



UNIVERSIDAD TECNOLÓGICA NACIONAL

Facultad Regional Santa Fe

DOCTORADO EN INGENIERÍA

MENCIÓN EN INGENIERÍA EN SISTEMAS DE INFORMACIÓN

Tesis Doctoral

**"DEFINICIÓN DE LA PERSPECTIVA DE RECURSOS EN
EL DESARROLLO DE SISTEMAS DE INFORMACIÓN
ORIENTADOS A PROCESOS DE NEGOCIO"**

Tesista: Ing. Luis Jesús Ramón Stroppi

Director: Dr. Pablo Villarreal

Co-Director: Dr. Omar Chiotti

Santa Fe, Argentina

Febrero 2015

Stropi, Luis Jesús Ramón

Definición de la perspectiva de recursos en el desarrollo de sistemas e información orientados a procesos de negocio. - 1a ed. - Santa Fe : el autor, 2015.

193 p. ; 29x21 cm.

ISBN 978-987-33-7495-1

1. Tesis Doctorales. I. Título

CDD 378.007

UNIVERSIDAD TECNOLÓGICA NACIONAL

Facultad Regional Santa Fe

Comisión de Posgrado

Se presenta esta Tesis en cumplimiento de los requisitos exigidos por la Universidad Tecnológica Nacional para la obtención del grado académico de Doctor en Ingeniería, mención Sistemas de Información

"DEFINICIÓN DE LA PERSPECTIVA DE RECURSOS EN EL DESARROLLO DE SISTEMAS DE INFORMACIÓN ORIENTADOS A PROCESOS DE NEGOCIO"

POR

ING. LUIS JESÚS RAMÓN STROPPI

DIRECTOR: DR. PABLO VILLARREAL

CO-DIRECTOR: DR. OMAR CHIOTTI

JURADOS DE TESIS

DR. DANIEL RIESCO

DR. CRISTIAN MATEOS DIAZ

DR. HORACIO LEONE

SANTA FE, ARGENTINA

FEBRERO 2015

A Dios y a mi familia.

1 Índice

1	Índice	i
2	Índice de Tablas	v
3	Índice de Figuras	vii
4	Resumen.....	ix
5	Prólogo.....	xi
6.	Agradecimientos	xvii
1	Introducción	1
1.1	Contexto	1
1.2	Motivación.....	3
1.3	Objetivos.....	5
1.3.1	Objetivos Específicos	5
1.4	Método de Investigación	6
1.5	Principales Contribuciones	8
1.6	Organización de la Tesis.....	9
2.	Marco Teórico.....	11
2.1.	Perspectiva de Recursos de Procesos de Negocio	11
2.1.1.	Definición de la Estructura de Recursos	12
2.1.2.	Definición de Políticas de Distribución de Trabajo	14
2.1.3.	Definición de Mecanismos de Autorización	15
2.2.	Soporte a la Perspectiva de Recursos por BPMN 2.0	16
2.2.1.	Trabajos Relacionados a la Definición de la Perspectiva de Recursos con BPMN	19
2.3.	Arquitecturas de WfMS.....	20
2.4.	Perspectiva de Recursos en Lenguajes de Ejecución de Procesos	23
2.4.1.	YAWL.....	23
2.4.2.	BPEL4People	26
2.5.	Soporte a la Perspectiva de Recursos por WfMSs	28
2.6.	Metodologías de Desarrollo de SIOPs	30
2.7.	Conclusiones.....	31

3	Marco de Trabajo para Definir la Perspectiva de Recursos.....	35
3.1	Estructura del Marco de Trabajo	35
3.2.	Metamodelo de la Perspectiva de Recursos	38
3.3.	Representación de Plataformas de Implementación de la Perspectiva de Recursos	41
3.3.1.	Metamodelo de Implementación de la Perspectiva de Recursos	42
3.3.2.	Modelo de Implementación de la Perspectiva de Recursos	46
3.4.	Extensiones de BPMN para Soportar la Perspectiva de Recursos	50
3.4.1.	Método para la Definición de Extensiones a BPMN 2.0	50
3.4.2.	Extensión de Estructura de Recursos y Autorización Estática.....	53
3.4.3.	Extensión de Distribución de Trabajo y Autorización Dinámica	56
3.5.	Conclusiones.....	59
4	Método para Definir la Perspectiva de Recursos en el Desarrollo de SIOPs	63
4.1	Etapas del Método	63
4.2	Etapa 1: Definir la Lógica del Proceso.....	65
4.3	Etapa 2: Definir los Requerimientos de la Perspectiva de Recursos.....	66
4.3.1	Definir el PI-RSM	67
4.3.2	Definir el PI-EPM	69
4.4	Etapa 3: Seleccionar la Plataforma de Implementación	71
4.5	Etapa 4: Definir la Implementación de la Perspectiva de Recursos	76
4.6	Etapa 5: Verificar y Validar la Implementación de la Perspectiva de Recursos	78
4.7	Etapa 6: Generar Especificaciones Ejecutables.....	81
4.8	Conclusiones.....	82
5	Herramienta para Definir la Perspectiva de Recursos en el Desarrollo de SIOPs.	85
5.1	Estructura y Funcionalidades de la Herramienta.....	85
5.2	Conclusiones.....	93
6	Evaluación	95
6.1	Modelado de Plataformas de Implementación	95
6.1.1	Modelo de Implementación de la Perspectiva de Recursos de YAWL ..	95

6.1.2	Modelo de Implementación de la Perspectiva de Recursos de WS-HumanTask	101
6.2	Aplicación del Método Propuesto en la Definición de la Perspectiva de Recursos en el Desarrollo de SIOPs	105
6.2.1	Caso de Estudio: Proceso de Gestión de Fondos para Obras Menores	105
6.2.2	Caso de Estudio: Proceso de Gestión de Presupuestos de Muebles a Medida	118
6.3	Soporte a los Patrones de Recursos	135
6.4	Conclusiones	140
7	Conclusiones	143
7.1	Contribuciones	143
7.1.1	Marco de Trabajo para Definir la Perspectiva de Recursos	144
7.1.2	Método para Definir la Perspectiva de Recursos en el Desarrollo de SIOPs	148
7.1.3	Herramienta	151
7.2	Limitaciones y Trabajos Futuros	151
8	Anexo I Definición y Validación de Modelos BPMN+X	153
8.1	Definición de un Modelo BPMN+X a partir de un Modelo Conceptual de Extensión	153
8.2	Verificación de Modelos BPMN+X	156
9	Anexo II Reglas de Verificación de PS-RSM y PS-EPM	159
10	Bibliografía	170

2 Índice de Tablas

Tabla 1. Notación gráfica de los Modelos de Estructura de Recursos.	54
Tabla 2. Propiedades de Evaluación de Estructura de Recursos y Autorización Estática.	72
Tabla 3. Tabla de Resultados de Evaluación de Estructura de Recursos y Autorización Estática.	73
Tabla 4. Propiedades de Evaluación de Distribución de Trabajo y Autorización Dinámica.....	73
Tabla 5. Tabla de Resultados de Evaluación de Distribución de Trabajo y Autorización Dinámica.....	74
Tabla 6. Propiedades de Evaluación de Entidades de Implementación a los Aspectos Estructura de Recursos y Autorización Estática.....	75
Tabla 7. Propiedades de Evaluación para Aspectos Distribución de Trabajo y Autorización Dinámica.....	75
Tabla 8. Resumen de Implementaciones Candidatas de Estructura de Recursos provistas por Bonita.....	111
Tabla 9. Resumen de Implementaciones Candidatas de Distribución de Trabajo Provistas por Bonita.	112
Tabla 10. Resumen de Implementaciones Candidatas de Estructura de Recursos provistas por YAWL.	124
Tabla 11. Resumen de Implementaciones Candidatas de Distribución de Trabajo Provistas por YAWL.	126
Tabla 12. Resumen de Implementaciones Candidatas de Estructura de Recursos provistas por Bonita.....	127
Tabla 13. Resumen de Implementaciones Candidatas de Distribución de Trabajo Provistas por Bonita.	128
Tabla 14. Patrones de Recursos Soportados por el Marco de Trabajo.....	139
Tabla 15. Posibles representaciones de una propiedad de un modelo conceptual en un modelo BPMN+X.....	154
Tabla 16. Posibles representaciones de una generalización de un modelo conceptual en un modelo BPMN+X.....	156

3 Índice de Figuras

Figura 1. Perspectiva de recursos en el desarrollo de SIOPs basados en WfMS.	12
Figura 2. Metamodelo de Recursos con Enfoque Tecnológico.....	12
Figura 3. Metamodelo de Recursos con Enfoque Organizacional.	13
Figura 4. Metamodelo de Asignación de Recursos de BPMN 2.0.....	17
Figura 5. Modelo de Referencia de WfMC	21
Figura 6. Modelo de Referencia propuesto por Pesic y van der Aals	22
Figura 7. Arquitectura de Infraestructura WS-HumanTask	23
Figura 8. Metamodelo de Recursos de YAWL	24
Figura 9. Metamodelo de Distribución de Trabajo de YAWL.....	25
Figura 10. Elementos de Asignación de Recursos de BPEL4People/WS-HumanTask	28
Figura 11. Metamodelos y Modelos del Marco de Trabajo Propuesto.	37
Figura 12. RPMeta – Aspectos Estructura de Recursos y Autorización Estática	39
Figura 13. RPMeta – Aspectos Distribución de Trabajo y Autorización Dinámica.	40
Figura 14. RPIimplMeta con los elementos de la Estructura de Recursos y Autorización Estática	43
Figura 15. RPMeta haciendo referencia a los elementos del RPIimplMeta (Estructura de Recursos y Autorización Estática).....	44
Figura 16. RPIimplMeta con los elementos de Distribución de Trabajo y Autorización Dinámica.....	45
Figura 17. RPIimplMeta haciendo referencia a los elementos del RPIimplMeta (Distribución de Trabajo y Autorización Dinámica).....	46
Figura 18. RPIM de los aspectos Estructura de Recursos y Autorización Estática del WfMS Bonita.....	47
Figura 19. RPIM de Distribución de Trabajo y Autorización Dinámica del WfMS Bonita.	49
Figura 20. Metamodelo del mecanismo de extensión de BPMN 2.0	51
Figura 21. Método para definir extensiones de BPMN 2.0.....	52
Figura 22. Perfil UML BPMN+X.	53
Figura 23. Extensión de Estructura de Recursos y Autorización Estática	55
Figura 24. Notación propuesta para la visualización de RSMs.....	56
Figura 25. Extensión de Distribución de Trabajo y Autorización Dinámica.	57
Figura 26. Ejemplo de EPM.	59
Figura 27. Etapas del método propuesto.	63
Figura 28. Arquitectura de la Herramienta basada en Oryx.....	86
Figura 29. Editor de Modelos de Lógica de Procesos de Negocio.....	86
Figura 30. Editor de Modelos de Estructura de Recursos.	87
Figura 31. Asistentes para Inicializar PI-EPMs.	88
Figura 32. Asistente para la Definición del Aspecto Distribución de Trabajo.....	88
Figura 33. Editor de Modelos de Proceso Extendidos Independientes de Plataforma..	89

Figura 34. Editor de Modelos de Implementacion de Perspectiva de Recursos (RPIM).	90
Figura 35. Selector de Modelos de Implementación de Perspectiva de Recursos.	90
Figura 36. Diálogo de Implementaciones Candidatas de RSMs.	91
Figura 37. Diálogo de Implementaciones Candidatas de EPMs.	91
Figura 38. Implementación de Privilegios de Recursos en un PS-RSM.	92
Figura 39. Diálogos de resultados de validación y verificación.....	92
Figura 40. RPIM de Aspectos Estructura de Recursos y Autorización Estática de YAWL.	97
Figura 41. RPIM de Aspectos Distribución de Trabajo y Autorización Dinámica de YAWL.	100
Figura 42. Modelo de Implementación de Perspectiva de Recursos de WS-HumanTask	104
Figura 43. BPLM Proceso de Gestión de Fondos para Obras Menores.	106
Figura 44. PI-RSM de los Recursos Involucrados en el Proceso de Gestión de Fondos.	107
Figura 45. PI-EPM del Proceso de Gestión de Fondos.	109
Figura 46. PS-RSM del Proceso de Gestión de Fondos.	114
Figura 47. PS-EPM del Proceso de Gestión de Fondos.	115
Figura 48. Proceso de Solicitud de Fondos Implementado en el WfMS Bonita.	117
Figura 49. BPLM de Proceso de Gestión de Presupuestos de Muebles a Medida.....	120
Figura 50. PI-RSM Proceso de Gestión de Presupuestos de Muebles.	121
Figura 51. PI-EPM del Proceso de Gestión de Presupuestos de Muebles a Medida..	122
Figura 52. PS-RSM del Proceso de Gestión de Presupuestos de Muebles a Medida.	130
Figura 53. PS-EPM del Proceso de Gestión de Presupuestos de Muebles a Medida..	131
Figura 54. Esquema de Recursos de YAWL del Proceso de Gestión de Presupuestos.	134
Figura 55. Especificación Ejecutable de Workflow del Proceso de Gestión de Presupuestos.	134

4 Resumen

Los Sistemas de Información Orientados a Procesos (SIOPs) son sistemas de software que, con base en modelos de procesos de negocio, gestionan y ejecutan operaciones que involucran personas, aplicaciones y/o fuentes de información. Los Sistemas de Gestión de Workflow (WfMSs) son sistemas de información de propósito general utilizados para desarrollar SIOPs. Los mismos permiten automatizar y ejecutar procesos de negocio que involucran recursos humanos y aplicaciones. La *perspectiva de recursos* se refiere a la parte de la definición de un proceso de negocio en la que se representan los recursos humanos involucrados en la ejecución del proceso, la distribución del trabajo (tareas a realizar) a los mismos y la gestión de sus interacciones con el SIOP.

Los WfMSs proveen un soporte dispar a la *perspectiva de recursos*. Emplean conceptos diferentes para su representación y proveen soporte a requerimientos diferentes. Por otra parte, el soporte provisto por los lenguajes de modelado de procesos existentes para definir esta perspectiva es limitado, no permitiendo expresar en modelos de procesos aspectos de la misma que tienen impacto en el desempeño de los procesos. En este contexto, se dificulta la especificación de los requerimientos de esta perspectiva, la selección de un WfMS apropiado para soportar estos requerimientos y validar que las soluciones tecnológicas resultantes sean consistentes con los requerimientos de la organización.

El objetivo de esta tesis es proveer soporte a la definición, implementación, verificación y validación de los requerimientos de la perspectiva de recursos en el desarrollo de SIOPs basados en WfMSs. Las principales contribuciones de esta tesis son: (1) un marco de trabajo basado en BPMN para la definición de la perspectiva de recursos en modelos de procesos y la representación de las entidades provistas por los WfMSs para implementar la misma, el cual se compone de metamodelos y de extensiones a BPMN; (2) un método de desarrollo dirigido por modelos, que hace uso del marco de trabajo propuesto, para definir la perspectiva de recursos durante el desarrollo de SIOPs; y (3) una herramienta que soporta el marco de trabajo y el método propuestos.

A través del uso del marco de trabajo y el método propuestos, es posible expresar los requerimientos de la perspectiva de recursos en modelos conceptuales de procesos de negocio, seleccionar un WfMS adecuado como plataforma de implementación del SIOP, definir la implementación de dichos requerimientos en modelos específicos de la plataforma y generar el código de las especificaciones ejecutables de procesos. De este modo se incorpora la definición de la perspectiva de recursos en las distintas etapas del desarrollo de SIOPs manteniendo la consistencia entre los modelos conceptuales de la misma y su implementación en las correspondientes especificaciones ejecutables de procesos.

5 Prólogo

La difusión de los enfoques de Gestión de Procesos de Negocio y Automatización de Procesos de Negocio tuvo como consecuencia un creciente interés de las organizaciones por crear representaciones explícitas de sus procesos, las cuales puedan ser empleadas para construir sistemas de información que soporten sus operaciones. Estos sistemas se conocen como Sistemas de Información Orientados a Procesos (SIOPs). Los SIOPs son sistemas de software que, con base en modelos de procesos de negocio, gestionan y ejecutan operaciones que involucran personas, aplicaciones y/o fuentes de información (Dumas, et al., 2005). Los Sistemas de Gestión de Workflow (WfMSs) son sistemas de información de propósito general utilizados para implementar SIOPs.

Durante las últimas dos décadas surgieron numerosos lenguajes para definir procesos de negocio en distintos niveles de abstracción. Los lenguajes de modelado de procesos permiten definir modelos conceptuales de procesos cuyo propósito es expresar el modo en que la organización gestiona o gestionará sus operaciones. Los lenguajes de ejecución de procesos permiten definir especificaciones de procesos cuyo propósito es la implementación de un SIOP, típicamente sobre la base de un Sistema de Gestión de Workflow (WfMS), para dar soporte a la ejecución de los procesos. Si bien la capacidad de estos lenguajes para representar la lógica de los procesos de negocio ha alcanzado un cierto grado de madurez, la capacidad de los mismos para expresar las relaciones entre las tareas de los procesos y los miembros de las organizaciones, lo cual se conoce como perspectiva de recursos, se encuentra menos estudiada.

La perspectiva de recursos define los recursos humanos responsables de las tareas de un proceso y sus interacciones con un SIOP para gestionar y ejecutar las mismas. La definición de esta perspectiva es relevante desde los puntos de vista organizacional, operativo y tecnológico. Desde el punto de vista organizacional, especifica la implementación en un SIOP de la asignación de responsabilidades y la jerarquía de control definidas en la estructura de la organización. Desde el punto de vista operativo, controla la asignación a los recursos humanos del trabajo a realizar en las tareas de los procesos. Desde el punto de vista tecnológico, define los esquemas de interacción entre los recursos humanos y el SIOP para completar el trabajo de los procesos.

La implementación de SIOPs en las organizaciones permite reducir los tiempos de transporte y espera entre las actividades de los procesos, y controlar la asignación de trabajo a los recursos humanos. Esto contribuye a mejorar el desempeño de los procesos de negocio, especialmente en procesos administrativos y de servicios conocidos como procesos de workflow, en los que alrededor del 60% del costo de ejecución corresponde a horas de trabajo de recursos humanos. Los SIOPs requieren, por lo tanto, de descripciones detalladas de la perspectiva de recursos, esto es de los recursos humanos, la distribución de

trabajo entre los mismos, y el modo en que éstos pueden interactuar con el entorno de ejecución para organizar y cumplimentar el trabajo que les es asignado.

Sin embargo la definición de la perspectiva de recursos durante el desarrollo de SIOPs presenta una serie de dificultades:

- Las capacidades limitadas de los lenguajes de modelado de procesos de negocio existentes para representar los diferentes tipos de requerimientos de la perspectiva de recursos. Este limitado soporte hace difícil representar la perspectiva de recursos de forma comprensible para los expertos de negocio de la organización, y también dificulta la comunicación de los requerimientos de esta perspectiva a los desarrolladores de SIOPs.
- El soporte dispar provisto por los WfMSs actuales a la implementación de los diferentes tipos de requerimientos de la perspectiva de recursos, lo cual dificulta la definición de esta perspectiva durante el desarrollo de SIOPs, conduciendo a procesos de desarrollo laboriosos y propensos a errores.
- La falta de metodologías que guíen a expertos de negocio y desarrolladores en la definición de esta perspectiva durante el desarrollo de SIOPs. Esto constituye un obstáculo para la definición de SIOPs que distribuyan el trabajo de las tareas de los procesos de negocio de manera consistente con los lineamientos definidos desde los puntos de vista organizacional y operacional. Lo anterior lleva a un pobre soporte de los SIOPs a los requerimientos de la perspectiva de recursos, lo cual impacta de forma negativa en el desempeño de los procesos implementados.

El objetivo de esta tesis es proveer métodos y herramientas para dar soporte a la definición de los requerimientos relacionados con la perspectiva de recursos durante el diseño de procesos de negocio y el desarrollo de SIOPs. En este sentido, las contribuciones de esta tesis consisten en un marco de trabajo, un método y una herramienta para la definición de la perspectiva de recursos en el desarrollo de SIOPs.

El marco de trabajo propuesto se basa en el metamodelo de BPMN 2.0, del cual se toman sus elementos generales usados para definir la perspectiva de recursos. El mismo está compuesto por metamodelos y extensiones a BPMN. El Metamodelo de la Perspectiva de Recursos define el modelo conceptual adoptado en esta tesis para representar tres aspectos de la perspectiva de recursos: Estructura de Recursos, Distribución de Trabajo y Autorización (Estática y Dinámica). Un segundo metamodelo propuesto es el denominado Metamodelo de Implementación de la Perspectiva de Recursos, el cual provee los elementos para describir las entidades provistas por los WfMSs para la implementación de esta perspectiva. Estas entidades se describen en Modelos de Implementación de la Perspectiva de Recursos (RPIM). Con base en estos metamodelos se desarrollaron también

dos extensiones al lenguaje de modelado BPMN 2.0: la Extensión de Estructura de Recursos y Autorización Estática, y la Extensión de Distribución de Trabajo y Autorización Dinámica. Por medio de la aplicación de estas extensiones, es posible definir dos tipos de artefactos de modelado para un proceso: un Modelo de Estructura de Recursos y un Modelo de Proceso Extendido. Estos artefactos permiten expresar los aspectos de la perspectiva de recursos de un proceso, como así también cómo implementarlos a través de la importación y referencia de los elementos del RPIM que representa la plataforma seleccionada.

La principal ventaja del marco de trabajo es que provee artefactos de alto nivel de abstracción para definir tanto modelos conceptuales de la perspectiva de recursos como modelos específicos de la plataforma de implementación, lo cual permite a analistas de negocio y desarrolladores de SIOPs tomar decisiones en conjunto acerca de los requerimientos asociados con esta perspectiva y simplifica la tarea de implementarlos por medio de un WfMSs adecuado. Además, al estar basado en un lenguaje ampliamente aceptado como BPMN 2.0, se reduce el tiempo requerido por las personas familiarizadas con este lenguaje para comprender los nuevos conceptos introducidos por el marco de trabajo, a la vez que se incrementa la posibilidad de reutilizar herramientas que soportan el lenguaje BPMN.

El método propuesto hace uso de los artefactos del marco de trabajo para proveer un enfoque sistemático que permite definir la perspectiva de recursos durante el desarrollo de SIOPs, con base en los principios del desarrollo dirigido por modelos. El método provee un conjunto de lineamientos y técnicas que guían a los usuarios durante la definición de los requerimientos de la perspectiva de recursos en modelos de procesos BPMN, la selección de un WfMS adecuado como plataforma de implementación, la definición en modelos de procesos de la implementación de los requerimientos por medio del WfMS seleccionado, la verificación y la validación de la implementación resultante, y la generación de las especificaciones ejecutables requeridas por la plataforma de implementación.

Como beneficios, el método provee lineamientos respecto del orden en que los metamodelos y extensiones a BPMN provistos por el marco de trabajo deben ser empleados durante la definición de cada uno de los modelos a ser creados por analistas de negocio o desarrolladores. Por otra parte, provee un conjunto de técnicas automatizables para: (1) evaluar el soporte provisto por un WfMS a los requerimientos de perspectiva de recursos definidos en modelos conceptuales de procesos, (2) verificar la adecuación de las implementaciones de los requerimientos a las restricciones impuestas por el WfMS seleccionado, (3) validar la consistencia entre los requerimientos definidos en modelos conceptuales y en modelos específicos de la plataforma, y (4) generar especificaciones ejecutables de la perspectiva de recursos. Esto permite reducir el tiempo requerido para estar en condiciones de generar las especificaciones ejecutables requeridas por el WfMS seleccionado. Y fundamentalmente, el método posibilita generar soluciones tecnológicas

que contienen definiciones de la perspectiva de recursos consistentes con aquellas de los modelos conceptuales que expresan la solución organizacional y operacional.

La herramienta desarrollada para dar soporte al marco de trabajo y método propuesto provee editores que permiten definir los distintos tipos de artefactos de modelado soportados por el marco de trabajo, así como asistentes que permiten almacenar y recuperar dichos artefactos de un repositorio de modelos e integrarlos para cumplimentar las distintas etapas del método propuesto. De este modo, la herramienta posibilita y facilita la aplicación del marco de trabajo y el método propuestos.

Los beneficios provistos por la herramienta consisten en el soporte a la automatización de: las tareas de inicialización de modelos, las técnicas de cálculo de tasas de soporte de un WfMS a los requerimientos de la perspectiva de recursos definidos para un proceso, la verificación y la validación de las implementaciones; y la generación de especificaciones ejecutables. La herramienta es extensible para soportar la generación de especificaciones ejecutables requeridas por WfMSs existentes y nuevos, por medio de la incorporación en el repositorio de modelos de RPIMs adicionales y de transformaciones modelo a código asociadas como complementos del lado del servidor.

Los resultados del trabajo realizado en esta tesis han sido divulgados a través de las siguientes publicaciones:

- Stroppi, L. J., Omar, C., & Villarreal, P. D. (2015). Defining the Resource Perspective in the Development of Processes-Aware Information Systems. *Information and Software Technology*, ELSEVIER, Volume 59, Pages 86-108, ISSN 0950-5849. Impact Factor: 1,328
- Stroppi, L. J., Villarreal, P. D., & Chiotti, O. (2013). Extending the WS-HumanTask Architecture to Support the Resource Perspective of BPEL Processes. *CLEI ELECTRONIC JOURNAL*, Volume 16, Number 1, Paper 3. ISSN 0717-5000.
- Stroppi, L. J., Chiotti, O., & Villarreal, P. D. (2012). Extended Resource Perspective Support for BPMN and BPEL. *Iberoamerican Conference on Software Engineering*. ISBN 978-987-1635-46-7.
- Stroppi, L. J., Chiotti, O., & Villarreal, P. D. (2011). Extending BPMN 2.0: Method and Tool Support. *Lecture Notes in Business Information Processing*. Vol. 95, pp 59-73. ISSN 1865-1348.
- Stroppi, L. J., Chiotti, O., & Villarreal, P. D. (2011). A BPMN 2.0 Extension to Define the Resource Perspective of Business Process Models. *Iberoamerican Conference on Software Engineering*. ISBN 978-85-8006-032-4.

- Stroppi, L. J., Chiotti, O., & Villarreal, P. D. (2009). Un Modelo de Referencia para Definir la Perspectiva Organizacional de Modelos de Workflows. *Congreso Argentino de Ciencias de la Computación*. ISBN 978-897-24068-3-9.

6. Agradecimientos

Quiero expresar mi gratitud a todos aquellos que me apoyaron durante estos años e hicieron posible que alcanzara esta meta tan importante.

Agradezco a mi director, el Dr. Pablo Villarreal, por su generosidad en brindar su tiempo y sus conocimientos para orientarme en mis actividades de investigación, permitiéndome desarrollarlas con total libertad. Quiero agradecerle también por sus valiosos y oportunos consejos tanto en el plano académico como en el personal.

Agradezco también a mi co-director, el Dr. Omar Chiotti, por la claridad de sus conceptos y su permanente predisposición para colaborar en mis actividades de investigación.

Quiero expresar mi más profunda gratitud a mi familia por darme su amor, aliento y contención; sin los cuales no hubiese podido realizar esta tesis. Agradezco a Marisú, mi esposa, y a Clara, mi hija, quienes dan sentido a mi vida. A Haydée, mi hermana y amiga incondicional. A Luis y Graciela, mis padres, quienes son mi *modelo en todos los aspectos* de la vida.

A todos los integrantes del Centro de Investigación y Desarrollo de Sistemas de Información (CIDISI) por el compañerismo, por el tiempo y las experiencias compartidas y por hacer del trabajo un lugar cálido y agradable.

Al Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) por proporcionarme el sustento económico para la realización de esta tesis, y a la Facultad Regional Santa Fe de la Universidad Tecnológica Nacional por la formación y los recursos brindados.

1 Introducción

En este capítulo se describe el contexto en el que se enmarca la tesis detallando los conceptos del dominio de desarrollo de Sistemas de Información Orientados a Procesos (Sección 1.1). Se describen la motivación y los problemas a resolver (Sección 1.2), se definen los objetivos (Sección 1.3) y se explicita el método de investigación seguido (Sección 1.4). Finalmente, se presentan las principales contribuciones (Sección 1.5) y la organización de la tesis (Sección 1.6).

1.1 Contexto

La Gestión de Procesos de Negocio es un enfoque de gestión organizacional que implica la identificación, diseño, ejecución, documentación, medición, monitoreo y control de procesos de negocio, con el fin de alcanzar las metas de negocio de una organización (CBOOK, 2009). Si bien este enfoque de gestión podría aplicarse sin el uso de Tecnologías de la Información, las organizaciones pueden obtener beneficios adicionales si los esfuerzos que realizan para organizar el trabajo que llevan a cabo son tomados como un punto de partida para el desarrollo de plataformas basadas en Tecnologías de la Información que lo soporten (Ter Hofstede, 2010). La Automatización de Procesos de Negocio agrupa distintos enfoques y metodologías para el diseño e implementación de estas plataformas.

Un proceso de negocio define un conjunto de tareas lógicamente relacionadas con el propósito de obtener un resultado de valor para la organización. Los procesos de negocio pueden clasificarse de acuerdo con la naturaleza de su resultado en *procesos de manufactura* o *procesos de workflow* (Reijers, 2003). Los *procesos de manufactura* tienen como propósito la producción de un bien material, mientras que los *procesos de workflow* producen como resultado información. Los procesos administrativos, de gestión de información y de servicios pertenecen a esta segunda clase.

Los Sistemas de Información Orientados a Procesos (SIOPs) son sistemas de software que soportan la ejecución de *procesos de workflow*. Con base en modelos de procesos, estos sistemas gestionan y ejecutan operaciones que involucran personas, aplicaciones y/o fuentes de información (Dumas, et al., 2005). El desarrollo, despliegue, ejecución y mantenimiento de estos sistemas involucra personas con distintas formaciones e intereses. Las personas involucradas incluyen los directivos y/o gerentes de la organización, los analistas de negocio, los desarrolladores del sistema y los usuarios. Estos últimos son quienes finalmente ejecutan las operaciones de la organización con el soporte del SIOP.

El desarrollo de SIOPs siguiendo un enfoque “top-down” requiere construir primero los modelos conceptuales de los procesos a ser soportados y luego sus especificaciones ejecutables. Los *modelos conceptuales de procesos* son representaciones de alto nivel de abstracción construidas generalmente por analistas de negocio con base en los requerimientos relevados y los lineamientos provistos por los directivos de la organización.

El propósito de estos modelos es permitir la comunicación y la toma de decisiones durante la definición de los procesos. Dichos modelos forman parte de una *solución de negocio u organizacional*, la cual describe la forma en que una organización gestiona o gestionará sus operaciones con independencia de cualquier tecnología de implementación. Estos modelos generalmente son definidos empleando lenguajes como BPMN (OMG, 2013), Diagramas de Actividad UML (OMG, 2009) o EPC (van der Aalst, 1999).

Las *especificaciones ejecutables de procesos* (también conocidas como *especificaciones de workflows*) son representaciones que pueden ser interpretadas, instanciadas y gestionadas por una determinada plataforma de implementación basada en Tecnologías de Información. Los Sistemas de Gestión de Workflow (WfMSs por sus siglas en inglés) son sistemas de información que se utilizan como plataformas de implementación de SIOPs. Los WfMSs consisten en un conjunto de componentes reutilizables, los cuales son configurados mediante las *especificaciones ejecutables de procesos*. Los WfMSs fueron gradualmente integrados con Sistemas de Integración de Aplicaciones Empresariales y Sistemas de Gestión de Reglas de Negocio para conformar lo que actualmente se conoce como Sistemas de Gestión de Procesos de Negocio (Harmon, 2003). Las *especificaciones ejecutables de procesos* constituyen las *soluciones tecnológicas* de los SIOPs. Dichas especificaciones son definidas empleando lenguajes de ejecución de procesos, los cuales en muchos casos son específicos del WfMS seleccionado, o estándares tales como WS-BPEL (Jordan, et al., 2007).

Los modelos conceptuales y especificaciones ejecutables de procesos son definidos en términos de distintas perspectivas: la perspectiva de control, la perspectiva de datos, la perspectiva operacional y la perspectiva de recursos (van Der Aalst, et al., 2003). La *perspectiva de control* describe las actividades de un proceso y el orden de ejecución de las mismas a través de constructores de flujo de control, como por ejemplo secuencia, flujos alternativos, paralelismo o sincronización. La *perspectiva de datos* define los documentos u objetos (información) que son pasados entre las actividades y las variables de los procesos, las cuales definen las precondiciones y poscondiciones de la ejecución de las actividades. La *perspectiva operacional* describe las actividades de un proceso que son ejecutadas por aplicaciones específicas. La *perspectiva de recursos* define los recursos humanos que tendrán a cargo la ejecución de las tareas del proceso y sus interacciones con el sistema para la ejecución de dichas tareas.

Los WfMSs proveen soporte a la *perspectiva de recursos* por medio de la presentación y asignación de ítems de trabajo a los recursos humanos involucrados en la ejecución del proceso. Un ítem de trabajo es una representación del trabajo a ser realizado por una persona en el contexto de una tarea en una instancia de un proceso (Hollingsworth & Hampshire, 1993). Este concepto permite separar la gestión de las interacciones con los recursos humanos de la gestión de las instancias de las tareas de los procesos.

Para maximizar los beneficios que una organización puede obtener de implementar un SIOP, es necesario que la forma de gestionar las operaciones sea consistente con el modo de asignar los ítems de trabajo a los recursos humanos (Abbott & Sunil, 1994). Para esto, la definición de la perspectiva de recursos en las *soluciones tecnológicas* debe ser consistente con los requerimientos definidos en las *soluciones de negocio*. Estos requerimientos incluyen directrices respecto del modo en que el trabajo de las tareas de los procesos debe ser asignado a los recursos humanos desde los puntos de vista operativo y organizacional. Desde el punto de vista operativo, el modo en que el trabajo es dado a conocer y asignado a los recursos humanos para su ejecución debe ser consistente con los modelos de planificación de los recursos (Reijers, 2003). Desde el punto de vista organizacional, la asignación del trabajo a los recursos humanos debe estar en línea con la estructura de la organización y la definición de responsabilidades sobre los procesos. Dado que la perspectiva de recursos define relaciones entre los modelos de proceso y los modelos organizacionales, en diversos trabajos se la refiere también como perspectiva organizacional (Zur Muehlen, 2004).

1.2 Motivación

El desarrollo de SIOPs basados en WfMSs puede contribuir a mejorar el desempeño de los procesos de negocio de la organización a través de la reducción de tiempos de transporte y espera entre las actividades de los procesos, y del control de la asignación de trabajo a los recursos humanos (Reijers & van der Aalst, 2005). Los procesos de workflow se caracterizan por un alto grado de participación de recursos humanos. Alrededor del 60% del costo de ejecución de estos procesos corresponde a horas de trabajo de recursos humanos (Dumas, et al., 2005). Los SIOPs requieren, por lo tanto, de descripciones detalladas de los recursos humanos, la distribución de trabajo entre los mismos, y el modo en que éstos pueden interactuar con el entorno de ejecución para organizar y cumplimentar el trabajo que les es asignado.

La tarea de definir la perspectiva de recursos durante el desarrollo de SIOPs en forma consistente con los requerimientos de las organizaciones presenta una serie de dificultades. Por un lado, los WfMSs se basan en conceptos diferentes y frecuentemente emplean terminologías completamente distintas (Pesic & van der Aalst, 2007). Los patrones de recursos de workflow (Russell, et al., 2005) identifican un conjunto de estrategias de distribución de ítems de trabajo entre recursos humanos soportadas recurrentemente por los WfMSs. Evaluaciones del soporte provisto a la perspectiva de recursos por diferentes WfMSs, basadas en estos patrones (Wohed, et al., 2009) (Russell & van der Aalst, 2007), muestran resultados dispares. Ninguno de los WfMSs evaluados mostró un soporte completo a los patrones, además cada WfMS evaluado soportó un conjunto diferente de patrones. Por otro lado, las evaluaciones realizadas con base en estos patrones a lenguajes de modelado de procesos de negocio de alto nivel de abstracción, tales como BPMN o

Diagramas de Actividad UML (Wohed, et al., 2006), (Russell, et al., 2006) muestran un limitado soporte de estos lenguajes a la representación de dicha perspectiva.

Este limitado soporte hace difícil representar la perspectiva de recursos de forma comprensible para los expertos de negocio de la organización. Sumado esto al hecho que los distintos WfMSs se basan en conceptos diferentes, la tarea de implementar los requerimientos definidos en términos de un WfMS suele ser laboriosa y propensa a errores. Esto produce una brecha comunicacional entre los expertos de negocio y los profesionales de sistemas involucrados en el desarrollo de los SIOPs.

Otra consecuencia de este limitado soporte provisto por los lenguajes de modelado es la omisión de la definición de la perspectiva de los recursos en las primeras etapas de diseño de los procesos y desarrollo de los SIOPs, lo cual implica que se relega la definición de los requerimientos de esta perspectiva a una etapa tardía del desarrollo. Esto es, se definen directamente durante la etapa de implementación de los SIOPs, en la cual se definen las especificaciones ejecutables de procesos para el WfMS seleccionado. La selección del WfMS previa a la definición de los requerimientos de la perspectiva de recursos incrementa el riesgo de elegir una plataforma que no se adecua a las necesidades de la organización (Weske, 2012). Esto constituye un obstáculo para la definición de SIOPs que distribuyan el trabajo de las tareas de los procesos de negocio de manera consistente con los lineamientos definidos desde los puntos de vista organizacional y operacional.

Lo anterior también da lugar a una definición demasiado simplista de la perspectiva de los recursos, lo cual deriva en el desarrollo de SIOPs que pueden resultar inflexibles o liberales (Pesic & van der Aalst, 2005), (Kumar, et al., 2002). Los SIOPs inflexibles pueden impactar negativamente en el desempeño de la organización debido a que, cuando los recursos humanos asignados a una determinada tarea están sobrecargados o ausentes, bloquean la ejecución del proceso hasta que las personas asignadas puedan completar dicha tarea, aún cuando estén disponibles otras personas capaces de cumplimentarla. Por otra parte, los SIOPs liberales habilitan a más personas de las necesarias a acceder y realizar las tareas, lo cual trae aparejado frecuentes problemas de seguridad.

Del análisis previo se desprende la necesidad de contar con métodos de diseño y desarrollo, y herramientas de software que posibiliten: (1) la definición de modelos de procesos de negocio de alto nivel de abstracción, los cuales contemplen las definiciones de las estructuras organizacionales de recursos humanos y de las políticas de distribución de trabajo de la perspectiva de los recursos; (2) la refinación de estos modelos hasta obtener especificaciones ejecutables de procesos alineadas con la solución definida a nivel de negocio u organizacional.

1.3 Objetivos

El objetivo de la presente tesis es desarrollar métodos y herramientas para dar soporte a la definición de los requerimientos de la perspectiva de recursos durante el diseño de procesos de negocio y el desarrollo de SIOPs. El propósito es proveer a las organizaciones de métodos y herramientas que les permitan: obtener especificaciones ejecutables de procesos alineadas y consistentes con la solución definida a nivel de negocio u organizacional; mejorar la eficiencia de los procesos a ser implementados en SIOPs a través de una asignación adecuada de las tareas de los procesos a los recursos humanos (personas), de acuerdo a la estructura de recursos de la organización; guiar la definición, verificación, validación e implementación de la perspectiva de recursos en las etapas del desarrollo de SIOPs; y reducir el costo y esfuerzo necesarios para adoptar e implementar soluciones tecnológicas basadas en SIOPs que contemplen las definiciones de la perspectiva de recursos.

1.3.1 Objetivos Específicos

Los objetivos específicos contemplados en esta tesis son los siguientes:

- Desarrollar metamodelos y extensiones a lenguajes de modelado que permitan expresar los diferentes tipos de requerimientos de la perspectiva de recursos en modelos de procesos (artefactos), sobre la base de lenguajes de modelado de procesos ampliamente aceptados. El propósito es crear modelos de procesos que puedan ser utilizados en las diferentes fases del desarrollo de SIOPs, para entender, analizar e implementar la perspectiva de recursos.
- Definir un método que haga uso de los principios y beneficios del Desarrollo Dirigido por Modelos, con el propósito de proveer un enfoque sistemático para:
 - La definición de los requerimientos de la perspectiva de recursos en modelos conceptuales de procesos de negocio, que representen la solución de negocio u organizacional, con independencia de la plataforma de implementación.
 - La evaluación y selección de plataformas de implementación (WfMSs) adecuadas para dar soporte a los requerimientos de la perspectiva de recursos definidos en los modelos conceptuales de procesos.
 - La representación de dicha perspectiva en modelos de procesos específicos de la plataforma de implementación seleccionada.
 - La validación y verificación de las soluciones tecnológicas (modelos específicos de la plataforma) generadas respecto de los requerimientos de la perspectiva de recursos definidos en los modelos de procesos y las restricciones impuestas por la plataforma de implementación seleccionada.

- La generación automática de especificaciones ejecutables de procesos, las cuales definen la implementación de los requerimientos de la perspectiva de recursos en un SIOP con base en la plataforma seleccionada.
- Desarrollar una herramienta basada en un repositorio de modelos que implemente e integre los metamodelos y el método propuestos, para dar soporte a la definición de la perspectiva de recursos en modelos de proceso, su almacenamiento, evolución y la generación automática de especificaciones ejecutables de los mismos.
- Validar y evaluar las propuestas anteriores a través del desarrollo de casos de estudio y también el análisis de las mismas respecto a patrones de referencia de la perspectiva de recursos de procesos de negocio.

1.4 Método de Investigación

Para el desarrollo de esta tesis se aplicó la metodología de investigación de Ciencia de Diseño propuesta por (Peffer, et al., 2007). Esta metodología fue seleccionada dado que provee un marco de trabajo ampliamente aceptado para llevar a cabo investigaciones en sistemas de información, destinadas a resolver problemas a través del diseño y desarrollo de artefactos. Esta metodología también provee lineamientos para la evaluación y presentación de los resultados de la investigación.

Como parte de la aplicación de dicha metodología, esta investigación se inició siguiendo un enfoque centrado en el problema (Peffer, et al., 2007). La misma comenzó con la observación de:

- Las capacidades limitadas de los lenguajes de modelado de procesos de negocio existentes para representar los requerimientos de la perspectiva de recursos.
- El soporte dispar provisto por los WfMSs actuales a la implementación de los requerimientos de la perspectiva de recursos, lo cual dificulta la definición de la misma durante el desarrollo de SIOPs.
- El pobre soporte de los SIOPs a los requerimientos de la perspectiva de recursos, lo cual impacta de forma negativa en el desempeño de los procesos implementados.

A continuación se describen las actividades llevadas a cabo en esta tesis, de acuerdo a las etapas propuestas por la metodología de investigación aplicada.

Etapas 1 – Identificación de Problemas:

- Se analizaron y clasificaron diferentes tipos de requerimientos de la perspectiva de recursos. Se identificaron problemas en la representación de

estos requerimientos durante el desarrollo de SIOPs, tanto a nivel conceptual como de implementación.

Etapa 2 – Definición de los Objetivos:

- Los objetivos para la presente tesis, presentados en la sección 1.2, fueron derivados a partir de los problemas identificados en la definición de la perspectiva de recursos en el desarrollo de SIOPs.

Etapa 3 – Diseño y Desarrollo:

- Análisis de los elementos provistos por el lenguaje de modelado de procesos BPMN 2.0 (OMG, 2013) para la representación de la perspectiva de recursos. Este lenguaje fue seleccionado como base para el desarrollo de los artefactos propuestos en esta tesis por las siguientes razones: (i) es un lenguaje ampliamente aceptado que provee un vocabulario común, orientado tanto a analistas de negocio como a expertos en sistemas y software; (ii) su metamodelo provee los elementos básicos para definir la perspectiva de recursos, lo cual simplifica la comprensión de nuevos elementos que extiendan el lenguaje; (iii) provee un mecanismo de extensión que permite introducir nuevos elementos al lenguaje manteniendo la portabilidad de los modelos resultantes entre distintas herramientas, (iv) es posible reutilizar herramientas existentes para dar soporte a los nuevos artefactos propuestos.
- Desarrollo de un marco de trabajo basado en BPMN para dar soporte a la definición de la perspectiva de recursos de procesos de negocio. Esta actividad incluyó el desarrollo de los siguientes artefactos:
 - Un metamodelo que representa los conceptos empleados para la definición de la perspectiva de recursos en modelos de procesos, el cual complementa a los elementos provistos por el metamodelo de BPMN 2.0 con este propósito.
 - Un metamodelo para representar la implementación de la perspectiva de recursos en distintas plataformas de WfMSs. El mismo fue definido a partir de los requerimientos identificados para esta perspectiva y los elementos provistos por BPMN 2.0.
 - Un conjunto de extensiones al lenguaje BPMN, definidas empleando el mecanismo de extensión provisto por este lenguaje, las cuales permiten definir los requerimientos de la perspectiva de recursos en modelos conceptuales de procesos, y su posterior mapeo a modelos de procesos específicos de la plataforma. Estas extensiones fueron definidas siguiendo un método desarrollado como parte de la presente

investigación, el cual provee un perfil UML llamado BPMN+X, que permite la definición de extensiones BPMN en forma gráfica.

- Desarrollo de un método que hace uso del marco de trabajo propuesto, con el propósito de dar soporte a la definición, verificación, validación e implementación de la perspectiva de recursos durante el desarrollo de SIOPs. El mismo está basado en los principios de la Arquitectura Orientada a Modelos (MDA) (OMG, 2003) y el Desarrollo Dirigido por Modelos (MDD) (Selic, 2003).

Etapa 4 y 5 – Demostración y Evaluación:

- La aplicabilidad y efectividad del marco de trabajo y el método propuestos fueron demostradas por medio del desarrollo de una herramienta que los soportan, como así también su aplicación a casos de estudio, los cuales proveyeron una base para la evaluación de su efectividad para abordar los problemas definidos. Además, la expresividad y completitud del marco de trabajo propuesto para representar los requerimientos de la perspectiva de recursos fue evaluada en términos de los patrones de recursos de workflow.

Etapa 6 – Comunicación:

- El problema y su importancia tratado en esta tesis, los resultados obtenidos, su utilidad y originalidad, y su relevancia científica y profesional fueron comunicados en diferentes trabajos en revistas y congresos, los cuales se mencionan en el prólogo de la tesis.

1.5 Principales Contribuciones

Las principales contribuciones de la presente tesis son las siguientes:

- Un marco de trabajo para la definición de la perspectiva de recursos, el cual abarca tres aspectos identificados para la misma: Estructura de Recursos, Distribución de Trabajo y Autorización. El marco de trabajo se basa en el metamodelo de BPMN 2.0, del cual se toman sus elementos generales usados para definir la perspectiva de recursos, y está compuesto de metamodelos y extensiones a BPMN. Como parte del marco de trabajo se desarrolló:
 - El Metamodelo de la Perspectiva de Recursos (RPMeta), el cual define los elementos necesarios para representar los aspectos identificados de la perspectiva de recursos.

- El Metamodelo de Implementación de la Perspectiva de Recursos (RPIImplMeta). El mismo permite representar los elementos provistos por distintos WfMSs para la implementación de los aspectos definidos de la perspectiva de recursos.
- La extensión a BPMN 2.0 para definir los aspectos Estructura de Recursos y Autorización Estática (RSExt) en Modelos de Estructura de Recursos (RSMs).
- La extensión a BPMN 2.0 para definir los aspectos Distribución de Trabajo y Autorización Dinámica (WDExt) en Modelos de Proceso Extendidos (EPMs).
- Un método basado en los principios del desarrollo dirigido por modelos y en el marco de trabajo propuesto, para la definición de la perspectiva de recursos en el desarrollo de SIOPs, el cual contempla:
 - La definición de los requerimientos de la perspectiva de recursos de procesos de negocio en modelos conceptuales y específicos de la plataforma.
 - La selección del WfMS adecuado para su implementación.
 - La generación automatizada de especificaciones ejecutables de procesos de negocio para el WfMS seleccionado.
 - La validación y la verificación de las implementaciones resultantes.
- Una herramienta basada en el editor y repositorio de modelos de procesos Oryx, la cual da soporte al marco de trabajo y al método propuesto para la definición de la perspectiva de recursos en modelos de procesos de negocio.
- Un repositorio de Modelos de Implementación de Perspectiva de Recursos (RPIM), los cuales son instancias del RPIImplMeta, que representan el soporte provisto por distintos WfMSs tales como YAWL u otros basados en BPEL4People/WS-HumanTask a la perspectiva de recursos.
- Un método basado en los principios del desarrollo dirigido por modelos y una herramienta basada en la plataforma Eclipse, para la definición y generación de extensiones al metamodelo de BPMN 2.0.

1.6 Organización de la Tesis

Esta tesis se estructura de la siguiente manera. El capítulo 2 presenta y discute trabajos previos desarrollados en el dominio abordado. El capítulo 3 presenta el marco de trabajo

desarrollado para definir la perspectiva de recursos. El capítulo 4 presenta un método basado en los principios del Desarrollo Dirigido por Modelos definido con el propósito de guiar la definición de dicha perspectiva durante el desarrollo de SIOPs basados en WfMSs. El capítulo 5 presenta las herramientas desarrolladas para proveer soporte a los artefactos y métodos propuestos. El capítulo 6 presenta un conjunto de casos de estudios desarrollados para mostrar y evaluar la utilidad y aplicabilidad de los artefactos y métodos propuestos. Finalmente, el capítulo 7 presenta las conclusiones y trabajos futuros.

2. Marco Teórico

En este capítulo se presenta el marco teórico y los trabajos relacionados con la tesis. Se describen los conceptos de la perspectiva de recursos (Sección 2.1) y el soporte provisto por el lenguaje BPMN para su representación y las propuestas existentes de extensiones a dicho lenguaje con este propósito (Sección 2.2). Se introducen arquitecturas (Sección 2.3), lenguajes (Sección 2.4) y plataformas (Sección 2.5) para la implementación de esta perspectiva. Se describen los enfoques metodológicos existentes para la definición de la misma durante el desarrollo de SIOPs (Sección 2.6). Finalmente se discute el estado del arte en la definición de esta perspectiva (Sección 2.7).

