symbolic model checking for real-time systems”, *Information and Computation*, 111(2):193-244, 1994.
- [17] Olivero. Modélisation et analyse de systèmes temporisés et hybrides. Thèse, Institut National Polytechnique de Grenoble, France, September 1994.
- [18] Sifakis and Yovine. Compositional Specifications of Timed Systems. STACS '96.