2.1. Perspectiva de Recursos de Procesos de Negocio

Los SIOPs que soportan la ejecución de *procesos de workflow* generalmente se implementan sobre la base de WfMSs (Figura 1). El desarrollo de un SIOP implica crear representaciones del comportamiento de un proceso (definición del proceso) y definir los recursos involucrados en su ejecución (definición de los recursos). Con base en estas definiciones, el WfMS coordina la ejecución de las tareas del proceso por parte de los recursos. Para que esta coordinación sea posible, la definición del proceso incluye las políticas de distribución de trabajo que especifican los recursos responsables de la ejecución de cada tarea.

En tiempo de ejecución, los WfMSs gestionan las instancias o casos de procesos controlando la activación de sus tareas en el orden especificado por la definición del proceso. Las tareas que involucran personas requieren además de mecanismos que permitan coordinar las interacciones de las mismas con el sistema. Los WfMSs soportan estos mecanismos por medio de la creación de *ítems de trabajo*. Un *ítem de trabajo* representa el trabajo a ser realizado por uno o más recursos en el contexto de una instancia de una tarea, como parte a su vez de una instancia del proceso. Cuando una tarea es activada, un ítem de trabajo es presentado a uno o más recursos para habilitar la ejecución de la tarea. Luego de ello, los recursos asignados finalizan el ítem de trabajo ejecutando operaciones autorizadas a los mismos con ese propósito. Finalmente, el resultado del ítem de trabajo es copiado como salida a la instancia de la tarea.

La perspectiva de recursos se centra en la definición de los principios que regulan el modo en que los WfMS distribuyen los ítems de trabajo entre los recursos y permiten la interacción con los mismos. En las siguientes secciones se describen los distintos elementos y enfoques existentes para su definición.

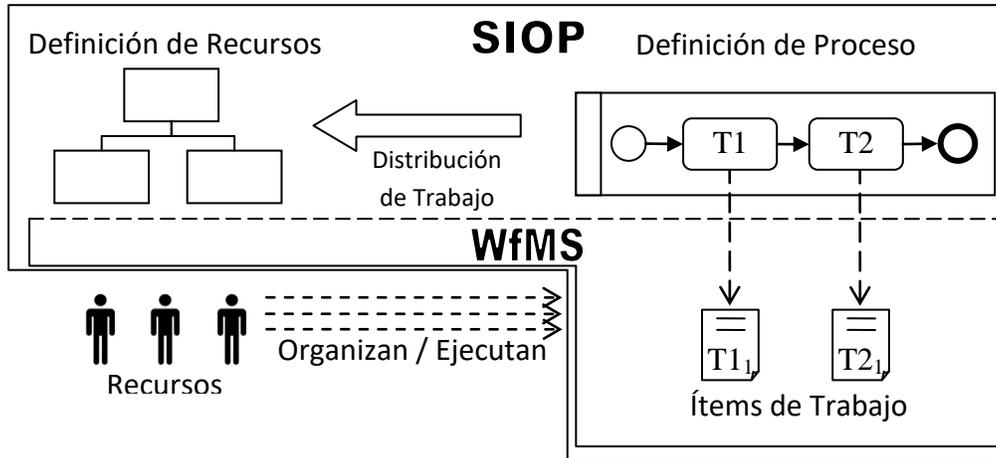


Figura 1. Perspectiva de recursos en el desarrollo de SIOPs basados en WfMS.

2.1.1. Definición de la Estructura de Recursos

A fin de permitir la asignación de trabajo entre los miembros de una organización, un WfMS debe tener acceso a información sobre los mismos. Dicha información está contenida en la estructura de los recursos, la cual es representada en modelos (o esquemas) de recursos. Los modelos de recursos incluyen elementos que permiten representar recursos humanos individuales, sus características y distintas maneras en que los mismos pueden ser agrupados o clasificados. Se pueden distinguir dos enfoques básicos para la definición de los modelos de recursos: el *enfoque tecnológico* y el *enfoque organizacional* (Zur Muehlen, 1999).

El *enfoque tecnológico* aborda el modelado de recursos en términos de un problema de autorización de acceso a usuarios. Los metamodelos que soportan este enfoque (Figura 2) representan a los recursos humanos como *usuarios* del sistema e incorporan los conceptos de *rol* y *grupo* para definir la clasificación de los mismos. Un *rol* define un conjunto de usuarios autorizados a ejecutar una tarea de un proceso. Un *grupo* representa un conjunto de usuarios definido con base en criterios tales como su ubicación o pertenencia a una determinada división organizacional. Un usuario puede desempeñar distintos roles y pertenecer a distintos grupos. De este modo, los modelos definidos con este enfoque proveen información a los WfMS para restringir el conjunto de recursos autorizados a ejecutar el trabajo de una determinada tarea en distintos escenarios.



Figura 2. Metamodelo de Recursos con Enfoque Tecnológico.

Cuando se adopta un *enfoque tecnológico puro*, los roles y conjuntos de recursos son derivados de los modelos de procesos. Esto se conoce como *modelado de recursos dirigido*

por el workflow. Esto permite obtener modelos de recursos simples dado que sólo se representan los recursos implicados en el proceso en consideración. Sin embargo, cambios en los modelos de los procesos pueden afectar el modelo de recursos con facilidad. La introducción de nuevas tareas en un proceso puede requerir la definición de nuevos roles que expresen la autorización de los recursos para ejecutarlas.

El *enfoque organizacional* tiene como propósito capturar de manera fidedigna los niveles de control y los agrupamientos definidos en la estructura de la organización. La Figura 3 muestra un ejemplo de metamodelo de recursos que soporta este enfoque. El mismo emplea el concepto *persona* para abstraer a los recursos humanos de la organización en forma individual. Las características propias de las personas pueden expresarse por medio del concepto *capacidad*. Por otra parte, el concepto *posición* representa un trabajo, función o cargo desempeñado por una o más personas en una organización. Las posiciones definen líneas de reporte que representan niveles de control en una estructura organizacional. Por último, el concepto *unidad organizacional* representa un agrupamiento de posiciones organizacionales con un propósito común. Ejemplos de unidades organizacionales son un departamento, una cátedra o una división. Una unidad organizacional puede pertenecer a otra unidad y agrupar múltiples unidades. Esto permite expresar la desagregación de una organización en unidades menores.

La adopción de un *enfoque organizacional puro* puede resultar en modelos de recursos innecesariamente complejos al capturar detalles de la estructura organizacional que no sean de relevancia para soportar la asignación de trabajo correspondiente a los procesos en consideración. Sin embargo, estos modelos pueden ser mantenidos con mayor independencia de los modelos de procesos, en comparación con aquellos modelos basados en un enfoque tecnológico. En el enfoque organizacional, los modelos resultantes generalmente son más comprensibles para los miembros de la organización y pueden ser reutilizados por múltiples modelos de procesos.

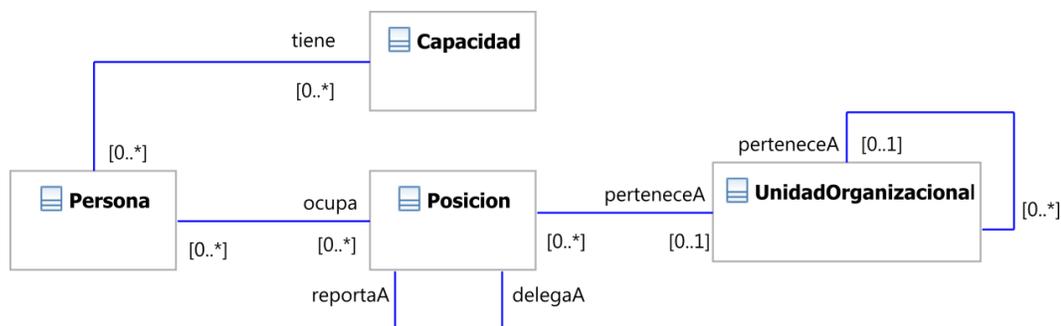


Figura 3. Metamodelo de Recursos con Enfoque Organizacional.

2.1.2. Definición de Políticas de Distribución de Trabajo

La distribución de trabajo se refiere al modo en que los ítems de trabajo de las tareas de los procesos son dados a conocer y asignados a los recursos para su ejecución (Russell, et al., 2005). Hay dos tipos básicos de estrategias de distribución de trabajo: las estrategias de oferta o *pull* y las estrategias de asignación o *push*. Las *estrategias de oferta* dan a conocer los ítems de trabajo a los recursos, los cuales asumen voluntariamente la responsabilidad de su ejecución. Esto implica que el sistema oferta ítems de trabajos y los recursos son quienes seleccionan el ítem, y el momento en que lo ejecutarán. Las *estrategias de asignación* definen mecanismos por los cuales el sistema identifica el o los recursos responsables de la ejecución de los ítems de trabajo correspondientes a una tarea. Esto implica que el sistema asigna el ítem de trabajo a un recurso.

Una *estrategia de distribución de trabajo* incluye la definición de una asignación de recursos que especifica el conjunto de recursos responsables de la ejecución de los ítems de trabajo de una tarea (Awad, et al., 2009). La asignación de recursos generalmente se define como una consulta de información almacenada en un repositorio de recursos o se encuentra disponible en el ámbito de una instancia del proceso. Dentro de las categorías de los patrones de recursos, los patrones de creación (Russell, et al., 2005) capturan requerimientos recurrentes de asignación de recursos encontrados en workflows:

- La *asignación directa* se refiere a la especificación en tiempo de diseño de la identidad de la persona responsable de ejecutar los ítems de trabajo de las instancias de una tarea.
- La *asignación basada en roles* consiste en la especificación en tiempo de diseño de la distribución de los ítems de trabajo de una tarea entre los recursos que comparten un determinado rol definido en un modelo de recursos empleando el enfoque tecnológico descrito en la sección 2.1.
- La *asignación diferida* se presenta en casos donde la identidad del responsable por la ejecución de los ítems de trabajo de una instancia de una tarea se determina en tiempo de ejecución, con base en datos disponibles en el alcance de la instancia de la tarea.
- La *asignación basada en la historia* se refiere a la distribución de los ítems de trabajo de una tarea entre los recursos con base en sus antecedentes de ejecución de ítems de trabajo correspondientes con las instancias de la misma.
- La *asignación organizacional* consiste en la especificación en tiempo de diseño de la distribución de ítems de trabajo de una tarea entre los recursos

que pertenecen a una determinada posición o unidad organizacional definida en un modelo de recursos basado en un enfoque organizacional.

La asignación de recursos también puede incluir restricciones como el enlace o la separación de responsabilidades (Bertino, et al., 1999), (Hummer, et al., 2013). Estas restricciones, también referidas como restricciones de resolución son frecuentemente empleadas durante la especificación de estrategias de distribución de trabajo de tipo *push* con el objeto de asegurar que los ítems de trabajo de la tarea sean asignados a un único recurso para su ejecución. Estas restricciones pueden ser duras o blandas (*hard* o *soft*). Las primeras son siempre impuestas por el sistema, las segundas describen restricciones deseables que pueden ser violadas durante determinadas condiciones (Stefansen, et al., 2008).

Otros requerimientos de distribución de trabajo más avanzados incluyen cambios en la distribución de los ítems de trabajo ante la ocurrencia de eventos, o como consecuencia de una operación ejecutada por un recurso. La categoría de patrones de recursos denominada patrones de desvío (Russell, et al., 2005) identifican requerimientos recurrentes de este tipo. Una *escalada* define un desvío iniciado por el WfMS ante la ocurrencia de un evento disparado en el ámbito del ítem de trabajo. Ejemplos de estos eventos son “startDeadline” y “completionDeadline” definidos por WS-HumanTask (Agrawal, et al., 2010). Los mismos permiten modificar la asignación de un ítem de trabajo de una tarea en caso que el mismo no sea respectivamente comenzado a ejecutar o completado dentro de un plazo estipulado. Las escaladas son generalmente definidas con el objetivo de reducir el tiempo requerido para completar ítems de trabajo cuando una fecha límite no es cumplida (van der Aalst, et al., 2007). Los desvíos iniciados por recursos humanos como el rechazo, la reasignación o la delegación (Schefer-Wenzl, et al., 2012) dan algún grado de libertad a los recursos para cambiar la asignación de ítems de trabajo de un recurso a otros.

2.1.3. Definición de Mecanismos de Autorización

La perspectiva de recursos también incluye la definición e implementación de mecanismos de autorización y control de acceso. En un SIOP, la función de autorización se refiere al control sobre las operaciones que pueden realizar los recursos para acceder y ejecutar los ítems de trabajo que le son asignados (Leitner & Rinderle-Ma, 2014). Hay tres tipos principales de mecanismos de control de acceso: control de acceso basado en roles, control de acceso basado en tareas; y control de acceso multi-nivel.

El control de acceso basado en roles (Sandhu, et al., 1996) es un mecanismo de autorización adoptado con frecuencia para gestionar la autorización de usuarios en sistemas de información de distintos tipos. Este mecanismo emplea el concepto de rol, el cual consiste en un conjunto de permisos. Un permiso es una autorización para realizar una determinada acción, tal como ejecutar una tarea. Los usuarios pueden asociarse con uno o

más roles. Los roles pueden estructurarse en jerarquías, es decir, es posible definir sub-roles que hereden los permisos de super-roles. Este mecanismo fue adaptado para incorporar elementos específicos de SIOPs (Wainer, et al., 2003), tales como instancias de procesos o tareas.

El control de acceso basado en tareas (Thomas & Sandhu, 1994) es un mecanismo de autorización para tareas individuales. Los modelos de autorización de este tipo permiten definir la autorización para ejecutar determinada tarea durante la ejecución de una instancia de proceso (Atluri & Huang, 1996). Los derechos de acceso son otorgados cuando la tarea es activada y son revocados cuando la misma es completada.

Los mecanismos de control de acceso multi-nivel especifican niveles de seguridad (confidencial, secreto, público) para procesos o tareas, los cuales definen niveles de autorización (Atluri & Huang, 1997). Los mismos evitan dependencias de los niveles superiores respecto de los inferiores. Por ejemplo, las tareas confidenciales no deben ser seguidas de tareas públicas.

También existen propuestas de modelos de delegación, los cuales permiten expresar la posibilidad de transferir la autorización para ejecutar una determinada tarea (Crampton & Khambhammettu, 2008). La delegación puede convertirse en un problema de seguridad para SIOPs. Si los recursos delegan tareas a otros que no tienen las atribuciones adecuadas, pueden ocurrir violaciones de seguridad como accesos no autorizados. Esto es abordado por modelos de delegación que extienden los mecanismos de autorización introducidos arriba con esquemas de especificación y revocación de delegaciones (Wainer, et al., 2007).

2.2.Soporte a la Perspectiva de Recursos por BPMN 2.0

Business Process Model and Notation (BPMN) (OMG, 2013) es un lenguaje de modelado de procesos de negocio ampliamente aceptado, el cual fue declarado en 2013 como un estándar ISO (ISO/IEC 19510:2013). Uno de sus principales objetivos es proveer un mecanismo estandarizado y de fácil comprensión para definir y visualizar modelos de procesos de negocio. Este lenguaje provee un conjunto de diagramas diseñados para reducir la brecha en el modo en que analistas de negocio y desarrolladores de sistemas entienden estos modelos. El propósito de BPMN es posibilitar la colaboración de distintos tipos de profesionales en la definición de procesos de negocio, lo cual no es soportado por lenguajes como WS-BPEL (Jordan, et al., 2007) que están destinados a la implementación y ejecución de procesos de negocio basados en servicios Web.

La primera versión de este lenguaje fue publicada en 2004 bajo el nombre de Business Process Modeling Notation por una organización llamada Business Process Management Initiative (BPMI). BPMI fue absorbida por el Object Management Group (OMG) en 2005, el cual comenzó a publicar la especificación del lenguaje en 2006. En 2009, el nombre de del lenguaje es cambiado a Business Process Model and Notation manteniendo la sigla

BPMN y en 2010 se publica la segunda versión del lenguaje, la cual incluye un metamodelo especificando la estructura de sus elementos.

El metamodelo de BPMN define dos tipos de tarea que involucran personas en la ejecución de procesos: Tarea Manual (ManualTask) y Tarea de Usuario (UserTask). Una tarea manual representa el trabajo llevado a cabo por recursos humanos sin la asistencia de aplicaciones de software. Una tarea de usuario representa el trabajo llevado a cabo por personas a través de la interacción con un sistema gestor de tareas. Los elementos provistos por el metamodelo de BPMN para definir la asignación de recursos a las tareas de procesos se muestran en la Figura 4.

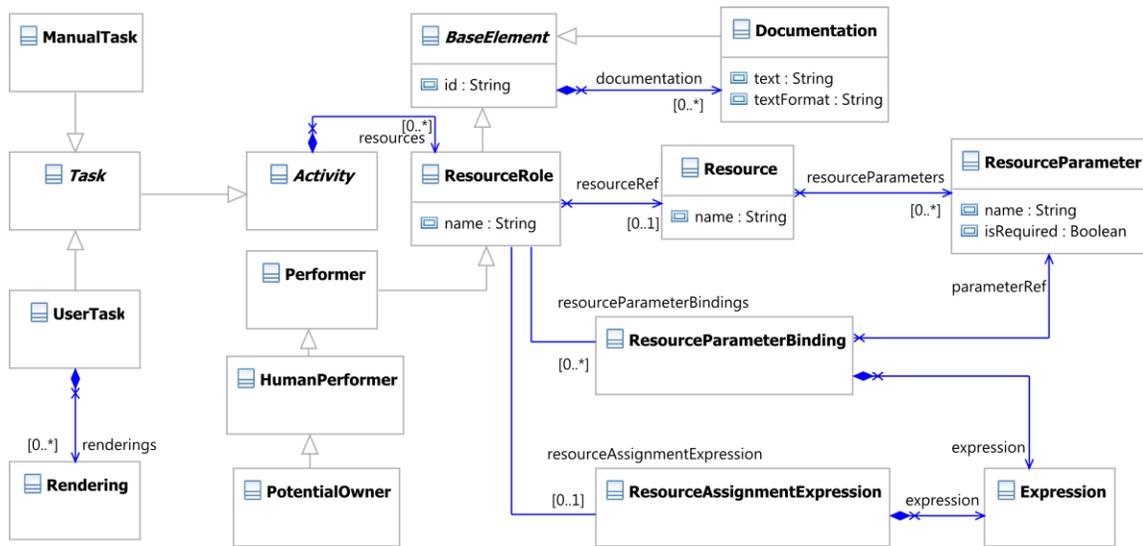


Figura 4. Metamodelo de Asignación de Recursos de BPMN 2.0

Los elementos Resource y ResourceParameter representan conceptos de la estructura de recursos. El elemento Resource define una representación abstracta de un recurso humano o no humano que puede ser referenciado por una actividad de un proceso. Un elemento Resource puede contener cero o más elementos ResourceParameter, los cuales definen información acerca de los recursos. Dicha información puede ser consultada en tiempo de ejecución para realizar la asignación de recursos (OMG, 2013). Sin embargo, la definición de la estructura de recursos está fuera del alcance de BPMN, ya que el lenguaje asume que la misma está definida en un modelo de recursos (Sección 2.1.1) separado. Actualmente, no existe una especificación estándar para la definición de la estructura de recursos en conjunto con el lenguaje BPMN.

En BPMN, la distribución de trabajo es representada por medio de la asignación de recursos a roles que pueden ser desempeñados en cada tarea. Un rol (elemento ResourceRole) define un conjunto de interacciones que un recurso puede llevar a cabo para realizar el trabajo de una tarea por medio de la ejecución de operaciones sobre los ítems de

trabajo. Ejemplo de tales operaciones son: *claim*, *start* y *complete*. Las mismas son generalmente soportadas por WfMSs para permitir a los recursos asumir la responsabilidad por la ejecución de un ítem de trabajo, comenzar su ejecución y notificar su compleción, respectivamente.

La asignación de recursos a roles se define por medio de dos mecanismos: expresiones de asignación de recursos (elemento `ResourceAssignmentExpression`) o consultas parametrizadas de recursos (elemento `Resource Parameter Binding`). Una *expresión de asignación de recursos* permite definir los recursos a ser asignados a un rol a través de la evaluación de datos que están en el alcance de una tarea. Una consulta parametrizada de recursos permite definir una asignación de recursos a través de la evaluación de parámetros de recursos, típicamente almacenados en un repositorio organizacional, contra valores de datos que están definidos en el alcance de una tarea.

BPMN define una jerarquía de roles que inicia en el elemento `ResourceRole` y termina en el elemento `PotentialOwner`, que es el único rol específico definido por el lenguaje. Un elemento `PotentialOwner` en una tarea de usuario indica la asignación de recursos que pueden reclamar los ítems de trabajo de la misma para realizarlos.

El tipo de estrategia de distribución de trabajo depende del significado del rol asignado. El elemento `PotentialOwner` define una distribución de trabajo de tipo *pull*. BPMN no soporta la definición de estrategias de tipo *push*. La especificación del lenguaje prevé la definición de roles adicionales por medio de la especialización del elemento `HumanPerformer`. Sin embargo, esto requiere introducir nuevos elementos al metamodelo del mismo, lo cual no es frecuentemente soportado por las herramientas de modelado existentes.

Lo anterior resulta en que a menudo los profesionales intercambian modelos que adoptan convenciones, como por ejemplo asumir una política de distribución de trabajo *push* cuando el rol `PotentialOwner` es asignado a un único recurso. Sin embargo, esto puede llevar a problemas de interpretación de los modelos y fundamentalmente de comunicación en caso que dichas convenciones no estén adecuadamente documentadas.

El elemento `Lane` es frecuentemente empleado en diagramas BPMN para representar asignaciones de recursos a través del agrupamiento de actividades por una determinada unidad o posición organizacional. Sin embargo, `Lane` es un elemento genérico provisto por el lenguaje para categorizar elementos de un diagrama de procesos con base en cualquier criterio. Por lo tanto esto es útil únicamente para propósitos de visualización.

El elemento `ResourceRole` y sus subclases también permiten representar algunos mecanismos de autorización. Sin embargo, el conjunto de operaciones de ítem de trabajo que un rol autoriza a los recursos debe ser inferido de descripciones en lenguaje natural

incluidas en la especificación del lenguaje. La existencia de un único rol (PotentialOwner) es también una limitación dado que se define un único conjunto de operaciones de ítem de trabajo.

2.2.1. Trabajos Relacionados a la Definición de la Perspectiva de Recursos con BPMN

Existen trabajos que abordan las limitaciones de BPMN para la representación de la perspectiva de recursos, los cuales se centran en la definición de políticas de distribución de trabajo y mecanismos de autorización. Los mismos emplean distintos enfoques para ello.

En un trabajo previo a la publicación de BPMN 2.0 (Grosskopf, 2007), se presenta una capa extendida de recursos para BPMN con base en la especificación Business Process Definition Metamodel (BPDM), la cual provee soporte a 13 de los 43 patrones de recursos. Otros trabajos previos a BPMN 2.0 proponen extensiones al lenguaje definiendo metamodelos con base en la descripción de los elementos de diagramas de BPMN 1. En (Awad, et al., 2009) se propone un metamodelo para representar restricciones de asignación de recursos empleando OCL en diagramas de proceso BPMN 1. Según los autores, la extensión soporta nueve patrones de recursos. En un trabajo similar (Wolter & Schaad, 2007), se propone un metamodelo de BPMN 1 para permitir la introducción de una notación formal para la definición de restricciones de autorización en tareas de usuario. Empleando este enfoque, en (Rodríguez, et al., 2007) se define un metamodelo para definir mecanismos de control de acceso y requerimientos de seguridad más detallados tales como no repudiación, integridad y privacidad.

Las propuestas posteriores a BPMN 2.0 emplean distintos enfoques para mejorar el soporte del lenguaje a la perspectiva de recursos. Un lenguaje específico de dominio llamado RAL para la definición de asignaciones de recursos embebidas en expresiones de asignación de recursos BPMN fue propuesto en (Cabanillas, et al., 2011) y formalizado en una ontología en (Cabanillas, et al., 2012). El mismo asume que los modelos de recursos son definidos con base en el metamodelo adoptado por los patrones de recursos. Este lenguaje permite definir asignaciones de recursos con un alto nivel de abstracción y analizar las mismas usando razonadores de lógica descriptiva. Sin embargo, su uso está restringido sólo a plataformas que soportan el metamodelo mencionado para la definición de modelos de recursos.

Existen también otros trabajos que proponen cambios al metamodelo de BPMN. En (Brucker, et al., 2012) se propone un enfoque llamado “SecureBPMN” para modelar e implementar requerimientos de control de acceso. El mismo propone la modificación del metamodelo de BPMN con el propósito de proveer un conjunto de constructores para la especificación de requerimientos de control de acceso basado en roles, separación de responsabilidades y enlace de responsabilidades. Un prototipo de implementación de estos

constructores fue propuesto con base en el WfMS Activity. Este enfoque tiene como desventaja que, a menos que los cambios propuestos sean adoptados por la OMG, los modelos de procesos resultantes no pueden ser intercambiados con otras herramientas que soportan BPMN dado que los constructores del lenguaje aparecen asociados directamente con elementos ajenos al mismo.

Otros trabajos se basan en la adopción de convenciones para la representación de requerimientos que no son directamente soportados por el lenguaje. En (Altuhhova, et al., 2012) se propone una alineación entre los conceptos del dominio de la gestión de riesgos de sistemas de información y los elementos de BPMN. Este enfoque emplea elementos Pool llamados Usuario y Violador para representar usuarios esperados e intrusos en el sistema. Este enfoque tiene como ventaja la portabilidad de los modelos resultantes. Sin embargo, el uso de elementos Pool implica que los usuarios son representados como entidades externas al proceso. Esto impacta de forma negativa en la legibilidad de modelos de procesos que definen diversas tareas de usuario que involucran a recursos de distintas áreas de la organización.

Finalmente, existen trabajos que proponen la representación de requerimientos adicionales en modelos separados a los soportados por BPMN. Estos modelos se basan en metamodelos construidos para representar los nuevos conceptos requeridos. Los mismos son vinculados en una etapa posterior con los elementos de los modelos de proceso por medio de diferentes mecanismos. El trabajo propuesto en (Leitner, et al., 2011) es un ejemplo de este enfoque. Dicha propuesta permite la definición de requerimientos de seguridad con base en los conceptos de responsabilidad, permiso y rol; los cuales se definen en forma separada de los procesos y luego se vinculan por medio de una herramienta desarrollada a tal fin. El empleo de este enfoque mantiene los modelos de procesos sin alteraciones. Sin embargo, este tipo de enfoque se justifica cuando el lenguaje de base no provee constructor para dar soporte a los nuevos conceptos requeridos.

2.3.Arquitecturas de WfMS

Los SIOPs implementan la perspectiva de recursos con base en los WfMSs, los cuales consisten en un conjunto de componentes de propósito general que son configurados para soportar los procesos específicos de una organización. Estos componentes son definidos en distintas arquitecturas y modelos de referencia.

El modelo de referencia propuesto por la organización Workflow Management Coalition (conocida como WfMC) en (Hollingsworth & Hampshire, 1993) se muestra en la Figura 5. El mismo define un servicio de instanciación de workflow (Workflow Enactment Service), el cual se comunica a través de una API (Application Programming Interface) que provee cinco interfaces que permiten la interacción con herramientas de definición de

proceso, aplicaciones o servicios (interfaz 3), con otros WfMS (interfaz 4), con herramientas de administración y monitoreo (interfaz 5) y con aplicaciones clientes de workflow (interfaz 2). Esta última interfaz tiene como propósito permitir la interacción del servicio de instanciación de procesos con aplicaciones que permitan a los recursos acceder y completar los ítems de trabajo que se les asignan.

La interfaz 2 fue desarrollada con base en el concepto de lista de trabajo. Este concepto se refiere a una cola de ítems de trabajo que son asignados a un recurso o grupo de recursos. La lista de trabajo es accedida por el WfMS con el propósito de asignar ítems de trabajo y por las aplicaciones cliente (también conocidas como manejadores de listas de trabajo) para permitir a los usuarios acceder y completar los ítems de trabajo. Esta interfaz define un conjunto de operaciones para soportar el establecimiento de sesiones, el acceso a la definición de las tareas y el estado del proceso; y funciones para el manejo de ítems y listas de trabajo.

En este modelo, la lista de trabajo y la información referente a los recursos son gestionadas por el WfMS. Al momento de su publicación, la mayoría de los vendedores de WfMSs proveían módulos de gestión de recursos propietarios. Los sistemas de este tipo no permiten compartir información sobre los recursos. Esto lleva a problemas en caso que distintos WfMS se empleen en la misma organización para implementar distintos procesos, dada la necesidad de mantener definiciones redundantes de la estructura organizacional (Zur Muehlen, 2004). Además, información sobre los recursos tal como la carga de trabajo no puede ser determinada sin consolidar la información de los distintos WfMSs.

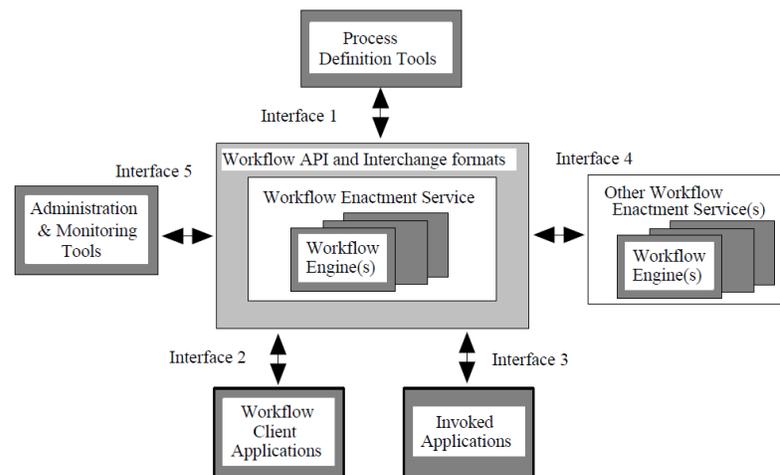


Figura 5. Modelo de Referencia de WfMC

La Figura 6 muestra un modelo de referencia propuesto en un trabajo posterior (Pesic & van der Aalst, 2007) en el que se identifican y definen las interacciones entre dos módulos de WfMSs empleando redes de Petri coloreadas: el de distribución de trabajo

(Work Distribution) y el de lista de trabajo (Work List). El módulo de distribución de trabajo gestiona la asignación de ítems de trabajo a los recursos con base en datos provenientes del esquema organizacional y del de autorización. El módulo de lista de trabajo recibe mensajes del módulo de distribución de trabajo y provee mecanismos para gestionar el inicio de sesiones por parte de los recursos y la posterior selección y ejecución de ítems de trabajo.

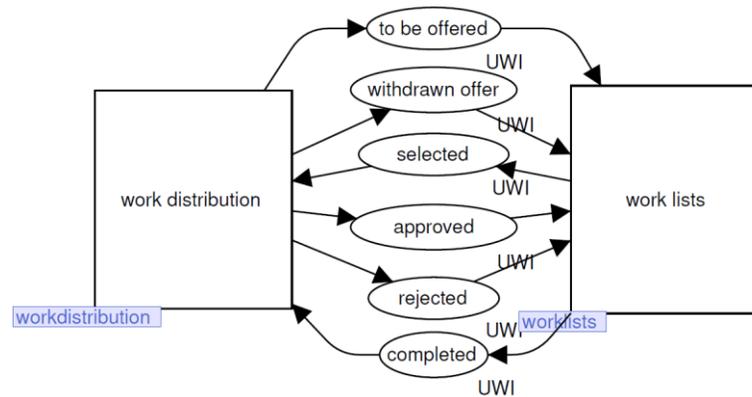


Figura 6. Modelo de Referencia propuesto por Pesic y van der Aals

La Figura 7 muestra la arquitectura propuesta en la especificación WS-HumanTask (Agrawal, et al., 2010). La misma fue definida para permitir la creación de SIOPs basados en WS-BPEL que soporten la perspectiva de recursos. En estos sistemas, los procesos son soportados a través de la orquestación de servicios Web definidos de acuerdo a una arquitectura orientada a servicios. Los motores de proceso WS-BPEL son capaces únicamente de gestionar interacciones con servicios Web. Para soportar la perspectiva de recursos, WS-HumanTask define una arquitectura en la que los componentes de WfMS encargados de gestionar instancias de tareas de usuario se encuentran desacoplados del motor de procesos y accesibles como un servicio Web. En dicha arquitectura, el componente de gestión de ítems de trabajo (Task Processor) permite a motores de proceso WS-BPEL (Task Parent) crear ítems de trabajo y recuperar sus resultados por medio de la invocación de sus operaciones. A fin de permitir la presentación de los ítems de trabajo a los recursos, WS-HumanTask provee el componente Task Client.

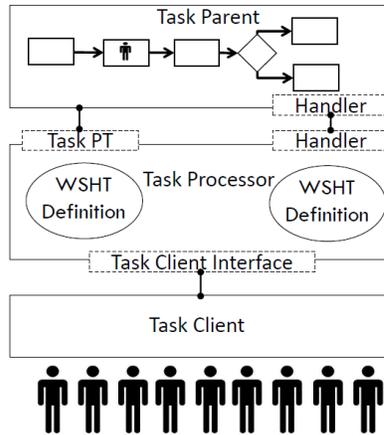


Figura 7. Arquitectura de Infraestructura WS-HumanTask

Los tres modelos de referencia descriptos asumen la existencia de un repositorio con información referente a los recursos. Sin embargo, ninguno de ellos define su arquitectura o las interfaces con el repositorio correspondiente.

2.4. Perspectiva de Recursos en Lenguajes de Ejecución de Procesos

Los WfMSs proveen soporte a la perspectiva de recursos por medio de la asignación de los ítems de trabajo a los recursos con base en los parámetros definidos en especificaciones ejecutables de los procesos. Estas especificaciones ejecutables se definen empleando lenguajes de ejecución de procesos, los cuales varían de un WfMS a otro. Las siguientes secciones presentan un análisis de distintos lenguajes de ejecución existentes.

2.4.1. YAWL

El lenguaje de ejecución de procesos Yet Another Workflow Language (YAWL) fue desarrollado como parte de la iniciativa académica Workflow Patterns Initiative con el propósito de mostrar la factibilidad de dar soporte a los patrones de workflow.

Las entidades provistas por YAWL para definir modelos de recursos (Tan & Aalst, 2006) se muestra en la Figura 8. Este lenguaje soporta un enfoque organizacional de modelado de recursos. El mismo asume que los recursos que llevan a cabo el trabajo asociado con las tareas de los procesos son miembros de una organización. En este contexto, el término organización se emplea para hacer referencia al agrupamiento formal de recursos que llevan a cabo el trabajo asociado con un conjunto común de metas de negocio.

Los recursos individuales, representados por medio del elemento Participant, pueden tener asociados cero o más posiciones en la organización. La mayoría de las características relevantes de un recurso en una organización se derivan en mayor medida de la posición que los mismos ocupan en la organización.

Cada posición pertenece a un grupo de la organización (OrgGroup), los cuales pueden ser temporales o permanentes. Un grupo organizacional puede representar una rama definiendo el agrupamiento de recursos con base en su ubicación; o una división que define el agrupamiento de recursos con base en una línea de negocio. En términos de la jerarquía organizacional, cada posición puede tener un número específico de relaciones con otras posiciones. La posición a la cual una posición reporta representa el recurso responsable por el trabajo ejecutado por el recurso del nivel inferior de la jerarquía.

El metamodelo de recursos de YAWL también contempla la representación de atributos inherentes a los recursos por medio de roles y capacidades. Un rol (elemento Role) provee otro mecanismo para agrupar recursos con posiciones o niveles de responsabilidad similares como gerentes o secretarios. Cada recurso puede tener uno o más roles. Un recurso también puede tener capacidades (elemento Capability), que expresan las aptitudes del mismo para determinado tipo de trabajo. Las capacidades pueden incluir calificaciones, habilidades o experiencia.

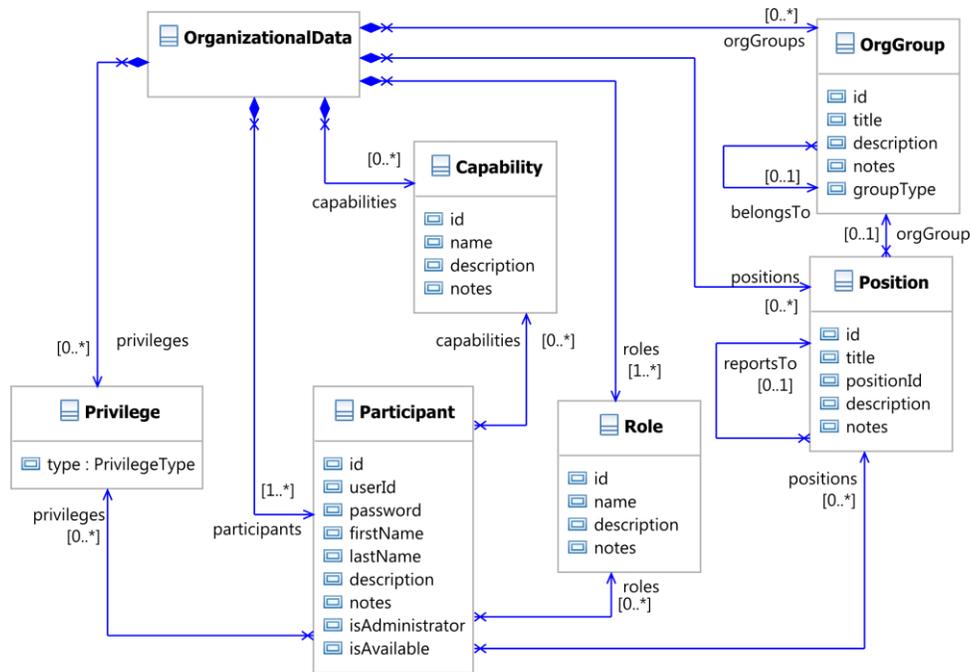


Figura 8. Metamodelo de Recursos de YAWL

La distribución de trabajo se representa en YAWL a través de la definición de una estrategia de interacción. Dicha estrategia consiste en la especificación para tres puntos de interacción (oferta, asignación e inicio) del agente responsable de determinar cuándo ocurre cada una de estas interacciones. Los posibles agentes de cada punto de interacción son el sistema o un recurso elegible.

El resultado del punto de interacción de oferta (elemento `ResourcingOfferFactsType`) consiste en un conjunto inicial de recursos a los cuales se da a conocer el ítem de trabajo correspondiente con la instancia de la tarea. En caso que este punto de interacción sea iniciado por el sistema, este conjunto de recursos debe expresarse definiendo los atributos *role* y *participant* del elemento `ResourcingSetFactsType`. En caso que este punto de interacción sea iniciado por el recurso, el conjunto de recursos iniciales es determinado por un administrador del WfMS.

El resultado del punto de interacción de asignación (elemento `ResourcingAllocateFactsType`) es la elección del recurso responsable por la elección de la tarea. En caso que este punto de interacción sea iniciado por el sistema, es necesario además especificar una estrategia de asignación (*allocator*) a fin de seleccionar uno de los recursos del conjunto inicial resultante de la oferta del ítem de trabajo. En caso que el punto de interacción sea iniciado por el recurso, la responsabilidad por la ejecución del ítem de trabajo es asumida voluntariamente por el primer recurso que lo reclame.

El resultado del punto de interacción de inicio es la ejecución del ítem de trabajo. Cuando este punto de interacción es iniciado por el sistema, el ítem de trabajo es comenzado a ejecutar inmediatamente luego de la asignación del recurso como resultado del punto de interacción anterior. En caso contrario, el ítem de trabajo se comienza a ejecutar cuando el usuario lo decida.



Figura 9. Metamodelo de Distribución de Trabajo de YAWL

2.4.2. BPEL4People

BPEL4People (Kloppmann, et al., 2005) es una extensión al lenguaje Web Service Business Process Execution Language (WS-BPEL) propuesto por el consorcio de industrias OASIS. BPEL4People soporta la integración de personas en la ejecución de procesos de negocio con base en la especificación WS-HumanTask. La Figura 10 muestra los elementos introducidos por estas especificaciones para soportar la perspectiva de recursos. BPEL4People permite introducir dos elementos en especificaciones de procesos BPEL: HumanInteractions y PeopleActivity.

El elemento HumanInteractions es un contenedor para definiciones de requerimientos de la perspectiva de recursos a ser referenciados en distintas partes del proceso por elementos de tipo PeopleActivity. HumanInteractions contiene declaraciones de tres elementos definidos por WS-HumanTask: LogicalPeopleGroup, Task y Notification. El elemento LogicalPeopleGroup especifica un grupo de personas que puede ser asignado a una o más tareas. El elemento Task define el tipo de resultado esperado de un trabajo a ser desarrollado por una o más personas, con base en información provista a las mismas para tal fin. El elemento Notification es un tipo específico de tarea a través de la cual se da a conocer a las personas una determinada condición del proceso sin que se requiera ninguna acción por parte de las mismas.

El elemento PeopleActivity indica las etapas del proceso en las que deben tener lugar las tareas que involucran personas. Las mismas son incluidas en procesos WS-BPEL por medio de elementos ExtensionActivity, el cual es provisto por el lenguaje para incluir nuevos tipos de actividad en especificaciones de procesos. ExtensionActivity es un contenedor para otro elemento definido en un espacio de nombres diferente, el cual especifica funcionalidades o características adicionales a ser soportadas por un motor de proceso WS-BPEL.

El elemento PeopleActivity define una referencia a una tarea WS-HumanTask (elemento Task) en la que se define el trabajo a ser ejecutado por personas. Dicho elemento permite especificar la asignación de recursos que desempeñan un rol (GenericHumanRole) en la tarea a través del elemento PeopleAssignments, por medio de distintos mecanismos como literales, expresiones o agrupamientos lógicos de personas. El primer mecanismo consiste en especificar los recursos a ser asignados directamente en la definición de la tarea. El segundo mecanismo permite definir expresiones que evalúan datos disponibles en el alcance de la tarea para determinar los recursos a ser asignados. El tercer mecanismo consiste en la ejecución de consultas definidas en elementos LogicalPeopleGroup a un repositorio organizacional (referido como People Directory). En cualquier caso, la asignación de recursos puede resultar en entidades de tipo User o Group, las cuales representan un recurso individual o un conjunto de recursos, respectivamente. Sin embargo, las especificaciones BPEL4People / WS-HumanTask no especifican la estructura de la

información a ser almacenada en el repositorio organizacional o el modo en que las consultas deben ser evaluadas.

WS-HumanTask provee un conjunto de roles genéricos (elemento `GenericHumanRole`), cada uno de los cuales permite a los recursos ejecutar un conjunto de operaciones para progresar o influir en el progreso de los ítems de trabajo de la tarea:

- Task Initiator es asignado por la persona que creó la instancia de la tarea. La asignación de este rol es opcional en WS-HumanTask y depende de la forma en que se creó la instancia de la tarea.
- Task Stakeholder se asigna a personas responsables de la supervisión y el resultado de la tarea. Los recursos con este rol pueden influenciar el progreso de los ítems de trabajo de la tarea adjuntando comentarios, reasignándolos, resolviendo plazos de ejecución vencidos, o simplemente observando los cambios de estados de los mismos.
- Potential Owners es desempeñado por personas que pueden reclamar la tarea para su ejecución. Un recurso asignado a este rol se convierte en el dueño real de un ítem de trabajo de la tarea (rol Actual Owner) reclamándolo.
- Excluded Owners se asigna a las personas que no pueden ser asignadas a los roles Potential Owner o Actual Owner.
- Actual Owner es desempeñado por la persona que está ejecutando un ítem de trabajo. En WS-HumanTask, no se puede definir la asignación de este rol. El mismo debe ser obtenido por los recursos con el rol Potential Owner por medio del reclamo del ítem de trabajo.
- Business Administrators permite a los recursos las mismas operaciones que Task Stakeholders, pero a nivel de la definición de tarea. Es decir que pueden acceder e influenciar tanto el estado de los ítems de trabajo de la tarea, como el de las notificaciones disparadas en el contexto de los mismos.
- Notification Recipients se asigna a las personas que reciben notificaciones referentes al estado de los ítems de trabajo de la tarea, tales como el vencimiento de una fecha límite. Dado que las notificaciones no requieren acciones por parte de los usuarios, este rol es de un carácter informativo.

Una tarea WS-HumanTask también puede definir fechas límite de comienzo y terminación, a través de los atributos *startDeadline* y *completionDeadline*. Este elemento también permite especificar el escalado de la tarea, lo cual significa reasignar el rol Potential Owner en caso que alguno de los plazos definidos por los atributos antes mencionados se haya vencido.

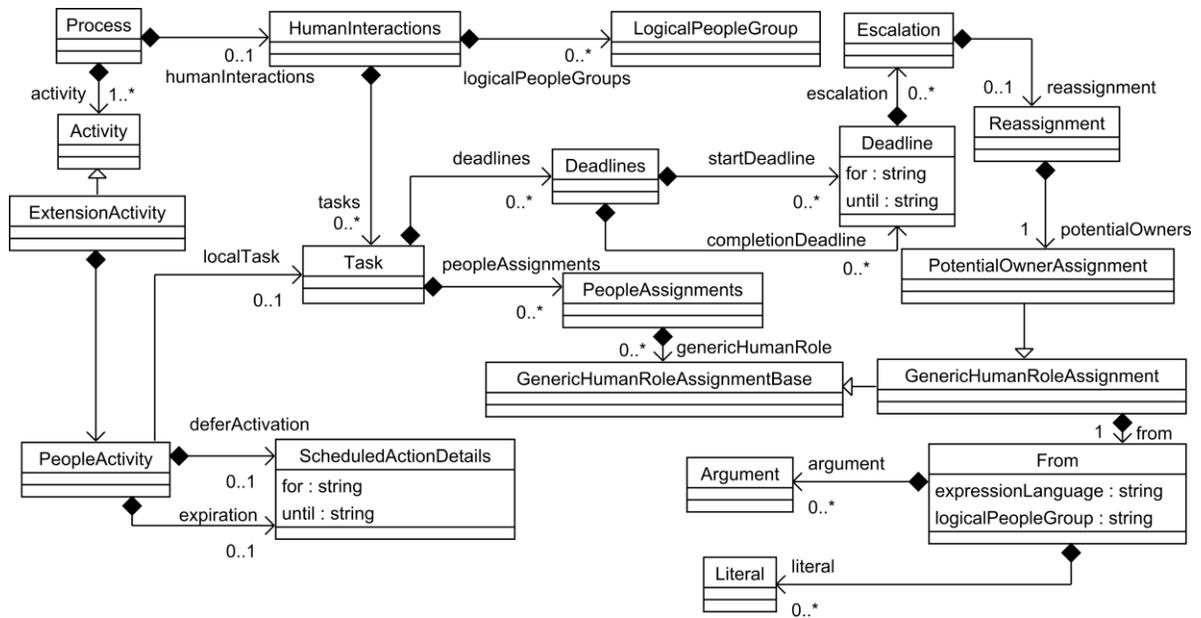


Figura 10. Elementos de Asignación de Recursos de BPEL4People/WS-HumanTask

2.5. Soporte a la Perspectiva de Recursos por WfMSs

Existen numerosos WfMSs, los cuales proveen soporte a la perspectiva de recursos implementando distintas variantes de las arquitecturas y lenguajes descriptos en las secciones anteriores. En general, los WfMSs actuales implementan los componentes que gestionan la asignación y ejecución de ítems de trabajo como componentes separados, los cuales proveen interfaces SOAP o REST para permitir la comunicación con los motores de procesos.

El WfMS YAWL¹ es la plataforma de ejecución desarrollada como parte de una iniciativa académica para demostrar la utilidad y aplicabilidad del lenguaje de ejecución con el mismo nombre. Este WfMS implementa las entidades para la definición de la perspectiva de recursos descriptas en la Sección 2.4.1. Incluye un editor de procesos que provee un asistente para la definición de políticas de distribución de ítems de trabajo. También provee un servicio de recursos, el cual permite gestionar información sobre los recursos con base en el metamodelo de la Figura 8. Sin embargo, no provee un entorno para diseñar modelos de recursos o definir la perspectiva de recursos en modelos de proceso de manera gráfica.

También existen iniciativas académicas que tienen como propósito proveer entornos de ejecución para la perspectiva de recursos con base en las especificaciones BPEL4People y WS-HumanTask. Por ejemplo, VieBOP (Holmes, et al., 2008) es una propuesta que permite implementar procesos BPEL4People reutilizando motores WS-BPEL estándar

¹ <http://www.yawlfoundation.org/>

como Apache ODE². El mismo define un enfoque de despliegue en el que los elementos PeopleActivity son reemplazados por elementos Invoke y Receive que permiten delegar la ejecución de dichas tareas en otro componente propuesto a tal fin. De manera similar, en (Gerhards, et al., 2011) se propone un marco de trabajo para la implementación de Procesadores de Tarea WS-HumanTask con base en un servicio separado que hace uso de Kerberos (Neuman, et al., 1994) para implementar mecanismos de autorización y LDAP (Zeilenga, 2006) para almacenar y permitir la consulta de información referente a los recursos de la organización.

En otro trabajo (Thomas, et al., 2007) se provee un enfoque que permite la integración de personas en la ejecución de procesos WS-BPEL con base en un servicio de gestión de tareas de usuario diferente del propuesto en WS-HumanTask. El mismo provee además una extensión a la sintaxis de WS-BPEL diferente a la propuesta por BPEL4People para definir requerimientos de la perspectiva de recursos. A fin de dar soporte a la ejecución de estas tareas extendidas, este trabajo provee una plantilla XSL (Clark, 1999) que permite transformar procesos WS-BPEL extendidos en especificaciones de procesos WS-BPEL estándar en las que los elementos PeopleActivity son reemplazados por elementos Invoke y Receive, los cuales permiten al motor BPEL interactuar con un gestor de tareas propuesto por los autores.

En lo que respecta a los WfMSs de uso comercial, los mismos también implementan distintas arquitecturas y lenguajes para dar soporte a la perspectiva de recursos. Por ejemplo, ActiveVOS³ es un WfMS que provee un editor BPMN para la definición de procesos, los cuales son luego implementados y ejecutados en una plataforma definida con base en las especificaciones WS-BPEL, BPEL4People y WS-HumanTask. Este WfMS provee también un servicio para el acceso a información sobre los recursos denominado Servicio de Identidad, el cual no es definido por dichas especificaciones. El mismo permite almacenar y recuperar información de los recursos con base en las entidades de rol e identidad de usuario.

Intalio⁴ es otro ejemplo de WfMS que provee un editor de procesos BPMN y un motor de procesos WS-BPEL. Sin embargo, este motor de procesos no adopta la especificación de BPEL4People. El mismo implementa la perspectiva de recursos por medio del servicio Tempo, el cual se basa en un metamodelo de tareas de usuario propietario. Al no implementar BPEL4People, esta plataforma de implementación requiere que los recursos de la organización sean representados como entidades externas al proceso que está siendo definido. De este modo, los distintos grupos de recursos involucrados en la ejecución de

² <http://ode.apache.org/>

³ <http://www.activevos.com/>

⁴ <http://www.intalio.com/>

procesos son representados por elementos de tipo Pool, que contienen las tareas asignadas a los mismos. El proceso que está siendo definido debe incluir mensajes que permitan activar y recuperar los resultados de dichas tareas.

Oracle Fusion Middleware⁵ es otra plataforma de implementación que permite la definición de procesos en BPMN y BPEL. Provee un servicio de tareas de usuario propietario con mecanismos de asignación, escalado, notificaciones, eventos y reglas de acceso. Este entorno de ejecución también provee un enfoque propio para permitir a las especificaciones de procesos BPEL incluir referencias a tareas que involucran personas.

Otros WfMSs proveen soporte a la implementación de procesos conforme con la semántica de ejecución definida en BPMN 2.0. Un ejemplo de este tipo de WfMS es Bonita⁶. Sin embargo, provee soporte a la perspectiva de recursos con base en conceptos distintos de los descritos en la sección 2.2. Este WfMS implementa la distribución de ítems de trabajo en términos de definiciones de asignaciones de actores y filtros de actores para las tareas de usuario. Provee además un metamodelo organizacional genérico que permite definir la membresía de usuarios a grupos de la organización por medio del desempeño de roles. De manera similar, Bizagi⁷ implementa las perspectivas de control y de datos con base en BPMN 2.0. En el caso de la perspectiva de recursos, este WfMS soporta la distribución de ítems de trabajo con base en reglas de asignación, métodos de asignación y precondiciones. También provee un repositorio organizacional basado en un metamodelo que incluye entidades como áreas, posiciones, roles, habilidades, grupos y propiedades.

2.6. Metodologías de Desarrollo de SIOPs

Existen antecedentes de trabajos que proporcionan guías metodológicas para abordar ciertos aspectos de la perspectiva de recursos durante el ciclo de vida de los procesos de negocio. Un trabajo propuesto en (Zur Muehlen, 2004) provee una descripción general de las actividades requeridas durante el análisis de entorno, diseño, implementación, ejecución y evaluación de procesos de workflow para gestionar dicha perspectiva. Sin embargo, existen pocas propuestas que provean enfoques sistemáticos para definir esta perspectiva en el desarrollo de SIOPs partiendo de modelos conceptuales definidos por expertos del dominio.

Una metodología presentada en (Reijers, 2003) propone un método para el diseño y control de procesos de workflow con base en conceptos frecuentemente empleados en la definición de procesos de manufactura. Esta propuesta incluye lineamientos para la

⁵ <http://www.oracle.com/us/products/middleware/overview/index.html>

⁶ <http://www.bonitasoft.com>

⁷ <http://www.bizagi.com>

definición de la perspectiva de recursos. Sin embargo, no proporciona directrices para su posterior implementación en SIOPs basados en WfMSs.

Otro enfoque presentado en (Weske, 2012) propone una metodología general para el desarrollo de SIOPs. La misma provee lineamientos generales respecto de las actividades a desarrollar a lo largo del ciclo de vida del desarrollo de SIOPs que consiste en las etapas de relevamiento, diseño, selección de la plataforma, implementación, testeo, despliegue, operación y control. Sin embargo, no provee técnicas a ser aplicadas para abordar perspectivas específicas, tales como la de control, datos o recursos a lo largo de dicho ciclo de vida.

Existen también enfoques basados en los principios del desarrollo dirigido por modelos que tienen como propósito permitir la generación automatizada de especificaciones ejecutables de la perspectiva de recursos. Por ejemplo, el enfoque propuesto en (Link, et al., 2008) provee un método MDA para definir la perspectiva de recursos a través de un perfil UML llamado “Human Task Profile”, el cual se basa en la especificación de WS-HumanTask. Por lo cual este enfoque se restringe a los requerimientos soportados por dicha plataforma. Otro trabajo (Tran, et al., 2008) propone un enfoque dirigido por modelos para definir las distintas perspectivas de los procesos como vistas de un metamodelo, el cual puede ser extendido empleando mecanismos de herencia para representar distintas plataformas de ejecución. Sin embargo, este trabajo no propone notaciones que permitan a los expertos de dominio comprender con facilidad las especificaciones de los procesos resultantes.

2.7.Conclusiones

De acuerdo con los antecedentes descriptos y analizados en las secciones anteriores, se ha identificado una serie de inconvenientes que surgen al momento de definir la perspectiva de recursos durante el desarrollo de SIOPs. Los tipos de requerimientos a ser representados en la perspectiva de recursos de procesos de negocio pueden ser clasificados en: estructura de recursos, distribución de trabajo y autorización (Sección 2.1).

Los lenguajes de modelado de procesos proveen un limitado soporte para la definición de esta perspectiva. Evaluaciones realizadas con base en los patrones de recursos a BPMN y a otros lenguajes de modelado de procesos de negocio de alto nivel de abstracción, tales como Diagramas de Actividad UML, muestran un limitado soporte de los mismos a la representación de la perspectiva de recursos.

El lenguaje de modelado BPMN, considerado un estándar por la industria y la academia, carece de constructores para dar soporte completo a los diferentes tipos de requerimientos de la perspectiva de recursos de procesos. No provee constructores para la definición de políticas de distribución de trabajo de tipo *push*, ni políticas con restricciones de resolución y desvíos. Si bien soporta la definición de mecanismos de autorización

basados en tareas, provee un único rol para ser utilizado en la definición de modelos de procesos. Excluye de su alcance la definición de modelos de recursos. Hasta el momento no ha sido definida una especificación estándar para la definición de modelos de recursos a ser empleada en conjunto con BPMN.

Existen antecedentes de trabajos que proponen enfoques para mejorar el soporte de BPMN para la representación de la perspectiva de recursos (Sección 2.2.1). Estos trabajos destacan la importancia de contar con representaciones de esta perspectiva en modelos de alto nivel de abstracción y proveen enfoques para la definición de restricciones en la asignación de ítems de trabajo, lenguajes para definir expresiones de asignación de recursos y mecanismos adicionales de autorización. Los enfoques propuestos soportan la definición de estas cuestiones en forma aislada y, en la definición de los requerimientos de la perspectiva de recursos, hacen suposiciones respecto a su posterior implementación, tales como la adopción de WS-HumanTask o de entidades para definir modelos de recursos que son específicas de una plataforma. Esto limita dichos enfoques a la representación de requerimientos soportados por tales plataformas. En lo que respecta a trabajos anteriores a BPMN 2.0, los mismos se basan en metamodelos construidos ad-hoc sobre la base de las descripciones de diagramas de BPMN 1, lo cual dificulta la adopción de dichos enfoques debido a que cada uno se basa en modelos conceptuales diferentes. En el caso de trabajos posteriores a la publicación de BPMN 2.0, las propuestas que introducen conceptos adicionales relacionados con esta perspectiva, no reutilizan el mecanismo de extensión provisto por el lenguaje para este propósito. Esto impide el intercambio de los modelos resultantes entre diferentes herramientas BPMN.

Los lenguajes de ejecución de procesos permiten la representación de la perspectiva de recursos por medio de distintos enfoques. Las especificaciones BPEL4People y WS-HumanTask proveen un enfoque que se basa en la asignación de recursos a roles genéricos. Dichas especificaciones no incluyen soporte para la representación de la estructura de recursos. Las evaluaciones de estas especificaciones en relación a los patrones de recursos muestran limitaciones en sus capacidades para soportar requerimientos de distribución de trabajo avanzados y mecanismos de autorización.

El lenguaje de ejecución de procesos YAWL provee un enfoque distinto para la definición de la perspectiva de los recursos. Se basa en la definición de estrategias de interacción del sistema con los recursos para gestionar la oferta o asignación de ítems de trabajo y proporciona un conjunto de entidades para la definición de modelos de recursos. Fue desarrollado con el propósito de dar soporte a los patrones de recursos, por lo cual proporciona mayor soporte a los mismos. Sin embargo, no contempla la definición de escalado de tareas ni la posibilidad de involucrar recursos en ítems de trabajo con roles distintos al de su ejecución.

Los WfMSs que implementan procesos de negocio que involucran personas en la ejecución de sus tareas presentan marcadas diferencias entre sí, de modo que no existe un estándar (Sección 2.5). Estas diferencias no se deben sólo a aspectos no cubiertos, tales como la definición de modelos de recursos, sino también a divergencias respecto de las especificaciones en las que se basan. Esto se refleja en los resultados dispares que se muestran en las evaluaciones realizadas con base en los patrones de recursos del soporte provisto a la perspectiva de los recursos por diferentes WfMS.

La limitada capacidad de los lenguajes de modelado para representar requerimientos de la perspectiva de recursos dificulta la definición de esta perspectiva en los modelos conceptuales. Como consecuencia, la mayor parte de estos requerimientos son representados directamente durante la definición de las especificaciones ejecutables requeridas por el WfMS seleccionado, los cuales no son fácilmente comprensibles por expertos del dominio. Dado que los WfMS utilizan conceptos y terminologías diferentes, es mayor el riesgo de optar por una plataforma de ejecución que no satisfaga las necesidades de la organización. En tal caso, la organización debería afrontar altos costos para desarrollar una implementación con base en otra plataforma. Todo esto constituye un obstáculo para la definición de SIOPs que distribuyan el trabajo de las tareas de los procesos de manera consistente con los lineamientos definidos desde los puntos de vista organizacional y operativo.

Otra consecuencia de lo anterior es, con frecuencia, que las definiciones de la perspectiva de los recursos en SIOPs basados en WfMSs son simplistas. Esto puede resultar en SIOPs inflexibles o liberales, cuyos inconvenientes fueron señalados (Sección 1.2).

Con el propósito de proveer soporte a la definición de la perspectiva de recursos de procesos de negocio durante el desarrollo de SIOPs, varios problemas deben ser resueltos:

- La definición de requerimientos de esta perspectiva en modelos conceptuales de procesos de negocio.
- La selección de un WfMS que soporte los requerimientos definidos en modelos conceptuales de procesos.
- La implementación de dichos requerimientos definiendo modelos de procesos específicos de plataforma y especificaciones ejecutables de workflow.
- El chequeo de la consistencia entre la implementación y los modelos conceptuales con el propósito de constatar que los requerimientos son correctamente implementados por la solución tecnológica.

Los capítulos siguientes presentan soluciones concretas a estos problemas.

3 Marco de Trabajo para Definir la Perspectiva de Recursos

En este capítulo se presenta el marco de trabajo propuesto para definir la perspectiva de recursos de procesos de negocio. Se presenta la estructura del marco de trabajo (Sección 3.1). Se describe el Metamodelo de la Perspectiva de Recursos (Sección 3.2). Se presenta el Metamodelo de Implementación de la Perspectiva de Recursos y se ilustra un ejemplo de un Modelo de Implementación de la Perspectiva de Recursos representando un WfMS (Sección 3.3). Se describe la Extensión de Estructura de Recursos y la Extensión de Distribución de Trabajo y se muestra un ejemplo de un Modelo de Estructura de Recursos y de un Modelo de Proceso Extendido, definidos aplicando dichas extensiones (Sección 3.4). Finalmente, se presentan las conclusiones del capítulo (Sección 3.5).

3.1 Estructura del Marco de Trabajo

En esta sección se presenta la estructura del marco de trabajo propuesto para la definición de la perspectiva de recursos durante el desarrollo de SIOPs basados en WfMSs. El objetivo del mismo es proveer metamodelos y extensiones a BPMN que permitan la definición de esta perspectiva con base en un único modelo conceptual, con el propósito de:

Dar soporte a la definición de modelos conceptuales de la perspectiva de recursos, los cuales sean comprensibles por expertos de dominio, analistas de negocio u organizacionales, y desarrolladores de SIOPs.

Permitir la representación de los requerimientos definidos en dichos modelos conceptuales en términos de los elementos provistos por distintos WfMSs, a fin de obtener modelos y especificaciones que definan implementaciones de SIOPs con base en dichos WfMSs.

A partir de los modelos de referencia y arquitecturas de WfMS analizados (Sección 2.3), se puede establecer que un WfMS es una plataforma que permite la implementación de la perspectiva de recursos de procesos de negocio por medio de tres tipos de componentes:

Repositorio organizacional: conocido como Organizational Repository o People Directory. Permite la gestión de la información referente a los recursos con base en un modelo o esquema de recursos, también conocido como esquema organizacional. Además, provee los mecanismos que permiten el acceso y consulta de la información sobre los recursos en tiempo de ejecución.

Componente de distribución de trabajo: conocido como Work Distribution Component o Task Processor. Ejecuta las políticas de distribución de trabajo definidas para los ítems de trabajo de una tarea, consultando la información almacenada en el repositorio organizacional o disponible en el alcance de las instancias de tarea. Mantiene listas de ítems de trabajo, las cuales son colecciones de datos que definen el trabajo a ser cumplimentado

por cada recurso registrado en el repositorio organizacional. Este componente implementa los mecanismos de autorización para las operaciones que afectan el estado de los ítems de trabajo. En resumen, es el encargado de gestionar el ciclo de vida de los mismos.

Manejador de lista de tareas: conocido como Worklist Manager o Task Client. Implementa los mecanismos de autenticación de recursos, provee las interfaces de usuario requeridas para acceder, organizar y cumplimentar los ítems de trabajo incluidos en las listas de ítems trabajo. Implementa los mecanismos de autorización para las operaciones que permiten a los recursos organizar sus listas de trabajo.

Con base en estos componentes se identificaron tres aspectos a ser representados para capturar la información requerida por los WfMSs para gestionar la distribución de ítems de trabajo y las posteriores interacciones de los recursos con los mismos. Dichos aspectos, definidos en (Stroppi, et al., 2011), son: Estructura de Recursos, Autorización y Distribución de Trabajo.

El aspecto Estructura de Recursos define la información sobre los recursos humanos a ser almacenada por el repositorio organizacional para ser consultada durante la distribución de ítems de trabajo. La definición de este aspecto resulta en un modelo de estructura de recursos. El mismo define la caracterización y la clasificación de los recursos. La caracterización es la descripción de las propiedades de los recursos que son relevantes para la asignación de ítems de trabajo a los mismos. Ejemplo de tales propiedades son: especialidades, ubicación, años de experiencia, horarios o turnos de trabajo. La clasificación consiste en la asociación de recursos con un concepto, por ejemplo, una determinada unidad organizacional, un puesto de trabajo, un grupo de usuarios o un equipo de trabajo. Esto permite referir grupos de recursos y representar sus características comunes.

El aspecto Distribución de Trabajo define el modo en que los ítems de trabajo son dados a conocer y también asignados a los recursos para su ejecución (Russell, van der Aalst, ter Hofstede, & Edmond, 2005). Este aspecto resulta en la definición de políticas de distribución de trabajo para cada tarea de usuario de un proceso. Dichas políticas especifican los principios con base en los cuales el componente de distribución de trabajo asigna recursos para la ejecución de los ítems de trabajo de una tarea.

El aspecto Autorización define los privilegios concedidos a los recursos para las operaciones de ítem de trabajo y las operaciones de lista de trabajo. Este aspecto se divide en Autorización Estática y Autorización Dinámica. El aspecto Autorización Estática se refiere a los privilegios otorgados en tiempo de diseño a los recursos. Este resulta en la definición de privilegios como parte del modelo de estructura de recursos. El aspecto Autorización Dinámica se refiere a los privilegios que los recursos tienen para interactuar con los ítems de trabajo de una tarea en tiempo de ejecución, es decir, cuando las instancias

de los procesos son gestionadas. El mismo resulta en la definición de información acerca de estos privilegios en la especificación de las tareas de un proceso.

El marco de trabajo propuesto permite la representación de estos aspectos en diferentes modelos a ser usados como artefactos en el desarrollo de SIOPs, posibilitando de esta manera la definición e implementación de los requerimientos de la perspectiva de recursos. Para ello, provee metamodelos que permiten definir estos modelos (Figura 11).

El marco de trabajo toma como base los conceptos provistos por el metamodelo BPMN 2.0 para la definición de la perspectiva de recursos (Sección 2.2). El propósito es posibilitar la definición de modelos (artefactos) basados en BPMN durante el desarrollo de SIOPs. A partir de los conceptos de BPMN 2.0, se definieron dos metamodelos (Figura 11): el Metamodelo de la Perspectiva de Recursos denominado *Resource Perspective Metamodel (RPMeta)*, y el Metamodelo de Implementación de la Perspectiva de Recursos denominado *Resource Perspective Implementation Metamodel (RPIImplMeta)*. También se definieron dos extensiones a BPMN que permiten introducir los conceptos de estos metamodelos en modelos definidos con este lenguaje.

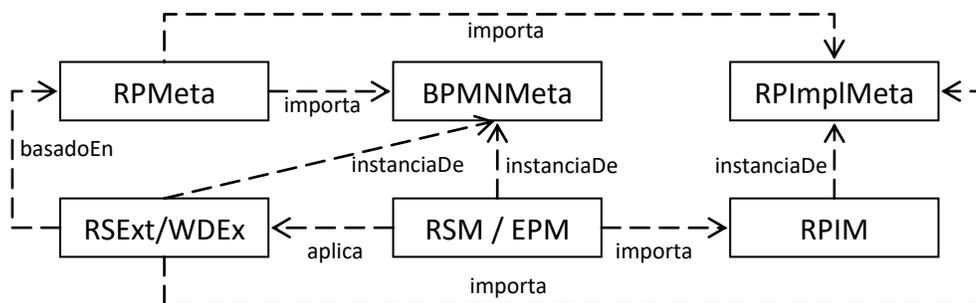


Figura 11. Metamodelos y Modelos del Marco de Trabajo Propuesto.

El RPMeta representa el modelo conceptual de la perspectiva de recursos propuesto en esta tesis. Define los elementos que permiten expresar los requerimientos de esta perspectiva. Este metamodelo permite describir los distintos tipos de requerimientos de la perspectiva de recursos (Sección 2.1) con base en los elementos genéricos provistos por el Metamodelo de BPMN (BPMNMeta). De este modo, permite subsanar las limitaciones de los lenguajes de modelado de procesos actuales, reutilizando además los conceptos provistos por el estándar BPMN. Con este propósito, el RPMeta importa elementos del BPMNMeta y define un conjunto de elementos adicionales que completan los conceptos requeridos para definir los aspectos de la perspectiva de recursos.

El RPIImplMeta, basado en los conceptos del RPMeta, define los elementos que permiten representar las entidades provistas por los WfMSs para implementar los aspectos de la perspectiva de recursos. De este modo, permite describir los WfMSs con base en un único modelo conceptual, el cual es independiente de las diferentes arquitecturas, lenguajes

o enfoques en los que se basan los WfMSs. Utilizando este metamodelo se pueden instanciar Modelos de Implementación de la Perspectiva de Recursos, denominados Resource Perspective Implementation Model (RPIM). Un RPIM representa una plataforma de implementación o WfMS. Describe las entidades provistas por un WfMS para implementar los requerimientos de la perspectiva de recursos.

Con el propósito de permitir la inclusión de los elementos definidos en el RPMeta y el RPImplMeta en modelos de proceso definidos con BPMN, el marco de trabajo provee dos extensiones a BPMN, definidas de acuerdo con el mecanismo de extensión provisto por este lenguaje: La *Extensión de Estructura de Recursos* (RSExt) y la *Extensión de Distribución de Trabajo* (WDExt). Mediante estas extensiones, el marco de trabajo permite la creación de dos modelos de la perspectiva de recursos de procesos de negocio (Figura 11): el *Modelo de Estructura de Recursos*, denominado *Resource Structure Model (RSM)*, y el *Modelo de Proceso Extendido*, denominado *Extended Process Model (EPM)*. Un RSM define los aspectos Estructura de Recursos y de Autorización Estática. Un EPM define los aspectos Distribución de Trabajo y de Autorización Dinámica. Ambos modelos pueden ser utilizados como modelos conceptuales independientes de la plataforma, o bien como modelos de implementación basados en un WfMS. Para esto último, un RSM y un EPM pueden importar un RPIM que representa las entidades de un WfMS, las cuales pueden ser referenciadas por sus elementos para indicar cómo se implementan los mismos según la plataforma de WfMS seleccionada.

3.2. Metamodelo de la Perspectiva de Recursos

El Metamodelo de la Perspectiva de Recursos (RPMeta) define el modelo conceptual propuesto en esta tesis para la representación de los aspectos Estructura de Recursos, Distribución de Trabajo y Autorización que conforman la perspectiva de recursos. El RPMeta reusa los conceptos genéricos provistos por BPMN 2.0 para representar esta perspectiva. Las Figuras 12 y 13 muestran los elementos del RPMeta y sus relaciones. Las clases de color gris representan los elementos definidos por BPMN que son reusados; las clases de color blanco representan los nuevos elementos definidos.

La Figura 12 muestra los elementos del RPMeta que permiten expresar los aspectos Estructura de Recursos y Autorización Estática. La definición del aspecto Estructura de Recursos es abordada en esta tesis como un problema de clasificación y caracterización de recursos. Este enfoque permite asociar conjuntos de recursos con un concepto y definir propiedades a los mismos en forma individual o colectiva por medio de su pertenencia a una clase. Esto otorga al enfoque un carácter genérico, dado que permite expresar modelos de recursos siguiendo tanto el *enfoque organizacional* como el *enfoque tecnológico* (Sección 2.1.1) a través de la asociación de recursos con conceptos pertenecientes a uno u otro dominio. El aspecto Autorización Estática, por su parte, se aborda en términos de la

asignación de privilegios a los recursos en forma individual o colectiva, para la ejecución de operaciones sobre cualquier ítem de trabajo que les sea asignado.

Para representar el aspecto Estructura de Recursos, el metamodelo introduce los conceptos de recurso humano (elemento *HumanResource*) y clasificador de recurso (elemento *ResourceClassifier*) como especializaciones del elemento *Resource* de BPMN, el cual representa cualquier tipo de recurso que pueda ser referenciado por una tarea de un proceso. El elemento *HumanResource* representa recursos humanos en forma individual; mientras que el elemento *ResourceClassifier* representa un conjunto de recursos agrupados con base en un criterio determinado. Ejemplos de clasificadores de recursos son: un departamento o posición organizacional, un turno de trabajo, un equipo de trabajo o un grupo de usuarios del sistema.

El RPMeta también permite definir referencias entre los clasificadores de recursos (elemento *ResourceReference*). Una relación de recursos (*ResourceRelationship*) es un tipo de referencia que indica una asociación dirigida entre clasificadores de recursos. Ejemplos de relaciones entre clasificadores de recursos son las líneas de reporte entre empleados y jefes de área o entre jefes de área y gerentes de departamentos que se presentan en los modelos organizacionales. El otro tipo de referencia entre clasificadores de recursos es la inclusión (elemento *Subsumtion*), la cual representa la incorporación de la población del clasificador de origen en la población del clasificador de destino, junto con la herencia en el clasificador de origen de los parámetros y privilegios definidos por el clasificador de destino. Ejemplos de este tipo de relación son las relaciones entre los departamentos y áreas de una organización.

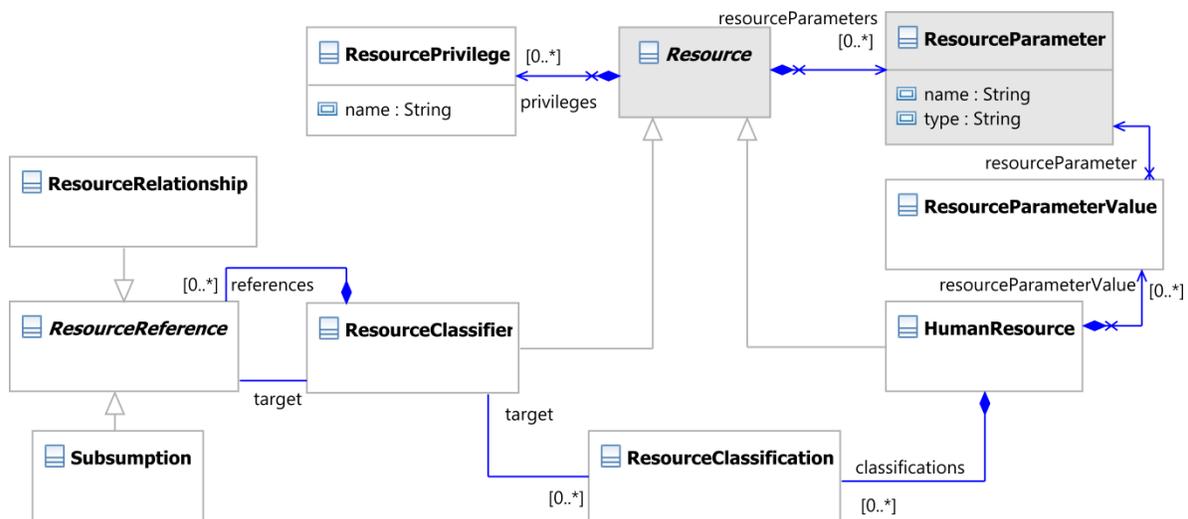


Figura 12. RPMeta – Aspectos Estructura de Recursos y Autorización Estática

La membresía de un recurso humano a un determinado clasificador de recurso se define usando relaciones de clasificación (elemento *ResourceClassification*). Por medio de

una relación de clasificación, las propiedades y privilegios definidos por un determinado clasificador de recurso son asociadas con un recurso humano.

La caracterización de recursos se expresa por medio del elemento *ResourceParameter* de BPMN. Este elemento permite representar la información requerida para distribuir ítems de trabajo entre los recursos, tal como sus capacidades, especialidades o años de experiencia en forma individual o colectiva.

Por último, el aspecto Autorización Estática es representado por medio de la definición de privilegios (elemento *ResourcePrivilege*) para los recursos humanos o clasificadores de recursos.

La Figura 13 muestra los elementos provistos por el RPMeta para definir los aspectos Distribución de Trabajo y Autorización Dinámica. El metamodelo utiliza el elemento *ResourceRole* provisto por BPMN 2.0 para la distribución y asignación de tareas a recursos. Este elemento define el modo en que los recursos pueden interactuar con los ítems de trabajo correspondientes con las instancias de las tareas de usuario de un proceso. Distintos roles pueden ser desempeñados en una tarea de usuario. La asignación de recursos a un rol es realizada por medio de dos mecanismos: las consultas parametrizadas de recursos y las expresiones de asignación de recursos. El primero consiste en definir una referencia (*resourceRef*) a un elemento *Resource* y cero o más enlaces o referencias a parámetros del recurso (elemento *ResourceParameterBinding*). El segundo consiste en definir expresiones (elemento *ResourceAssignmentExpression*) que evalúan datos en el alcance de la tarea y devuelven los identificadores de los recursos a ser asignados al rol.

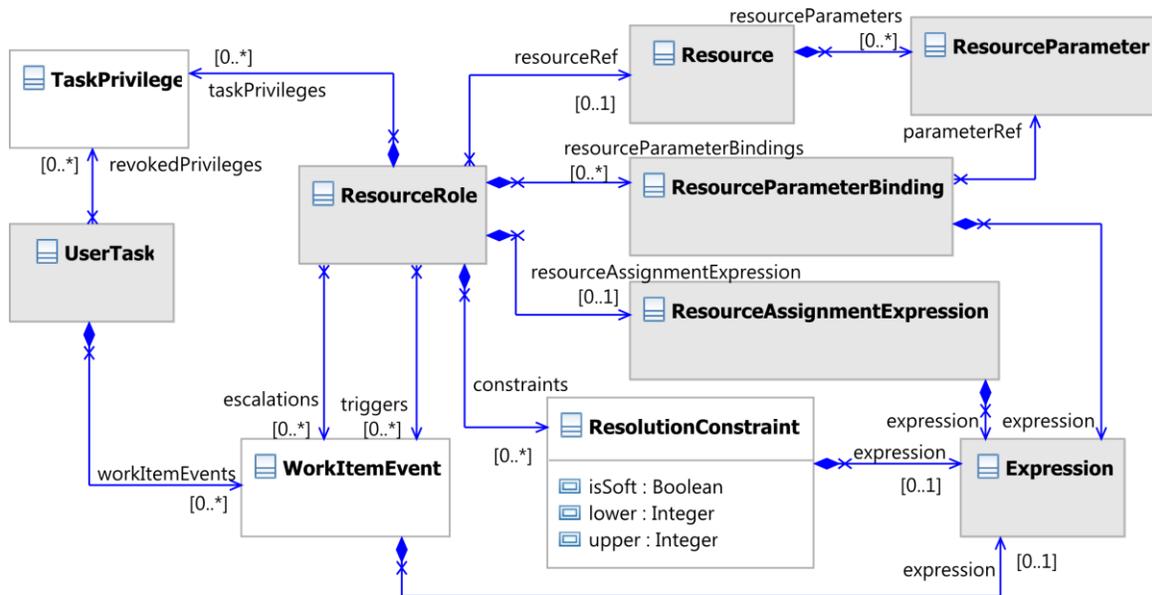


Figura 13. RPMeta – Aspectos Distribución de Trabajo y Autorización Dinámica.

Con el propósito de dar soporte a la necesidad de definir restricciones sobre los conjuntos de recursos resultantes, el RPMeta introduce el elemento *ResolutionConstraint*, el cual representa restricciones sobre el conjunto de roles asignados al rol de recurso, tales como el enlace o la separación de responsabilidades que se introdujeron en el Capítulo 2. Una restricción de resolución se expresa en términos de un nombre (atributo *name*), las cantidades mínima y máxima de recursos asignados al rol luego de su aplicación (atributos *lower* y *upper*) y si la misma puede ser violada en determinadas condiciones definidas en una expresión asociada (atributo *isSoft*).

El aspecto Autorización Dinámica también se define a partir del elemento *ResourceRole* de BPMN. Como se indicó en la Sección 2.1, BPMN define un único rol llamado *PotentialOwner*, lo cual es una limitación desde el punto de vista de la autorización. A fin de mejorar el soporte provisto a este aspecto, se introduce el concepto de privilegio de tarea (elemento *TaskPrivilege*). Por medio de este elemento es posible describir, a nivel de modelo, los privilegios que un rol de recurso otorga a los recursos para ejecutar operaciones sobre los ítems de trabajo de una tarea. Estas operaciones pueden tener como consecuencia modificaciones en la asignación de recursos a los roles definidos para la tarea. Además, por medio del atributo *revokedPrivileges*, es posible restringir los privilegios asignados a los recursos por los roles en el contexto de una tarea de usuario. De este modo, el conjunto de privilegios de tarea otorgados a los recursos es el resultante de abstraer los privilegios revocados por la tarea de los privilegios otorgados por el rol.

Con el propósito de dar soporte a la definición de desvíos iniciados por el sistema, se introduce el concepto de evento de ítem de trabajo (elemento *WorkItemEvent*), el cual define un evento que es disparado en el contexto de un ítem de trabajo de una tarea de usuario y manejado por una operación de ítem de trabajo. Los eventos de ítem de trabajo pueden ser referenciados por roles de recursos definiendo sus atributos *trigger* y *escalation*. El atributo *trigger* referencia un evento de ítem de trabajo que causa la asignación de los recursos al rol. Esto permite definir requerimientos de distribución tardía (Russell, et al., 2005). Se asume que un rol sin un *trigger* definido es distribuido cuando el ítem de trabajo es activado. El atributo *escalation* referencia un evento ante el cual el rol es removido a los recursos. Las fechas límite son ejemplos típicos del empleo de eventos de ítem de trabajo. Los atributos *trigger* y *escalation* en conjunto permiten definir la reasignación de los recursos sin la necesidad de interrumpir la tarea de usuario.

3.3. Representación de Plataformas de Implementación de la Perspectiva de Recursos

En esta sección se presentan los metamodelos y artefactos provistos por el marco de trabajo con el objetivo de permitir la descripción del soporte provisto por distintos WfMSs a los conceptos de la perspectiva de recursos descritos en la sección anterior.

3.3.1. Metamodelo de Implementación de la Perspectiva de Recursos

El Metamodelo de Implementación de la Perspectiva de Recursos (RPIImplMeta) fue definido con base en el RPMeta (Sección 3.2). Cada elemento del RPIImplMeta representa la implementación de un concepto del RPMeta. Los atributos de los elementos del RPIImplMeta definen restricciones respecto del modo en que los distintos conceptos de la perspectiva de recursos son soportados por los WfMSs. De este modo, los RPIMs definidos a partir de este metamodelo pueden ser empleados para evaluar y comparar el soporte que los WfMSs proporcionan a los distintos conceptos de la perspectiva de recursos.

La Figura 14 muestra los elementos del RPIImplMeta que representan las entidades de implementación referidas a los aspectos Estructura de Recursos y Autorización Estática. El elemento *ResourceImpl* define la implementación de un elemento *Resource* de BPMN en un WfMS. Hay dos tipos de implementación de este elemento: *HumanResourceImpl* define la implementación de un elemento que representa un recurso humano en forma individual; *ResourceClassifierImpl* define la implementación de un elemento que representa un conjunto de recursos con características comunes. Estos elementos heredan tres atributos de *ResourceImpl*. El atributo *isReferenceable* especifica si un recurso puede ser referenciado por una consulta de recursos parametrizada. El atributo *isBindable* define cuándo el recurso puede ser devuelto como parte de una expresión de asignación de recursos. Por último, el atributo *bindingType* define si el elemento recurso puede ser evaluado contra un valor literal, una variable o una expresión combinando variables y funciones.

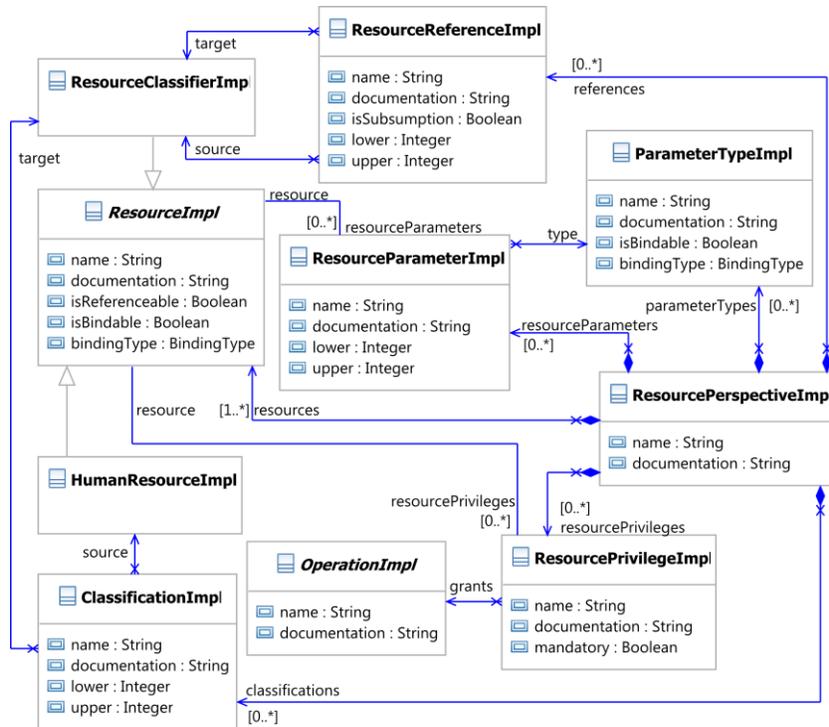


Figura 14. RPIImplMeta con los elementos de la Estructura de Recursos y Autorización Estática

El elemento *ResourceParameterImpl* representa una implementación de un elemento *ResourceParameter* de BPMN en un WfMS. El mismo especifica el tipo de parámetro que una plataforma de implementación permite definir para los recursos, junto con las cardinalidades mínimas y máximas (atributos *lower* y *upper*). Un tipo de parámetro (*ParameterTypeImpl*) define los atributos *isBindable* y *bindingType*, que permiten especificar si los parámetros de ese tipo pueden evaluarse contra un valor literal, una variable o una expresión.

El RPIImplMeta también provee elementos para definir las relaciones entre los elementos *HumanResourceImpl* y *ResourceClassifierImpl*. El elemento *ClassificationImpl* es una relación por la cual un recurso humano es incluido en la población de un clasificador de recurso. El elemento *ResourceReferenceImpl* especifica una relación dirigida entre dos clasificadores de recursos. Este elemento provee el atributo booleano *isSubsumption* para indicar si el clasificador de origen es un sub-clasificador del clasificador de destino o no. Ambos tipos de relación permiten definir cardinalidades máximas y mínimas (atributos *lower* y *upper*).

El elemento *ResourcePrivilegeImpl* especifica un privilegio a ser otorgado de manera estática a los recursos. Este tipo de privilegio puede ser otorgado a dos tipos de operaciones expresadas por los elementos *WorklistOperationImpl* y *WorkItemOperationImpl* (Figura 16). Una operación de lista de trabajo (elemento *WorklistOperationImpl*) especifica una

operación que permite a los recursos visualizar y organizar la lista de ítems de trabajo que le fueron asignados. Una operación de ítem de trabajo (elemento *WorkItemOperationImpl*) permite a los recursos modificar el estado de un ítem de trabajo. La modificación puede consistir en el establecimiento de su resultado, la asignación de recursos al mismo u otros cambios como la creación de comentarios o el adjuntado de archivos al mismo. El atributo *escalates* define los roles que una operación quita a los recursos. El atributo *triggers* define los roles que una operación asigna a los recursos.

La Figura 15 muestra cómo los elementos del RPMeta de la Figura 12 referencian los elementos del RPIImplMeta de la Figura 14 (los cuales se muestran dentro de los recuadros de línea discontinua) a fin de proveer un mecanismo para definir la implementación de aspectos Estructura de Recursos y Autorización Estática. De este modo, cada elemento del RPMeta tiene asociado un elemento del RPIImplMeta por medio del cual se define su implementación.

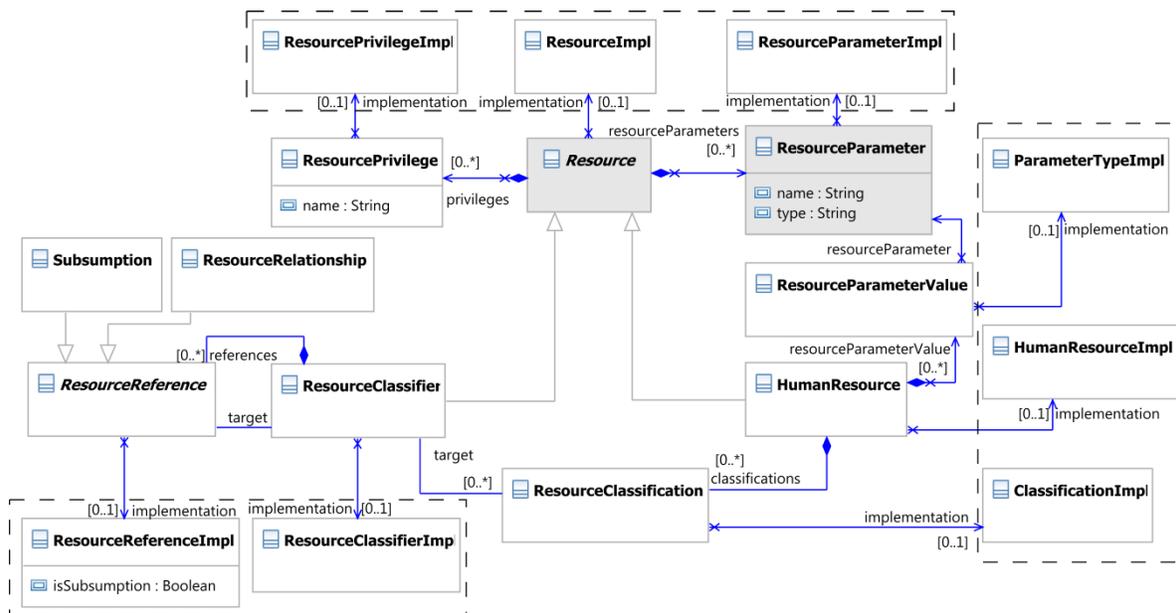


Figura 15. RPMeta haciendo referencia a los elementos del RPIImplMeta (Estructura de Recursos y Autorización Estática).

La Figura 16 muestra los elementos provistos por el RPIImplMeta para describir el soporte de un WfMS a los aspectos Distribución de Trabajo y Autorización Dinámica. El elemento *ResourceRoleImpl* especifica la implementación en un WfMS de un elemento *ResourceRole* de BPMN. Un rol de recurso es definido en términos de un conjunto de privilegios de tarea (elemento *TaskPrivilegeImpl*), los que son otorgados a los recursos para ejecutar operaciones de ítem de trabajo (elemento *WorkItemOperationImpl*). El elemento *ResourceRoleImpl* también tiene atributos que definen el modo en que un rol debe ser asignado a los recursos. El atributo *required* especifica si la asignación del rol es obligatoria para todas las tareas de usuario definidas en un modelo de proceso. El atributo

isAssignable especifica si está permitido o no definir una asignación de recursos para el rol en consideración. Si este atributo tiene un valor falso, el rol únicamente puede ser obtenido por los recursos por medio de la ejecución de operaciones de ítem de trabajo permitidas por otros roles. Finalmente, el atributo *constraintCategories* define un conjunto de elementos *RoleConstraintCategoryImpl* que especifican grupos de restricciones de resolución a ser aplicadas durante la asignación de recursos para el rol, con indicación de cardinalidades mínimas y máximas, y si está o no permitida la repetición de restricciones con la misma implementación.

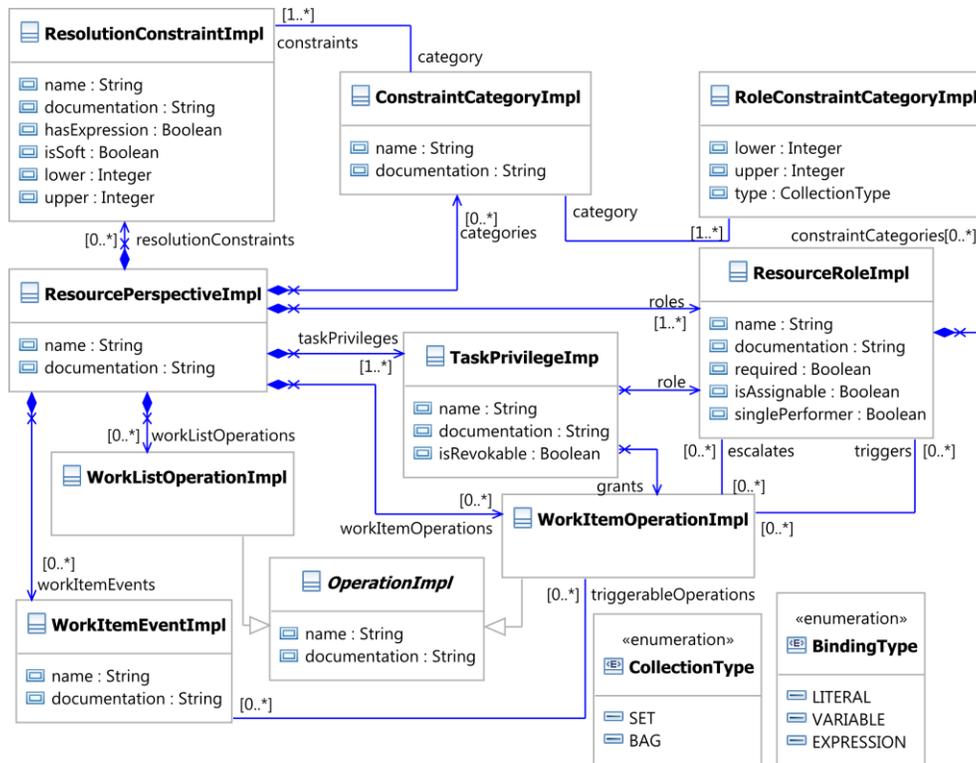


Figura 16. RPIImplMeta con los elementos de Distribución de Trabajo y Autorización Dinámica

El elemento *ResolutionConstraintCategoryImpl* define una estrategia para restringir los recursos que resultan de una consulta parametrizada de recursos o una expresión de asignación de recursos. El mismo incluye un atributo *hasExpression* para especificar si las instancias de las restricciones definidas tienen asociadas una expresión o no. El atributo *isSoft* permite distinguir entre restricciones de recursos duras y blandas. Este elemento también define cardinalidades mínimas y máximas para los conjuntos de recursos resultantes y una referencia al elemento *ConstraintCategoryImpl* para incluir la restricción en una categoría.

Finalmente, el elemento *WorkItemEventImpl* define un evento que se dispara y maneja en el alcance del ítem de trabajo. El mismo permite definir un conjunto de operaciones que

pueden ser ejecutadas como consecuencia de un evento con el propósito de modificar la asignación de recursos.

La Figura 17 muestra cómo los elementos del RPMeta de la Figura 13 referencian los elementos del RPIImplMeta de la Figura 16 (los cuales se encuentran dentro de los recuadros de línea discontinua) para permitir la implementación de los requerimientos de distribución de trabajo y autorización dinámica.

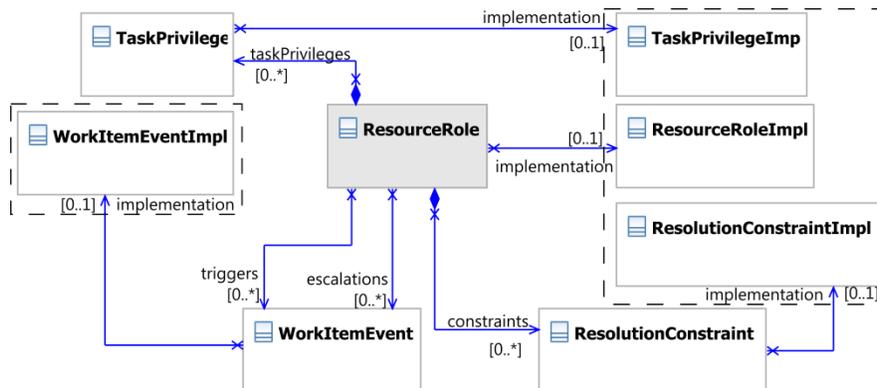


Figura 17. RPIImplMeta haciendo referencia a los elementos del RPIImplMeta (Distribución de Trabajo y Autorización Dinámica).

3.3.2. Modelo de Implementación de la Perspectiva de Recursos

Un Modelo de Implementación de la Perspectiva de Recursos (RPIM) es una instancia del RPIImplMeta, la cual representa las entidades provistas por un WfMS para implementar los aspectos Estructura de Recursos, Distribución de Trabajo y Autorización. Por medio de la definición de RPIMs, las distintas entidades empleadas por los WfMSs para dar soporte a esta perspectiva son representadas en términos del modelo conceptual definido por el RPMeta.

Un RPIM se construye por medio del análisis de las capacidades de un WfMS para dar soporte a la perspectiva de recursos. Este análisis puede efectuarse con base en la documentación de la plataforma o en forma directa a través de la instalación y empleo de las distintas funcionalidades provistas por la misma para dar soporte a dicha perspectiva. En el caso de WfMSs de código abierto, esto puede efectuarse también por medio del estudio de su implementación.

La Figura 18 muestra un ejemplo de RPIM que representa las entidades provistas por el WfMS Bonita⁸ para definir los aspectos Estructura de Recursos y Autorización Estática. Este modelo fue construido por medio del análisis de la documentación provista por el WfMS y de la evaluación directa del mismo por medio de su instalación y uso. El análisis

⁸ <http://documentation.bonitasoft.com/>

comenzó con la identificación de las entidades provistas para representar recursos humanos en forma individual. Dichas entidades son User y Administrator. Estas entidades, las cuales representan respectivamente a usuarios regulares y administradores del sistema, fueron definidas en el RPIM como elementos *HumanResourceImpl*.

El análisis de Bonita continuó con la identificación de los tipos de parámetros que pueden ser definidos, y de los privilegios que son o pueden ser concedidos a cada recurso por medio de las entidades User y Administrator. En este caso, Bonita provee dos tipos de parámetros, los cuales se representan por medio de elementos *ParameterTypeImpl*: Property y UserRef. Property representa cualquier propiedad de un recurso relevante para asignar trabajo al mismo. UserRef permite indicar una referencia a otro recurso del modelo en forma individual. Estos tipos de parámetros pueden ser definidos a usuarios y administradores por medio de elementos *ResourceParameterImpl*, representados en el diagrama de la Figura 18 mediante una línea continua con un rombo en su extremo inicial. En lo que respecta a los privilegios de recursos, estos se representan mediante flechas desde un elemento *ResourceImpl* a un elemento *WorklistOperationImpl* o *WorkItemOperationImpl*. La línea de la flecha es continua cuando el privilegio es obligatorio (atributo *mandatory* con valor verdadero) o discontinua cuando el privilegio es revocable (atributo *mandatory* con valor falso). La operación de lista de trabajo Reorder es permitida a todos los usuarios y administradores por medio de la implementación de privilegio de recursos reorder. Bonita no provee privilegios de recursos revocables. Adicionalmente, las operaciones de ítem de trabajo AssignTo y Unassign son permitidas a todos los administradores por medio de los privilegios de recurso assign y unassign.

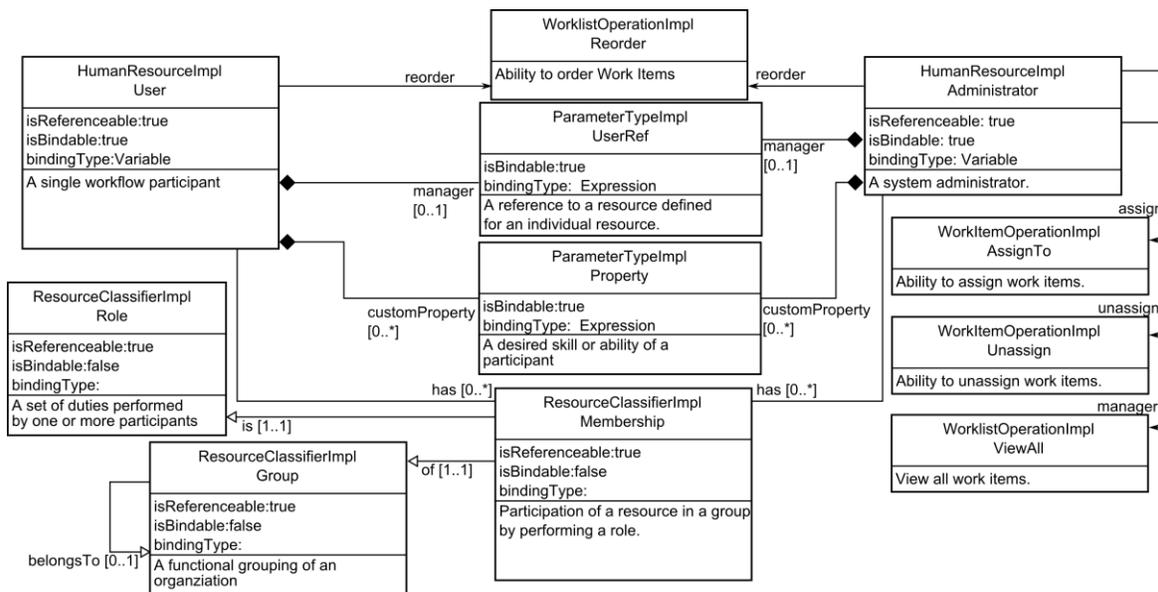


Figura 18. RPIM de los aspectos Estructura de Recursos y Autorización Estática del WfMS Bonita.

La construcción de un RPIM continúa con la identificación de las entidades provistas para representar agrupamientos de recursos, es decir los elementos *ResourceClassifierImpl*. Bonita provee tres entidades con este propósito: Group, Role y Membership. La entidad Group representa un agrupamiento funcional de la organización. La entidad Role representa un conjunto de responsabilidades de uno o más recursos. Finalmente, la entidad Membership representa el desempeño de un rol en un grupo.

Una vez identificadas las implementaciones de los clasificadores de recursos (elemento *ResourceClassifierImpl*), se definen los parámetros (elemento *ResourceParameterImpl*) y privilegios (elemento *ResourcePrivilegeImpl*) que pueden ser asignados a los recursos por medio de estos clasificadores. También se definen las relaciones (elemento *ResourceReferenceImpl*) que pueden presentarse entre clasificadores de recursos de distinto tipo y la forma en que los recursos pueden incluirse en la población de los tipos de clasificador identificados (elemento *ResourceClassificationImpl*). Bonita sólo permite definir parámetros y privilegios para los recursos en forma individual, esto se refleja en que no hay elementos *ResourceParameterImpl* o *ResourcePrivilegeImpl* asociados con los elementos *ResourceClassifierImpl* definidos. Los usuarios y administradores pueden ser clasificados únicamente por medio de la entidad Membership. Esto se define por medio de elementos *ResourceClassificationImpl* representados como líneas continuas. Los usuarios y administradores son incluidos en la población de los roles y grupos en forma indirecta a través de las relaciones de inclusión, las cuales se muestran en la figura mediante flechas con punta hueca. Esta plataforma también permite especificar jerarquías de grupos a través de la relación de inclusión *belongsTo*.

Posteriormente se debe evaluar el modo en que las entidades provistas para la estructura de recursos pueden ser empleadas para definir la asignación de recursos a las tareas de usuario de los procesos. En este caso, dado que las entidades de Bonita para representar recursos humanos en forma individual pueden ser referenciadas directamente en asignaciones de recursos o enlazadas como resultado de la evaluación de expresiones, las mismas presentan sus atributos *isReferenceable* e *isBindable* con valor verdadero. Estas entidades pueden ser evaluadas contra variables definidas en el alcance de las tareas, por lo cual su atributo *bindingType* se establece con el valor *Variable*. Las entidades provistas por Bonita para representar agrupamientos de recursos presentan por su parte el atributo *isReferenceable* con valor verdadero, por lo que pueden ser referenciadas por asignaciones de recursos definidas a nivel de tarea. Finalmente, los parámetros de tipo *Property* pueden ser evaluados en expresiones que incluyan variables y funciones, por lo que el valor del atributo *bindingType* fue establecido como *Expression*.

Luego, se continúa con el análisis de las entidades provistas por el WfMS para dar soporte a la implementación de los aspectos Distribución de Trabajo y Autorización

Dinámica. Las entidades provistas por Bonita con este propósito se muestran en la Figura 19. La definición de esta segunda parte del RPIM comienza con la creación de roles de recursos (elementos *ResourceRoleImpl*), por medio de los cuales es posible expresar los distintos tipos de política de distribución de trabajo. Con el propósito de determinar el tipo de política de distribución (*push* o *pull*) soportada por cada rol, se identifican las operaciones de ítem de trabajo (elementos *WorkItemOperationImpl*) autorizadas al rol. Las mismas se definen a través de privilegios de tarea (elementos *TaskPrivilegeImpl*), los cuales se representan mediante flechas con línea sólida o discontinua, según tengan el atributo *mandatory* con valor verdadero o falso, respectivamente.

Finalmente, se evalúa el modo en que cada implementación de rol de recurso puede ser asignada a los recursos. Esto se determina a través de las restricciones de resolución y los eventos de ítem de trabajo que pueden ser asociados con cada elemento *ResourceRoleImpl*, así como la definición de valores para los atributos *required*, *singlePerformer* e *isAssignable* del mismo.

En Bonita, el rol Actor representa una estrategia de distribución de trabajo de tipo *pull*. Esto se debe a que otorga privilegios de tarea para ejecutar las operaciones de ítem de trabajo *take* y *do*, las cuales permiten a los recursos asumir y comenzar la ejecución de ítems de trabajo voluntariamente. Este rol es asignable a múltiples recursos (el valor del atributo *singlePerformer* es *false*) y la asignación de recursos a este rol puede también ser limitada especificando restricciones de tipo *ActorFiltering*. Tal como se indica mediante la convención *Set(0..1)*, esta implementación de rol admite la especificación de una colección de tipo *Set* (sin elementos repetidos) de restricciones de categoría *ActorFiltering*, con una cardinalidad mínima 0 (la aplicación de restricciones es opcional) y una cardinalidad máxima de 1.

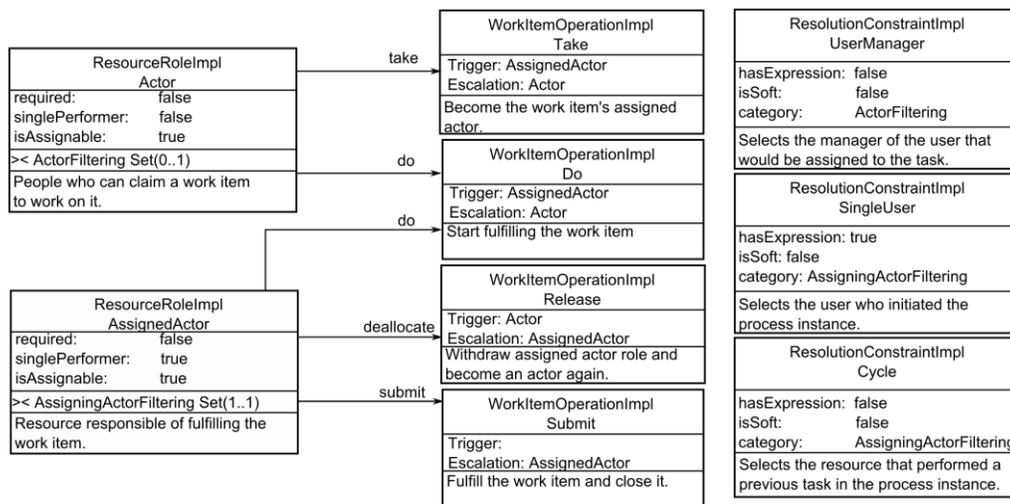


Figura 19. RPIM de Distribución de Trabajo y Autorización Dinámica del WfMS Bonita.

El segundo rol provisto por Bonita es *AssignedActor*, el cual solamente provee operaciones para cumplimentar ítems de trabajo. Por lo tanto, este rol representa una política de distribución de trabajo de tipo *push*. Este rol únicamente es asignable especificando restricciones de categoría *AssigningActorFiltering*. Esto se indica mediante la especificación de una cardinalidad mínima de 1 para las restricciones de esta categoría en la implementación de rol *AssignedActor*. Esto significa que este rol no puede ser asignado en un modelo de proceso sin especificar una restricción de esta categoría. Sin embargo, dicho rol puede ser obtenido en tiempo de ejecución por los recursos asignados al rol *Actor* por medio de la ejecución de las operaciones *take* o *do*, las cuales quitan a quien las ejecuta el rol *Actor* (atributo *escalation* del elemento *WorkItemOperationDefinition*) y asignan el rol *AssignedActor* (atributo *trigger* del elemento *WorkItemOperationDefinition*) permitidas por el rol *Actor*.

3.4. Extensiones de BPMN para Soportar la Perspectiva de Recursos

En esta sección se presentan las extensiones de Estructura de Recursos y Distribución de Trabajo definidas a BPMN con base en el RPMeta, las cuales permiten expresar los aspectos de la perspectiva de recursos en modelos de procesos BPMN. Las extensiones fueron desarrolladas empleando el método para extender BPMN propuesto en (Stropi, et al., 2011). A continuación se describe dicho método y posteriormente las extensiones.

3.4.1. Método para la Definición de Extensiones a BPMN 2.0

El metamodelo de BPMN 2.0 fue diseñado para ser extensible. El mismo provee un conjunto de elementos que permiten agregar atributos a los elementos predefinidos del lenguaje. Este mecanismo permite incorporar nuevos elementos en modelos BPMN manteniendo su portabilidad entre distintas herramientas compatibles con el lenguaje. Este enfoque de extensión, referido como extensión por adición o anotación, es distinto del provisto por lenguajes como UML, en donde las extensiones se basan en la especialización de sus elementos usando perfiles o metamodelos. Los elementos del mecanismo de extensión de BPMN 2.0 se muestran en la Figura 20.



Figura 20. Metamodelo del mecanismo de extensión de BPMN 2.0

Una definición de extensión (elemento *ExtensionDefinition*) consiste en un conjunto de atributos agrupados bajo un nombre, los cuales pueden ser agregados a cualquier elemento de BPMN. Cada atributo (elemento *ExtensionAttributeDefinition*) especifica un nombre y un tipo. El tipo puede ser obtenido desde cualquier espacio de nombres, incluyendo el de BPMN. De esta manera, a través de las definiciones de extensión con sus atributos se permite que los elementos de BPMN contengan nuevos elementos o atributos de tipos definidos tanto en el metamodelo de BPMN como en otros metamodelos.

Todo modelo BPMN consiste en una única instancia del elemento *Definitions*, el cual es el contenedor superior de todos los elementos del modelo, los cuales son instancias de subclases de *BaseElement*. A fin de aplicar una definición de extensión en un modelo BPMN, la misma debe ser importada y enlazada en el mismo por medio del elemento *Extension*. Este elemento permite incluir los elementos *ExtensionDefinition* y *ExtensionAttributeDefinition* de una extensión en el modelo. De este modo, cada elemento del modelo puede referenciar las definiciones de extensión (atributo *extensionDefinitions* de *BaseElement*) y definir valores para sus atributos por medio de elementos *ExtensionAttributeValue* (atributo *extensionValues* de *BaseElement*).

El método propuesto en (Stroppi, et al., 2011) permite definir extensiones a BPMN respetando su mecanismo de extensión. Este método (Figura 21) provee lineamientos para

que, partiendo de modelos conceptuales de la extensión, se obtenga una representación gráfica de la extensión en términos de los elementos provistos por el mecanismo de extensión de BPMN, y luego su posterior transformación automática en documentos XML Schema conformes al estándar BPMN. Estos documentos XML Schema son requeridos para que puedan ser procesados por herramientas definidas de acuerdo con el estándar BPMN 2.0.

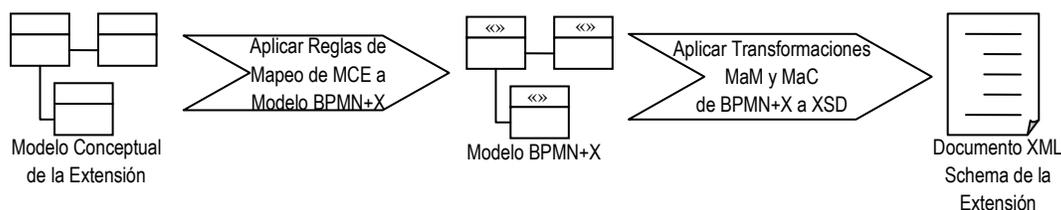


Figura 21. Método para definir extensiones de BPMN 2.0.

El primer paso del método consiste en definir un Modelo Conceptual de la Extensión (MCE) en el que se describen los conceptos del dominio a ser representados en modelos BPMN extendidos. El MCE es representado en diagramas de clases UML. En este modelo, las clases definidas son clasificadas como conceptos de BPMN o conceptos de la extensión.

El segundo paso del método consiste en desarrollar un modelo BPMN+X basado en el MCE resultante del paso anterior. BPMN+X es un lenguaje para la definición de extensiones a BPMN desarrollado como un perfil UML en (Stroppi, et al., 2011). Los estereotipos definidos en dicho perfil especializan elementos UML para representar los elementos del mecanismo de extensión de BPMN (Figura 22).

El perfil BPMN+X se basa en los elementos del mecanismo de extensión definido por el metamodelo de BPMN (Figura 20). El estereotipo *ExtensionModel* se aplica a un paquete UML a fin de emplearlo como contenedor de todos los elementos de una definición de extensión. Los estereotipos *BPMNEnum* y *ExtensionEnum* se aplican a enumeraciones para representar conjuntos de literales definidos en el metamodelo de BPMN y en el modelo de la extensión, respectivamente. Tres estereotipos fueron definidos para el elemento *Class* con el propósito de representar elementos de extensiones de BPMN: *BPMNElement*, *ExtensionDefinition* y *ExtensionElement*. *BPMNElement* representa un elemento original del metamodelo de BPMN. *ExtensionElement* define un nuevo elemento en el modelo de extensión. *ExtensionDefinition* representa un grupo de atributos a ser agregados a elementos BPMN existentes. Los elementos *ExtensionAttributeDefinition* del mecanismo de extensión de BPMN son representados en BPMN+X por medio de propiedades UML tanto pertenecientes a elementos *ExtensionDefinition* o navegables por medio de asociaciones. Las propiedades de los elementos *ExtensionDefinition* y *ExtensionElement* pueden ser de cualquiera de los tipos definidos por clases aplicando los estereotipos *BPMNElement* o *ExtensionElement*; enumeraciones aplicando los estereotipos *BPMNEnum*

o *ExtensionEnum*; o bien un tipo primitivo de UML. Por último, el estereotipo *ExtensionRelationship* especializa el elemento *Dependency* para especificar una relación conceptual entre un elemento *BPMN* y un elemento *ExtensionDefinition* que lo extiende.

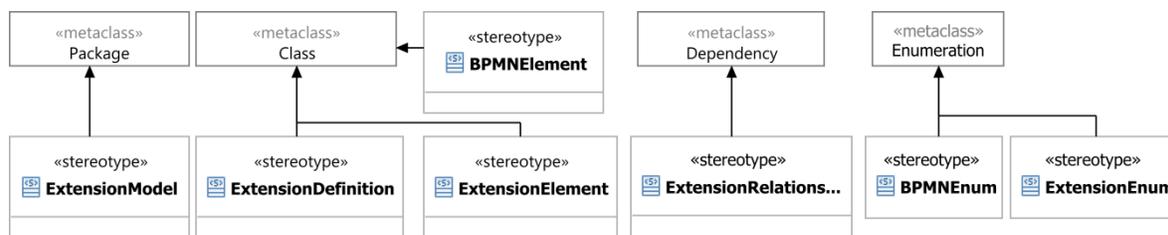


Figura 22. Perfil UML BPMN+X.

El método propuesto provee un conjunto de reglas que permiten a los desarrolladores de extensiones a BPMN determinar cómo se deben representar los distintos elementos de un modelo conceptual de la extensión en un modelo BPMN+X. El mismo provee además un conjunto de reglas OCL que permiten validar la corrección de las extensiones definidas desde el punto de vista estructural. Con base en los modelos BPMN+X resultantes, el método permite generar documentos XML Schema de las extensiones, los cuales definen el formato de intercambio para los modelos extendidos, y pueden ser compartidos entre distintas herramientas que implementen BPMN 2.0. En el Anexo I se incluyen los detalles acerca de las reglas de mapeo de MCE a BPMN+X, las reglas de verificación OCL y las transformaciones modelo a modelo y modelo a código definidas.

La aplicación de este método para extender BPMN consistió en tomar el RPMeta (Sección 3.2) como modelo conceptual de la extensión, el cual define los elementos de BPMN a reusar y los nuevos elementos propuestos para representar la perspectiva de recursos. A partir del mismo se definió un modelo BPMN+X por cada extensión propuesta, el cual expresa gráficamente la extensión respetando el mecanismo de extensión de BPMN. A continuación se describen las dos extensiones definidas como parte del marco de trabajo propuesto en esta tesis.

3.4.2. Extensión de Estructura de Recursos y Autorización Estática

La Extensión de Estructura de Recursos y Autorización Estática, denominada Resource Structure Extension (RSExt), tiene como propósito posibilitar la definición de los respectivos aspectos, por medio de la extensión de los elementos *Resource* y *ResourceParameter* de BPMN (Figura 23). La misma fue definida a partir de los elementos del RPMeta indicados en las Figuras 12 y 15.

El elemento *Resource* es extendido mediante la definición de extensión *ResourceBase*, como lo indica la relación de extensión entre los mismos. Esta definición de extensión agrega el atributo *resourcePrivileges* al elemento *Resource*, para representar el aspecto Autorización Estática. Las definiciones de extensión *HumanResource* y *ResourceClassifier*

especializan la definición de extensión *ResourceBase*. Las mismas heredan el atributo *resourcePrivileges* y añaden dos atributos: *type* e *implementation*. El primero es utilizado para anotar los elementos *Resource* indicando si los mismos representan un elemento *HumanResource* o un elemento *ResourceClassifier*. El segundo especifica la implementación de dicho elemento en un WfMS. La definición de extensión *HumanResource* añade además el atributo *parameterValues*, el cual almacena los valores de los parámetros que definen la caracterización de los recursos.

Tres definiciones de extensión fueron especificadas para el elemento *ResourceParameter*: *ValuedParameter*, *ResourceReference* y *ResourceClassification*. La definición de extensión *ValuedParameter* especifica la implementación de los parámetros que representan la caracterización de recursos. La definición de extensión *ResourceReference* especifica las relaciones de recursos (atributo *isSubsumption* con valor falso) y de inclusión (atributo *isSubsumption* con valor verdadero). La definición de extensión *ResourceClassification* especifica la pertenencia de un recurso humano a un clasificador de recurso. La representación de los elementos *ResourceReference* y *ResourceClassification* como definiciones de extensión sobre *ResourceParameter* (en lugar de como elementos de extensión) permite su posterior consulta empleando elementos *ResourceParameterBinding* de BPMN, de la misma forma que los parámetros con valor (*ValuedParameter*).

Aplicando esta extensión se obtiene un Modelo de Estructura de Recursos (RSM). Un RSM es un modelo BPMN que consiste de elementos *Resource* y *ResourceParameter* anotados empleando el mecanismo de extensión provisto por este lenguaje, donde dichas anotaciones corresponden a las extensiones propuestas. Esto hace posible importar un RSM en un modelo BPMN y definir asignaciones de recursos con base en los elementos del RSM, empleando a la vez los elementos estándares provistos por BPMN.

Un RSM es además un artefacto diseñado con el fin de facilitar la comunicación de los requerimientos relacionados con los aspectos Estructura de Recursos y Autorización Estática entre expertos de dominio, analistas funcionales y desarrolladores. Dado que BPMN no proporciona una notación para representar los elementos *Resource* y *ResourceParameter* en diagramas y que no existe una especificación estándar para la representación de dichos aspectos, en esta tesis se propone una notación desarrollada específicamente para permitir la visualización de RSMs, la cual se muestra en la Tabla 1.

Tabla 1. Notación gráfica de los Modelos de Estructura de Recursos.

Elemento	Notación
HumanResource	

ResourceClassifier	 NewResourceClassifier <<Implementation>> Documentation
ResourceParameter	 newResourceParameter
ResourcePrivilege	 newResourcePrivilege
ResourceClassification	
ResourceRelationship	
Subsumption	

Un RSM puede emplearse para definir requerimientos de estructura de recursos y autorización estática tanto a nivel conceptual (independiente de la plataforma) como de implementación (específico de la plataforma). Cuando se emplea este modelo a nivel conceptual, el significado de sus elementos es descripto en lenguaje natural, definiendo valores para los atributos *name* y *documentation* que los elementos *Resource* y *ResourceParameter* heredan de *BaseElement*. Por otra parte, cuando se define un RSM específico de la plataforma, la implementación de sus elementos con base en un WfMS se define haciendo referencia a los elementos de un RPIM, los cuales son importados como elementos de extensión (elementos enmarcados por líneas discontinuas en la Figura 23).



Figura 23. Extensión de Estructura de Recursos y Autorización Estática

La Figura 24 muestra un ejemplo de un RSM representando parte de la estructura de recursos de una empresa, la cual es un fabricante de muebles a medida. El RSM define un departamento de fabricación como un clasificador de recurso. Este departamento incluye los clasificadores de diseñador y de gerente de fabricación, los cuales representan puestos

de trabajo en la empresa. Esta inclusión se representa mediante flechas con punta hueca y corresponde a una referencia de recurso (atributo *isSubsumption* con valor verdadero). También se define una línea de reporte desde el clasificador diseñador al de gerente de fabricación por medio de una relación de recursos (atributo *isSubsumption* con valor falso), la cual se representa mediante una flecha con punta sólida.

Los miembros del clasificador diseñador se caracterizan por medio del parámetro especialidad. Esto permite distinguir a los diseñadores de acuerdo a su especialidad, es decir, según el tipo de mueble en cuyo diseño son más expertos. La membresía de los recursos humanos a los clasificadores de recursos que se mencionan arriba se especifican empleando relaciones de clasificación que se representan como líneas sólidas.

Este modelo también define el aspecto Autorización Estática por medio de la definición de privilegios de recursos. El marco de trabajo permite la definición de cualquier privilegio de acuerdo con los requerimientos de las organizaciones. En el ejemplo de la Figura 24, se representaron privilegios que corresponden con algunos patrones de recursos de workflow (Russell, et al., 2005). El privilegio de recursos *autonomiaDeSeleccion* es otorgado a los diseñadores y al gerente de fabricación para permitirles priorizar y organizar sus listas de trabajo, tal como lo indica el patrón de recursos número 26 (Autonomía de Selección). Además, el privilegio de recursos *verAsignados* es otorgado al gerente de fabricación para permitir a Juan supervisar a los diseñadores accediendo las listas de ítems de trabajo asignados a los mismos. Este privilegio fue definido con base en el patrón de recursos número 41 (Visibilidad Configurable de Ítems de Trabajo Asignados).

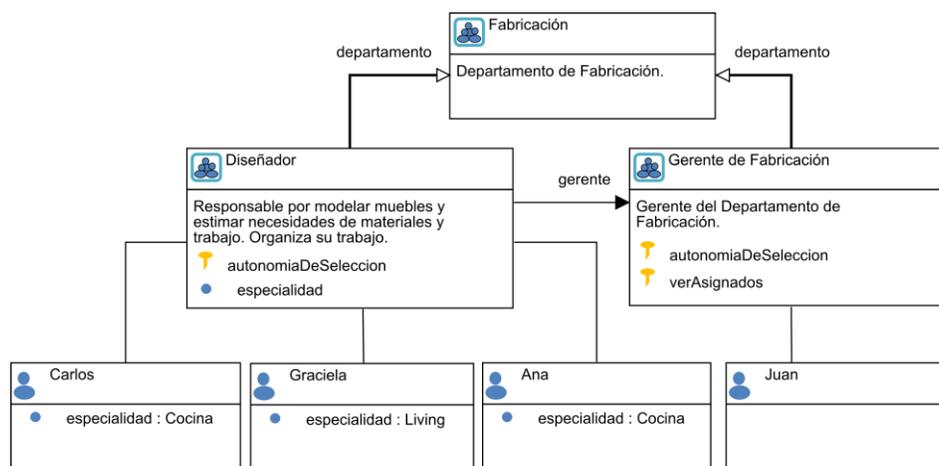


Figura 24. Notación propuesta para la visualización de RSMs.

3.4.3. Extensión de Distribución de Trabajo y Autorización Dinámica

La extensión de Distribución de Trabajo y Autorización Dinámica, denominada Work Distribution Extension (WText), define estos aspectos a través de atributos adicionales que

extienden los elementos *UserTask* y *ResourceRole* de BPMN (Figura 25). La misma fue definida a partir de los elementos del RPMeta indicados en las Figuras 13 y 17.

El elemento *UserTask* es extendido por medio de la definición de extensión *DynamicAuthorization*, la cual agrega los atributos *revokedPrivileges* y *workItemEvents*. El primero expresa los privilegios de tarea que no pueden ser otorgados a los recursos a través de su asignación a roles. El segundo define los eventos que pueden modificar la asignación de roles a recursos.

El elemento *ResourceRole* es extendido por medio de la definición de extensión *WorkDistribution*. Esta extensión especifica los atributos *taskPrivileges*, *constraints*, *triggers* y *escalations* introducidos en el RPMeta. Estos atributos definen: los privilegios concedidos por el rol, las restricciones sobre los recursos a ser asignados al mismo, y los eventos que disparan su asignación y la revocación de su asignación respectivamente.

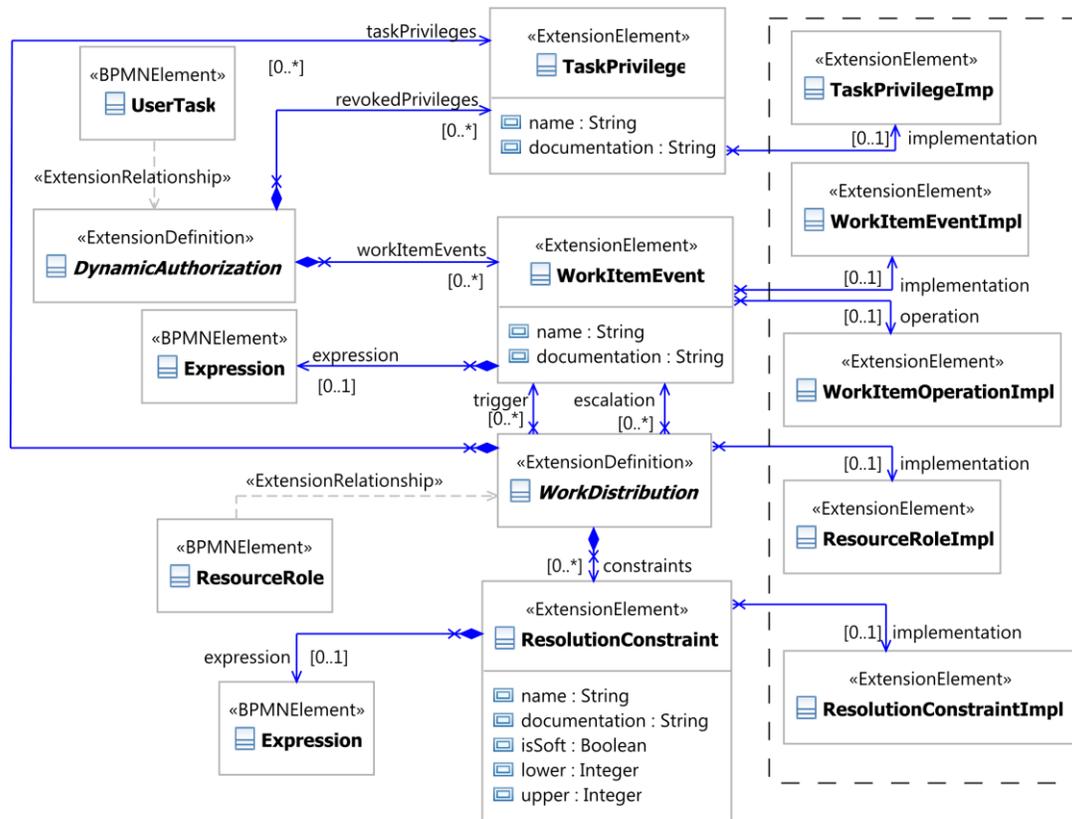


Figura 25. Extensión de Distribución de Trabajo y Autorización Dinámica.

Aplicando esta extensión se obtiene un Modelo de Proceso Extendido (EPM). Un EPM es un modelo de proceso BPMN en el que se definen los aspectos Distribución de Trabajo y Autorización Dinámica para las tareas de usuario definidas (elementos *UserTask*) por medio de las extensiones definidas para los elementos *UserTasks* y *ResourceRole* (Figura

25). Un EPM puede ser definido a nivel conceptual (o independiente de la plataforma) o a nivel de implementación (o específico de la plataforma). A nivel conceptual, dichos aspectos son descriptos por medio de los atributos *name* y *documentation* de los distintos elementos representando roles, privilegios de tarea, restricciones de resolución y eventos de ítem de trabajo definidos. A nivel de implementación, estos aspectos son descriptos referenciando los elementos de un RPIM (elementos enmarcados en líneas discontinuas en la Figura 25).

El EPM tiene como propósito permitir la comunicación de requerimientos y la toma de decisiones respecto de los aspectos Distribución de Trabajo y Autorización Dinámica de un proceso. Dado que BPMN no proporciona una notación gráfica para el elemento *ResourceRole*, y con el propósito de evitar la introducción de nuevos artefactos de notación en modelos de procesos BPMN, se adoptó una convención para la visualización de la definición de dichos aspectos, la cual se muestra junto con un ejemplo en la Figura 26. Dicha convención no adiciona nuevos elementos visuales a BPMN sino que hace uso del mecanismo de anotaciones de BPMN asociadas a elementos del modelo.

La Figura 26 muestra un ejemplo de EPM representando un proceso de gestión de presupuesto para la empresa fabricante de muebles a medida, cuyo RSM se describió anteriormente. Como ejemplo de la definición de los requerimientos de distribución de trabajo empleando los elementos de la extensión propuesta, se describe la tarea Elaborar Lista de Materiales. Los requerimientos a representar en esta tarea son:

1. La elaboración de la lista de materiales debería ser asignada a aquellos diseñadores especializados en el tipo de mueble solicitado.
2. La asignación de esta tarea a los diseñadores de una especialidad debería ser efectuada al recurso con menor cantidad de trabajo pendiente.
3. Un diseñador podría rechazar una asignación en esta tarea. En dicho caso se le asigna la misma al próximo diseñador en una lista.
4. Si el diseñador que tiene asignado la tarea no puede completarla en el término de dos días, se debe ofrecer la misma a todos los diseñadores, para que aquél que voluntariamente lo decida, pueda llevar adelante la ejecución de la tarea.

Para dar soporte a estos requerimientos en el EPM, se definen dos roles para esta tarea de usuario: Responsable y Candidato. La documentación de los mismos expresa informalmente sus semánticas.

El rol Responsable especifica que el diseñador asignado es responsable por la elaboración de la lista de materiales. Por lo tanto, éste representa una política de distribución de trabajo de tipo *push*. Este rol también permite a los recursos rechazar la responsabilidad por la ejecución del ítem de trabajo. El rol Responsable es asignado inicialmente a un diseñador cuya especialidad coincide con el tipo de mueble indicado en la

especificación del mueble solicitado. Esto es definido por medio de una expresión de asignación de recursos (RAE) y una restricción de resolución. La expresión de asignación permite obtener el conjunto de recursos pertenecientes al grupo de diseñadores correspondientes con el tipo de mueble especificado. La restricción (constraint) *RecursoMasLibre*, definida a partir del patrón de recursos número 17, expresa la asignación del ítem de trabajo al recurso con menor cantidad de ítems de trabajo en su lista de tareas. En caso que el recurso asignado rechace la responsabilidad por la ejecución de un ítem de trabajo de esta tarea, el mismo es asignado al siguiente recurso en el ciclo, de acuerdo con la restricción *RecursoMasLibre*.

El rol *Candidato* especifica que los diseñadores pueden asumir la responsabilidad por la ejecución de los ítems de trabajo libremente. Por lo tanto, éste define una estrategia de distribución de trabajo de tipo *pull*. Este rol es asignado a partir de la ocurrencia del evento especificado como disparador de la asignación (*Trigger*), con el propósito de representar el cambio de la distribución de los ítems de trabajo de la tarea en tiempo de ejecución, desde un recurso asignado al rol *Responsable* a los recursos asignados al rol *Candidato*. Para esto también se define como atributo del rol *Responsable* el evento que causa el cambio del rol. De esta manera, se expresa que si la tarea no se completa en dos días, este evento causa la revocación del rol *Responsable* y dispara la asignación del rol *Candidato* para todos los diseñadores sin tener en cuenta su especialidad.

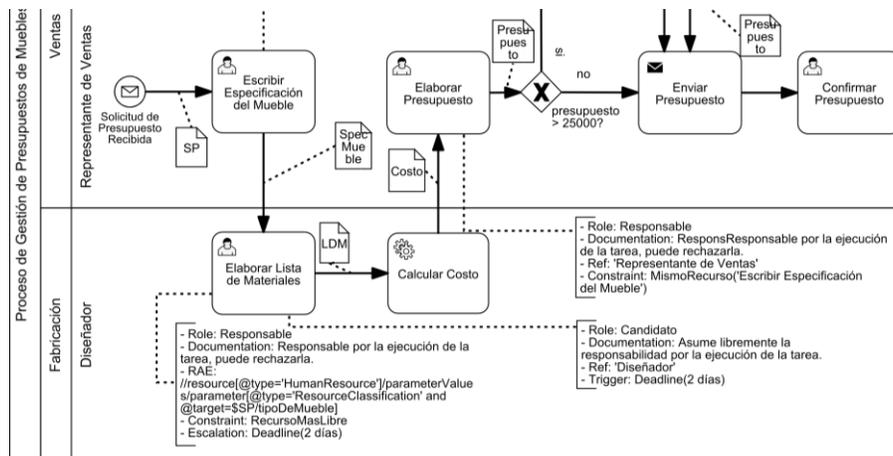


Figura 26. Ejemplo de EPM.

3.5. Conclusiones

El marco de trabajo presentado en este capítulo provee un modelo conceptual de la perspectiva de recursos de procesos de negocio, el cual se basa en tres aspectos identificados: Estructura de Recursos, Distribución de Trabajo y Autorización.

Los elementos del Metamodelo de la Perspectiva de Recursos (RPMeta) no incluyen conceptos que representan requerimientos concretos de la perspectiva de recursos, tales

como el rol Potential Owner de BPMN 2.0, el cual sólo define políticas de distribución de trabajo de tipo *pull*. En su lugar, cualquier tipo de roles, eventos de ítems de trabajo, privilegios y restricciones pueden ser descriptos en los modelos de procesos, a partir de los conceptos provistos en el RPMeta. Esto le confiere un carácter genérico, lo cual incrementa significativamente la cantidad de requerimientos de la perspectiva de recursos que pueden ser representados.

El Metamodelo de Implementación de la Perspectiva de Recursos (RPIimplMeta) definido como parte del marco de trabajo permite describir las entidades provistas por distintos WfMSs para implementar esta perspectiva con base en los conceptos del RPMeta. Cada elemento del RPIimplMeta permite definir el modo en que un concepto del RPMeta es implementado en distintos WfMSs. De este modo, el marco de trabajo brinda un mecanismo para mapear los diversos conceptos utilizados en los WfMSs para implementar la perspectiva de recursos con los del modelo conceptual propuesto.

Los tres tipos de artefactos que el marco de trabajo permite crear con base en RPMeta y RPIimplMeta son: Modelo de Estructura de Recursos (RSM), Modelo de Proceso Extendido (EPM) y Modelo de Implementación de la Perspectiva de Recursos (RPIM). Estos permiten representar requerimientos de la perspectiva de recursos y su implementación durante el desarrollo de SIOPs basados en WfMSs.

Las dos extensiones desarrolladas, Extensión de Estructura de Recursos (RSExt) y Extensión de Distribución de Trabajo (WDExt), permiten la instanciación de los elementos del RPMeta en MERs y MEPs. Además, los modelos resultantes son intercambiables entre distintas herramientas compatibles con BPMN 2.0.

La RSExt soporta la definición de los aspectos Estructura de Recursos y Autorización Estática, los cuales no caen dentro del alcance de BPMN. Esta extensión permite subsanar la falta de una especificación estándar para la representación de estos aspectos en modelos de procesos definidos con BPMN. Los RSMs creados mediante esta extensión pueden ser fácilmente importados y referenciados en modelos de procesos BPMN para permitir la definición de los restantes aspectos de la perspectiva de recursos. Al basarse en una extensión definida empleando el mecanismo de extensión de BPMN, un RSM es un modelo BPMN que consiste de elementos *Resource* y *ResourceParameter* ya provistos por dicho lenguaje. Esta extensión de BPMN permite representar un amplio rango de estructuras organizacionales en términos de dos conceptos genéricos: Recurso Humano y Clasificador de Recurso, de acuerdo a lo definido en el RPMeta. Dichas estructuras de recursos pueden ser representadas y descriptas a nivel conceptual y luego pueden ser mapeadas a elementos específicos de la plataforma por medio de la importación de un RPIM y la referencia a sus elementos. La notación gráfica propuesta para visualizar RSMs permite una mejor comunicación y toma de decisiones respecto de estos aspectos.

La WDExt permite definir los aspectos Distribución de Trabajo y Autorización Dinámica en EPMs. La misma provee un medio para describir los roles desempeñados por recursos humanos en tareas de usuario empleando el elemento *ResourceRole* ya provisto por BPMN. Esta extensión permite representar diferentes estrategias de distribución de trabajo por medio de la descripción de los distintos roles de recursos en cada modelo de procesos, de acuerdo a lo requerido por el proceso de negocio que está siendo definido. No define de antemano un conjunto restringido de roles de recursos para ser usados en los procesos. De este modo, es posible representar cualquier tipo de política de distribución de trabajo sin la necesidad de agregar un nuevo elemento al metamodelo y extensiones de BPMN por cada una de ellas. Esta extensión también permite definir requerimientos detallados de distribución de trabajo y autorización por medio de la definición de restricciones de resolución, eventos de ítem de trabajo y privilegios de tarea en MPEs. Estos requerimientos también pueden ser definidos a nivel conceptual o específico de una plataforma de implementación (WfMS) por medio de la importación de un RPIM y la referencia de sus elementos. Los requerimientos definidos en un EPM son visualizados en diagramas de modelos de procesos empleando artefactos de anotación de texto asociados con las tareas de usuario. Por medio del uso de esta convención de BPMN, se permite la visualización de los elementos adicionales introducidos por la WDExt sin la necesidad de crear nuevos artefactos de notación a BPMN.

Las instancias del RPIimplMeta denominadas Modelos de Implementación de la Perspectiva de Recursos (RPIMs) proporcionan una descripción estructurada de las entidades provistas por un WfMS para soportar la perspectiva de recursos. De este modo, un RPIM permite indicar el soporte que un WfMS proporciona a los requerimientos de la perspectiva de recursos definidos en un RSM y un EPM. Este artefacto puede además ser de ayuda en la evaluación y comprensión del funcionamiento de un determinado WfMS. El RPIimplMeta puede incluso ser empleado para describir distintos tipos de WfMS o plataformas de implementación de la perspectiva de recursos. El marco de trabajo provee una base para el desarrollo de un repositorio de RPIMs, con el propósito de ser usados para evaluar, comparar y emplear dichas plataformas en múltiples procesos por diversas organizaciones.

4 Método para Definir la Perspectiva de Recursos en el Desarrollo de SIOPs

En este capítulo se presenta un método para la definición de la perspectiva de recursos de procesos de negocio en el desarrollo de SIOPs. Se presenta una descripción general de las etapas del método (Sección 4.1). Se describe la etapa de definición de la lógica del proceso (Sección 4.2), la etapa de definición de los requerimientos de la perspectiva de recursos (Sección 4.3), la etapa de selección de la plataforma de implementación (Sección 4.4), la etapa de definición de la solución tecnológica (Sección 4.5), la etapa de verificación y validación de la solución tecnológica (Sección 4.6) y la etapa de generación de la especificación ejecutable (Sección 4.7). Finalmente se presentan las conclusiones del capítulo (Sección 4.8).

4.1 Etapas del Método

El método propuesto guía la definición de la perspectiva de recursos de procesos de negocio en las distintas etapas del desarrollo de SIOPs. El método está basado en los principios del desarrollo dirigido por modelos (Selic, 2003)(OMG, 2003) con dos propósitos: el de utilizar modelos de procesos de negocio y modelos que representan las plataformas de WfMS como principales artefactos del desarrollo; y el de automatizar la generación de especificaciones ejecutables a ser interpretadas por WfMS para implementar SIOPs a partir de dichos modelos. El método se basa en el marco de trabajo propuesto en el capítulo anterior para definir y generar los artefactos (modelos) resultantes de cada etapa del mismo. El objetivo del método es proveer un enfoque sistemático para agregar representaciones de los aspectos de la perspectiva de recursos en modelos conceptuales de procesos definidos con BPMN; y obtener especificaciones ejecutables de workflow y esquemas de recursos, los cuales constituyen los artefactos a ser importados en un WfMS para implementar un SIOP que soporte la ejecución los procesos definidos. Las etapas del método y sus artefactos resultantes se muestran en la Figura 27.

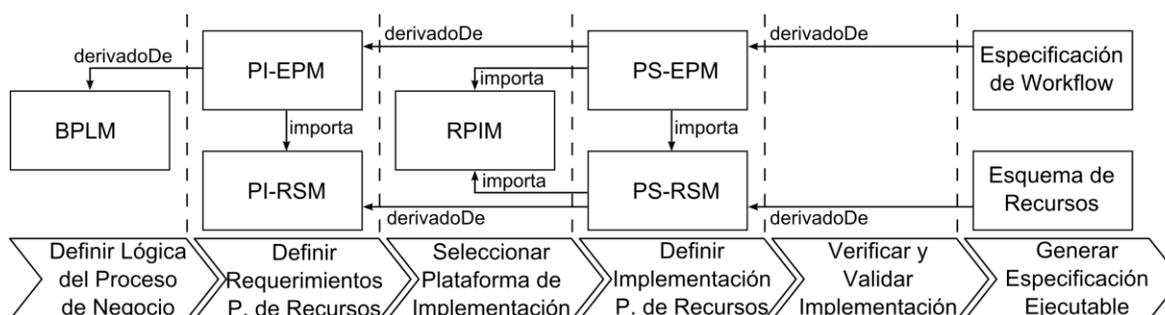


Figura 27. Etapas del método propuesto.

Las etapas del método propuesto pueden resumirse de la siguiente manera:

- **Etapa 1:** *Definir la Lógica del Proceso.* Consiste en identificar las tareas que requieren la intervención de personas en un proceso de negocio nuevo o existente, para el cual se requiere su implementación y ejecución a través de un SIOP. En esta etapa se representan primero las perspectivas de control y de datos del proceso en un modelo de proceso BPMN estándar, el cual se denomina Modelo de la Lógica del Proceso de Negocio o Business Process Logic Model (BPLM). Luego, se identifican las tareas del BPLM para las cuales se deben definir los aspectos de la perspectiva de recursos.
- **Etapa 2:** *Definir los Requerimientos de la Perspectiva de Recursos.* Consiste en definir los requerimientos de la perspectiva de recursos del proceso desde un punto de vista conceptual e independiente de la plataforma de implementación. Para ello se aplican las extensiones a BPMN propuestas como parte del marco de trabajo (Sección 3.4). Los recursos involucrados en la ejecución de las tareas del proceso y sus agrupaciones en clases de recursos (aspecto Estructura de Recursos) son definidos en un Modelo de Estructura de Recursos Independiente de la Plataforma, denominado Platform-Independent Resource Structure Model (PI-RSM). Con base en este modelo, la asignación de roles a recursos para gestionar y ejecutar los ítems de trabajo de las tareas (aspecto Distribución de Trabajo) es definida en un Modelo de Proceso Extendido Independiente de la Plataforma, denominado Platform-Independent Extended Process Model (PI-EPM). Este modelo es derivado a partir del BPLM, para respetar la lógica de este último, e importa el PI-RSM para referenciar los recursos y sus propiedades. La definición de los requerimientos de la perspectiva de recursos se completa incluyendo en ambos modelos información referente al modo en que los recursos pueden organizar y cumplimentar los ítems de trabajo (aspecto Autorización), por medio de la descripción de privilegios de recursos, roles y privilegios de tarea.
- **Etapa 3:** *Seleccionar la Plataforma de Implementación.* Consiste en seleccionar la plataforma de WfMS más adecuada para implementar los requerimientos de la perspectiva de recursos definidos en la etapa previa. El soporte provisto a estos requerimientos por diferentes WfMSs, cada uno de ellos representado en un Modelo de Implementación de la Perspectiva de Recursos (RPIM), es evaluado con base en un conjunto de propiedades estructurales. Dicha evaluación permite determinar el conjunto de entidades provistas por cada WfMS que son candidatas para implementar cada elemento de los modelos independientes de la plataforma. Además, la misma permite calcular tasas de soporte del WfMS al PI-RSM y al PI-EPM, lo cual provee una

base cuantitativa para la comparación de las plataformas y la selección de la plataforma más adecuada.

- **Etapa 4:** *Definir la Implementación de la Perspectiva de Recursos.* Consiste en definir cómo serán implementados los requerimientos de la perspectiva de recursos definidos en la segunda etapa, en términos de las entidades de implementación provistas por el WfMS seleccionado en la etapa anterior. Esto se define por medio de la creación de un Modelo de Estructura de Recursos Específico de la Plataforma, denominado Platform-Specific Resource Structure Model (PS-RSM), y un Modelo de Proceso Extendido Específico de la Plataforma (PS-EPM), denominado Platform-Specific Extended Process Model (PS-EPM). Ambos modelos son inicializados respectivamente a partir del PI-RSM y el PI-EPM. Los elementos de estos modelos específicos de la plataforma especifican referencias a los elementos del RPIM que representa la plataforma seleccionada.
- **Etapa 5:** *Verificar y Validar la Implementación de la Perspectiva de Recursos.* Consiste en constatar la adecuación del PS-RSM y el PS-EPM a las restricciones especificadas en el RPIM seleccionado (verificación); y en evaluar la consistencia del PS-RSM y el PS-EPM con los requerimientos definidos en el PI-RSM y el PI-EPM (validación). El método provee reglas y procedimientos a tal fin.
- **Etapa 6:** *Generar Especificaciones Ejecutables.* Consiste en la generación automática de especificaciones ejecutables de la perspectiva de recursos a partir del PS-RSM y el PS-EPM para el WfMS seleccionado. Esto requiere contar con transformaciones de modelo a código para el RPIM que representa dicho WfMS. Por medio de estas transformaciones es posible producir el código correspondiente a la especificación ejecutable del proceso según el formato requerido por el WfMS, a partir del PS-RSM y el PS-EPM.

En las siguientes secciones se describe cada una estas etapas en detalle.

4.2 Etapa 1: Definir la Lógica del Proceso

En esta etapa se define la lógica del proceso de negocio y se determinan las tareas que requieren la intervención de personas. La lógica del proceso de negocio es una representación de las posibles secuencias de tareas de un proceso y los documentos u objetos (información) pasados entre las tareas (Silver, 2011). Ésta involucra la definición de las perspectivas de control y de datos.

Las perspectivas de control y datos son representadas en un modelo de proceso BPMN estándar, al cual se lo denomina Modelo de la Lógica del Proceso de Negocio (BPLM). La definición detallada de dichas perspectivas cae fuera del alcance del método propuesto en esta tesis. Estas perspectivas pueden ser definidas con base en los lineamientos provistos

por metodologías existentes para definir la lógica de los procesos, tales como las propuestas en (Silver, 2011) o en (White, 2008). En esta etapa también se pueden aplicar enfoques para la verificación de la perspectiva de control como el propuesto en (Dijkman, et al., 2008) a fin de obtener un BPLM libre de errores, es decir, con ausencia de bloqueos (deadlocks) y ciclos infinitos (liveloops).

Las tareas del proceso que requieren la intervención de personas son aquellas definidas como tareas de usuario de BPMN, las cuales se diferencian de las tareas de tipo servicio, script o regla. Estas últimas representan la ejecución de una tarea en forma automática por una aplicación o motor de reglas. Las tareas de usuario son las que requieren la definición de los aspectos de la perspectiva de recursos. La información a ser presentada a los recursos por medio de la asignación de ítems de trabajo, los resultados esperados de su ejecución, y la información que puede ser empleada para la distribución de los mismos en tiempo de ejecución, es representada en el BPLM por medio de los objetos de datos asociados con los flujos de entrada y salida de las tareas de usuario.

Un BPLM puede también incluir elementos de tipo Lane, con el fin de indicar las tareas que se ejecutan en el contexto de una misma unidad organizacional o departamento de la organización. Estos elementos permiten representar visualmente el alcance del proceso en términos de las unidades organizacionales o departamentos involucrados en su ejecución y conocer los responsables últimos por la ejecución del proceso de principio a fin, lo cual se conoce como definición del gobierno del proceso. Si bien el elemento Lane no tiene como propósito específico representar conceptos de la perspectiva de recursos (Capítulo 2.2), este agrupamiento de tareas también puede ser tomado como base para definir los aspectos de la perspectiva de recursos como se discute en la siguiente sección.

4.3 Etapa 2: Definir los Requerimientos de la Perspectiva de Recursos

El propósito de esta etapa es obtener y definir los requerimientos de la perspectiva de recursos con analistas de negocio y expertos del dominio del proceso. El objetivo es crear representaciones de esta perspectiva que sean comprensibles por las distintas personas involucradas en el proceso, y que puedan ser tomadas como base para el desarrollo del SIOP que lo soporte. Durante esta etapa, los requerimientos de la perspectiva de recursos son definidos con base en los elementos provistos por el marco de trabajo descrito en el Capítulo 3, sin tener en cuenta aún la plataforma de implementación. Como resultado de esta etapa se obtienen dos modelos: un Modelo de Estructura de Recursos Independiente de la Plataforma (PI-RSM) y un Modelo de Proceso Extendido Independiente de la Plataforma (PI-EPM), los cuales son definidos empleando las extensiones a BPMN definidas en la Sección 3.4.

En la Sección 4.3.1, se provee una guía para la definición de un PI-RSM. En la Sección 4.3.2 se proveen lineamientos para la definición de un PI-EPM con base en el BPLM y el PI-RSM definidos previamente.

4.3.1 Definir el PI-RSM

Un PI-RSM define una representación conceptual de los aspectos Estructura de Recursos y Autorización Estática. En este modelo se emplean clasificadores de recursos y relaciones para capturar los agrupamientos de los recursos humanos. La caracterización de los mismos se define por medio de parámetros de recursos. El aspecto Autorización Estática se define por medio de los privilegios de los recursos. Tanto los parámetros como los privilegios de los recursos pueden ser especificados en los recursos humanos en forma individual o en los clasificadores de recursos que agrupan un conjunto de recursos humanos. Un PI-RSM puede definirse por medio de los siguientes pasos:

1. Definir los clasificadores de recursos.
2. Caracterizar los clasificadores de recursos.
3. Definir el aspecto Autorización Estática.
4. Definir relaciones entre clasificadores de recursos.
5. Identificar la población (recursos humanos) de los clasificadores de recursos.

Definir los Clasificadores de Recursos

La definición de un PI-RSM comienza con la identificación de las clases de recursos involucrados en el proceso. Esto puede ser efectuado siguiendo alguno de los enfoques básicos para la definición de modelos de recursos: el *enfoque dirigido por el workflow* o el *enfoque dirigido por la organización* (Zur Muehlen, 2004).

El *enfoque dirigido por el workflow*, se aplica generalmente en organizaciones que no cuentan con una estructura previamente formalizada. Cuando se aplica este enfoque, cada uno de los clasificadores de recursos definidos representa el grupo de recursos responsables de la ejecución de una tarea de usuario del BPLM. De este modo, inicialmente se define un clasificador de recurso por cada tarea de usuario del proceso. Posteriormente, el conjunto de clasificadores de recursos resultante se refina agrupando clasificadores con correspondientes a tareas que requieran competencias comunes a los recursos que las llevan a cabo; o bien desglosando clasificadores correspondientes a tareas que involucren distintos recursos dependiendo del contexto en que se ejecuten.

El *enfoque dirigido por la organización*, se aplica generalmente en organizaciones con estructuras previamente definidas. En este caso, los clasificadores de recursos definidos representan agrupamientos de personas basados en la descripción de las unidades de la organización y las posiciones o puestos de trabajo que las personas que desempeñan en las mismas. La aplicación de este enfoque no implica la necesidad de representar la estructura completa de la organización en el PI-RSM. El alcance del mismo puede restringirse para

representar únicamente las unidades organizacionales involucradas en el proceso definido. En el caso que el BPLM incluya elementos Lane representando el gobierno del proceso, dichos elementos pueden ser tomados como base para este propósito.

Ambos enfoques presentan ventajas y desventajas. El *enfoque dirigido por el workflow* conduce generalmente a PI-RSMs simples y alineados con los BPLMs. Sin embargo, este enfoque conlleva frecuentemente la necesidad de cambiar el PI-RSM ante cambios en la lógica del proceso. Por otro lado, los modelos resultantes suelen ser más difíciles de reutilizar en otros modelos de procesos. El *enfoque dirigido por la organización* generalmente resulta en modelos de estructura de recursos más complejos, los cuales son en cambio más fáciles de reutilizar y evolucionan con más independencia de los procesos. Por lo tanto, la selección de alguno de estos enfoques puros o de algún enfoque intermedio debe realizarse teniendo en cuenta la existencia o no de una estructura organizacional definida y la necesidad futura de reutilizar el PI-RSM para definir la perspectiva de recursos de otros procesos.

Caracterizar los Clasificadores de Recursos

La definición de un PI-RSM continúa con la caracterización de los clasificadores de recursos por medio de la definición de parámetros de recursos, para representar las propiedades de los recursos que resultan relevantes a los efectos de distribuir trabajo entre los mismos. Los parámetros de recursos se emplean para definir atributos que permiten determinar cuál de los recursos pertenecientes a un determinado clasificador de recurso es más adecuado para ejecutar un ítem de trabajo en el contexto de una determinada instancia de tarea de usuario. Ejemplos de propiedades de caracterización de los recursos son: el conocimiento de un determinado lenguaje, su ubicación geográfica, su horario de trabajo, o el monto máximo autorizado para realizar una determinada transacción.

Definir el aspecto Autorización Estática

Luego, los requerimientos de autorización de los recursos pertenecientes a cada clasificador de recurso son expresados por medio de la definición de privilegios de recursos, los cuales determinan las operaciones que son permitidas a los mismos para gestionar los ítems de trabajo de cualquier tarea que se les asigne (aspecto Autorización Estática). Ejemplos de tales requerimientos son la capacidad de los recursos para: organizar los ítems de trabajo, de aceptar o rechazar la responsabilidad por la ejecución de un ítem de trabajo, o bien de delegar su ejecución en otros recursos. Los requerimientos recurrentes de este tipo identificados por los Patrones de Recursos de Workflow (Russell, et al., 2005) pueden ser tomados como vocabulario de base para definir estos requerimientos.

Definir relaciones entre clasificadores de recursos

Una vez identificados los clasificadores de recursos del PI-RSM y definidos los parámetros y privilegios de los recursos pertenecientes a los mismos, se deben definir las

dependencias entre estos clasificadores. Estas dependencias se definen por medio de dos tipos de referencias: la relación de recursos y la de inclusión. Los vínculos de los recursos pertenecientes a un clasificador con los recursos pertenecientes a otro que resulten de relevancia para la distribución de ítems de trabajo, tales como las líneas de reporte, son representados por medio de relaciones de recursos. Por otra parte, las inclusiones se emplean para definir la incorporación de la población de un clasificador de orden inferior, en la de un clasificador de orden superior, como sucede por ejemplo en las jerarquías de unidades y sub-unidades tales como departamentos y sub-departamentos; o bien para agrupar conjuntos de parámetros y privilegios comunes de dos o más clasificadores en otro de orden superior.

Identificar la Población de los Clasificadores de Recursos

Finalmente, la población de los clasificadores definidos es representada por medio de recursos humanos conectados con los respectivos clasificadores por medio de relaciones de clasificación. El propósito de representar los miembros de los clasificadores en esta instancia es permitir a los expertos de dominio visualizar el agrupamiento y la caracterización de recursos resultante, como así también proveer una base para la validación de las políticas de distribución de trabajo definidas como se discute más adelante.

4.3.2 Definir el PI-EPM

Un PI-EPM define una representación conceptual de los aspectos Distribución de Trabajo y Autorización Dinámica. Estos aspectos se definen por medio de la asignación de roles a recursos. Los tipos de política de distribución de trabajo (*push* o *pull*) dependen del significado de los roles asignados. La definición de estos aspectos puede llevarse a cabo por medio de los siguientes pasos para cada tarea de usuario definida en el BPLM:

1. Identificar y describir los roles de recursos.
2. Definir la asignación de recursos a cada rol.
3. Definir restricciones de resolución.
4. Definir eventos de ítem de trabajo.

Identificar y Describir los Roles de Recursos

La definición de los aspectos Distribución de Trabajo y Autorización Dinámica de una tarea de usuario comienza con la identificación y descripción de los roles a ser desempeñados por los recursos en la misma. Esto implica describir las responsabilidades de los recursos en relación con los ítems de trabajo correspondientes con las instancias de la tarea y el modo en que estas responsabilidades son establecidas para los mismos. En esta etapa, los roles de recursos son descriptos de manera informal por medio de los atributos *name* y *documentation* del elemento *ResourceRole*.

El uso más frecuente del rol de recurso es definir la asignación de la responsabilidad por la ejecución de los ítems de trabajo de una tarea. Sin embargo, este elemento puede emplearse para definir otros tipos de responsabilidades o papeles a ser desempeñados por las personas en los ítems de trabajo. Estos papeles pueden incluir el desempeño de funciones de índole administrativa, de supervisión, de asesoramiento o de monitoreo de eventos relevantes en la ejecución de los ítems de trabajo.

La descripción de un rol de recurso de una tarea también debe incluir lineamientos respecto del modo en que la responsabilidad descrita es establecida para los recursos y la forma en que los recursos pueden influir en la asignación de responsabilidades sobre el ítem de trabajo. Esto es, la descripción del rol de recurso debe indicar si el SIOP resultante asignará dicha responsabilidad en forma directa (política de distribución de trabajo de tipo *push*) o si la responsabilidad es ofrecida por el sistema y asumida en forma voluntaria por los recursos (política de distribución de trabajo *pull*); y si los recursos son capaces de modificar dicha asignación de responsabilidades por medio de la ejecución de desvíos tales como la delegación, el rechazo o la reasignación descritas por los patrones de recursos de workflow (Russell, et al., 2005).

Definir la Asignación de Recursos a cada Rol

La asignación de recursos a los roles identificados es expresada inicialmente por medio de los mecanismos provistos por BPMN: consultas parametrizadas de recursos y expresiones de asignación de recursos.

Las consultas parametrizadas de recursos permiten definir la asignación de recursos referenciando clasificadores de recursos definidos en el PI-RSM y estableciendo condiciones sobre los valores de sus parámetros; o bien referenciando un recurso humano en forma directa. Este tipo de asignación de recursos también se conoce como asignación basada en la estructura de recursos.

Las expresiones de asignación de recursos permiten determinar la identidad de los recursos a ser asignados a un rol por medio de la evaluación de datos definidos en el alcance de la instancia de la tarea. Esto se conoce como asignación basada en datos.

Definir Restricciones de Resolución

La asignación de recursos definida puede ser refinada por medio de la definición de restricciones de resolución, las cuales permiten limitar el conjunto de recursos asignados a un rol de la tarea con base en distintos criterios. Ejemplos de este tipo de restricción son la separación de responsabilidades, en la que los recursos que desempeñaron un rol en una tarea de usuario previa son excluidos del rol siendo asignado. Otro tipo de restricciones de resolución son aquellas que permiten seleccionar un único recurso del conjunto definido inicialmente, tales como la asignación cíclica, aleatoria o del recurso con menor carga de trabajo.

Definir Eventos de Ítem de Trabajo

Finalmente, se definen eventos que puedan causar cambios en los roles asignados a recursos para los ítems de trabajo de una tarea de usuario. Esto se expresa por medio de la definición, a nivel de la tarea de usuario, de eventos que afecten la asignación de los roles. El ejemplo típico de evento de este tipo son las fechas límite. Los distintos roles definidos para una tarea pueden referenciar a los eventos definidos para la tarea por medio de los atributos *escalation* o *trigger*. Ante la ocurrencia en tiempo de ejecución de un evento para un determinado ítem de trabajo, los roles que lo referencien como *escalation* serán retirados a los recursos, por lo cual el ítem de trabajo ya no será accesible a dichos recursos. Por otra parte, los roles que referencien al evento como *trigger* serán asignados a los recursos únicamente ante la ocurrencia del mismo.

4.4 Etapa 3: Seleccionar la Plataforma de Implementación

Esta etapa tiene como propósito la selección de un WfMS apropiado para implementar el proceso en consideración. Para ello, el método provee una técnica que permite evaluar y comparar el soporte provisto por distintos WfMSs a los requerimientos de la perspectiva de recursos definidos en el PI-RSM y el PI-EPM. Esta técnica requiere la representación de cada WfMS a ser evaluado en un RPIM, como el que se describió en el Capítulo 3, Sección 3.3.2 (Figuras 18 y 19). El RPIM de cada WfMS a ser evaluado puede ser construido en esta etapa o bien tomado de un repositorio de modelos si ya estuviera definido.

La técnica propuesta para evaluar un WfMS se basa en un *conjunto de propiedades de evaluación* a ser aplicadas a: los elementos de los modelos independientes de la plataforma (PI-RSM y PI-EPM) resultantes de la etapa anterior, y los elementos de los RPIMs que representan las plataformas de implementación candidatas. Esto es posible dado que el marco de trabajo propuesto permite describir los requerimientos de la perspectiva de recursos y su implementación en términos de los mismos conceptos.

El conjunto de propiedades de evaluación con valor verdadero para un elemento de un PI-RSM o PI-EPM se denomina *conjunto de propiedades requeridas*. El conjunto de propiedades de evaluación con valor verdadero para un elemento de un RPIM se denomina *conjunto de propiedades soportadas*. Si el conjunto de propiedades requeridas de un elemento de un PI-RSM o PI-EPM está incluido en el conjunto de propiedades soportadas por un elemento de un RPIM, se dice que el elemento del PI-RSM o PI-EPM es soportado por el WfMS representado por el RPIM. Por lo tanto, es posible calcular una *tasa de soporte* de un RPIM a un PI-RSM o PI-EPM como el cociente entre el número de elementos soportados y el número de elementos evaluados del PI-RSM o PI-EPM. Esto provee una base para la comparación de diferentes WfMSs y la selección de uno como plataforma de implementación para el SIOP siendo desarrollado. Con base en estas propiedades, es posible seleccionar el WfMS por medio de los siguientes pasos:

1. Determinar los conjuntos de propiedades requeridas.
2. Determinar los conjuntos de propiedades soportadas.
3. Evaluar y seleccionar la plataforma de implementación.

Determinar los Conjuntos de Propiedades Requeridas

Determinar, para cada uno de los elementos del PI-RSM y el PI-EPM, su *conjunto de propiedades requeridas*. Por medio de su definición, se provee el criterio para determinar si una determinada entidad de implementación definida en un RPIM puede o no ser considerada como candidata para implementar el elemento del PI-RSM o PI-EPM en consideración.

La Tabla 2 muestra las propiedades de evaluación definidas para los aspectos Estructura de Recursos y Autorización Estática. Las mismas se evalúan para cada uno de los recursos humanos y clasificadores de recursos (elementos *Resource* de tipo *HumanResource* o *ResourceClassifier*) definidos en un PI-RSM importado por un PI-EPM. El resultado de la evaluación de dichas propiedades para cada elemento del PI-RSM es almacenado en el encabezado (en negrita) de una fila de una tabla de resultados de evaluación, cuya estructura se muestra en la Tabla 3.

Tabla 2. Propiedades de Evaluación de Estructura de Recursos y Autorización Estática.

Propiedad	Condición de Verdad en elemento Resource de un PI-RSM importado por un PI-EPM
<i>isResource</i>	Es de tipo <i>HumanResource</i> .
<i>isClassifier</i>	Es de tipo <i>ResourceClassifier</i> .
<i>isReferenceable</i>	Es objeto de una referencia <i>resourceRef</i> de un elemento <i>ResourceRole</i> del PI-EPM.
<i>isBindable</i>	Existe al menos un elemento <i>ResourceAssignmentExpression</i> en el PI-EPM.
<i>subsumes</i>	Es destino de una relación de inclusión (<i>ResourceReference</i> con atributo <i>isSubsumption</i> con valor verdadero).
<i>isSubsumed</i>	Es origen de una relación de inclusión (<i>ResourceReference</i> con atributo <i>isSubsumption</i> con valor verdadero).
<i>isTarget</i>	Es destino de una relación de recursos (<i>ResourceReference</i> con atributo <i>isSubsumption</i> con valor falso).
<i>isSource</i>	Es origen de una relación de recursos (<i>ResourceReference</i> con atributo <i>isSubsumption</i> con valor falso).
<i>hasPopulation</i>	Es de tipo <i>ResourceClassifier</i> y es destino de una relación <i>ResourceClassification</i>
<i>isMember</i>	Es de tipo <i>HumanResource</i> y es origen de una relación <i>ResourceClassification</i> .
<i>hasPrivileges</i>	Tiene un conjunto de <i>resourcePrivileges</i> mayor que cero.
<i>hasParameters</i>	Tiene un conjunto de <i>resourceParameters</i> mayor que cero.
<i>hasBindableParams</i>	Al menos uno de los elementos <i>ResourceParameter</i> en el conjunto de <i>resourceParameters</i> se encuentra en el conjunto <i>parameterRef</i> de un elemento <i>ResourceParameterBinding</i> .

Tabla 3. Tabla de Resultados de Evaluación de Estructura de Recursos y Autorización Estática.

	isResource	isClassifier	isReferenceable	isBindable	subsumes	isSubsumed	isSource	isTarget	hasPopulation	isMember	hasPrivileges	hasParameters	hasBindableParams
Elemento Resource 1 Elemento ResourceImpl 1 Elemento ResourceImpl n													
Elemento Resource 2 Elemento ResourceImpl 1 Elemento ResourceImpl n													
Otros ResourceImpl Elemento ResourceImpl m													

La Tabla 4 muestra las propiedades de evaluación definidas para los aspectos Distribución de Trabajo y Autorización Dinámica. Estas propiedades son evaluadas para los roles de recursos (elemento *ResourceRole*) definidos en el PI-EPM. El resultado de la evaluación de estas propiedades para cada elemento del PI-EPM es almacenado en el encabezado (en negrita) de una fila de una tabla de resultados de evaluación, cuya estructura se muestra en la Tabla 5.

Tabla 4. Propiedades de Evaluación de Distribución de Trabajo y Autorización Dinámica

Propiedad	Condición de Verdad en elemento ResourceRole de un PI-EPM
<i>isAssignable</i>	Para todo ResourceRole.
<i>multiplePerformers</i>	No define una referencia a un único elemento Resource de tipo HumanResource ni define una ResolutionConstraint con el atributo <i>upper</i> igual a uno.
<i>singlePerformer</i>	Define una referencia a un único elemento Resource de tipo HumanResource o define una ResolutionConstraint con el atributo <i>upper</i> igual a uno.
<i>hasHardConstraints</i>	Tiene al menos un elemento ResolutionConstraint con el atributo <i>isSoft</i> con valor falso en el conjunto de constraints.
<i>hasSoftConstraints</i>	Tiene al menos un elemento ResolutionConstraint con el atributo <i>isSoft</i> con valor verdadero en el conjunto de constraints.
<i>hasTriggers</i>	Tiene al menos un elemento en el conjunto trigger.
<i>hasEscalations</i>	Tiene al menos un elemento en el conjunto escalation.
<i>revokablePrivileges</i>	Pertenece a una tarea de usuario con al menos un elemento en el conjunto de revokedPrivileges.

Tabla 5. Tabla de Resultados de Evaluación de Distribución de Trabajo y Autorización Dinámica.

	isAssignable	multiplePerformers	singlePerformer	hasHardConstraints	hasSoftConstraints	hasTriggers	hasEscalations	revokablePrivileges
Elemento ResourceRole 1 Elemento ResourceRoleImpl 1 Elemento ResourceRoleImpl n								
Elemento ResourceRole 2 Elemento ResourceRoleImpl 1 Elemento ResourceRoleImpl n								
Otros ResourceRoleImpl Elemento ResourceRoleImpl m								

Determinar los Conjuntos de Propiedades Soportadas

Evaluar, para cada WfMS candidato, el soporte que cada una de las entidades de implementación definidas en el correspondiente RPIM proporciona a cada elemento definido en el PI-RSM y el PI-EPM resultantes de la segunda etapa. Por medio de esta evaluación se define el conjunto de *entidades de implementación candidatas* provistas por cada WfMS para los elementos del PI-RSM y el PI-EPM, y se reúne la información necesaria para calcular la tasa de soporte de cada WfMS a cada uno de estos modelos.

Para cada WfMS evaluado, se completan las tablas de resultados de evaluación correspondientes con los aspectos Estructura de Recursos y Autorización Estática; y con los aspectos Distribución de Trabajo y Autorización Dinámica. El resultado de la evaluación de las propiedades definidas para cada entidad provista por el WfMS representado por el RPIM para implementar recursos y roles de recursos (elementos ResourceImpl y ResourceRoleImpl) es empleado para completar la información de dichas tablas. De este modo, cuando un elemento *ResourceImpl* o *ResourceRoleImpl* evaluado satisfaga el *conjunto de propiedades requeridas* de un elemento *Resource* o *ResourceRole*, su nombre será colocado en el cuerpo de la fila correspondiente a este último, junto con los valores de sus propiedades.

Una vez completadas estas tablas, las mismas constituyen un resumen del conjunto de implementaciones candidatas provistas por un WfMS para los elementos del PI-RSM y el PI-EPM.

El soporte de un RPIM a los aspectos Estructura de Recursos y Autorización Estática se calcula con base en el conjunto de propiedades de evaluación que se muestran en la

Tabla 6. Estas propiedades fueron definidas para los elementos *ResourceImpl* (de tipo *HumanResourceImpl* o *ResourceClassifierImpl*) presentes en los RPIMs.

Tabla 6. Propiedades de Evaluación de Entidades de Implementación a los Aspectos Estructura de Recursos y Autorización Estática.

Propiedad	Condición de Verdad en elemento ResourceImpl de un RPIM
<i>isResource</i>	Es de tipo <i>HumanResourceImpl</i> .
<i>isClassifier</i>	Es de tipo <i>ResourceClassifierImpl</i> .
<i>isReferenceable</i>	Tiene el atributo <i>isReferenceable</i> con valor verdadero.
<i>isBindable</i>	Tiene el atributo <i>isBindable</i> con valor verdadero
<i>Subsumes</i>	Es destino de una implementación de relación de inclusión (<i>ResourceReferenceImpl</i> con atributo <i>isSubsumption</i> con valor verdadero).
<i>isSubsumed</i>	Es origen de una implementación de relación de inclusión (<i>ResourceReferenceImpl</i> con atributo <i>isSubsumption</i> con valor verdadero).
<i>isTarget</i>	Es destino de una relación de recursos (<i>ResourceReferenceImpl</i> con atributo <i>isSubsumption</i> con valor falso).
<i>isSource</i>	Es origen de una relación de recursos (<i>ResourceReferenceImpl</i> con atributo <i>isSubsumption</i> con valor falso).
<i>hasPopulation</i>	Es de tipo <i>ResourceClassifierImpl</i> y es destino de una relación <i>ResourceClassificationImpl</i> .
<i>isMember</i>	Es de tipo <i>HumanResourceImpl</i> y es origen de una relación <i>ResourceClassification</i> .
<i>hasPrivileges</i>	Tiene un conjunto de <i>resourcePrivileges</i> mayor que cero.
<i>hasParameters</i>	Tiene un conjunto de <i>resourceParameters</i> mayor que cero.
<i>hasBindableParams</i>	Al menos uno de los elementos <i>ResourceParameterImpl</i> en el conjunto de <i>resourceParameters</i> tiene el atributo <i>isBindable</i> con valor verdadero.

El soporte provisto por un WfMS a los aspectos Distribución de Trabajo y Autorización Dinámica es evaluado a través de las propiedades que se muestran en la Tabla 7. El valor de dichas propiedades es calculado para los elementos *ResourceRoleImpl* del RPIM.

Tabla 7. Propiedades de Evaluación para Aspectos Distribución de Trabajo y Autorización Dinámica

Propiedad	Condición de Verdad en elemento ResourceRoleImpl de un RPIM
<i>isAssignable</i>	Tiene el atributo <i>isAssignable</i> con valor verdadero
<i>multiplePerformers</i>	Tiene el atributo <i>singlePerformer</i> con valor falso.
<i>singlePerformer</i>	Tiene el atributo <i>singlePerformer</i> con valor verdadero.
<i>hasHardConstraints</i>	Tiene al menos un elemento <i>ResolutionConstraintImpl</i> con el atributo <i>isSoft</i> con valor falso en el conjunto de <i>constraints</i> asociado por medio de <i>ConstraintCategoryImpl</i> .
<i>hasSoftConstraints</i>	Tiene al menos un elemento <i>ResolutionConstraintImpl</i> con el atributo <i>isSoft</i> con valor verdadero en el conjunto de <i>constraints</i> asociado por

	medio de <i>ConstraintCategoryImpl</i> .
<i>hasTriggers</i>	Está incluido en el conjunto <i>trigger</i> de al menos un elemento <i>WorkItemOperationImpl</i>
<i>hasEscalations</i>	Está incluido en el conjunto <i>trigger</i> de al menos un elemento <i>WorkItemOperationImpl</i>
<i>revokablePrivileges</i>	Tiene al menos un elemento en el conjunto <i>taskPrivileges</i> con atributo <i>isRevokable</i> con valor verdadero.

Evaluar y Seleccionar la Plataforma de Implementación

Este paso consiste en: calcular las *tasas de soporte* de cada WfMS a los aspectos Estructura de Recursos y Autorización Estática (PI-RSM) y a los aspectos Distribución de Trabajo y Autorización Dinámica (PI-EPM) del proceso modelado; y en seleccionar como plataforma de implementación al WfMS que provea mejor soporte a los mismos.

Los valores de estas tasas de soporte del RPIM respecto al PI-RSM y al PI-EPM se calculan por separado para cada uno de estos modelos. Para cada tabla de resultados de evaluación, la tasa de soporte se calcula de la siguiente manera: se divide la cantidad de filas de la tabla con al menos una entidad de implementación candidata (cantidad de elementos soportados), por la cantidad total de filas (cantidad total de elementos). El valor resultante de esta tasa proporciona una base cuantitativa para comparar el soporte provisto por distintos WfMSs a los requerimientos de perspectiva de recursos definidos en el PI-RSM y el PI-EPM, permitiendo seleccionar la plataforma que en promedio brinde mejor soporte a ambos modelos, es decir, que tenga la mayor tasa de soporte.

El resultado de esta etapa consiste en un WfMS seleccionado y un resumen de las implementaciones candidatas para cada elemento evaluado del PI-RSM (Tabla 3) y PI-EPM (Tabla 5). La implementación efectiva de estos elementos es decidida en la siguiente etapa teniendo en cuenta el detalle de los requerimientos definidos y el significado de los elementos del RPIM. Una tasa de soporte menor que uno no implica la imposibilidad de implementar los requerimientos de un PI-RSM o PI-EPM con un WfMS determinado, sino que la implementación de los requerimientos en modelos específicos de plataforma puede requerir la introducción de elementos adicionales en dichos modelos, tal como se discute más adelante.

4.5 Etapa 4: Definir la Implementación de la Perspectiva de Recursos

El propósito de esta etapa es definir una solución tecnológica, basada en el WfMS seleccionado en la etapa anterior, que especifique la implementación de los requerimientos de la perspectiva de recursos definidos en la segunda etapa. Como resultado de esta etapa se obtienen dos modelos: el Modelo de Estructura de Recursos Específico de Plataforma (PS-RSM) y el Modelo de Proceso Extendido Específico de Plataforma (PS-EPM). Los mismos se derivan del PI-RSM y el PI-EPM respectivamente.

El PS-RSM y el PS-EPM se inicializan como una copia de los respectivos modelos independientes de la plataforma. Luego, cada uno de sus elementos es mapeado al WfMS de destino especificando su atributo *implementation*, el cual define una referencia a un elemento del RPIM. Esto es llevado a cabo con base en las implementaciones candidatas identificadas en la etapa anterior.

La selección de la plataforma llevada a cabo en la etapa anterior se basa en un conjunto propiedades de evaluación de carácter estructural, las cuales no consideran el significado de los elementos de los modelos independientes de plataforma ni del RPIM. Durante esta etapa, se evalúa el significado de estos elementos con el propósito de decidir cuál de los elementos candidatos del RPIM seleccionado debe ser referenciado por cada elemento de los modelos específicos de la plataforma.

Un PS-RSM se define mapeando los elementos de recurso humano (*HumanResource*) y los clasificadores de recurso (*ResourceClassifier*) a las instancias de los elementos *HumanResourceImpl* y *ResourceClassifierImpl* del RPIM importado. Para cada elemento *HumanResource*, es posible que existan cero, uno o más implementaciones candidatas. En los dos últimos casos, la implementación se decide con base en el significado de los elementos, los cuales se expresan en sus atributos *documentation*. En caso que el WfMS seleccionado no provea elementos candidatos para implementar un elemento *HumanResource*, es posible evaluar implementaciones alternativas con base en otros elementos provistos por la plataforma. Las mismas consisten generalmente en la asociación del elemento *HumanResource* con elementos *ResourceClassifier* adicionales. En el caso de los elementos *ResourceClassifier*, existe otro escenario posible, el cual se presenta cuando un WfMS soporta su implementación por medio de varios elementos *ResourceClassifierImpl*. En este caso, el elemento *ResourceClassifier* debe ser reemplazado por los elementos *ResourceClassifier* que sean necesarios para permitir referenciar su implementación en el WfMS.

Una vez definida la implementación de los elementos *Resource* de tipo *HumanResource* y *ResourceClassifier*, se prosigue con la especificación de la implementación para las instancias de *ResourceParameter*, *ResourcePrivilege*, *ResourceClassification* y *ResourceReference* (relación de recursos o de inclusión). Para estos elementos, el *conjunto de implementaciones candidatas* provistas por el RPIM se determina con base en la implementación definida para los elementos *Resource* con los cuales se encuentran asociados.

De este modo, un elemento *ResourceParameter* o *ResourcePrivilege* únicamente podrá ser implementado por medio de una entidad de implementación definida en un elemento *ResourceParameterImpl* o *ResourcePrivilegeImpl*, si dicha entidad se encuentra asociada en el RPIM con el elemento *HumanResourceImpl* o *ResourceClassifierImpl* que

implementa al elemento *Resource* que lo contiene. Del mismo modo, una relación de tipo *ResourceClassification* o *ResourceReference* sólo podrá ser implementada por medio de entidades definidas en elementos *ClassificationImpl* o *ResourceReferenceImpl* con el valor requerido para el atributo *isSubsumption* en caso que dichas entidades de implementación aparezcan asociadas en el RPIM con las implementaciones definidas para los correspondientes elementos *Resource* en el PS-RSM.

Una vez determinado el conjunto de implementaciones candidatas para estos elementos, sus implementaciones se especifican considerando el significado de los elementos del PS-RSM y del RPIM, expresados en sus atributos de nombre y documentación.

Un PS-EPM se define especificando la implementación de sus elementos *ResourceRole* en términos de los elementos *ResourceRoleImpl* del RPIM importado. Esto también se lleva a cabo con base en las implementaciones candidatas identificadas para cada elemento en la etapa anterior, y teniendo en cuenta el significado de los elementos en consideración.

Con base en la implementación definida para cada elemento *ResourceRole*, se determinan los conjuntos de implementaciones candidatas para los elementos *ResolutionConstraint*, *TaskPrivilege* y *WorkItemEvent* definiendo *triggers* y *escalations* para el mismo. Del mismo modo que se describió para el PS-RSM, esto se determina con base en los elementos *ResolutionConstraintImpl*, *TaskPrivilegeImpl* y *WorkItemEventImpl* que aparecen asociados en el RPIM, con el elemento *ResourceRoleImpl* que implementa al elemento *ResourceRole* en consideración.

En caso que el RPIM no provea una implementación candidata para *triggers*, *escalations* o *constraints*, las implementaciones alternativas pueden definirse únicamente introduciendo cambios en las perspectivas de control y datos. Es decir, se debe modificar el flujo de control o los datos para representar la semántica necesaria de la perspectiva de recursos, agregando tareas que soporten los requerimientos a satisfacer. Las asignaciones de recursos de un PS-EPM se definen con base en el PS-RSM. Por lo tanto en caso que el PS-RSM presente variaciones respecto de su PI-RSM de origen, las asignaciones de recursos deben ser modificadas acorde con las mismas.

4.6 Etapa 5: Verificar y Validar la Implementación de la Perspectiva de Recursos

Los WfMSs con frecuencia imponen restricciones sobre el modo en que los requerimientos de la perspectiva de recursos pueden ser implementados. Por lo tanto, los modelos PS-RSM y PS-EPM pueden requerir la incorporación de variaciones respecto de los modelos PI-RSM y PI-EPM a fin de ajustarse a dichas restricciones. Esto introduce la posibilidad de generar implementaciones que no sean completamente consistentes con los

requerimientos definidos. Esta etapa tiene como propósito comprobar que el PS-RSM y el PS-EPM respetan las restricciones impuestas por la plataforma de implementación seleccionada y proveen soporte a los requerimientos definidos en los modelos independientes de la plataforma.

La verificación de un PS-RSM consiste en chequear que sus relaciones de clasificación, relaciones de recursos, relaciones de inclusión, parámetros de recursos y privilegios de recursos respetan las restricciones y entidades definidas en el RPIM del WfMS seleccionado. Las reglas de verificación descritas debajo fueron formalizadas en OCL y se adjuntan en el Anexo II.

- R1: Para toda relación de clasificación de recursos *rc*, la implementación de su recurso humano de origen y la de su clasificador de recurso de destino debe coincidir respectivamente con las entidades de implementación de recurso humano (elemento *HumanResourceImpl*) y de clasificador de recurso (elemento *ResourceClassifierImpl*) especificadas en la definición de la entidad de implementación de relación de clasificación (elemento *ResourceClassificationImpl*) que implementa *rc*.
- R2: Para toda referencia de recursos (relación de recursos o de inclusión) *rr*, la implementación de sus clasificadores de recursos de origen y destino debe coincidir con las entidades de implementación de clasificador de recurso (elementos *ResourceClassifierImpl*) especificadas como origen y destino en la definición de la entidad de implementación de referencia de recursos (elemento *ResourceReferenceImpl*) que implementa *rr*.
- R3: Para toda referencia de recursos, el valor de su atributo *isSubsumption* debe coincidir con el valor del atributo *isSubsumption* de la entidad de implementación de referencias de recursos (elemento *ResourceReferenceImpl*) que la implementa.
- R4: Para todo parámetro de recursos *rp*, las implementaciones del recurso que lo contiene y del valor del parámetro asociado con el mismo deben coincidir con los atributos *resource* y *type* de la entidad de implementación de parámetro de recursos (elemento *ResourceParameterImpl*) que implementa *rp*.
- R5: Para todo privilegio de recursos, la implementación del recurso que lo contiene debe coincidir con la referencia *resource* de la entidad de implementación de privilegio de recursos (elemento *ResourcePrivilegeImpl*) referenciado.

La verificación de un PS-RSM también incluye el chequeo de las cardinalidades mínimas y máximas definidas en el RPIM por los elementos *ResourceClassificationImpl* (R12), *ResourceReferenceImpl* (R13) y *ResourceParameterImpl* (R14); y que todos los privilegios de recursos obligatorios son definidos para los recursos correspondientes (R15).

La verificación de un PS-EPM es efectuada por medio de la comprobación de las consultas parametrizadas de recursos, expresiones de asignación de recursos, eventos de ítem de trabajo, privilegios revocados y restricciones de resolución definidas para cada rol de recurso. Las siguientes reglas fueron definidas:

- R6: Para toda consulta parametrizada de recursos, el recurso referenciado debe ser implementado por un elemento *ResourceImpl* con el atributo *isReferenceable* con valor verdadero, y los parámetros de recursos enlazados deben ser de un tipo enlazable.
- R7: La existencia de una expresión de asignación de recursos implica que la implementación de todos los recursos del PS-RSM importado refieran a una entidad de implementación de recursos (elemento *ResourceImpl*) con valor verdadero para el atributo *isBindable*.
- R8: Para toda restricción de resolución, sólo se debe definir una expresión cuando la entidad que la implementa (elemento *ResolutionConstraintImpl*) define un valor verdadero para el atributo *hasExpression*; y el valor de su atributo *isSoft* debe coincidir con el atributo de igual nombre en la entidad de implementación.
- R9: Para todo *trigger* definido para un rol de recurso dado, la implementación del evento de ítem de trabajo referenciado (elemento *WorkItemEventImpl*) debe poder disparar la asignación del rol de recurso por medio de alguna de las operaciones (elemento *WorkItemOperationImpl*) especificadas por la referencia *triggers*.
- R10: Para todo *escalation* definido para un rol de recurso dado, la implementación del evento de ítem de trabajo referenciado (elemento *WorkItemEventImpl*) debe poder quitar el rol de recurso por medio de alguna de las operaciones (elemento *WorkItemOperationImpl*) especificadas por la referencia *escalates*.
- R11: Para todo privilegio de tarea revocado, su implementación (elemento *TaskPrivilegeImpl*) debe definir un valor verdadero para el atributo *isRevokable*.

La verificación también comprueba que la cardinalidad inferior y superior de cada categoría de restricción es respetada (R16), y que todos los roles de recurso requeridos definidos por el RPIM importado son asociados con tareas de usuario (R17).

Por otra parte, la validación de un PS-RSM contra un PI-RSM consiste en un procedimiento mediante el cual se constata que ambos modelos definen los mismos recursos humanos, y que ambos modelos definan los mismos parámetros y privilegios para los mismos en forma directa o a través de su pertenencia a clasificadores de recursos.

La validación de un PS-EPM consiste en un procedimiento mediante el cual se determina que sus tareas de usuario, roles de recursos, restricciones de resolución, eventos de ítem de trabajo y asignaciones de recursos son consistentes con aquellos definidos en el PI-EPM. Esto es realizado en tres pasos. Primero, se comprueba que se definan las mismas tareas de usuario en ambos modelos. Luego, se chequea que cada tarea de usuario en el PS-EPM defina una implementación por cada rol de recurso, restricción de resolución y evento de ítem de trabajo definido en su contraparte del PI-EPM. Finalmente, se comprueba que las asignaciones de recursos definidas para cada rol de recurso en estos modelos resultan en los mismos recursos humanos. Esto puede ser chequeado con base en la información de los modelos en caso que las asignaciones no incluyan variables. En caso contrario, es necesario proveer valores para las mismas. Las restricciones de resolución deben ser chequeadas en forma manual dado que se definen en lenguaje natural en el PI-EPM.

4.7 Etapa 6: Generar Especificaciones Ejecutables

El propósito de la última etapa del método consiste en generar especificaciones ejecutables de la perspectiva de recursos para la plataforma de implementación seleccionada. El objetivo es implementar el SIOP que soporte los requerimientos definidos en el PS-RSM y el PS-EPM sobre la base del WfMS elegido. Los modelos específicos de la plataforma (PS-RSM y PS-EPM) resultantes de la etapa anterior son tomados en esta etapa como base para producir especificaciones (típicamente documentos XML con una estructura definida en un documento XML Schema) que puedan ser interpretados por el WfMS.

El resultado de esta etapa puede variar dependiendo de la plataforma de implementación seleccionada. En términos generales, se obtiene un documento definiendo el esquema de recursos y un documento de especificación del proceso incluyendo los detalles de la perspectiva de recursos como parte de la definición de las tareas de usuario, o bien en un documento separado. Empleando las técnicas del desarrollo dirigido por modelos (OMG, 2003; Selic, 2003), estos documentos pueden ser obtenidos a partir de la información contenida en el PS-RSM y el PS-EPM. Dichas técnicas consisten en la aplicación de un conjunto de reglas de transformación, las cuales tomen como entrada los elementos de los modelos específicos de plataforma y produzcan como salida los elementos y/o fragmentos correspondientes de dichos documentos.

El método propuesto no se restringe a una tecnología específica para la implementación de estas reglas. Las mismas pueden ser implementadas por medio de lenguajes específicos para su definición, tales como QVT (OMG, 2005) o ATL (Jouault, et al., 2006), o bien pueden ser implementadas en lenguajes de programación de propósito general como Java o Python. Independientemente de su implementación, estas reglas de transformación pueden ser definidas con base en la información contenida en el RPIM, el cual especifica el modo en que las distintas entidades de implementación provistas por un

WfMS se representan en términos de los elementos del marco de trabajo propuesto. De este modo, las transformaciones definidas con base en un RPIM pueden ser reutilizadas en múltiples proyectos para obtener especificaciones ejecutables a partir de modelos PS-RSM y PS-EPM que importen dicho RPIM. En el siguiente capítulo se describe el enfoque propuesto para permitir la definición y ejecución de dichas transformaciones de modelos independientemente del lenguaje con el que hayan sido definidas.

4.8 Conclusiones

El método propuesto en esta sección provee un enfoque sistemático para definir la perspectiva de recursos durante el desarrollo de SIOPs con base en los elementos provistos por el marco de trabajo presentado en el Capítulo 3.

Los lineamientos provistos por el método permiten partir de un modelo de proceso BPMN estándar referido como BPLM y describir requerimientos respecto al modo en que los ítems de trabajo de las tareas de usuario del proceso deben ser asignados a los recursos y posteriormente gestionados y ejecutados por los mismos. Dichos requerimientos son representados en un Modelo de Estructura de Recursos Independiente de la Plataforma (PI-RSM) y en un Modelo de Proceso Extendido Independiente de la Plataforma (PI-EPM). En estos modelos, los elementos genéricos provistos por el marco de trabajo son descritos en lenguaje natural, lo cual permite una mejor comunicación entre analistas de negocio y desarrolladores. De esta manera se obtienen modelos conceptuales que proveen descripciones estructuradas de los requerimientos de perspectiva de recursos.

Mediante la aplicación de un conjunto de propiedades de evaluación propuesta por el método es posible seleccionar el WfMS adecuado. Dichas propiedades permiten determinar, desde un punto de vista estructural, si las entidades de implementación provistas por un WfMS representado en un Modelo de Implementación de la Perspectiva de Recursos (RPIM), soportan los requerimientos de perspectiva de recursos expresados en los modelos conceptuales. Con base en dichas propiedades es posible calcular tasas de soporte, las cuales proporcionan una base cuantitativa para evaluar el grado en que los requerimientos definidos en los modelos independientes de la plataforma pueden ser representados en modelos específicos de plataforma sin introducir variaciones en los mismos. La aplicación de dichas propiedades de evaluación permite obtener un conjunto de entidades de implementación candidatas para cada elemento de los modelos independientes de la plataforma, lo cual proporciona un punto de partida para la definición de los correspondientes modelos específicos de plataforma.

Los modelos específicos de la plataforma PS-RSM y PS-EPM se crean respectivamente a partir de los modelos independientes de la plataforma PI-RSM y PI-EPM e importando el RPIM seleccionado. La implementación de los requerimientos es definida en estos modelos especificando, por cada uno de sus elementos, una referencia a una

entidad de implementación del tipo correspondiente definida en el RPIM. Esto se hace con base en los conjuntos de especificaciones candidatas y en el significado de los elementos de los modelos específicos de plataforma y de las entidades de implementación. En caso que una plataforma de implementación no provea soporte directo a los requerimientos de perspectiva de recursos definidos, es posible tomar acciones correctivas, las cuales consisten en introducir modificaciones a los elementos que definen la perspectiva de recursos en primera instancia, y a los elementos que definen las perspectivas de control de flujo y datos en una segunda instancia. En caso que las restricciones impuestas por los elementos del RPIM o que el significado de las entidades de implementación provistas haga imposible dar soporte a los requerimientos definidos con base en el WfMS seleccionado, es posible tomar otro de los WfMSs evaluados e intentar implementar dichos requerimientos de la manera descripta. Aún en el peor de los casos, el método permite lidiar con las limitaciones impuestas por las plataformas de implementación en etapas tempranas del desarrollo de los SIOPs. La detección de dichas limitaciones en las etapas finales tiene un costo significativamente mayor.

El conjunto de reglas de verificación y validación provistas por el método permiten mantener la consistencia de las representaciones de los requerimientos de perspectiva de recursos definidas a nivel de implementación con las definidas a nivel conceptual. Las mismas permiten constatar que, luego de introducidas las modificaciones necesarias, los modelos específicos de plataforma siguen siendo consistentes con las restricciones impuestas por la plataforma (verificación) y se mantienen alineados con los modelos conceptuales (validación).

Finalmente, por medio de transformaciones de modelo a código definidas con base en la información provista por RPIMs, el método permite obtener especificaciones ejecutables de los procesos de negocio para el WfMS seleccionado. Dichas reglas de transformación pueden ser reutilizadas para distintos PS-RSMs y PS-EPMs que definan la implementación de los requerimientos con base en un mismo RPIM. Esto permite reducir el esfuerzo, tiempo, costo y errores en la definición de la perspectiva de recursos en el desarrollo de SIOPs con base en WfMSs.

5 Herramienta para Definir la Perspectiva de Recursos en el Desarrollo de SIOPs

En este capítulo se presenta la herramienta desarrollada para proveer soporte al marco de trabajo y el método propuestos. Se describe la plataforma de modelado, la estructura de la herramienta y sus funcionalidades (Sección 5.1). Finalmente, se presentan las conclusiones del capítulo (Sección 5.2).

5.1 Estructura y Funcionalidades de la Herramienta

La herramienta desarrollada se basa en la plataforma de modelado denominada Oryx (Decker, et al., 2008). La misma fue seleccionada dado que proporciona una implementación de BPMN 2.0 que puede ser reutilizada y extendida. Además, dicha plataforma provee un repositorio que permite el almacenamiento, recuperación e integración de los distintos tipos de modelo que se definen durante las diferentes etapas del método propuesto.

Oryx es una aplicación Web con una arquitectura cliente-servidor tradicional. El servidor de Oryx provee una interfaz REST que permite almacenar y recuperar información del repositorio de modelos implementado sobre un sistema de gestión de bases de datos relacional. El cliente de Oryx es un editor de modelos desarrollado en Javascript y ejecutado desde un navegador Web. Dicho editor puede ser configurado para dar soporte a distintos lenguajes de modelado por medio de la carga de documentos JSON llamados stencilsets, los cuales definen los elementos de un lenguaje, sus propiedades y la notación gráfica asociada con los mismos.

Las funcionalidades provistas por el editor de Oryx para el manejo de modelos son implementadas como complementos o plugins. Dicho editor provee un complemento de guardado, el cual emplea la interfaz REST provista por el servidor para almacenar y recuperar los modelos del repositorio. La plataforma permite el desarrollo de complementos tanto en el lado del cliente como en el del servidor. Los complementos del cliente se definen en Javascript empleando una API definida como parte de la plataforma. Los complementos del servidor se implementan como Servlets de Java. Los complementos del cliente implementan funcionalidades relacionadas con la manipulación y visualización de modelos en el editor. Los mismos permiten invocar las operaciones definidas por complementos del servidor a los efectos de implementar procesamientos más complejos sobre los modelos definidos, tales como su transformación en otros modelos o especificaciones de algún tipo.

La arquitectura de la herramienta desarrollada se muestra en la Figura 28. La misma reutiliza el stencilset de BPMN 2.0 provisto por Oryx y provee tres stencilsets adicionales, los cuales implementan la Extensión de Estructura de Recursos, la Extensión de Distribución de Trabajo y el Metamodelo de Implementación de la Perspectiva de

Recursos. La herramienta incluye un editor de la perspectiva de recursos desarrollado como un complemento de cliente, el cual provee las funcionalidades requeridas para llevar a cabo las distintas etapas del método descrito en el Capítulo 4. La herramienta también incluye un conjunto de componentes de servidor, los cuales implementan las transformaciones de modelos requeridas.

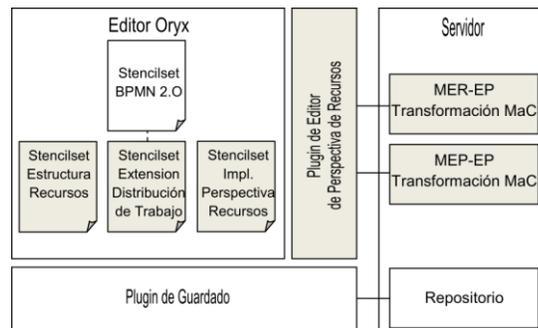


Figura 28. Arquitectura de la Herramienta basada en Oryx.

La Figura 29 muestra la barra de herramientas del complemento de edición de la perspectiva de recursos desplegada en el editor de modelos de Oryx. A la derecha, en la lista del menú desplegable, se muestran los ítems de menú que se corresponden con los comandos o funciones que dan soporte a las etapas del método propuesto.

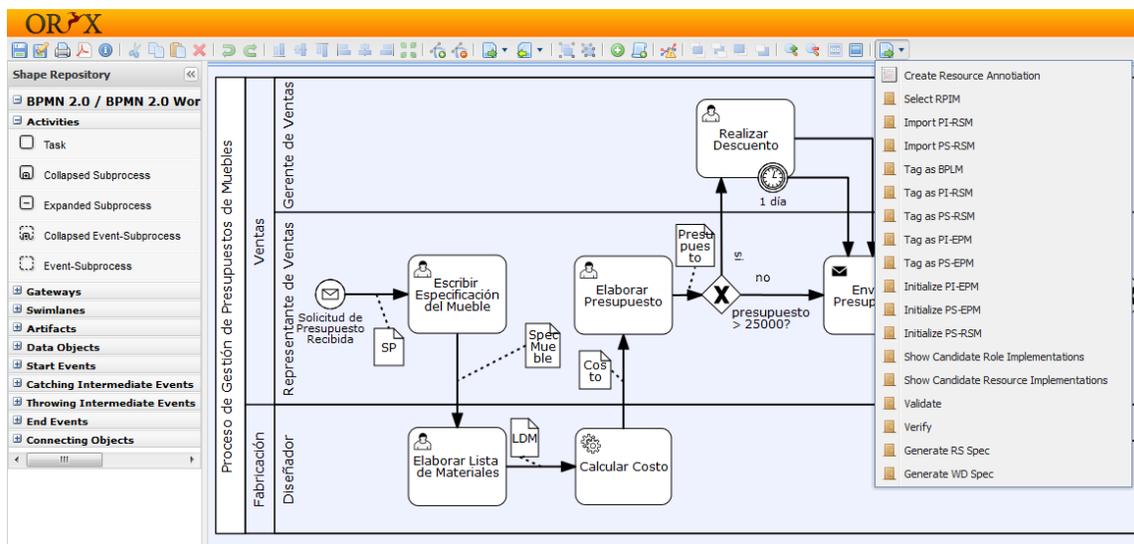


Figura 29. Editor de Modelos de Lógica de Procesos de Negocio.

Con el propósito de permitir la identificación de los distintos tipos de artefactos resultantes de las etapas del método propuesto, la herramienta hace uso del mecanismo de definición de etiquetas (tags) soportado por el repositorio de Oryx. Las etiquetas permiten clasificar a un modelo como de un tipo determinado. Con este fin, el complemento desarrollado define las etiquetas BPLM, PI-RSM, PI-EPM, RPIM, PS-RSM y PS-EPM; las

cuales corresponden a los artefactos propuestos en esta tesis. Estas etiquetas son almacenadas en el repositorio por medio de métodos provistos por el complemento de guardado de Oryx. Como ejemplo, un modelo BPMN 2.0, luego de definido, puede ser etiquetado y almacenado en el repositorio como modelo BPLM, según la denominación dada en el método propuesto.

La herramienta también provee un editor que permite definir y visualizar RSMs, los cuales son almacenados en el repositorio de modelos para ser referenciados por uno o más modelos de procesos etiquetados como EPM. Este editor se muestra en la Figura 30. A la derecha e izquierda de dicha pantalla se muestran respectivamente la paleta de elementos y la barra de propiedades del editor, las que son generadas automáticamente por Oryx a partir del stencilset de estructura de recursos. Con el propósito de implementar la notación propuesta para la representación de RSMs, fue necesario además definir un conjunto de documentos SVG, cada uno de los cuales define el constructor gráfico de un elemento del RSM que se muestra en la parte central de la figura. Los PI-RSMs creados mediante este editor, los cuales no importan ningún RPIM (ver el valor vacío de la propiedad RPIM en la Figura 30), luego son etiquetados como PI-RSM. De lo contrario, si hay un RPIM seleccionado, son etiquetados como PS-RSM.

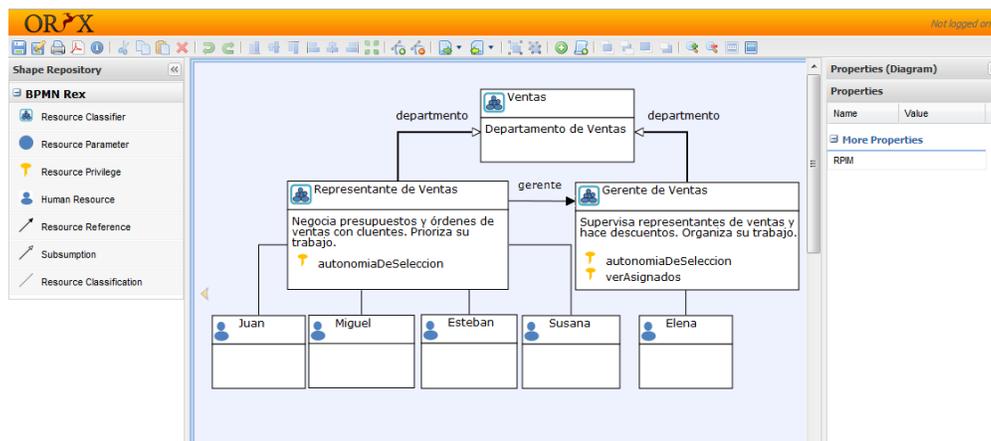


Figura 30. Editor de Modelos de Estructura de Recursos.

Para definir un PI-EPM a partir de un BPLM, el primero es inicializado con los mismos elementos del BPLM por medio del diálogo que se muestra en la parte izquierda de la Figura 31, al cual se accede mediante el comando Initialize PI-EPM (menú desplegable de la barra de herramientas de la Figura 29). Luego de ello, un PI-RSM es importado en el PI-EPM siendo definido por medio del cuadro de diálogo que se muestra en la parte derecha de la Figura 31, al cual se accede mediante el comando Import PI-RSM. Finalmente, la asignación de recursos a los distintos roles definidos para cada tarea de usuario es definida por medio del cuadro de diálogo que se muestra en la Figura 32. Cada una de las asignaciones definidas para una tarea puede ser visualizada en el diagrama del

PI-EPM mediante anotaciones de texto (Figura 33), las cuales pueden ser generadas automáticamente por medio de la selección de la tarea en el editor y la ejecución del comando *Create Resource Annotation*.

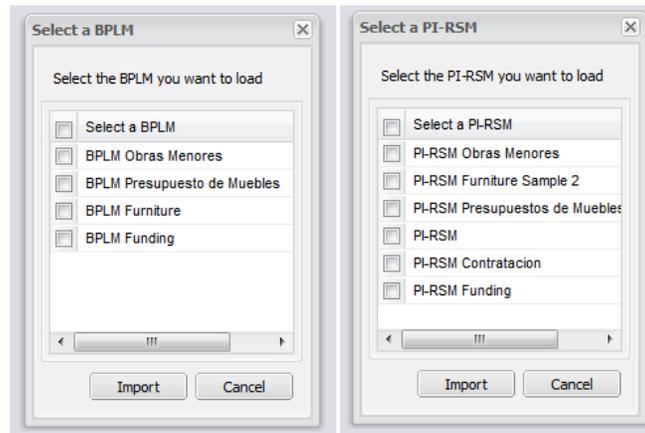


Figura 31. Asistentes para Inicializar PI-EPMs.

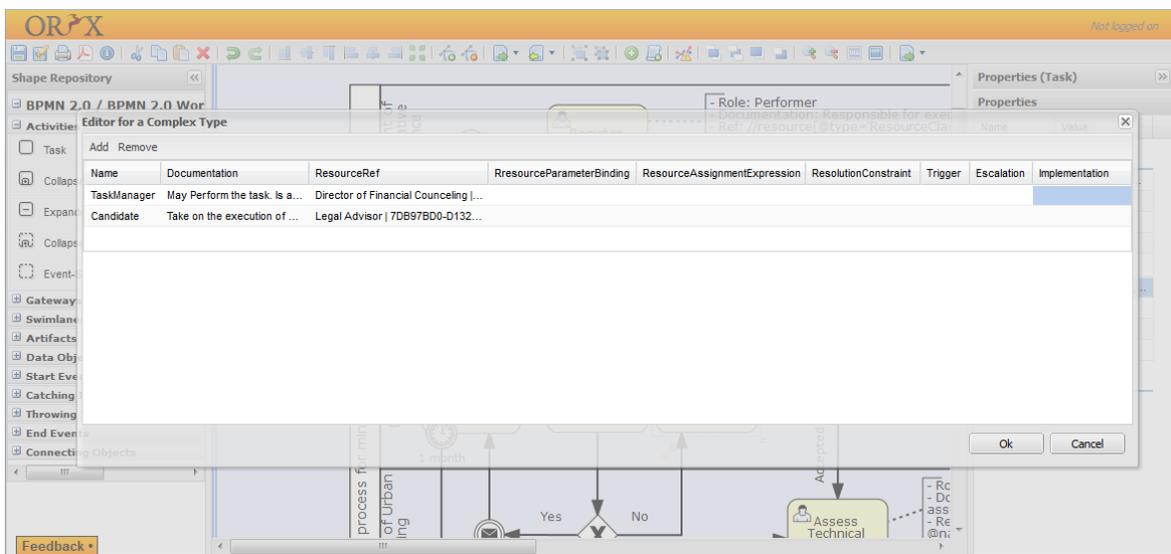


Figura 32. Asistente para la Definición del Aspecto Distribución de Trabajo.

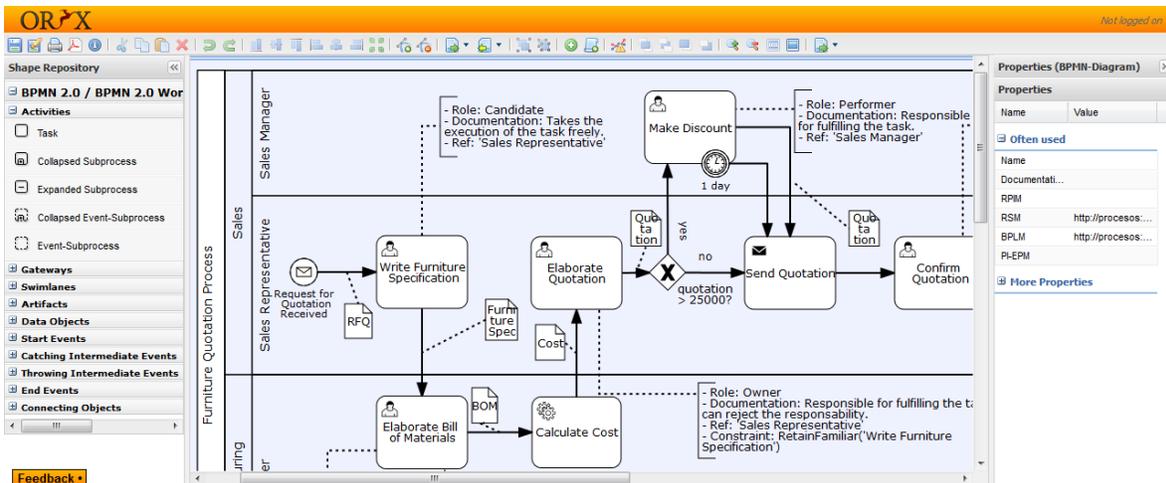


Figura 33. Editor de Modelos de Proceso Extendidos Independientes de Plataforma

La herramienta permite también la creación, almacenamiento y evaluación de RPIMs. La Figura 34 muestra el editor desarrollado con base en el stencilset de implementación de la perspectiva de recursos. Del mismo modo que para el editor de RSMs, se definieron documentos SVG que definen los constructores gráficos de cada uno de los elementos del RPIMeta. Por medio de este editor, los RPIMs son creados, etiquetados como RPIM y almacenados en el repositorio.

La herramienta implementa las propiedades de evaluación descritas en la Sección 4.4, las cuales permiten evaluar a los RPIMs en términos del soporte que proveen a los elementos definidos en un PI-RSM y un PI-EPM. La evaluación de dichas propiedades en un PI-RSM o en un PI-EPM puede ser realizada ejecutando el comando *Select RPIM* (Figura 29), el cual abre el cuadro de diálogo que se muestra en la Figura 35. Dicho cuadro de diálogo muestra los RPIM correspondientes a diferentes plataformas de WfMS que están almacenados en el repositorio. El usuario puede seleccionar uno de ellos, ejecutar la evaluación a través del botón *Calculate support ratio*, y obtener el resultado de la misma.

A partir de este cuadro de diálogo, también es posible inicializar un modelo específico de la plataforma (PS-RSM o un PS-EPM) a partir del correspondiente modelo independiente de la plataforma para el cual se realizó la evaluación e importando el RPIM seleccionado. Esto se realiza ejecutando el comando *Initialize PSM*.

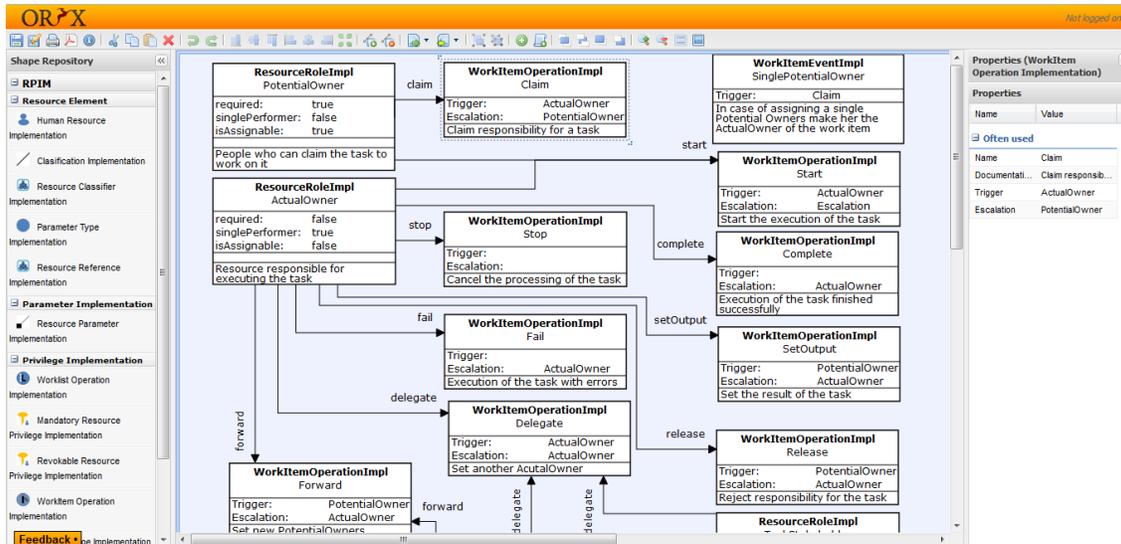


Figura 34. Editor de Modelos de Implementacion de Perspectiva de Recursos (RPIM).

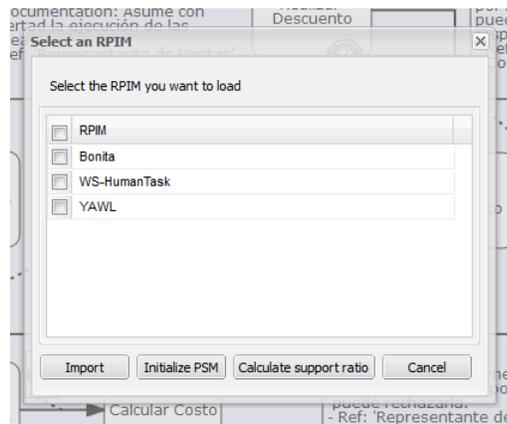


Figura 35. Selector de Modelos de Implementación de Perspectiva de Recursos.

La herramienta posibilita definir la implementación de los elementos de los modelos específicos de plataforma por medio de dos asistentes, los cuales se muestran en las Figuras 35 y 36. Estos asistentes muestran, para un elemento *Resource* o *ResourceRole* seleccionado en un PS-RSM o PS-EPM respectivamente, el conjunto de implementaciones candidatas provistas por el RPIM importado. Estos cuadros de diálogos son accesibles seleccionando el elemento para el cual se quiere definir la implementación y luego el comando *Show Candidate Resource Implementations* o *Show Candidate Role Implementations*, según corresponda.

Dichos asistentes muestran el nombre del elemento seleccionado y una grilla en la que se muestra una fila por cada implementación candidata y una columna por cada propiedad de evaluación. Las propiedades requeridas son marcadas con un asterisco en el encabezado de la columna. Aquellas implementaciones candidatas que provean soporte a todas las

propiedades requeridas muestran un valor 1 en la última columna de la derecha (*providesSupport*). Una vez identificado el elemento más adecuado para implementar el elemento del modelo específico de la plataforma, se marca la casilla a la izquierda de la fila correspondiente y se presiona el botón *Select*. La implementación de los restantes elementos de dichos modelos es definida en forma directa desde la barra de propiedades seleccionando la opción que corresponda en el menú desplegable *Implementation* (ver Figura 38).

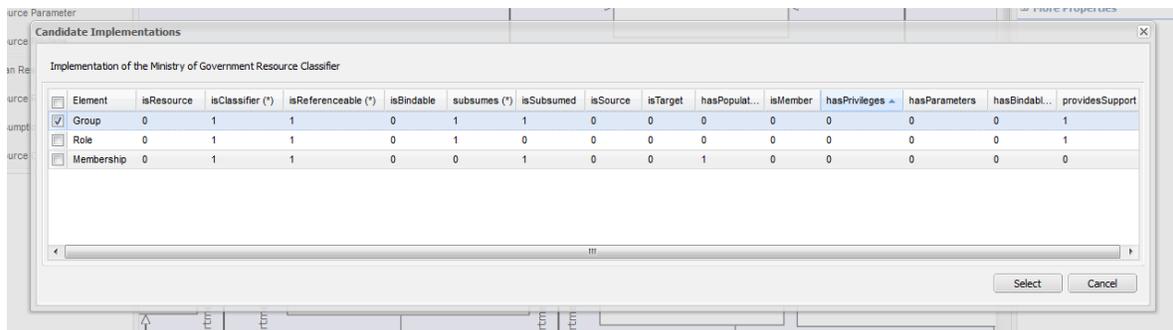


Figura 36. Diálogo de Implementaciones Candidatas de RSMs.

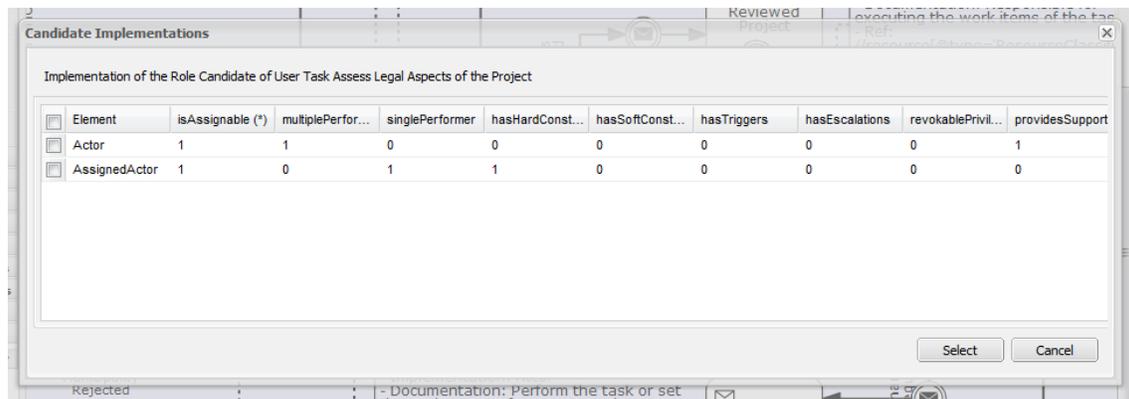


Figura 37. Diálogo de Implementaciones Candidatas de EPMs.

La herramienta también da soporte a los métodos de verificación y de validación del método propuesto. Estos métodos fueron desarrollados en Javascript con base en el API del editor de Oryx, los cuales implementan las reglas OCL de verificación y el procedimiento de validación descritos en la Sección 4.6. Los mismos son accesibles por medio de los comandos *Verify* y *Validate*. El resultado de los mismos es visualizado por medio de los cuadros de diálogos que se muestran en la Figura 39.

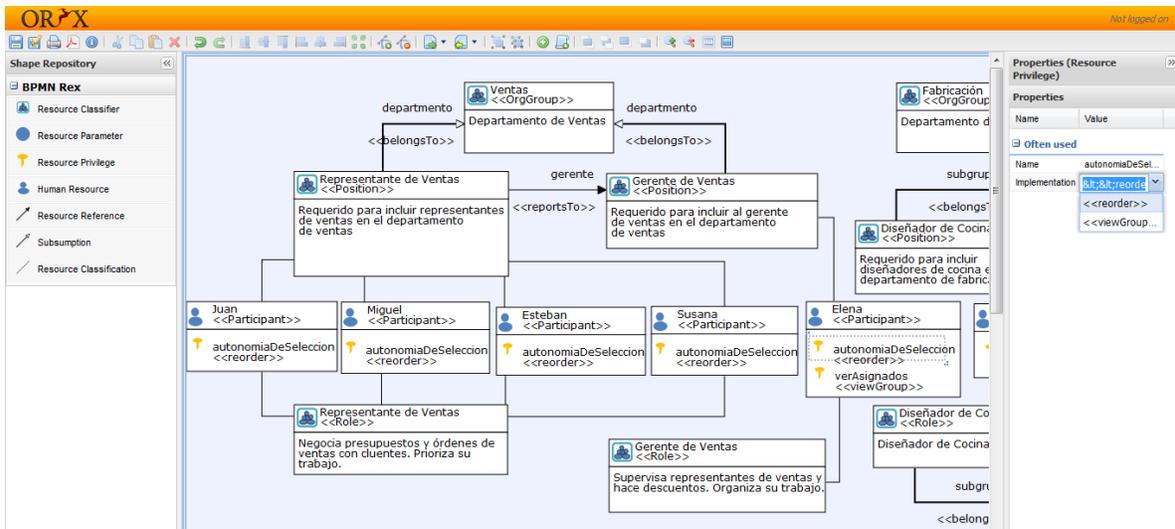


Figura 38. Implementación de Privilegios de Recursos en un PS-RSM.

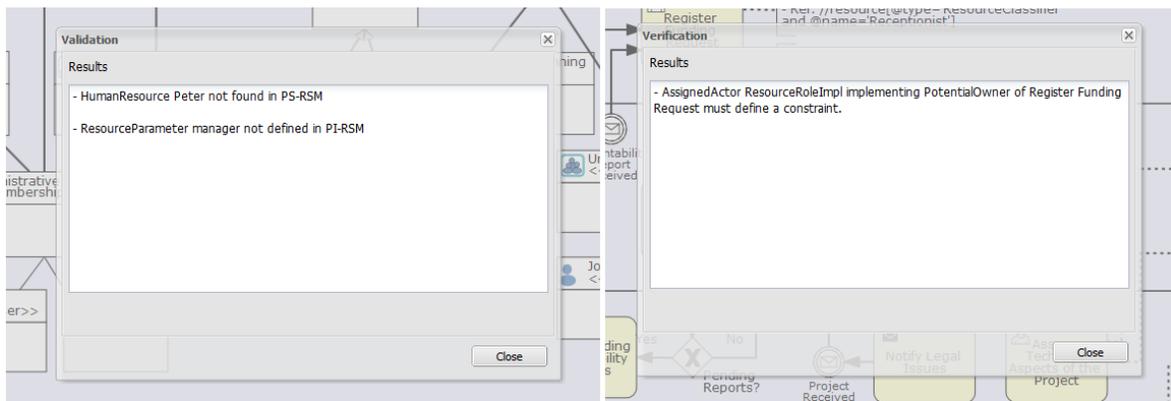


Figura 39. Diálogos de resultados de validación y verificación.

Por último, la herramienta provee soporte a transformaciones de modelo a código para generar implementaciones en diferentes plataformas de WfMS. Estas transformaciones o máquinas de transformación de modelo a código son desarrolladas como complementos del lado del servidor y registradas para cada RPIM almacenado en el repositorio. Las transformaciones disponibles pueden ser ejecutadas por medio de los comandos Generate RS Spec y Generate WD Spec. Ambos permiten obtener las especificaciones ejecutables basadas en el WfMS representado por el RPIM seleccionado a partir de los elementos definidos en un PS-RSM o un PS-EPM, respectivamente.

Los stencilsets, complementos y transformaciones de modelo a código desarrolladas se encuentran disponibles en <https://code.google.com/p/bpmn-rex>.

5.2 Conclusiones

La herramienta presentada en este capítulo demuestra la factibilidad de implementar editores para los artefactos definidos en el marco de trabajo propuesto: BPLM, PI-RSM, PI-EPM, RPIM, PS-RSM y PS-EPM. Estos editores fueron desarrollados por medio de la definición de un conjunto de stencilsets y complementos de editor de la plataforma Oryx.

Los stencilsets correspondientes con los editores de PI-RSMs, PS-RSMs y RPIMs fueron definidos con base en los elementos de la Extensión de Estructura de Recursos y el Metamodelo de Implementación de la Perspectiva de Recursos. Junto con estos stencilsets fueron creados un conjunto de documentos SVG definiendo el aspecto de los constructores gráficos propuestos para permitir la visualización de estos modelos. El stencilset correspondiente con el editor de PI-EPMs y PS-EPMs fue definido como una extensión al stencilset de BPMN 2.0 provisto por Oryx. Dicho stencilset permite incorporar los atributos definidos en la Extensión de Distribución de Trabajo. Los valores de estos atributos son visualizados por medio de los artefactos de anotación de texto provistos por BPMN.

Los distintos tipos de modelo definidos empleando estos editores son almacenados en un repositorio, lo cual simplifica la integración de los mismos durante el desarrollo de SIOPs. Con este propósito, la herramienta provee un conjunto de asistentes que guían a los usuarios a través de las distintas etapas del método propuesto en el Capítulo 4. Estos asistentes, implementados como complementos de cliente de Oryx, consumen las interfaces provistas por la plataforma para proveer funcionalidades que automatizan las tareas de: inicialización de modelos, aplicación de las propiedades de evaluación, cálculo de tasas de soporte de RPIMs a PI-RSMs y PI-EPMs, aplicación de reglas de verificación y ejecución de los procedimientos de validación de PS-RSMs y PS-EPMs definidas como parte del método.

La herramienta es extensible para soportar la implementación de requerimientos de perspectiva de recursos de procesos de negocio empleando WfMSs existentes o futuros. Esto es soportado a través de la definición y almacenamiento en el repositorio de nuevos RPIMs y del desarrollo de nuevas transformaciones de modelo a código registradas como complemento de servidor para cada uno de los RPIMs desarrollados. En este sentido, un trabajo en curso consiste en el incremento de la cantidad de RPIMs definidos para WfMSs de uso comercial y académico y su publicación con el propósito de permitir la comparación de herramientas existentes y fomentar el desarrollo de nuevas herramientas que provean soporte a la perspectiva de recursos de procesos de negocio.

6 Evaluación

En este capítulo se muestra la aplicabilidad y utilidad del marco de trabajo y el método propuestos, y se evalúa su capacidad para expresar requerimientos recurrentes de la perspectiva de recursos. Se muestra cómo el RPIimplMeta puede ser empleado para representar plataformas de implementación ampliamente difundidas (Sección 6.1). Se presenta la aplicación del método propuesto en dos casos de estudio, uno desarrollado para un gobierno provincial y otro para una empresa privada (Sección 6.2). Se presenta una evaluación del marco de trabajo propuesto respecto de los patrones de recursos de workflow (Sección 6.3). Finalmente, se discuten los resultados de la evaluación (Sección 6.4).

6.1 Modelado de Plataformas de Implementación

6.1.1 Modelo de Implementación de la Perspectiva de Recursos de YAWL

Yet Another Workflow Language (YAWL) es un WfMS desarrollado con el propósito de proveer una plataforma de ejecución que dé soporte a los patrones de workflow. Según sus desarrolladores, este WfMS soporta a 34 de los 43 patrones de recursos (Ter Hofstede, 2010). Dicho WfMS, cuyo metamodelo (Figuras 8 y 9) se presentó y discutió en el Capítulo 2, se basa en un modelo conceptual diferente del definido en el marco de trabajo propuesto en esta tesis. Con el propósito de mostrar la aplicabilidad y utilidad del RPIimplMeta para representar distintos tipos de WfMSs en términos del modelo conceptual propuesto, se presenta un RPIM construido para representar esta plataforma de implementación.

Aspectos Estructura de Recursos y Autorización Estática en YAWL

La Figura 40 muestra la porción del RPIM que representa las entidades provistas por YAWL para dar soporte a los aspectos Estructura de Recursos y Autorización Estática. Estas entidades constituyen los esquemas de recursos de YAWL, los cuales se almacenan en el repositorio organizacional provisto por esta plataforma. Un recurso humano puede ser representado por medio de las entidades Participant y Administrator, las cuales representan usuarios ordinarios y administradores del sistema. Los recursos humanos representados mediante estas entidades pueden ser referenciados directamente desde las tareas de usuario (atributo *isReferenceable* con valor verdadero), o bien pueden ser asignados a un rol de recurso en un ítem de trabajo de una instancia de una tarea con base en el valor de una variable en el alcance de la misma (atributo *isBindable* con valor verdadero y *bindingType* de tipo *Variable*).

En YAWL, las características de los recursos pueden ser expresadas por medio de capacidades, las cuales son definidas para cada recurso humano en forma individual. Durante la asignación de recursos a tareas, estas capacidades pueden ser empleadas para filtrar el conjunto de recursos a ser asignados a las tareas. El valor de dichas capacidades puede únicamente ser comparado con valores constantes. En el RPIM, esto se representa

por medio del tipo de parámetro (elemento *ParameterTypeImpl*) *Capability*, el cual tiene los atributos *isBindable* con valor verdadero y *binningType* con valor *Literal*. Este tipo de parámetro está asociado con las entidades *Participant* y *Administrator* por medio de la implementación de parámetro (elemento *ResourceParameterImpl*) llamada *possesses*.

En lo que respecta al aspecto Autorización Estática, YAWL permite definir privilegios de recursos sobre las operaciones de lista de trabajo *Choose*, *StartConcurrently*, *Reorder*, *ViewGroup*, *ManageCase* y *Chain*. Las mismas permiten a los recursos seleccionar el próximo ítem de trabajo a ejecutar, iniciar la ejecución de más de un ítem de trabajo en simultáneo, cambiar el orden de los ítems de trabajo que les fueron asignados, ver los ítems de trabajo asignados a recursos pertenecientes a un mismo grupo, reclamar automáticamente los ítems de trabajo de la instancia de un proceso a medida que se crean y comenzar automáticamente la ejecución del ítem de trabajo correspondiente a la tarea siguiente de la completada, respectivamente. En el RPIM, estas operaciones son representadas por medio de implementaciones de operación de lista de tarea (elemento *WorklistOperationImpl*), mientras que la posibilidad de otorgar o revocar privilegios para ejecutar las mismas se define mediante implementaciones de privilegio de recursos (elemento *ResourcePrivilegeImpl*) con el atributo *mandatory* con valor falso. Estos privilegios, representados mediante las flechas con línea discontinua, son *choose*, *startConcurrently*, *reorder*, *viewGroup*, *manageCase* y *chain*.

La plataforma otorga además un conjunto de privilegios de recursos (elemento *ResourcePrivilegeImpl*) de manera obligatoria a los recursos administradores del sistema (entidad *Administrator*). Estos privilegios son *viewUnoffered*, *viewOffered*, *offer*, *allocate* y *start*. Los mismos se representan gráficamente como flechas con líneas continua dado que presentan un valor verdadero para el atributo *mandatory*. Los privilegios *viewOffered* y *viewUnoffered* indican que los administradores pueden ejecutar las operaciones de lista de trabajo (elemento *WorklistOperationImpl*) *ViewWorklisted* y *ViewUnoffered*. Las mismas permiten a los recursos ver cualquier ítem de trabajo de cualquier tarea, esté ofrecido o no. Los privilegios *offer*, *allocate* y *start* indican que los administradores también pueden ejecutar las operaciones de ítem de trabajo *Offer*, *Allocate* y *Start*. Estas operaciones permiten asignar a los ítems de trabajo los recursos para desempeñar respectivamente los roles de recurso de trabajo *PotentialOwner*, *Owner* y *Executor*, que se describen en la siguiente subsección.

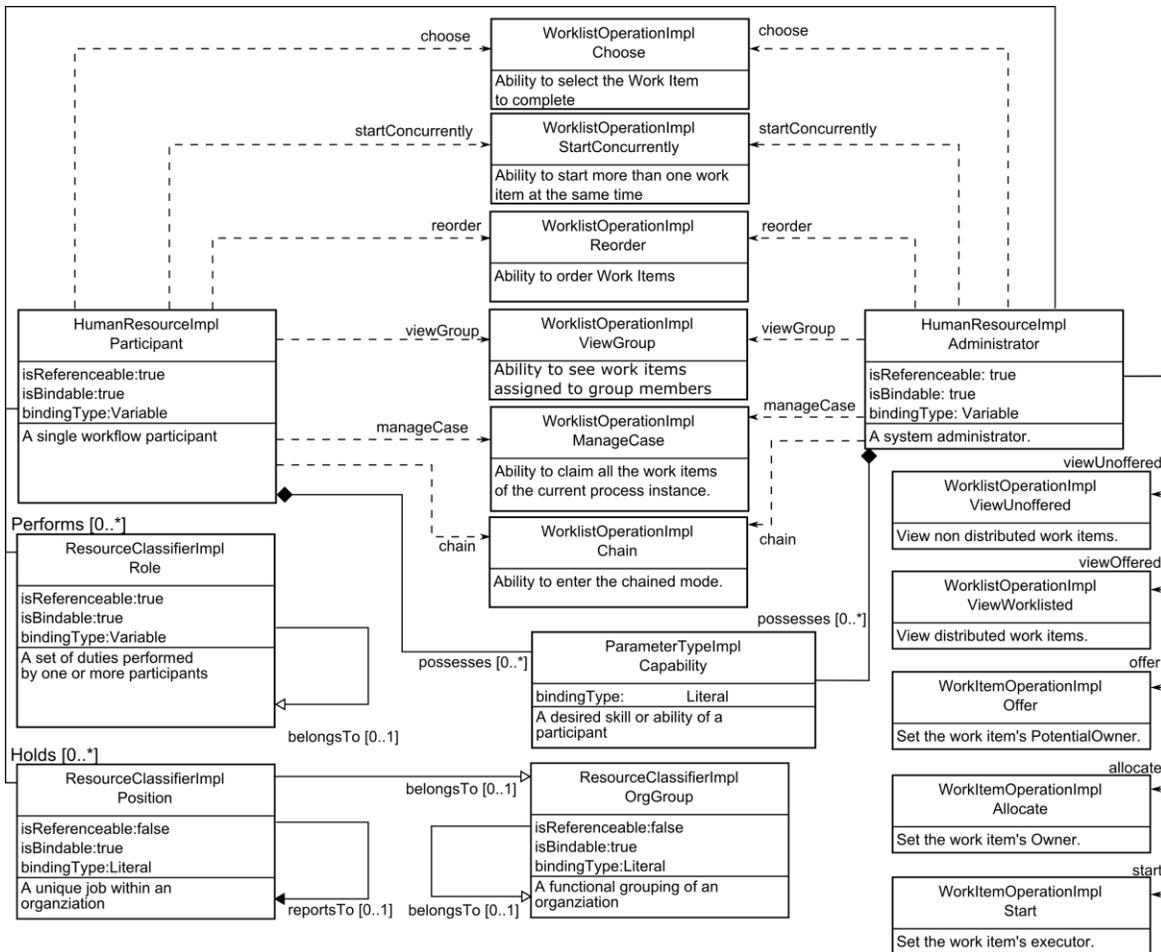


Figura 40. RPIM de Aspectos Estructura de Recursos y Autorización Estática de YAWL.

En lo que respecta a la clasificación de los recursos humanos, YAWL define tres entidades (elemento *ResourceClassifierImpl*) con este propósito: Role, Position y OrgGroup.

A diferencia de BPMN, YAWL emplea el concepto de rol (Role) como parte del aspecto Estructura de Recursos, pero no como parte del aspecto Distribución de Trabajo. Por medio de este concepto, la plataforma provee soporte al *enfoque tecnológico* para la definición de modelos de recursos descrito en la Sección 2.1.1. En YAWL, la entidad Role se emplea para representar las responsabilidades de un conjunto de recursos humanos representados como Participant o Administrator dentro de una organización. En YAWL, un recurso puede asociarse con múltiples roles y pueden existir jerarquías de roles y subroles. Esto es representado en el RPIM por medio de implementaciones de relaciones de clasificación (elemento *ClassificationImpl*) llamadas Performs, las cuales se representan mediante líneas continuas; y por medio de la implementación de relación de inclusión (elemento *ResourceReferenceImpl* con valor verdadero para el atributo *isSubsuption*)

llamada *belongsTo*, representada mediante una flecha con punta hueca. YAWL permite referenciar estos roles (entidad *Role*) en forma directa, o bien enlazarlos en forma tardía con base en el valor de variables para definir la asignación de recursos a las tareas. En el RPIM esto se define indicando valores verdaderos para los atributos *isReferenceable* e *isBindable* de la entidad *Role*, e indicando un valor *Variable* para su atributo *bindingType*.

YAWL también provee dos implementaciones de clasificador de recurso (elemento *ResourceClassifierImpl*) para dar soporte al *enfoque organizacional* para el modelado de estructuras de recursos. Una posición (entidad *Position*) representa un cargo o puesto de trabajo dentro de la organización, mientras que un grupo organizacional (entidad *OrgGroup*) representa un agrupamiento funcional de recursos dentro de la organización. Una posición organizacional puede reportar a otra posición organizacional, lo cual se expresa mediante la entidad de implementación de relación de recursos (elemento *ResourceReferenceImpl* con valor falso para el atributo *isSubsuption*) *reportsTo*, representada mediante una flecha con punta sólida. Esto permite expresar la jerarquía de control de la organización. La posibilidad de asociar recursos con posiciones se representa por medio de las implementaciones de relación de clasificación de recursos (elemento *ClassificationImpl*) llamadas *Hold*s que van desde *Participant* y *Administrator* a *Position*. Una posición también puede pertenecer a un grupo organizacional, lo cual se representa mediante la implementación de relación de inclusión *belongsTo* definida entre estas entidades. De este modo, los recursos humanos asociados con estas posiciones son incluidos en la población de los grupos organizacionales correspondientes. Además, pueden definirse jerarquías de grupos organizacionales mediante la implementación de relación de inclusión *belongsTo* definida en forma recursiva para *OrgGroup*.

Las entidades *Position* y *OrgGroup* pueden ser enlazadas en consultas parametrizadas o expresiones de asignación con valores constantes (atributo *isBindable* con valor verdadero y *bindingType* con valor *Literal*). Sin embargo, no pueden ser referenciadas directamente desde las tareas de usuario para definir la asignación de recursos a un ítem de trabajo (atributo *isReferenceable* con valor falso).

Aspectos Distribución de Trabajo y Autorización Dinámica en YAWL

Las políticas de distribución de trabajo son definidas en YAWL con base en la especificación de un conjunto de recursos elegibles para una tarea y una estrategia de interacción que define si cada uno de los tres puntos de interacción provistos por el sistema es disparado por los recursos o el sistema. Los puntos de interacción son: *Offer*, *Allocate* y *Start*. Los mismos llevan a los ítems de trabajo del estado inicial a los estados *Offered*, *Allocated* y *Started*. Estos estados corresponden a los momentos en que se determinó el conjunto de recursos elegibles para ejecutar el ítem de trabajo (*Offered*), se asignó un único

recurso para la ejecución del ítem de trabajo (Allocated) y el recurso asignado comenzó a ejecutar el ítem de trabajo (Started).

En el modelo conceptual de YAWL, una política de distribución de trabajo de tipo *pull* es definida indicando que es el recurso quien inicia los puntos de interacción Allocate y Start. En cuanto a las políticas de tipo *push*, es posible definir las en dos niveles. En el primero, el usuario decide cuándo comenzar a ejecutar el ítem de trabajo que le fue asignado, esto se define indicando que el punto de interacción Allocate es iniciado por el sistema y el punto de interacción Start iniciado por el recurso. En el segundo, el sistema dispara los puntos Allocate y Start, lo cual indica que el recurso debe ejecutar dicho ítem de trabajo en ese momento. Finalmente, cuando el punto de interacción Offer es iniciado por el usuario, el ítem de trabajo queda en espera que un recurso indicado como Administrator lo asigne a un recurso por medio de alguna de las operaciones de ítem de trabajo Offer, Allocate o Start que se muestran en la Figura 40.

Con el propósito de representar las políticas de distribución de trabajo de YAWL en términos del marco de trabajo propuesto, el RPIM de YAWL (Figura 41) define tres implementaciones de rol de recurso (elemento *ResourceRoleImpl*): PotentialOwner, Owner y Executor. Estas entidades representan el conjunto de operaciones de ítem de trabajo que pueden ser ejecutadas por los recursos sobre los ítems de trabajo cuando los mismos se encuentran en los estados Offered, Allocated y Started, respectivamente.

La implementación de rol de recurso PotentialOwner define privilegios para ejecutar las operaciones de ítem de trabajo (elemento *WorkItemOperationImpl*) Select y StartOffered, las cuales equivalen a disparar los puntos de interacción Allocate y Start. Los privilegios para ejecutar estas operaciones pueden ser revocados a nivel de tarea, tal como lo especifican las implementaciones de privilegio de tarea (elemento *TaskPrivilegeImpl*) select y startOffered, las cuales tienen el atributo *mandatory* con valor falso, por lo que se representan mediante flechas con líneas discontinuas.

La implementación de rol de recurso Owner define privilegios para ejecutar las operaciones de ítem de trabajo StartAllocated y Deallocate, las cuales llevan respectivamente a los ítems de trabajo a los estados Started y Offered. La operación StartAllocated es permitida de forma obligatoria por medio de la implementación de privilegio de tarea startAllocated, la cual tiene el atributo *mandatory* con valor verdadero. La operación Deallocate, la cual hace que el ítem de trabajo vuelva a ser visible por todos los recursos elegibles (estado Offered), puede ser revocada a nivel de tarea según lo define el atributo *mandatory* con valor falso para la implementación de privilegio de tarea deallocate.

Finalmente, la implementación de rol de recurso *Executor* permite ejecutar las operaciones de ítem de trabajo *Edit* y *Complete*, las cuales se encuentran habilitadas para los ítems de trabajo en el estado *Started*, en forma no revocable. Esto se define por medio de las implementaciones de privilegios de tarea *edit* y *complete*, los cuales tienen valor verdadero para el atributo *mandatory*.

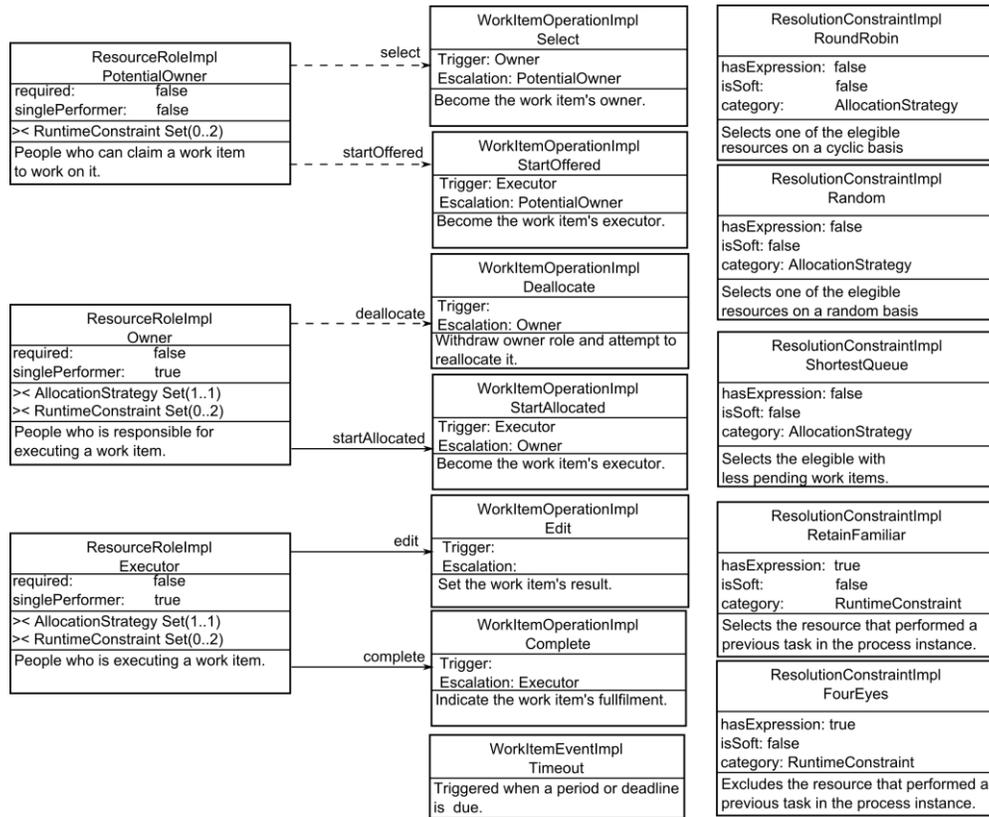


Figura 41. RPIM de Aspectos Distribución de Trabajo y Autorización Dinámica de YAWL.

Por medio de estas implementaciones de roles de recursos, es posible definir las estrategias de interacción de YAWL en términos de los conceptos provistos por el marco de trabajo propuesto. A continuación se explica cómo definir cada una de dichas estrategias de interacción en un PS-EPM importando el RPIM de YAWL:

- La especificación de asignaciones de recursos para un rol de recurso de una tarea, en donde la implementación del rol referencia a cualquiera de los roles del RPIM de YAWL (indicados en la Figura 41), equivale a especificar que el punto de interacción *offer* es disparado por el sistema. En caso de no especificar ningún rol de recurso para la tarea o definir un rol de recurso sin recursos asignados, se indica que el punto de interacción *offer* es disparado por el recurso. En tal caso, los ítems de trabajo serán asignados con el estado *unoffered* a los recursos indicados como *Administrator*, quienes se encargan de

asignarlos en el estado que decidan por medio de las operaciones *offer*, *allocate* y *start*.

- La especificación de asignaciones de recursos para un rol de recurso de una tarea, en donde la implementación del rol referencia al rol *Owner* o *Executor* del RPIM de YAWL, equivale a definir que el punto de interacción *allocate* es disparado por el sistema. Este punto de interacción es disparado por el recurso en caso que se definan asignaciones de recursos para un rol de recurso implementado por medio del rol *PotentialOwner*. Finalmente, el punto de interacción *start* es disparado por el sistema en caso que el rol de recurso de la tarea tenga como implementación el rol *Executor*, o que tenga el rol *PotentialOwner* y se revoque el privilegio *select*.

YAWL soporta dos tipos de restricción de resolución: estrategias de asignación (*AllocationStrategy*) y restricciones de tiempo de ejecución (*RuntimeConstraint*). Las mismas fueron representadas por medio de categorías de restricción de resolución. *RoundRobin* y *Random*, las cuales se representan como elementos de tipo *ResolutionConstraintImpl*, son estrategias de asignación. Ambas son restricciones duras. La definición de una estrategia de asignación es obligatoria para los roles *Owner* y *Executor*. *RetainFamiliar* y *FourEyes* son restricciones de tiempo de ejecución. Estas restricciones requieren definir una expresión con el propósito de especificar una tarea de usuario previa cuyo ejecutor debe ser asignado o excluido del rol siendo distribuido. Las restricciones de tiempo de ejecución pueden ser definidas para cualquiera de los roles *PotentialOwner*, *Owner* o *Executor*.

YAWL también define un evento de ítem de trabajo *Timeout* (representado por medio de un elemento *WorkItemEventImpl*), el cual es disparado cuando un período o fecha límite está vencido. Este evento permite realizar diferentes acciones de compensación ante su ocurrencia. Sin embargo, YAWL no permite disparar una operación que cambie la distribución de los ítems de trabajo. Esto se lo definió como una futura mejora a la plataforma (van der Aalst, et al., 2007).

6.1.2 Modelo de Implementación de la Perspectiva de Recursos de WS-HumanTask

Las especificaciones de *BPEL4People* y *WS-HumanTask* descritas en la Sección 2.4.2 proveen un enfoque para representar la perspectiva de recursos en procesos definidos empleando el lenguaje *WS-BPEL*. *WS-HumanTask* provee un modelo conceptual basado en roles para definir los aspectos *Distribución de Trabajo* y *Autorización Dinámica*. Este modelo conceptual fue adoptado y tomado como base por distintos *WfMSs* para definir los elementos a ser empleados para la especificación de dichos aspectos. Los aspectos *Estructura de Recursos* y *Autorización Estática* no son cubiertos por *WS-HumanTask*. Esta sección presenta un RPIM que representa las entidades de implementación especificadas

por WS-HumanTask con el propósito de mostrar la aplicabilidad y utilidad del RPIImplMeta para la descripción de WfMSs basados en esta especificación. El RPIM de la Figura 42 representa los siete tipos de rol de recurso definidos por WS-HumanTask, los cuales son referidos en la especificación del lenguaje como roles genéricos de recursos humanos.

El rol genérico de recursos Potential Owner soporta un enfoque *pull* de distribución de trabajo mediante un conjunto de permisos sobre operaciones que permiten a los recursos asumir voluntariamente la ejecución de los ítems de trabajo. El mismo es representado en el RPIM por medio de la implementación de rol de recurso (elemento *ResourceRoleImpl*) PotentialOwner, el cual define las implementaciones de privilegios de tarea (elemento *TaskPrivilegeImpl*) claim y start, sobre las implementaciones de operación de ítem de trabajo (elemento *WorkItemOperationImpl*) Claim y Start. Las mismas permiten a los recursos asumir la responsabilidad y comenzar la ejecución de un ítem de trabajo de la tarea mediante la adquisición del rol genérico de recursos ActualOwner, tal como lo muestran los atributos *trigger* y *escalation* definidos para estas operaciones. Un rol de recurso implementado como PotentialOwner puede ser asignado a múltiples recursos (atributo *singleResource* con valor falso). Además, los procesos implementados con plataformas basadas en WS-HumanTask tienen que definir, para cada una de sus tareas de usuario, un rol de recurso implementado como PotentialOwner, según lo indica el valor verdadero para el atributo *required* en esta entidad.

El rol genérico de recursos Actual Owner es provisto por WS-HumanTask para otorgar a los recursos privilegios sobre operaciones que les permiten completar los ítems de trabajo. Esto es representado en el RPIM mediante la entidad ActualOwner. Esta implementación de rol de recurso define un conjunto de privilegios no revocables sobre las implementaciones de operación de ítem de trabajo (elemento *WorkListOprationImpl*) Start, Stop, SetOutput, Fail y Complete. El rol genérico de recursos Actual Owner define además privilegios que permiten al recurso asignar otro recurso al rol Actual Owner (implementación de operación de ítem de trabajo Delegate), volver a hacer el ítem de trabajo accesible a los recursos asignados al rol PotentialOwner (implementación de operación de ítem de trabajo Release) o asignar el rol PotentialOwner a un conjunto de recursos diferente del especificado en la definición de la tarea (implementación de operación de ítem de trabajo Forward). Las especificaciones de tarea WS-HumanTask no pueden definir asignación de recursos al rol Actual Owner (atributo *isAssignable* con valor falso). Este rol puede ser adquirido por un único recurso para un ítem de trabajo (atributo *singlePerformer* con valor verdadero) por medio de la ejecución de las operaciones permitidas por el rol genérico de recursos Potential Owner, o en caso que el rol Potential Owner sea asignado a un único recurso, tal como lo especifica la implementación de evento de ítem de trabajo (elemento *WorkItemOperationImpl*) SinglePotentialOwner. Dicho

evento dispara de manera automática la ejecución de la operación Claim para el recurso asignado al rol Potential Owner en dicha circunstancia.

WS-HumanTask no provee entidades que permitan definir restricciones de resolución (elemento *ResolutionConstraintImpl*). Con el propósito de restringir el conjunto de recursos asignados al rol Potential Owner en tiempo de ejecución, esta especificación define un rol sin privilegios de tarea llamado Excluded Owner, el cual se representa mediante la implementación de rol de recurso ExcludedOwner. Los recursos asignados a este rol no pueden ser asignados al rol Potential Owner ni al rol Actual Owner.

WS-HumanTask también provee un conjunto de roles genéricos de recursos que no están asociados directamente con la asignación de responsabilidades sobre la ejecución de los ítems de trabajo correspondientes con tareas de usuario. El rol Task Initiator, representado mediante la entidad TaskInitiator, puede ser asignado a un recurso con el propósito de permitirle activar en forma manual la inicialización del ítem de trabajo, lo cual se representa mediante la definición de privilegios sobre la implementación de operación de ítem de trabajo Activate. El rol Notification Recipient, el cual se representa en el RPIM mediante la entidad NotificationRecipient, no permite la ejecución de ninguna operación de ítem de trabajo. El mismo puede ser asignado a los recursos con el propósito que reciban notificaciones disparadas ante la reasignación de los roles Potential Owner o Actual Owner. Dicha reasignación puede ser disparada mediante la ocurrencia de eventos de ítem de trabajo implementados mediante las entidades StartDeadline y CompleteDeadline, los cuales permiten definir respectivamente el escalado de los roles Potential Owner y Actual Owner; y la asignación de un nuevo conjunto de recursos al rol Potential Owner.

WS-HumanTask provee dos roles que permiten a los recursos ejecutar operaciones de ítem de trabajo de carácter administrativo: Task Stakeholder y Business Administrator, los cuales se representan mediante las entidades TaskStakeholder y BusinessAdministartor. Estos roles permiten a los recursos activar manualmente la inicialización de ítems de trabajo por medio de la operación de ítem de trabajo Activate. Por otra parte, definen privilegios para ejecutar las operaciones Suspend y Resume mediante las cuales pueden indicar al recurso asignado al rol Actual Owner, cuándo suspender temporalmente y reanudar la ejecución del ítem de trabajo. Además, otorgan privilegios para ejecutar las operaciones Delegate y Forward, las cuales permiten a los recursos modificar los recursos asignados a los roles Potential Owner y Actual Owner. Finalmente, el rol Business Administrator tiene además privilegios para eliminar ítems de trabajo por medio de la operación Remove.

6.2 Aplicación del Método Propuesto en la Definición de la Perspectiva de Recursos en el Desarrollo de SIOPs

En esta sección se describe la aplicación del marco de trabajo y el método propuestos para definir la perspectiva de recursos en el desarrollo de SIOPs mediante dos casos de estudio. Para ambos casos, se introduce el propósito del caso de estudio y luego se detallan los resultados de cada una de las etapas del método propuesto.

6.2.1 Caso de Estudio: Proceso de Gestión de Fondos para Obras Menores

Los procesos administrativos de organizaciones burocráticas como los gobiernos constituyen un ejemplo frecuente de procesos de workflow, en los que la mayoría de las tareas son llevadas a cabo por personas. En este contexto, el desempeño de los procesos depende en gran parte del modo en que dichas tareas son distribuidas entre los recursos para su ejecución. La distribución de trabajo debe además llevarse a cabo teniendo en cuenta la estructura organizacional así como sus políticas y regulaciones internas. El marco de trabajo y el método propuestos fueron aplicados para rediseñar e implementar un proceso relacionado con la asignación de fondos para obras menores en un gobierno provincial. El objetivo de este proceso es gestionar las solicitudes de fondos recibidas por los gobiernos de municipios y comunas, con el propósito de ejecutar obras de infraestructura menores. Más de doscientas solicitudes son autorizadas cada año a las municipalidades y comunas de la provincia. Esto requiere la coordinación de diferentes áreas técnicas y administrativas del gobierno provincial, así como la coordinación y asignación de las tareas a las personas que trabajan en dichas áreas. Como parte de este caso de estudio se realizó el diseño del proceso mencionado y el desarrollo del SIOP que lo implementa, aplicando el método propuesto en esta tesis. A continuación se describen cada una de las etapas realizadas.

Definición de la Lógica del Proceso de Negocio

El proceso de asignación de fondos fue rediseñado. Para ello, primero se definió la lógica del proceso de negocio, indicando únicamente las tareas correspondientes con las funciones organizacionales a desempeñar; y excluyendo toda tarea que tenga que ver con aspectos de implementación, tales como aquellas definidas para indicar conversiones de datos requeridas o implementar una política específica de distribución de trabajo. El proceso rediseñado de asignación de fondos comienza con la recepción de una solicitud de fondos y termina con su rechazo o aceptación junto con el monto autorizado. Este proceso se muestra en el BPLM de la Figura 43.

Una solicitud de fondos recibida de una municipalidad o comuna, la cual contiene un proyecto describiendo la obra menor a ser realizada es derivada a la Dirección de Asistencia Administrativa donde un recepcionista administrativo asigna un número de expediente para la misma. Luego de ello, una tarea automática comprueba si el municipio tiene rendiciones de cuentas pendientes, en cuyo caso es notificado. En caso de existir

rendiciones pendientes, la municipalidad o comuna debe presentar dichas rendiciones de cuentas y notificar su recepción dentro del plazo de un mes a fin de evitar que el proceso termine con el estado Solicitud Rechazada.

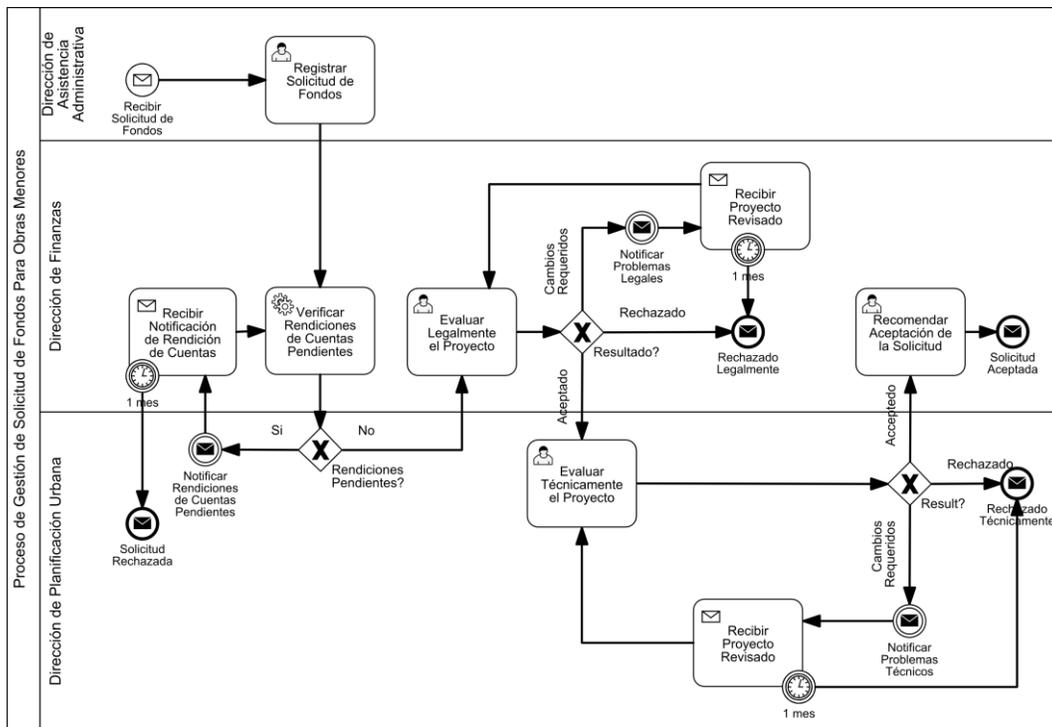


Figura 43. BPLM Proceso de Gestión de Fondos para Obras Menores.

Si la municipalidad no tiene rendiciones de cuentas pendientes, la solicitud es dirigida a la Dirección de Finanzas a fin de ser evaluada respecto de leyes nacionales y municipales por un asesor legal. En caso que el proyecto se encuentre en conflicto con alguna ley, el mismo es rechazado y el proceso finaliza con una notificación de este resultado. En caso que el proyecto pueda adecuarse a la ley con cambios menores, una notificación de los problemas legales que presenta es enviada a la municipalidad o comuna, la cual debe llevar a cabo los cambios requeridos en el transcurso de un mes para evitar el rechazo de la solicitud. Luego, el proyecto modificado es recibido y evaluado nuevamente por el asesor legal.

Si el proyecto pasa la evaluación legal, el mismo es evaluado desde el punto de vista técnico por un asesor de la Dirección de Planificación Urbana. Esta evaluación puede resultar en el rechazo, la aceptación sujeta a cambios o la aceptación del proyecto. En el último caso, la solicitud es derivada al director de la Dirección de Finanzas, quien emite una recomendación de aceptación para la solicitud de fondos, incluyendo el monto a ser asignado para la misma, la cual es elevada a las autoridades pertinentes con el propósito de ser efectivizada.

Definición de los Requerimientos de Perspectiva de Recursos

La definición de los requerimientos de la perspectiva de recursos comenzó con la representación en un PI-RSM de los recursos involucrados en el proceso. Este modelo fue definido con base en la estructura del gobierno, siguiendo un enfoque de modelado de recursos dirigido por la organización (Zur Muehlen, 2004). Por lo tanto, los recursos fueron clasificados empleando conceptos identificados en el análisis de la estructura del gobierno. El modelo de la Figura 44 fue creado por medio de la herramienta descrita en la sección 5.1.

La estructura del gobierno presenta tres niveles jerárquicos: ministerios, secretarías y direcciones. Estos niveles fueron definidos como elementos clasificadores de recursos conectados por relaciones de inclusión. El proceso involucra a la Dirección de Asistencia Administrativa y la Dirección de Planificación Urbana, las cuales pertenecen a la Secretaría de Municipios; y a la Dirección de Finanzas que es parte de la Secretaría de Administración. Ambas secretarías pertenecen al Ministerio de Gobierno.

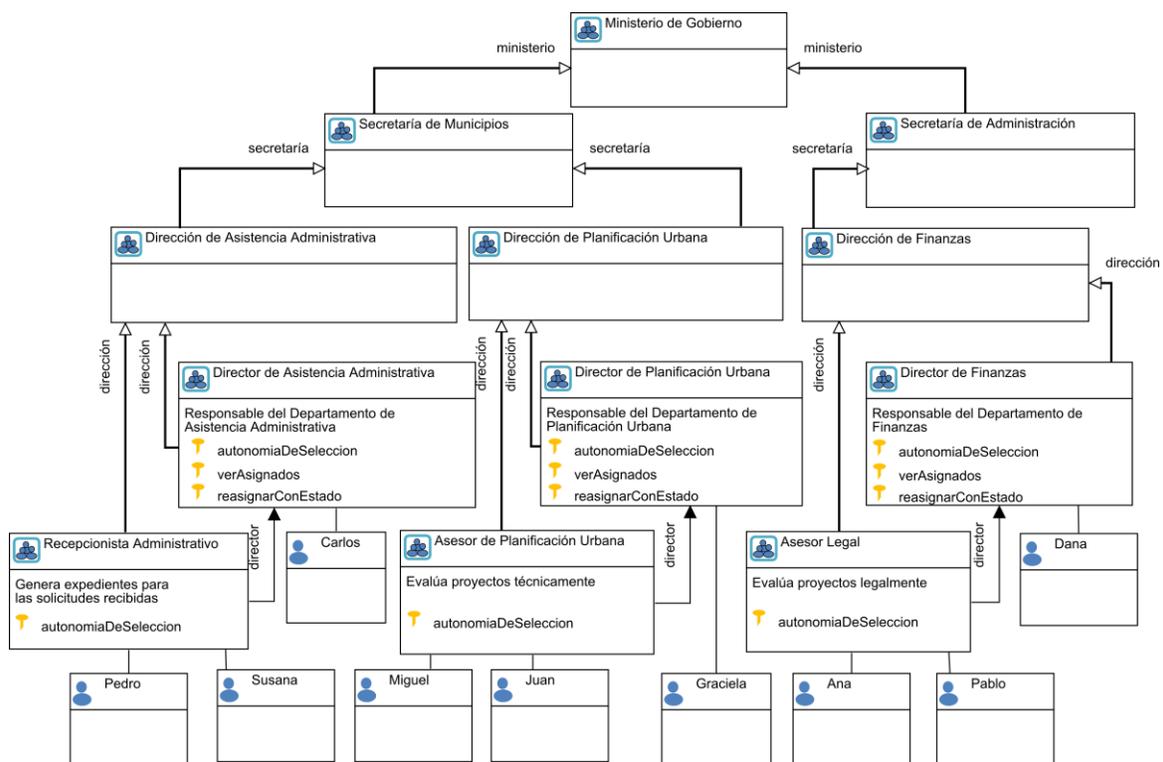


Figura 44. PI-RSM de los Recursos Involucrados en el Proceso de Gestión de Fondos.

Las direcciones definen puestos de trabajo o posiciones tales como director, recepcionista o asesor, los cuales también fueron definidos por medio de clasificadores de recursos. Estas posiciones se asocian por medio de líneas de reporte, tales como la representada por la relación de recursos director. Dichas posiciones son ocupadas por

personas representadas como recursos humanos asociados con las mismas por medio de relaciones de clasificación. El modelo de recursos incluye privilegios definidos para estas posiciones, los cuales fueron definidos empleando el vocabulario provisto por los patrones de recursos de workflow. El privilegio de recursos *autonomiaDeSeleccion* hace referencia al patrón Autonomía de Selección (patrón número 26), el privilegio *verAsignados* al patrón Visibilidad Configurable de Ítems de Trabajo Asignados (patrón número 41) y el privilegio *reasignarConEstado* al patrón Reasignación con Estado (patrón número 30).

Los requerimientos de distribución de trabajo para las tareas de usuario del proceso de gestión de fondos fueron definidos en el PI-EPM de la Figura 45, el cual fue derivado del BPLM de la Figura 43. La tarea Registrar Solicitud de Fondos es llevada a cabo por recepcionistas de la Dirección de Asistencia Administrativa. Un requerimiento para esta tarea fue la distribución equitativa de los ítems de trabajo entre los mismos. Para satisfacer este requerimiento, se definió una política de distribución de trabajo de tipo *push*, la cual asigna cada ítem de trabajo de la tarea a un recepcionista diferente de manera cíclica. Esto fue representado por medio del rol de recurso Responsable, el cual establece que el recurso asignado que tendrá a su cargo la ejecución del ítem de trabajo para la instancia de la tarea. El mismo es asignado a los recepcionistas por medio de la referencia de recursos (Ref) al clasificador de recurso Recepcionista Administrativo. La asignación del rol en forma cíclica es representada por medio de la restricción de resolución Ciclo.

Una política de distribución de trabajo diferente fue requerida para las tareas Evaluar Legalmente el Proyecto y Evaluar Técnicamente el Proyecto. Dado que el esfuerzo requerido para evaluar diferentes proyectos puede variar considerablemente, una política de distribución de trabajo *push* puede tener un impacto negativo en el desempeño del proceso. Por lo tanto, se definió una estrategia *pull* con el propósito de permitir a los asesores decidir qué proyectos evaluar y en qué orden. Otro requerimiento para estas tareas fue dar la posibilidad al Director de Finanzas y al Director de Planificación Urbana de asumir, supervisar y cambiar la asignación de los ítems de trabajo de estas tareas de usuario. Esto fue representado por medio de la asignación de dos roles para estas tareas. El primero es Candidato, el cual permite a los asesores reclamar y ejecutar ítems de trabajo. El mismo es asignado a los asesores por medio de la referencia de recursos (Ref) al clasificador de recurso Asesor Legal o Asesor de Planificación Urbana, respectivamente. El segundo es Administrador, que permite a los directores asumir y asignar los ítems de trabajo de estas tareas. Este rol de recurso es asignado por medio de la referencia de recursos (Ref) al clasificador de recurso Director de Finanzas o Director de Planificación Urbana según corresponda. De esta manera, cuando se crea un ítem de trabajo para alguna de estas tareas en el contexto de una instancia del proceso, dicho ítem de trabajo es ofrecido a todos los asesores del área correspondiente para que asuman su ejecución tanto en forma voluntaria como por indicación de su director, quien también puede asumir la ejecución del mismo.

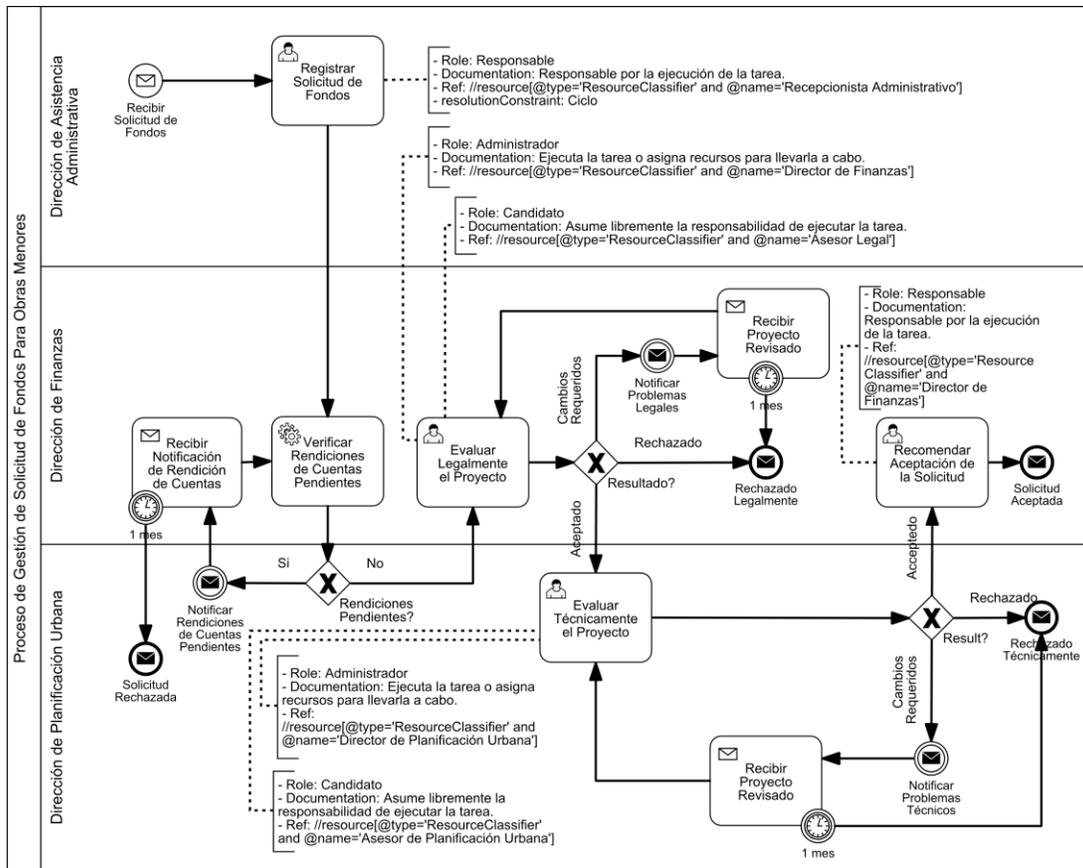


Figura 45. PI-EPM del Proceso de Gestión de Fondos.

Finalmente, una política de distribución de trabajo *push* fue definida para la tarea Recomendar Aceptación de la Solicitud, la cual tiene que ser llevada a cabo por el Director de Finanzas de acuerdo con las responsabilidades definidas para este cargo en el gobierno provincial. Con este propósito, el rol Responsable es asignado por medio de la definición de una referencia de recursos a este clasificador de recurso.

Selección de la Plataforma de Implementación

Un requerimiento al comienzo de este proyecto fue que el SIOP que dé soporte al proceso de gestión de solicitudes de fondos sea implementado con base en el WfMS Bonita. Por lo tanto, esta etapa consistió en la evaluación del soporte provisto por este WfMS a los requerimientos de la perspectiva de recursos identificados para este proceso en la etapa anterior.

La evaluación se realizó mediante la técnica propuesta y descrita en el Capítulo 4, Sección 4.4. Esta técnica se aplicó sobre el RPIM que representa la plataforma Bonita, el cual fue presentado en el Capítulo 3, Sección 3.3.2 (Figuras 18 y 19). El resultado de la

evaluación se resume en las Tablas 8 y 9. En las mismas se indica el conjunto de implementaciones candidatas para cada recurso humano, clasificador de recurso o rol de recurso definido en el PI-RSM y el PI-EPM junto con los valores obtenidos para las propiedades de evaluación en los mismos. En la última fila de dichas tablas se incluye además el listado de las entidades de implementación provistas por la plataforma que no proveyeron soporte a ninguno de los recursos y roles de recurso definidos en los modelos independientes de la plataforma, también acompañados con los valores obtenidos para cada propiedad de evaluación.

La Tabla 8 muestra los resultados de la evaluación respecto de los aspectos Estructura de Recursos y Autorización Estática. Los clasificadores de recursos Ministerio de Gobierno, Secretaría de Municipios, Secretaría de Administración, Dirección de Asistencia Administrativa y Dirección de Finanzas, los cuales definen la descomposición jerárquica del ministerio, pueden ser implementados a través del clasificador de recurso Group de Bonita. No obstante, ninguna de las implementaciones de clasificación de recursos provistas por este WfMS soporta las propiedades *isSource* de los clasificadores de recursos Recepcionista, Asesor Legal y Asesor Técnico; ni la propiedad *isTarget* de los clasificadores de recursos Director de Asistencia Administrativa, Director de Planificación Urbana y Director de Finanzas. Por lo tanto, Bonita no permite definir relaciones entre clasificadores de recursos, tales como la línea de reporte director entre Recepcionista y Director de Asistencia Administrativa. Finalmente, los recursos humanos definidos en el PI-RSM pueden ser implementados a través de las entidades User y Adminsitrator provistas por esta plataforma.

Como resultado de la evaluación de estos aspectos se obtuvo una tasa de soporte resultante de 0,71. Esto indica que Bonita no provee un soporte completo a la implementación del PI-RSM definido. No obstante, esto no implica que Bonita no puede ser utilizado, sino que habrá que introducir modificaciones en la próxima etapa a fin de implementar la caracterización y clasificación de los recursos agrupados por los clasificadores Recepcionista Administrativo, Asesor Legal, Asesor Técnico, Director de Asistencia Administrativa, Director de Planificación Urbana y Director de Finanzas, respetando las limitaciones impuestas por la plataforma.

Tabla 8. Resumen de Implementaciones Candidatas de Estructura de Recursos provistas por Bonita.

	isResource	isClassifier	isReferenceable	isBindable	subsumes	isSubsumed	isSource	isTarget	hasPopulation	isMember	Has Privileges	hasParameters	hasBindableParams
Min. de Gobierno	0	1	0	0	1	0	0	0	0	0	0	0	0
<i>Group</i>	0	1	1	0	1	1	0	0	0	0	0	0	0
Sec. Municipalidades	0	1	0	0	1	1	0	0	0	0	0	0	0
<i>Group</i>	0	1	1	0	1	1	0	0	0	0	0	0	0
Sec. Administración	0	1	0	0	1	1	0	0	0	0	0	0	0
<i>Group</i>	0	1	1	0	1	1	0	0	0	0	0	0	0
Dirección Asistencia Adm.	0	1	0	0	1	1	0	0	0	0	0	0	0
<i>Group</i>	0	1	1	0	1	1	0	0	0	0	0	0	0
Dirección Plan. Urbana	0	1	0	0	1	1	0	0	0	0	0	0	0
<i>Group</i>	0	1	1	0	1	1	0	0	0	0	0	0	0
Dirección de Finanzas	0	1	0	0	1	1	0	0	0	0	0	0	0
<i>Group</i>	0	1	1	0	1	1	0	0	0	0	0	0	0
Dir. Asistencia Admin.	0	1	1	0	0	1	0	1	1	0	1	1	0
Dir. Plan. Urbana	0	1	1	0	0	1	0	1	1	0	1	1	0
Dir. Finanzas	0	1	1	0	0	1	0	1	1	0	1	1	0
Recepcionista Admin.	0	1	1	0	0	1	1	0	1	0	1	1	0
Asesor Planif. Urbana	0	1	1	0	0	1	1	0	1	0	1	1	0
Asesor Legal	0	1	1	0	0	1	1	0	1	0	1	1	0
Recursos Individuales	1	0	0	0	0	0	0	0	0	0	1	0	0
<i>User</i>	1	0	1	1	0	0	1	1	0	1	0	1	1
<i>Administrator</i>	1	0	1	1	0	0	1	1	0	1	0	1	1
Otros ResourceImpl													
<i>Membership</i>	0	1	1	0	0	1	0	0	1	0	0	0	0
<i>Group</i>	0	1	1	0	1	1	0	0	0	0	0	0	0
<i>Role</i>	0	1	1	0	1	0	0	0	0	0	0	0	0

La Tabla 9 indica los resultados de la evaluación respecto de los aspectos Distribución de Trabajo y Autorización Dinámica. Esto es la evaluación respecto al soporte provisto por Bonita a estos aspectos definidos en el PI-EPM. Los roles de recursos Candidato y Administrador definidos en el PI-EPM pueden ser soportados por la entidad Actor de Bonita, la cual es un rol de recurso provisto por la plataforma. Esto se debe a que dicha entidad soporta las propiedades *isAssignable* y *multiplePerformers*, las cuales son requeridas por los roles Candidato y Administrador. El rol Responsable definido en el PI-EPM es soportado por la entidad AssignedActor de Bonita, ya que esta soporta las propiedades *isAssignable*, *singlePerformer* y *hasHardConstraints*. Como resultado de la evaluación de

estos aspectos se obtuvo una tasa de soporte resultante de 1. Esto indica que las entidades provistas por Bonita son suficientes para implementar los aspectos definidos en el PI-EPM.

Tabla 9. Resumen de Implementaciones Candidatas de Distribución de Trabajo Provistas por Bonita.

	isAssignable	multiplePerformers	singlePerformer	hasHardConstraints	hasSoftConstraints	hasTriggers	hasEscalations	revokablePrivileges
Responsable (Registrar Solicitud de fondos) <i>AssignedActor</i>	1 1	0 0	1 1	1 1	0 0	0 0	0 0	0 0
Candidato (Evaluar Legalmente el Proyecto) <i>Actor</i>	1 1	1 1	0 0	0 1	0 0	0 0	0 0	0 0
Administrador (Evaluar Legalmente el Proyecto) <i>Actor</i>	1 1	1 1	0 0	0 1	0 0	0 0	0 0	0 0
Candidato (Evaluar Técnicamente el Proyecto) <i>Actor</i>	1 1	1 1	0 0	0 1	0 0	0 0	0 0	0 0
Administrador (Evaluar Técnicamente el Proyecto) <i>Actor</i>	1 1	1 1	0 0	0 1	0 0	0 0	0 0	0 0
Responsable (Recomendar Aceptación Solicitud) <i>AssignedActor</i>	1 1	0 0	1 1	1 1	0 0	0 0	0 0	0 0

Definición de la Implementación de la Perspectiva de Recursos

La implementación de la perspectiva de recursos se definió en un PS-RSM y un PS-EPM. Los mismos se crearon a partir del PI-RSM y el PI-EPM definidos para este proceso, teniendo en cuenta los requerimientos expresados en estos últimos empleando lenguaje natural y los resultados de la evaluación de las entidades de implementación provistas por Bonita.

El PS-RSM resultante se muestra en la Figura 46. De acuerdo con lo indicado en la Tabla 8, los clasificadores de recursos que representan al ministerio, sus secretarías y direcciones se implementaron a través de la entidad Group de Bonita. Por su parte, las relaciones de inclusión entre estos clasificadores de recursos se implementaron a través de la entidad belongsTo. Los clasificadores de recursos que definen puestos de trabajo, tales como director y asesor de secretarías, fueron implementados mediante la entidad Membership, la cual refiere al único clasificador de recurso provisto por Bonita que puede tener una población de recursos, definida mediante relaciones de clasificación.

La entidad Membership de Bonita representa el desempeño por parte de un recurso humano (entidad User o Administrator) de un determinado rol (entidad Role) en el contexto

de una determinada unidad organizacional (entidad Group). Como se desprende del valor 1 para las cardinalidades mínimas de las entidades de implementación de relaciones de inclusión is y of (Figura 18 del RPIM de Bonita), la implementación de un clasificador de recurso a través de la entidad Membership requiere que dicho clasificador de recurso defina relaciones de inclusión con un clasificador de recurso implementado por la entidad Role y con otro clasificador de recurso implementado por la entidad Group. Para satisfacer esta restricción, las relaciones de inclusión entre los clasificadores de recursos que representan puestos de trabajo y los que representan direcciones se implementaron por medio de relaciones de inclusión of de Bonita. Además, fue necesario crear los clasificadores de recursos Recepcionista, Asesor y Director. Los mismos fueron implementados mediante Role, y son objeto de relaciones de inclusión is, desde los clasificadores de recursos correspondientes implementados por Membership.

Los recursos humanos que pertenecen a los clasificadores de recurso Recepcionista y Asesor fueron implementados por medio de la entidad User de Bonita. El parámetro de recursos denominado director fue agregado para estos recursos humanos con el fin de implementar las líneas de reporte con sus directores, lo cual no es soportado por bonita mediante relaciones de recursos. Por otro lado, los recursos humanos clasificados como Director fueron implementados por medio de la entidad Administrator, con el propósito de permitirles cambiar la distribución de los ítems de trabajo asignados a sus subordinados.

Bonita no soporta la definición de privilegios de recurso a nivel de clasificador de recurso. De este modo, los privilegios `autonomiaDeSelección`, `verAsignados` y `reassignarConEstado` implementados respectivamente mediante las entidades `reorder`, `manager` y `assign` fueron definidos para los recursos humanos correspondientes en forma individual. También fue necesario agregar un privilegio de recurso `quitar` implementado como `unassign` a los recursos implementados como Administrator dado que dicho privilegio es obligatorio en bonita para usuarios administradores.

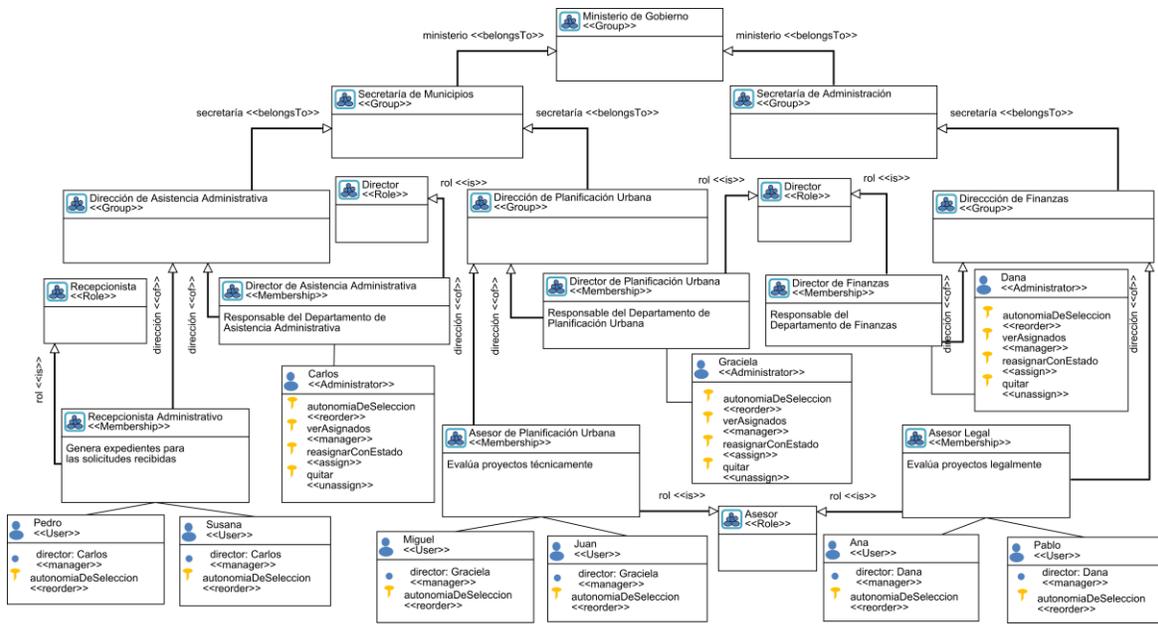


Figura 46. PS-RSM del Proceso de Gestión de Fondos.

El PS-EPM que define la implementación de los aspectos Distribución de Trabajo y Autorización Dinámica del proceso de solicitud de fondos se muestra en la Figura 47. Cada uno de los roles de recurso definidos en el PI-EPM se implementó de acuerdo a la entidad de Bonita indicada como candidata en la Tabla 9.

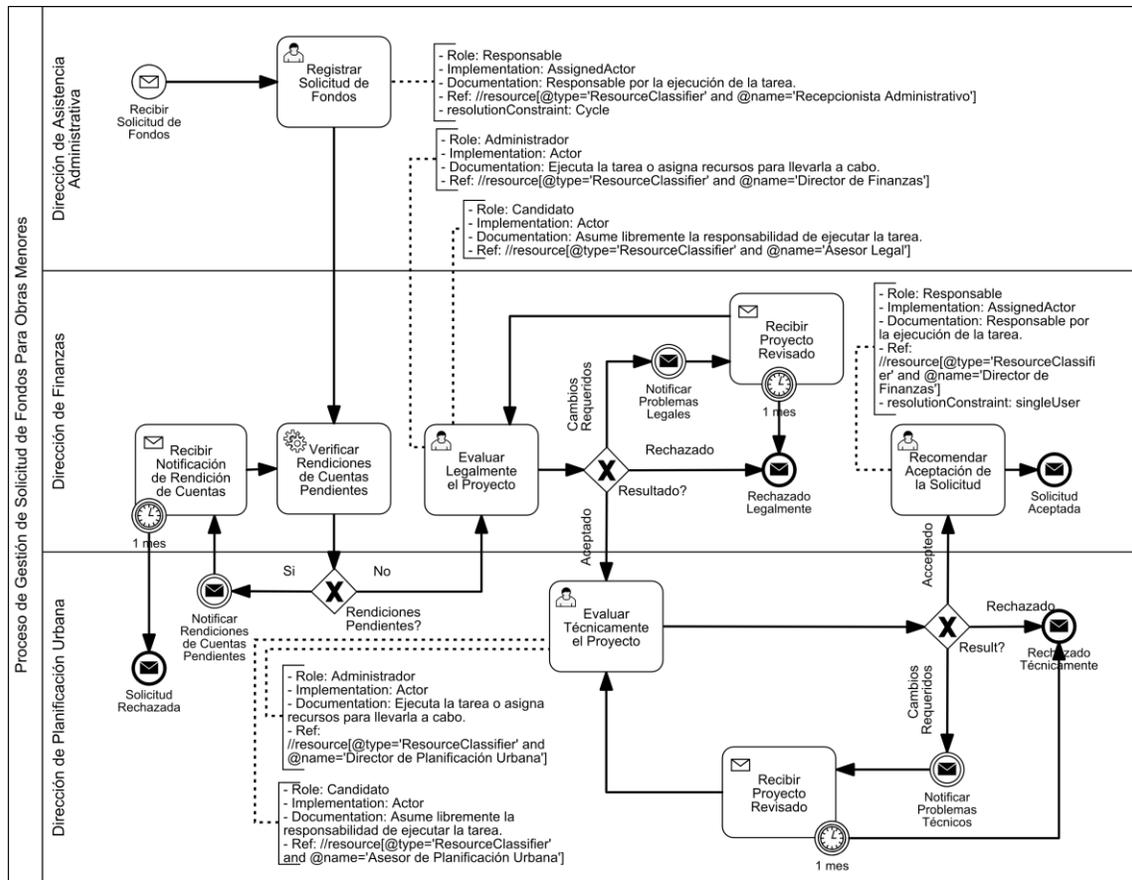


Figura 47. PS-EPM del Proceso de Gestión de Fondos.

Los roles de recurso Responsable de las tareas de usuario Registrar Solicitud de Fondos y Recomendar Aceptación de la Solicitud fueron implementados por medio de la entidad AssignedActor. Además de que esta entidad soporta las propiedades requeridas por estos roles (como se describió en la etapa anterior), la política de distribución de trabajo implementada por la entidad AssignedActor indica una estrategia *push*, la cual es la requerida para los roles Responsable de estas tareas.

La asignación de recursos para la tarea Registrar Solicitud de Fondos fue definida por medio de la referencia del clasificador Recepcionista Administrativo. La restricción de resolución Cycle no es soportada directamente por Bonita. La misma fue desarrollada en el contexto del proyecto como un filtro de asignación de actores de Bonita, haciendo uso de la API provista por esta plataforma. Este filtro de asignación de actores fue incluido en el RPIM de Bonita a fin de permitir su uso en esta instancia. La asignación de recursos para el rol Responsable de la tarea Recomendar Aceptación de la Solicitud fue definida por medio de la referencia al clasificador Director de Finanzas y la especificación de la restricción de resolución singleUser, la cual debió ser agregada para satisfacer la necesidad de definir una

restricción de la categoría `AssigningActorFiltering` para los roles de recurso implementados mediante la entidad `AssignedActor`.

Los roles de recurso `Candidato` y `Administrador` definidos para las tareas `Evaluar Aspectos Legales del Proyecto` y `Evaluar Aspectos Técnicos del Proyecto` fueron implementados por medio de la entidad `Actor`, debido a que se requiere implementar una estrategia de distribución de trabajo *pull* para estos roles. Los privilegios adicionales requeridos por los directores a ser asignados al rol de recurso `Administrador` para estas tareas fueron otorgados a los mismos por medio de su implementación como `Administrator`, dado que Bonita provee un enfoque de autorización estática para soportar este tipo de requerimiento.

Verificación y Validación de la Implementación de Perspectiva de Recursos

La implementación de los requerimientos de la perspectiva de recursos en la etapa anterior requirió la introducción de cambios en los modelos específicos de la plataforma con respecto de los modelos independientes de la plataforma dado que Bonita no provee un soporte directo a todos los requerimientos definidos en el PI-RSM. Esto hizo necesario aplicar las reglas de verificación y el procedimiento de validación (definidos en el Capítulo 4, Sección 4.6) con el propósito de constatar que el PS-RSM y el PS-EPM resultantes satisfacen las restricciones impuestas por el WfMS e implementan los requerimientos de perspectiva de recursos definidos de manera consistente.

Las reglas de verificación, las cuales se encuentran automatizadas en la herramienta desarrollada, fueron aplicadas en repetidas oportunidades sobre el PS-RSM y el PS-EPM resultantes, ya que los mismos fueron refinados hasta que el resultado de la aplicación de dichas reglas indicó que las restricciones impuestas por Bonita fueron satisfechas.

El procedimiento de validación también fue aplicado con el propósito de constatar la consistencia del PS-RSM y el PS-EPM respecto a los requerimientos definidos en el PI-RSM y el PI-EPM. La validación detectó inconsistencias originadas en la implementación de las relaciones de recursos director por medio de parámetros de recursos; y en la presencia de privilegios de recurso quitar implementados mediante la entidad `unassign`. Con base en estos resultados, se decidió continuar con el desarrollo del SIOP. Esto se debe a que el reemplazo de las relaciones de recurso director por parámetros de recurso con el mismo nombre implementados mediante la entidad `manager` fue una decisión de diseño tomada con el propósito de permitir la representación de las líneas de reporte definidas en el PI-RSM mediante las entidades de Bonita. En lo que respecta a la adición del privilegio de recurso quitar que fue requerida para los directores, esto no fue evaluado como un problema desde el punto de vista de la seguridad del sistema resultante.

Generación de Especificaciones Ejecutables

El PS-RSM y el PS-EPM de las Figuras 46 y 47 fueron transformados en tres documentos XML, los cuales fueron importados en el WfMS Bonita: un documento de proceso BPMN, un Esquema Organizacional de Bonita y un Documento de Mapeo de Actores. El documento de proceso BPMN fue generado empleando el complemento de exportación en formato BPMN 2.0 provisto por Oryx. Los documentos restantes fueron generados por medio de dos transformaciones de modelo a código desarrolladas para el RPIM de Bonita e incorporadas en la herramienta propuesta como complementos de servidor. La Figura 48 muestra una captura del esquema de recursos importado en el módulo de administración de modelos de recursos de Bonita y la especificación del proceso de gestión de fondos con la estrategia de distribución de trabajo para la tarea de usuario Registrar Solicitud de Fondos.

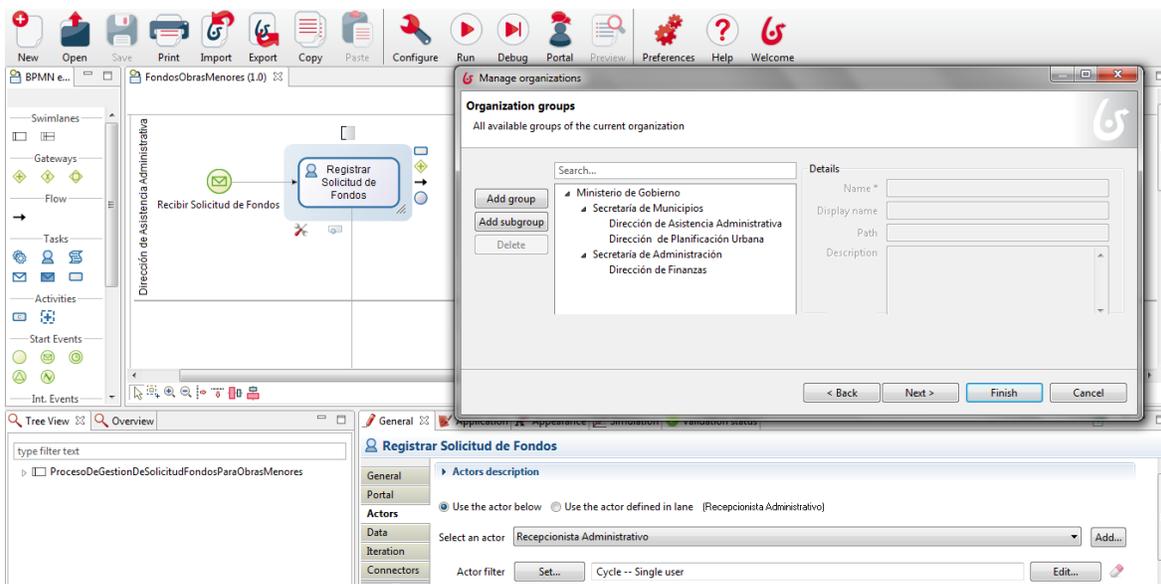


Figura 48. Proceso de Solicitud de Fondos Implementado en el WfMS Bonita.

Conclusiones del Caso de Estudio

La aplicación del marco de trabajo y el método propuestos para la definición de la perspectiva de recursos en el desarrollo de SIOPs para el presente caso de estudio mostró como resultado que:

- Los distintos artefactos de modelado contemplados por el método facilitaron la comunicación y el entendimiento de los requerimientos de perspectiva de recursos del proceso por los responsables de la organización y ejecutores de tareas, como así también de los desarrolladores. Los primeros trabajaron con los modelos independientes de la plataforma, creando representaciones estructuradas de sus requerimientos. Los segundos llevaron a cabo la definición

de modelos específicos de plataforma haciendo uso de la herramienta para verificar la adecuación de los mismos a las restricciones impuestas por Bonita y constatar la consistencia con los modelos independientes de la plataforma definidos.

- El método permitió un rediseño eficiente del proceso para incluir una definición detallada de los aspectos relacionados con la perspectiva de recursos, sin la necesidad de introducir tareas adicionales como parte de la lógica del proceso. Esto permite abstraerse de los detalles relacionados con esta perspectiva durante la definición de las restantes perspectivas del proceso.
- Aunque la plataforma de implementación ya estaba decidida por la organización, la aplicación de la técnica propuesta para la evaluación de plataformas de implementación respecto de los requerimientos de perspectiva de recursos definidos en los modelos independientes de la plataforma proveyó información que permitió guiar el proceso de desarrollo del SIOP en las etapas posteriores del método y obtener una solución adecuada respecto de las restricciones impuestas por la plataforma Bonita y los requerimientos definidos con los responsables de la organización.
- Los requerimientos referidos a los aspectos Distribución de Trabajo y Autorización Dinámica comprendían la definición de políticas de distribución *pull* y *push* incluyendo restricciones de resolución. Dichos requerimientos fueron soportados por Bonita. Sin embargo, en forma consistente con el valor obtenido para la tasa de soporte, varios cambios fueron requeridos en el PS-RSM para definir la implementación de los aspectos Estructura de Recursos y Autorización Estática en términos de las entidades de implementación provistas por esta plataforma.
- El uso de la herramienta posibilitó reducir los tiempos de desarrollo mediante la automatización de tareas de inicialización del PS-RSM y el PS-EPM a partir del PI-RSM y el PI-EPM. Más importante aún fue el soporte a la automatización de las técnicas de evaluación de plataformas, así como de las reglas y procedimientos de validación y verificación.
- Se logró como resultado un SIOP basado en Bonita, el cual implementa y ejecuta instancias del proceso de gestión de solicitud de fondos para obras menores, manteniendo un alto grado de consistencia con lo definido a nivel conceptual en la solución organizacional.

6.2.2 Caso de Estudio: Proceso de Gestión de Presupuestos de Muebles a Medida

Esta sección presenta un caso de estudio llevado a cabo en un fabricante de muebles a medida que opera en el noreste de la provincia de Santa Fe. El proyecto en consideración consistió en el desarrollo de un portal Web integrado con un SIOP que provea soporte a un

proceso de gestión de presupuestos de muebles a medida. Los objetivos del mismo fueron reducir el tiempo requerido para elaborar un presupuesto y expandir el área de operación del fabricante. Como parte del proyecto se realizó el diseño del proceso y el desarrollo del SIOPs que lo implementa, aplicando el método propuesto en esta tesis. A continuación se describen cada una de las etapas realizadas.

Definición de la Lógica del Proceso de Negocio

La Figura 49 muestra el BPLM en el que representa la lógica del proceso de gestión de presupuestos de este fabricante de muebles a medida. El mismo comienza cuando un cliente ingresa una solicitud de presupuesto (RFQ) por medio del sitio Web del fabricante. La solicitud consiste en una descripción informal del mueble a ser presupuestado e información de contacto incluyendo el número de teléfono del cliente, su dirección de correo electrónico y domicilio. La solicitud recibida es dirigida a un representante de ventas, quien se ocupa de contactar al cliente y elaborar una especificación detallada del mueble solicitado. La especificación del mueble es luego dirigida al departamento de fabricación, en el cual un diseñador prepara una lista de los materiales requeridos para producir el mueble. El costo de los materiales requeridos luego es calculado automáticamente.

Con base en el costo de fabricación, un representante de ventas elabora el presupuesto. En caso que el monto total del presupuesto supere los \$25.000, un descuento especial es elaborado por el gerente de ventas. En caso que el gerente de ventas no efectúe este descuento en el curso de un día, la tarea se cancela y el proceso continúa con el presupuesto elaborado inicialmente por el representante de ventas. En cualquier caso, el presupuesto es enviado por correo electrónico al cliente.

Finalmente, un representante de ventas contacta al cliente con el propósito de explicarle los detalles del presupuesto y solicitar su confirmación. El proceso finaliza con el presupuesto en el estado aceptado o rechazado con base en la respuesta del cliente.

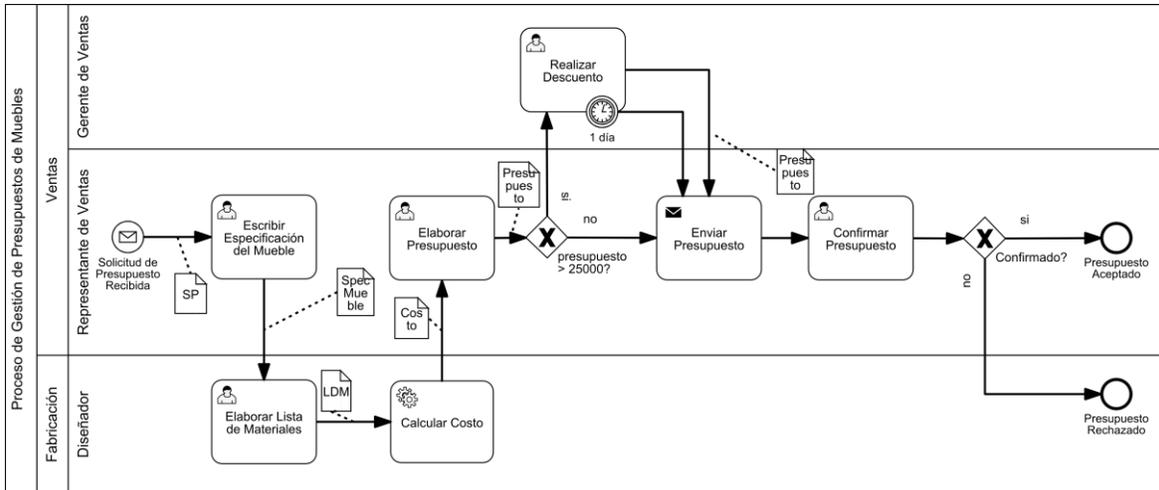


Figura 49. BPLM de Proceso de Gestión de Presupuestos de Muebles a Medida.

Definición de los Requerimientos de la Perspectiva de Recursos

La Figura 50 muestra el PI-RSM definido para este caso de estudio. Los departamentos de la organización involucrados en el proceso son los de Ventas y Fabricación, los cuales se representan como clasificadores de recursos. El departamento de Ventas define los puestos de trabajo Representante de Ventas y Gerente de Ventas, los cuales también se representan como clasificadores de recursos. Las poblaciones de estos clasificadores son incorporadas en la población del departamento de Ventas por medio de relaciones de inclusión. La línea de reporte desde el puesto Representante de Ventas al puesto Gerente de Ventas es representada por medio de la relación de recursos gerente.

El departamento de Fabricación cuenta con diseñadores encargados de modelar los muebles solicitados por los clientes y determinar las listas de materiales requeridos para su construcción. La empresa cuenta con diseñadores especializados en muebles de cocina y otros especializados en muebles de living. Esto se representó mediante una jerarquía de clasificadores de recursos vinculados por relaciones de inclusión, que indica que el departamento de Fabricación incluye diseñadores que pueden ser de dos grupos, Diseñador de Cocina o Diseñador de Living. En lo que respecta a los recursos humanos, se definió que la empresa cuenta con cuatro representantes de ventas, un gerente de ventas al que reportan, y tres diseñadores, uno de muebles de cocina y otros dos de muebles de living.

El PI-RSM de la Figura 50 también define requerimientos de Autorización Estática. El privilegio `autonomiaDeSeleccion` es otorgado a los diseñadores, a los representantes de ventas y al gerente de ventas, con el propósito de permitirles organizar su secuencia de trabajo, tal como se define por el patrón de recursos `Autonomía de Selección` (número 26). Esto implica que pueden decidir el orden en que seleccionan y ejecutan ítems de trabajo que les son presentados. El privilegio `verAsignados` relacionado con el patrón `Visibilidad`

Configurable de Ítems de Trabajo Asignados (número 41) y el privilegio reasignar relacionado con el patrón Reasignación con Estado (número 30) son otorgados además al Gerente de Ventas, para permitir al recurso Elena supervisar a los representantes de ventas a través del acceso a los ítems de trabajo que les fueron asignados y la posibilidad de modificar la asignación de los ítems de trabajo en caso de problemas en su ejecución o la ausencia de los recursos asignados.

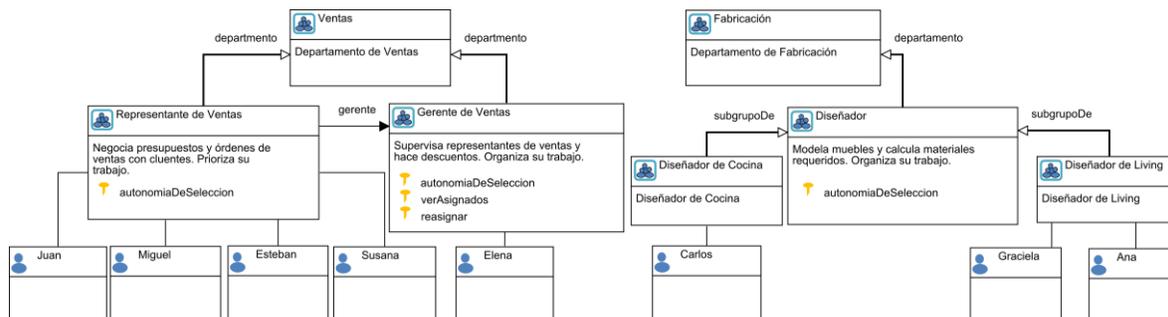


Figura 50. PI-RSM Proceso de Gestión de Presupuestos de Muebles.

La Figura 51 muestra el PI-EPM derivado del BPLM de la Figura 49. El mismo define los requerimientos de distribución de trabajo y autorización dinámica del proceso de Gestión de Presupuestos de Muebles a Medida.

El requerimiento para distribuir los ítems de trabajo de la tarea de usuario Escribir Especificación del Mueble entre los representantes de ventas consistió en dar libertad a los mismos de acordar y asumir la ejecución de los ítems de trabajo. La asignación de los ítems de trabajo de esta tarea a los representantes de ventas se realiza por medio de una consulta parametrizada de recursos, la cual especifica sólo una referencia de recursos (Ref) al clasificador de recurso Representante de Ventas, sin incluir en la consulta ningún parámetro. La libertad de los mismos para asumir la ejecución de los ítems de trabajo de la tarea fue expresada informalmente en la documentación del rol de recurso Candidato, el cual representa una política de distribución de trabajo de tipo *pull*.

En lo que respecta a la tarea Elaborar Lista de Materiales, los requerimientos de distribución de trabajo fueron que los ítems de trabajo de esta tarea se asignen a aquellos diseñadores especializados en el tipo de mueble solicitado. Con el propósito de reducir el tiempo transcurrido entre la asignación del ítem de trabajo y el comienzo de su ejecución por parte de un diseñador, el requerimiento para esta tarea es que cada ítem de trabajo de la misma debería asignarse al diseñador especializado en el tipo de mueble solicitado, y además, a aquel diseñador con menor cantidad de trabajo pendiente al momento de la creación del ítem de trabajo. Además, el diseñador seleccionado deberá poder rechazar una asignación del ítem de trabajo, en cuyo caso el ítem debe ser asignado al segundo diseñador con menor cantidad de trabajo pendiente. Por último, si el diseñador que tiene asignado la

tarea no puede completar el ítem de trabajo en el término de dos días, se debe ofrecer el mismo a todos los diseñadores, para que aquél que voluntariamente lo decida, pueda llevar adelante la ejecución de la tarea.

Para dar soporte a estos requerimientos en el PI-EPM, se definieron dos roles para esta tarea de usuario: Responsable y Candidato. La documentación de los mismos expresa informalmente sus semánticas. El rol Responsable especifica que el diseñador asignado es el encargado de la elaboración de la lista de materiales. Por lo tanto, éste representa una política de distribución de trabajo de tipo *push*. Este rol también permite a los recursos rechazar la responsabilidad por la ejecución del ítem de trabajo. El rol Responsable es asignado inicialmente a un diseñador cuya especialidad coincide con el tipo de mueble indicado en la especificación del mueble solicitado. Esto es definido por medio de una expresión de asignación de recursos (RAE) y una restricción de resolución. La expresión de asignación permite obtener el conjunto de recursos pertenecientes al grupo de diseñadores correspondientes con el tipo de mueble especificado. La restricción (constraint) *RecursoMasLibre*, definida a partir del patrón de recursos número 17, expresa la asignación del ítem de trabajo al recurso con menor cantidad de ítems de trabajo en su lista de tareas. En caso que el recurso asignado rechace la responsabilidad por la ejecución de un ítem de trabajo de esta tarea, el mismo es asignado al siguiente recurso en el ciclo, de acuerdo con la restricción *RecursoMasLibre*.

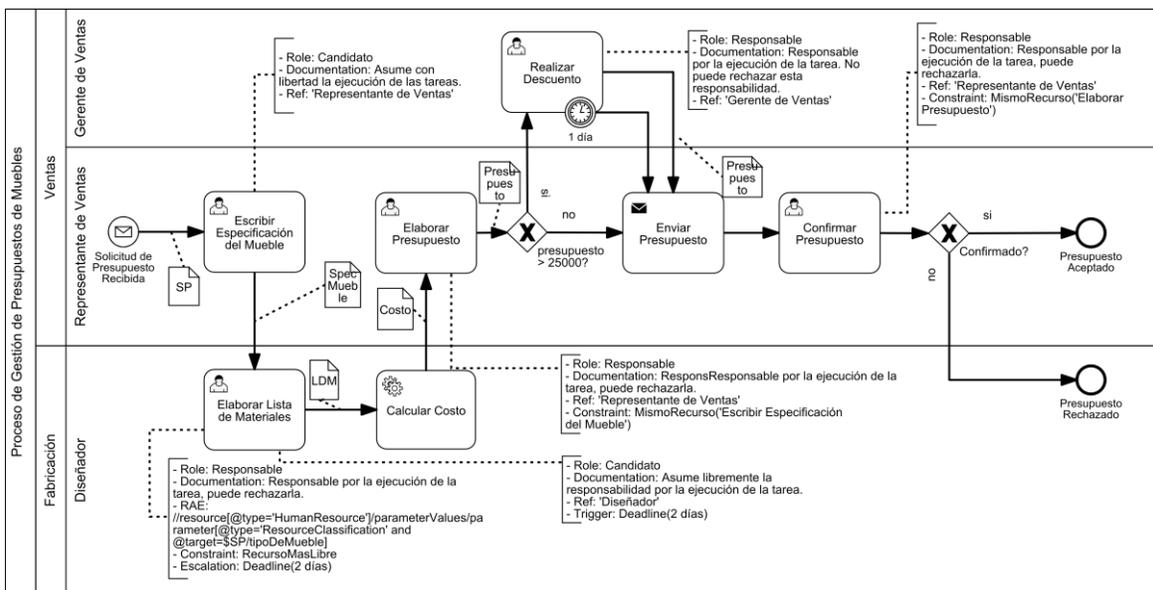


Figura 51. PI-EPM del Proceso de Gestión de Presupuestos de Muebles a Medida.

El rol Candidato de la tarea Elaborar Lista de Materiales especifica que los diseñadores pueden asumir la responsabilidad por la ejecución de sus ítems de trabajo libremente. Por lo

tanto, éste define una estrategia de distribución de trabajo de tipo *pull*. Este rol es asignado a partir de la ocurrencia del evento especificado como disparador de la asignación (Trigger), con el propósito de representar el cambio de la distribución de los ítems de trabajo de la tarea en tiempo de ejecución, desde un recurso asignado al rol Responsable a los recursos asignados al rol Candidato. Para esto también se define como atributo del rol Responsable el evento que causa el cambio del rol (Escalation). De esta manera, se expresa que si la tarea no se completa en dos días, éste evento causa la revocación del rol Responsable y dispara la asignación del rol Candidato para todos los diseñadores sin tener en cuenta su especialidad.

Con el propósito de permitir a los representantes de ventas construir relaciones más sólidas con los clientes, otro requerimiento de distribución de trabajo consistió en que para cada instancia del proceso, todas las tareas que requieran comunicarse con el cliente sean llevadas a cabo por el representante de ventas que ejecutó la primera tarea del proceso. Con este propósito, se definieron políticas de distribución de trabajo de tipo *push* para las tareas de usuario Elaborar Presupuesto y Confirmar Presupuesto. El rol Responsable definido para estas tareas es asignado al recurso clasificado como Representante de Ventas que completó la tarea de usuario Escribir Especificación del Mueble o Elaborar Presupuesto, respectivamente. Esto es efectuado por medio de la combinación de una referencia al clasificador de recurso Representante de Ventas con una restricción de resolución Mantener Familiaridad, la cual corresponde al patrón de recursos número 7.

Finalmente, el único miembro de la organización con la atribución de autorizar descuentos es el Gerente de Ventas. Para ello, se definió una política de distribución de trabajo *push* para la tarea de usuario Realizar Descuento mediante la asignación del rol Responsable, en cuya documentación se indica que no permite a los recursos rechazar la responsabilidad por la ejecución de los ítems de trabajo asignados. Esta asignación es expresada por medio de una consulta parametrizada de recursos referenciando al clasificador de recurso Gerente de Ventas. Notar que el evento temporizador definido para la tarea Realizar Descuento no forma parte de la definición de la perspectiva de recursos sino de la perspectiva de control de flujo. Este evento causa su cancelación de la instancia de la tarea si la misma no es completada en el curso de un día, lo cual tiene como consecuencia que el ítem de trabajo asociado con la instancia de la tarea sea destruido.

Selección de la Plataforma de Implementación

Las propiedades de evaluación presentadas en la Sección 4.4 fueron empleadas en este caso de estudio para comparar el soporte provisto por Bonita (RPIM de las Figuras 18 y 19) y el provisto por YAWL (RPIM de las Figuras 40 y 41) a los requerimientos de la

perspectiva de recursos definidos en la etapa anterior, con el propósito de proveer un criterio para la selección del WfMS más adecuado para la implementación del proceso.

Las Tablas 10 y 11 resumen el resultado de la evaluación de YAWL en términos del PI-RSM y el PI-EPM descritos en la sección anterior. Este WfMS provee dos alternativas para implementar los clasificadores de recursos Ventas y Fabricación: OrgGroup y Role. El clasificador de recurso Position de YAWL no resultó adecuado en este caso dado que no soporta la implementación de clasificadores de recursos que incluyan a otros clasificadores.

Ninguna de las implementaciones de clasificador de recurso provistas por YAWL resultó adecuada para implementar los clasificadores Representante de Ventas, Gerente de Ventas y Diseñador. La propiedad de evaluación *hasPrivileges* es requerida en estos clasificadores del PI-RSM. Sin embargo esta propiedad no es soportada por YAWL dado que no permite definir privilegios a nivel de clasificador de recurso.

Los clasificadores de recursos Diseñador de Cocina y Diseñador de Living pueden ser soportados por las entidades Position y Role provistas por YAWL. La entidad OrgGroup no resultó adecuada en este caso por no permitir la asociación de recursos humanos como población mediante relaciones de clasificación.

Finalmente, las entidades Participant y Administrator provistas por YAWL para representar recursos humanos resultaron adecuadas para todos los recursos definidos en el PI-RSM. Como resultado de la evaluación, la tasa de soporte de YAWL a los aspectos Estructura de Recursos y Autorización Estática definidos en el PI-RSM de la Figura 50 fue de 0.8.

Tabla 10. Resumen de Implementaciones Candidatas de Estructura de Recursos provistas por YAWL.

	isResource	isClassifier	isReferenceable	isBindable	subsumes	isSubsumed	isSource	isTarget	hasPopulation	isMember	hasPrivileges	hasParameters	hasBindableParams
Ventas	0	1	0	0	1	0	0	0	0	0	0	0	0
<i>OrgGroup</i>	0	1	0	1	1	1	0	0	0	0	0	0	0
<i>Role</i>	0	1	1	1	1	1	0	0	1	0	0	0	0
Fabricación	0	1	0	0	1	0	0	0	0	0	0	0	0
<i>OrgGroup</i>	0	1	0	1	1	1	0	0	0	0	0	0	0
<i>Role</i>	0	1	1	1	1	1	0	0	1	0	0	0	0
Representante de Ventas	0	1	1	0	0	1	1	0	1	0	1	0	0
Gerente de Ventas	0	1	1	0	0	1	0	1	1	0	1	0	0
Diseñador	0	1	1	0	1	1	0	0	0	0	1	0	0

Diseñador de Cocina	0	1	0	1	0	1	0	0	1	0	0	0	0
<i>Position</i>	0	1	0	1	0	1	1	1	1	0	0	0	0
<i>Role</i>	0	1	0	1	1	1	0	0	1	0	0	0	0
Diseñador de Living	0	1	0	0	0	1	0	0	1	0	0	0	0
<i>Position</i>	0	1	0	1	0	1	1	1	1	0	0	0	0
<i>Role</i>	0	1	0	1	1	1	0	0	1	0	0	0	0
Juan/Miguel/Esteban/Elena	1	0	0	0	0	0	0	0	0	1	0	0	0
<i>User</i>	1	0	1	1	0	0	0	0	0	1	1	1	1
<i>Administrator</i>	1	0	1	1	0	0	0	0	0	1	1	1	1
Carlos/Graciela/Ana	1	0	0	0	0	0	0	0	0	1	0	1	1
<i>User</i>	1	0	1	1	0	0	1	1	0	1	1	1	1
<i>Administrator</i>	1	0	1	1	0	0	1	1	0	1	1	1	1
Otros ResourceImpl													
<i>Position</i>	0	1	0	1	0	1	1	1	1	0	0	0	0
<i>OrgGroup</i>	0	1	1	1	1	1	0	0	0	0	0	0	0
<i>Role</i>	0	1	0	1	1	1	0	0	1	0	0	0	0

En lo que respecta al soporte provisto por YAWL a los aspectos Distribución de Trabajo y Autorización Dinámica definidos en el PI-EPM de la Figura 51, la implementación de rol de recurso PotentialOwner definida en el RPIM de YAWL (Figura 41) soporta la implementación del rol de recurso Candidato definido en la tarea Escribir Especificación del Mueble del proceso. Sin embargo, esta implementación de rol no resultó adecuada para implementar el rol de recurso con el mismo nombre definido para la tarea Elaborar Lista de Materiales, el cual no es asignado a los recursos hasta que ocurra el evento de ítem de trabajo que tiene asociado como trigger. Este requerimiento, el cual se refleja en el valor verdadero de la propiedad de evaluación *hasTriggers* para el rol de recurso Candidato de Elaborar Lista de Materiales, no es soportado por la entidad de implementación PotentialOwner, dado que presenta un valor falso para esta propiedad de evaluación.

Las implementaciones de roles de recursos Owner y Executor definidas en el RPIM de YAWL soportan la implementación del rol de recurso Responsable definido para las tareas Elaborar Presupuesto, Realizar Descuento y Confirmar Presupuesto. Sin embargo, estas entidades no resultaron adecuadas para implementar el rol Responsable definido para la tarea Elaborar Lista de Materiales, el cual es quitado a los recursos ante la ocurrencia del evento de ítem de trabajo referido como escalation. Este requerimiento, el cual se refleja en el valor verdadero de la propiedad de evaluación *hasEscalations* para el rol de recurso Responsable de Elaborar Lista de Materiales, no es soportado por las entidades de implementación Owner y Executor que presentan un valor falso para la propiedad de evaluación *hasEscalations* requerida en este caso. Como resultado de la evaluación, se obtuvo una tasa de soporte de YAWL a los aspectos Distribución de Trabajo y Autorización Dinámica del proceso de 0,66.

Tabla 11. Resumen de Implementaciones Candidatas de Distribución de Trabajo Provistas por YAWL.

	isAssignable	multiplePerformers	singlePerformer	hasHardConstraints	hasSoftConstraints	hasTriggers	hasEscalations	revokablePrivileges
Candidato (Escribir Especificación del Mueble)	1	1	0	0	0	0	0	0
<i>PotentialOwner</i>	1	1	0	1	0	0	0	1
Responsable (Elaborar Lista de Materiales)	1	0	1	1	0	0	1	0
Candidato (Elaborar Lista de Materiales)	1	1	0	0	0	1	0	0
Responsable (Elaborar Presupuesto)	1	0	1	1	0	0	0	0
<i>Owner</i>	1	0	1	1	0	0	0	1
<i>Executor</i>	1	0	1	1	0	0	0	0
Responsable (Realizar Descuento)	1	0	1	1	0	0	0	0
<i>Owner</i>	1	0	1	1	0	0	0	1
<i>Executor</i>	1	0	1	1	0	0	0	0
Responsable (Confirmar Presupuesto)	1	0	1	1	0	0	0	0
<i>Owner</i>	1	0	1	1	0	0	0	1
<i>Executor</i>	1	0	1	1	0	0	0	0
Otros ResourceRoleImpl	1	0	1	1	0	0	1	0
<i>PotentialOwner</i>	1	1	0	1	0	0	0	1
<i>Owner</i>	1	0	1	1	0	0	0	1
<i>Executor</i>	1	0	1	1	0	0	0	0

El segundo WfMS considerado como plataforma candidata para la implementación de los requerimientos definidos en el PI-RSM y el PI-EPM presentados en la sección anterior fue Bonita. Los resultados de su evaluación en término de dichos modelos se resumen en las Tablas 13 y 14.

La implementación de los clasificadores de recursos que representan los departamentos de Ventas y Fabricación es soportada por la entidad de implementación Group provista por Bonita. Esta plataforma no provee entidades que posibiliten implementar a los clasificadores de recursos Representante de Ventas, Gerente de Ventas y Diseñador, los cuales presentan la propiedad requerida *hasPrivileges*. Por otra parte, dado que Bonita carece de entidades de implementación que provean soporte a la propiedad *isBindable*, los clasificadores de recursos Diseñador de Cocina y Diseñador de Living tampoco son soportados por la misma. Finalmente, es posible implementar los recursos individuales definidos en el PI-RSM por medio de las entidades de implementación User y Administrator de Bonita. Como resultado de la evaluación, se obtuvo una tasa de soporte de 0,66 de la plataforma Bonita a los aspectos Estructura de Recursos y Autorización Estática definidos en el PI-RSM.

Tabla 12. Resumen de Implementaciones Candidatas de Estructura de Recursos provistas por Bonita.

	isResource	isClassifier	isReferenceable	isBindable	subsumes	isSubsumed	isSource	isTarget	hasPopulation	isMember	hasPrivileges	hasParameters	hasBindableParams
Ventas	0	1	0	0	1	0	0	0	0	0	0	0	0
<i>Group</i>	0	1	1	0	1	1	0	0	0	0	0	0	0
Fabricación	0	1	0	0	1	0	0	0	0	0	0	0	0
<i>Group</i>	0	1	1	0	1	1	0	0	0	0	0	0	0
Representante de Ventas	0	1	1	0	0	1	1	0	1	0	1	0	0
Gerente de Ventas	0	1	1	0	0	1	0	1	1	0	1	0	0
Diseñador	0	1	1	0	1	1	0	0	0	0	1	0	0
Diseñador de Cocina	0	1	0	1	0	1	0	0	1	0	0	0	0
Diseñador de Living	0	1	0	1	0	1	0	0	1	0	0	0	0
Juan/Miguel/Esteban/Elena	0	1	1	0	0	1	1	0	1	0	0	1	0
<i>User</i>	0	1	1	0	0	1	0	0	1	0	1	0	0
<i>Administrator</i>	0	1	1	0	1	1	0	0	0	0	1	0	0
Carlos/Graciela/Ana	1	0	0	0	0	0	0	0	0	0	0	0	0
<i>User</i>	1	0	1	1	0	0	1	1	0	1	1	1	1
<i>Administrator</i>	1	0	1	1	0	0	1	1	0	1	1	1	1
Otros ResourceImpl													
<i>Membership</i>	0	1	1	0	0	1	0	0	1	0	0	0	0
<i>Group</i>	0	1	1	0	1	1	0	0	0	0	0	0	0
<i>Role</i>	0	1	1	0	1	0	0	0	0	0	0	0	0

En lo que respecta a los requerimientos de Distribución de Trabajo y Autorización Dinámica, el rol de recurso Candiato de la tarea Escribir Especificación del Mueble (PI-EPM de la Figura 51) puede ser implementado por la entidad Actor de Bonita, ya que las propiedades de evaluación que son requeridas por estos roles son soportadas por esta entidad. Sin embargo, la entidad Actor no puede ser usada para implementa el rol de recurso Candidato definido para la tarea Elaborar Lista de Materiales, el cual tiene un evento de ítem de trabajo asociado como *trigger*. Esto se debe a que la entidad Actor presenta un valor falso para la propiedad de evaluación *hasTriggers*, la cual tiene un valor verdadero para el rol Candidato de Elaborar Lista de Materiales.

Por otra parte, la entidad AssignedActor de Bonita provee soporte a los roles de recursos Responsable definidos para las tareas Elaborar Presupuesto, Realizar Descuento y Confirmar Presupuesto, ya que las propiedades de evaluación que son requeridas por estos roles son soportadas por esta entidad. Sin embargo, esta entidad no soporta la

implementación del rol de recurso Responsable de la tarea Elaborar Lista de Materiales, dado que dicho rol presenta un valor verdadero para la propiedad de evaluación *hasEscalations*, la cual no es soportada por la entidad *AssignedActor*. Como resultado de la evaluación, se obtuvo una tasa de soporte de Bonita a los requerimientos definidos en el PI-EPM de 0,66.

Tabla 13. Resumen de Implementaciones Candidatas de Distribución de Trabajo Provistas por Bonita.

	isAssignable	multiplePerformers	singlePerformer	hasHardConstraints	hasSoftConstraints	hasTriggers	hasEscalations	revokablePrivileges
Candidato (Escribir Especificación del Mueble)	1	1	0	0	0	0	0	0
<i>Actor</i>	1	1	0	1	0	0	0	0
Responsable (Elaborar Lista de Materiales)	1	0	1	1	0	0	1	0
Candidato (Elaborar Lista de Materiales)	1	1	0	0	0	1	0	0
Responsable (Elaborar Presupuesto)	1	0	1	1	0	0	0	0
<i>AssignedActor</i>	1	0	1	1	0	0	0	0
Responsable (Realizar Descuento)	1	0	1	1	0	0	0	0
<i>AssignedActor</i>	1	0	1	1	0	0	0	0
Responsable (Confirmar Presupuesto)	1	0	1	1	0	0	0	0
<i>AssignedActor</i>	1	0	1	1	0	0	0	0
Otros ResourceRoleImpl								
<i>Actor</i>	1	1	0	1	0	0	0	0
<i>AssignedActor</i>	1	0	1	1	0	0	0	0

Finalmente, en esta etapa se compararon los resultados obtenidos en las evaluaciones efectuadas a los WfMSs YAWL y Bonita con el propósito de seleccionar el más adecuado de ellos como plataforma de implementación para el SIOP a ser desarrollado. Dado que la tasa de soporte a los aspectos de Estructura de Recursos y Autorización Estática provista por YAWL (0,8) fue mayor a la provista por Bonita (0,66) y que ambas plataformas mostraron una tasa de soporte de 0,66 a los aspectos de Distribución de Trabajo y Autorización Dinámica, la plataforma seleccionada para la implementación de los requerimientos definidos para este proceso fue YAWL.

Definición de la Implementación de la Perspectiva de Recursos

La implementación de la perspectiva de recursos se definió en modelos PS-RSM y PS-EPM (Figuras 52 y 53). Estos se derivaron respectivamente de los modelos PI-RSM y PI-EPM (Figuras 50 y 51), y a partir de las entidades candidatas indicadas en las Tablas 10 y 11, correspondientes al RPIM de YAWL (Figuras 40 y 41).

El PS-RSM resultante, el cual define la implementación de los aspectos Estructura de Recurso y Autorización Estática se muestra en la Figura 52. En el mismo, los clasificadores de recursos Ventas y Fabricación fueron implementados por medio de la entidad OrgGroup de YAWL. Esta entidad fue elegida por sobre Role dado que la misma tiene como propósito describir unidades organizacionales.

Los clasificadores de recursos Representante de Ventas, Gerente de Ventas y Diseñador no son soportados por YAWL dado que esta plataforma no permite definir privilegios de recursos a nivel de clasificador de recurso. Con el propósito de implementar los requerimientos sin violar esta restricción, los privilegios de recursos definidos fueron movidos a los recursos humanos agrupados por estos clasificadores. Una vez hecho esto, la propiedad de evaluación *hasPrivileges* dejó de ser requerida para estos clasificadores, lo cual permitió implementarlos con la entidad de implementación Role, la cual tiene valor verdadero para la propiedad de evaluación *isReferenceable*. Los clasificadores de recursos Diseñador de Cocina y Diseñador de Living también fueron implementados por medio de la entidad Role e incluidos en el clasificador de recurso Diseñador por medio de una relación de inclusión *belongsTo* de YAWL.

Otra restricción impuesta por YAWL consiste en la falta de una entidad que soporte la inclusión de los recursos humanos agrupados por un clasificador de recurso implementado como Role, dentro de la población de un clasificador de recurso implementado como OrgGroup. Con el propósito de implementar la inclusión de los recursos humanos definidos en el PS-RSM en los departamentos de Ventas y Fabricación sin violar esta restricción, fue necesario agregar clasificadores de recursos implementados como Position, definiendo las respectivas relaciones de inclusión (implementadas como *belongsTo*) a los clasificadores Ventas y Fabricación.

Respecto a los requerimientos de Autorización Estática definidos en el PI-RSM, se debió trasladar los privilegios de recursos definidos en los clasificadores de recursos de dicho modelo a los recursos humanos a fin de poder emplear los privilegios de recursos del modo soportado por YAWL. Esto se debe a que YAWL sólo permite especificar privilegios de recursos para recursos humanos en forma individual. El privilegio de recursos *autonomiaDeSeleccion*, fue implementado por medio de la entidad *choose* de YAWL y otorgado a los recursos humanos clasificados por Representante de Ventas y Diseñador, los cuales fueron implementados mediante la entidad *Participant*. Los privilegios de recursos *autonomiaDeSeleccion*, *verAsignados* y *reasignar* fueron implementados respectivamente por medio de las entidades *choose*, *viewOffered* y *allocate* de YAWL y asignados al recurso clasificado como Gerente de Ventas (Elena), el cual fue implementado como *Administrator* con el propósito de permitirle supervisar a los representantes de ventas, tal como lo indica el clasificador de recurso Gerente de Ventas

definido en el PI-RSM. Además, fue necesario otorgar a Elena los privilegios verNoAsignados, ofrecer y asignarIniciado, implementados mediante las entidades viewUnoffered, offer y start, los cuales son obligatorios en YAWL para los recursos implementados como Participant.

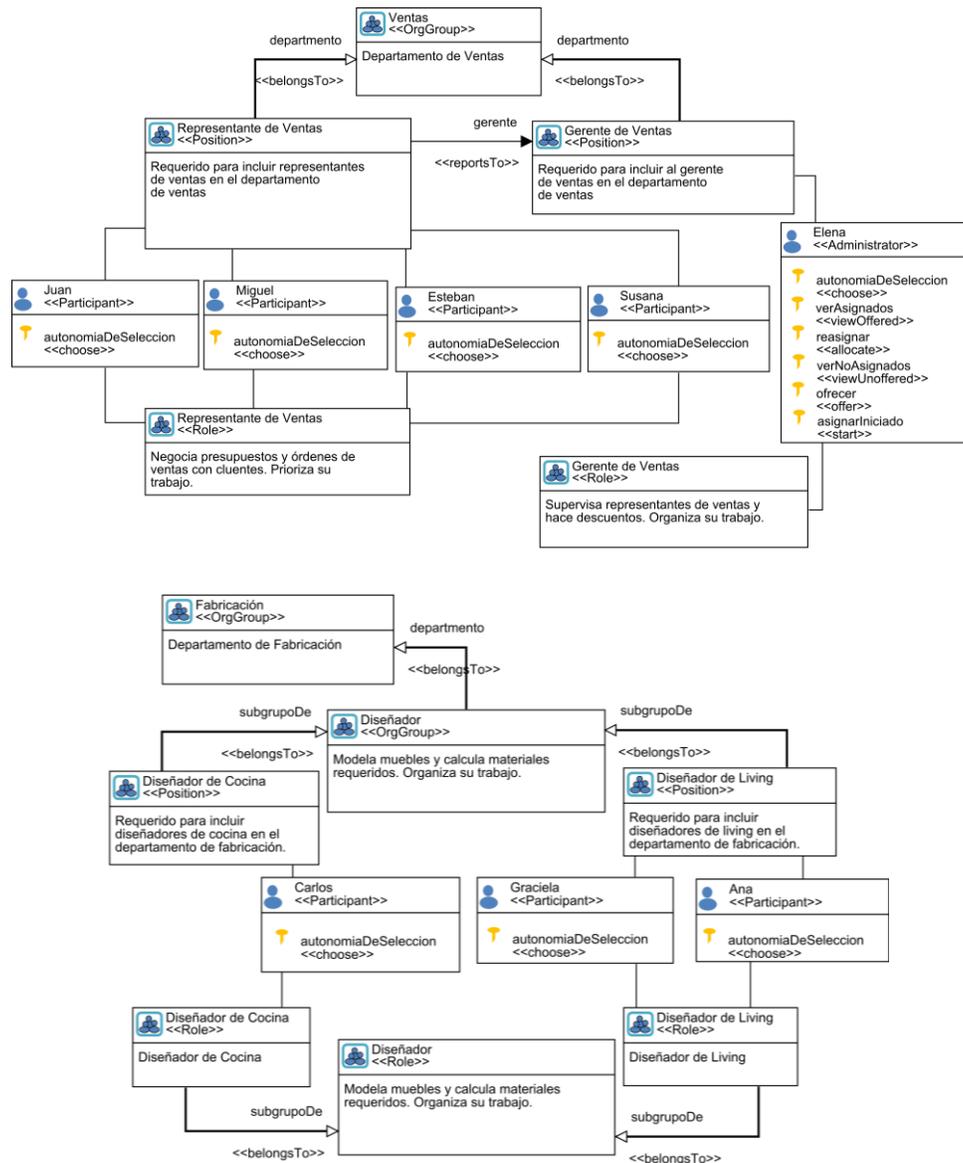


Figura 52. PS-RSM del Proceso de Gestión de Presupuestos de Muebles a Medida.

El PS-EPM resultante que define la implementación de los aspectos Distribución de Trabajo y Autorización Dinámica se muestra en la Figura 53. El rol Candidato de la tarea Escribir Especificación del Mueble fue implementado por medio de la entidad PotentialOwner de YAWL. Esto se debe a que PotentialOwner es la única entidad

candidata de implementación para este rol y además expresa una política de distribución de trabajo *pull*. La consulta parametrizada de recursos definida en el PI-EPM para este rol de recurso se mantiene sin cambios dado que referencia al clasificador de recurso Representante de Ventas, el cual es implementado en forma directa por medio de la entidad Role.

Los roles Responsable y Candidato de la tarea Elaborar Lista de Materiales no son soportados por YAWL. Esto se debe a que este WfMS no provee una operación de ítem de trabajo que permita el escalado del rol Owner y la asignación del rol PotentialOwner ante la ocurrencia de un evento de ítem de trabajo. Una acción correctiva posible en este caso sería introducir modificaciones en la definición de la lógica de control de flujo con el propósito de subsanar esta limitación. Dicha modificación consistiría en la adición de un evento de temporización a la tarea Elaborar Lista de Materiales con el propósito de dirigir el control a otra tarea de usuario, en la que se defina la asignación para este segundo rol en caso que la misma no se complete en dos días. Sin embargo, esto no es equivalente a la especificación de un evento de ítem de trabajo, dado que el ítem de trabajo de la tarea Elaborar Lista de Materiales sería descartado y su estado se perdería cuando la fecha límite se alcance. Además, cambios en la lógica del proceso requerirían que el BPLM, el PI-EPM y el PS-EPM sean modificados en consecuencia. Como solución de compromiso, se optó por implementar únicamente el rol Candidato que define una política de distribución de trabajo de tipo *pull*, la cual es menos restrictiva. Se optó por esta solución con el propósito de evitar bloqueos en el proceso ante la ausencia del recurso asignado inicialmente al rol Responsable. De este modo, el rol de recurso Candidato fue implementado mediante la entidad PotentialOwner omitiendo la la referencia *trigger* definida para el mismo en el PI-EPM.

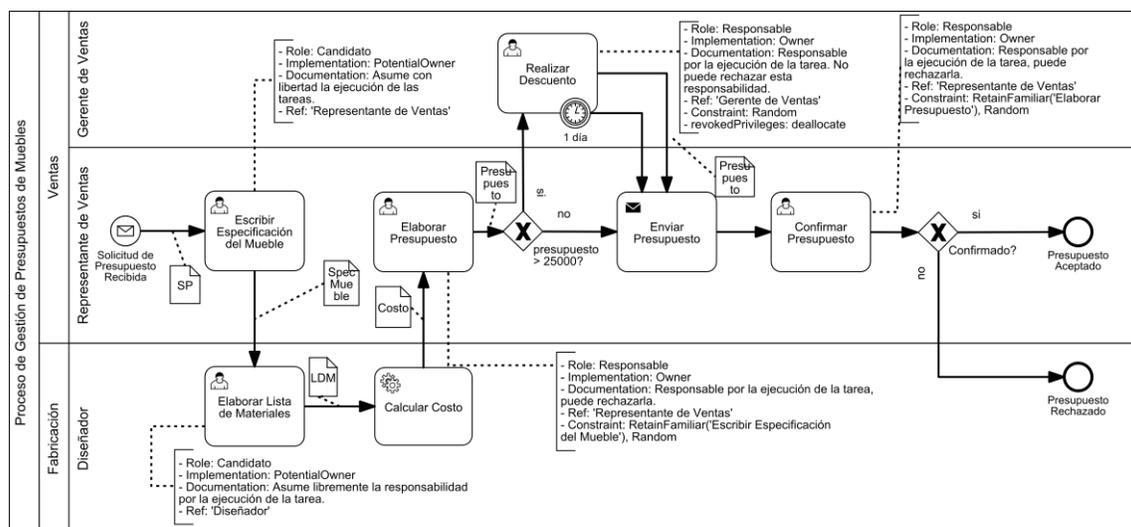


Figura 53. PS-EPM del Proceso de Gestión de Presupuestos de Muebles a Medida.

Los roles Responsable de las tareas Elaborar Presupuesto y Confirmar Presupuesto fueron implementados mediante el rol Owner definido en el RPIM de YAWL, dado que esta entidad representa una estrategia de distribución de trabajo *push*. Las referencias de recursos definidas para asignar estos roles no requirieron cambios respecto de las definidas en el PI-EPM de origen. Las restricciones de resolución MismoRecurso especificadas para los estos roles de recursos fueron implementados mediante la entidad RetainFamiliar de YAWL. Fue necesario además el agregado de restricciones de resolución implementadas mediante la entidad Random a fin de satisfacer la restricción especificada en el RPIM de YAWL respecto de la necesidad de definir una restricción de la categoría AllocationStrategy para los roles de recursos implementados como Owner o Executor.

El rol Responsable de la tarea Realizar Descuento también fue implementado por medio de la entidad Owner, para implementar la estrategia de distribución de trabajo *push*. Tal como se explicó en el párrafo anterior, la implementación de este rol mediante esta entidad de implementación requirió agregar la restricción de resolución de YAWL Random. Con el propósito de impedir el rechazo de los ítems de trabajo de esta tarea, el privilegio de tarea deallocate de YAWL fue definido como un privilegio revocado en esta tarea, por medio de su especificación a través del atributo *revokedPrivileges*.

Verificación y Validación de la Implementación de Perspectiva de Recursos

La implementación de los requerimientos de la perspectiva de recursos en la etapa anterior requirió la introducción de cambios en los modelos específicos de plataforma respecto de los modelos independientes de la plataforma, dado que YAWL no provee soporte directo a todos los requerimientos definidos en estos modelos. Las reglas de verificación (definidas en capítulo 4, sección 4.6, e implementadas por la herramienta, capítulo 5, sección 5.1) se aplicaron con el propósito de constatar que el PS-RSM y el PS-EPM resultantes se adecúan a las restricciones impuestas por el RPIM de YAWL. El resultado obtenido de la verificación indicó que todas las reglas fueron satisfechas. Lo cual indica que el PS-RSM y el PS-EPM definidos son consistentes con las entidades provistas por la plataforma YAWL.

El procedimiento de validación también fue aplicado. El mismo detectó inconsistencias en el PS-RSM ya que los privilegios de recursos implementados mediante las entidades viewUnoffered, offer y start en el PS-RSM no fueron definidos en el PI-RSM. En PS-RSM estos privilegios fueron agregados debido a que son obligatorios para los recursos implementados mediante la entidad Administrator, según se define en el RPIM de YAWL. El proceso de validación también detectó inconsistencias en el PS-EPM, debido a que no se definió una implementación para el rol Responsable de la tarea Elaborar Lista de

Materiales. Con base en estos resultados de validación, se procedió a evaluar la conveniencia de proceder a generar la solución tecnológica en base a la implementación evaluada. Teniendo en cuenta que estos privilegios de recursos adicionales requeridos por la plataforma no constituyen una amenaza desde el punto de vista de la seguridad, y que la política de distribución de trabajo definida para la tarea Elaborar Lista de Materiales es lo suficientemente flexible, se decidió continuar con el desarrollo del SIOP con base en esta plataforma.

Generación de Especificaciones Ejecutables

El PS-RSM y el PS-EPM de las Figuras 52 Y 53 fueron transformados respectivamente en un esquema de recursos de YAWL y la correspondiente especificación de workflow por medio de dos transformaciones de modelo a código desarrolladas para el RPIM de YAWL. Estas transformaciones toman como entrada los elementos del PS-RSM y el PS-EPM, junto con los del RPIM de YAWL, y generan como salida los documentos a ser importados en el servicio de recursos y en el diseñador de procesos de YAWL.

La Figura 54 muestra una captura de pantalla del esquema de recursos generado luego de su importación en el servicio de recursos de YAWL. La Figura 55 muestra la especificación de workflow YAWL resultante para el proceso de gestión de presupuestos de muebles, en donde se observa una ventana con información acerca de la estrategia de distribución de trabajo implementada para la tarea Escribir Especificación del Mueble.

Participant: Juan, Juan First Name: Juan Last Name: Juan User ID: Juan <input type="checkbox"/> Administrator Description: Notes:	Privileges <input type="checkbox"/> Choose Which Work Item to Start <input type="checkbox"/> Start Work Items Concurrently <input checked="" type="checkbox"/> Reorder Work Items <input type="checkbox"/> View All Work Items of Team <input type="checkbox"/> View All Work items of Org Group <input type="checkbox"/> Chain Work Item Execution <input type="checkbox"/> Manage Cases						
<table border="1"> <tr> <th>Roles</th> <th>Positions</th> <th>Capabilities</th> </tr> <tr> <td> Owns: Representante de Vent </td> <td> <input type="button" value=">"/> <input type="button" value="<"/> </td> <td> Available: Diseñador Diseñador de Cocina Diseñador de Living Gerente de Ventas Representante de Vent </td> </tr> </table>	Roles	Positions	Capabilities	Owns: Representante de Vent	<input type="button" value=">"/> <input type="button" value="<"/>	Available: Diseñador Diseñador de Cocina Diseñador de Living Gerente de Ventas Representante de Vent	Password New: _____ Confirm: _____ <input type="button" value="Save"/> <input type="button" value="New"/> <input type="button" value="Reset"/> <input type="button" value="Remove"/>
Roles	Positions	Capabilities					
Owns: Representante de Vent	<input type="button" value=">"/> <input type="button" value="<"/>	Available: Diseñador Diseñador de Cocina Diseñador de Living Gerente de Ventas Representante de Vent					

Figura 54. Esquema de Recursos de YAWL del Proceso de Gestión de Presupuestos.

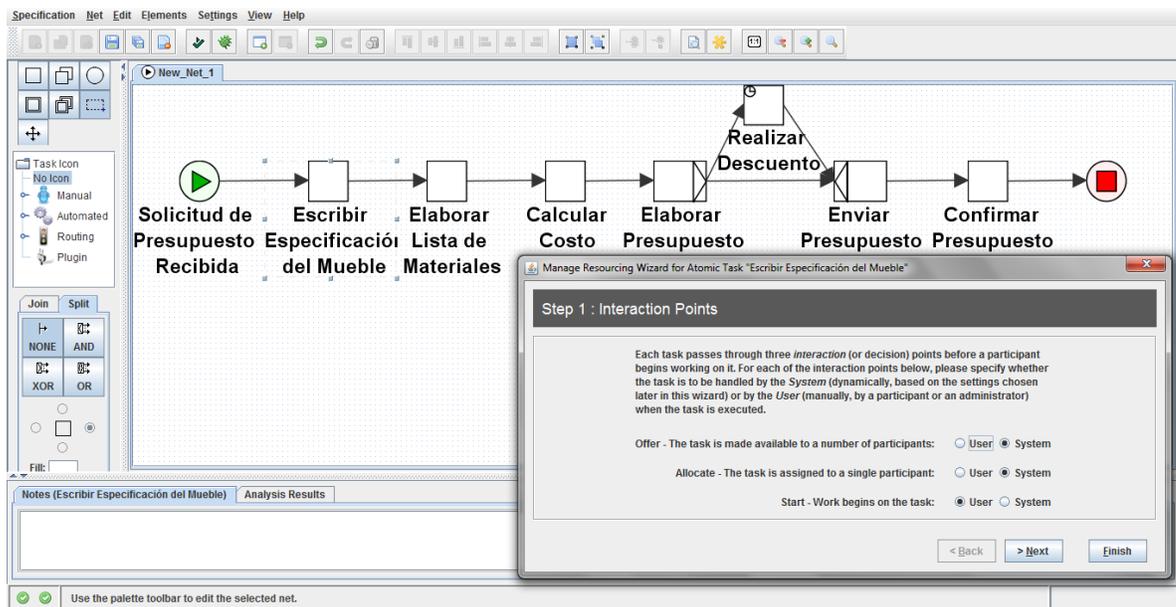


Figura 55. Especificación Ejecutable de Workflow del Proceso de Gestión de Presupuestos.

Conclusiones del Caso de Estudio

El marco de trabajo y el método propuestos permitieron la definición de la perspectiva de recursos en el desarrollo del SIOP presentado en este caso de estudio, siendo posible destacar que:

- La definición de los aspectos Estructura de Recursos y Autorización estática en un PI-RSM permitió representar y visualizar la estructura de las áreas de la empresa fabricante de muebles a medida, la cual no se encontraba definida formalmente.
- La definición de los aspectos Distribución de Trabajo y Autorización Dinámica en un PI-EPM permitió a las personas involucradas analizar y definir sus requerimientos en etapas tempranas del desarrollo del SIOP. Además, esto permitió la definición de políticas de distribución de trabajo avanzadas, tales como estrategias de distribución de trabajo de tipo *push* con estrategias de asignación basadas en la carga de trabajo de los recursos y definición de eventos para la reasignación de roles a recursos.
- La aplicación de la técnica de evaluación de plataformas de implementación permitió el cálculo de tasas de soporte a los requerimientos de perspectiva de recursos provistos por los WfMSs Bonita y YAWL, y proveyó una base cuantitativa para seleccionar el WfMS más adecuado de ellos como plataforma de implementación del SIOP.
- Los requerimientos de perspectiva de recursos que pueden ser implementados dependen en definitiva de las entidades de implementación provistas soporte provisto por el WfMS seleccionado. La aplicación del método permitió detectar estas limitaciones en etapas tempranas del desarrollo del SIOP, evaluar acciones correctivas posibles y tomar una decisión respecto de la implementación de los requerimientos afectados por dichas limitaciones.
- Las reglas de verificación y validación descritas permitieron detectar errores de implementación e inconsistencias entre los requerimientos definidos en modelos conceptuales de esta perspectiva y su posterior implementación. Finalmente, las transformaciones modelo a código desarrolladas para el RPIM de YAWL e incorporadas en la herramienta permitieron generar los documentos requeridos por este WfMS para implementar el SIOP desarrollado.

6.3 Soporte a los Patrones de Recursos

El marco de trabajo con sus metamodelos y extensiones también fue evaluado en términos de su expresividad para representar los requerimientos de la perspectiva de recursos capturados en los patrones de recursos de workflow (WRPs por sus siglas en inglés) (Russell, et al., 2005). Estos patrones han sido ampliamente utilizados en la

literatura para evaluar lenguajes y WfMSs, y por lo tanto, constituyen un marco de evaluación adecuado para nuevas propuestas de modelado de la perspectiva de recursos de procesos de negocio.

Los patrones de recursos están organizados en ocho categorías definidas con base en los estados de un modelo de referencia del ciclo de vida de los ítems de trabajo. Los Patrones de Creación (WRP-1 a WRP-11), los Patrones Push (WRP-12 a WRP-20), Escalado (WRP-28) y dos de los Patrones de Inicio Automático (WRP-36 y WRP-37) describen requerimientos recurrentes que pueden aparecer durante la creación, publicación y asignación de ítems de trabajo a los recursos. Por lo tanto, estos patrones caen dentro del alcance del aspecto Distribución de Trabajo. Los Patrones Pull (WRP-21 a WRP-26), los Patrones de Desvíos iniciados por el usuario (WRP-27 y WRP-29 a WRP-35), los Patrones de Visibilidad (WRP-40 y WRP-41) y los Patrones de Recursos Adicionales (WRP-42 y WRP-43) describen diferentes operaciones que pueden ser ejecutadas por los recursos para organizar los ítems de trabajo y modificar su estado. Por lo tanto, los mismos caen dentro del alcance del aspecto Autorización. La Tabla 14 resume los patrones que pueden ser representados para definir los aspectos Distribución de Trabajo, Autorización Estática y Autorización Dinámica, haciendo uso de los elementos de modelado definidos en los metamodelos y extensiones a BPMN del marco de trabajo propuesto.

Los patrones WRP-01, WRP-02, WRP-08 y WRP-10 son soportados en tareas de usuario mediante la definición de consultas parametrizadas de recursos, las cuales consisten en una referencia de recursos (`resourceRef`) desde un elemento `ResourceRole` a un elemento `Resource`, y cero o más elementos `ResourceParameterBinding` que definen condiciones sobre los parámetros del recurso. Cada uno de estos patrones puede representarse de la siguiente manera:

- WRP-01: definición de una referencia a un elemento `Resource` de tipo `HumanResource`.
- WRP-02: definición de una referencia a un elemento `Resource` de tipo `ResourceClassifier`, el cual representa un agrupamiento funcional de recursos.
- WRP-08: definición de una referencia a un elemento `Resource` de tipo `ResourceClassifier`, junto con un conjunto de elementos `ResourceParameterBinding` que definen las condiciones sobre los parámetros que describen capacidades de los recursos.
- WRP-10: definición de una referencia a un elemento `Resource` de tipo `ResourceClassifier` que representa una unidad o posición de la organización.

Los patrones WRP-03 y WRP-09 son soportados en tareas de usuario mediante la definición de expresiones de asignación de recursos (elemento `ResourceAssignmentExpression`), las cuales permiten determinar la identidad de los

recursos a ser asignados a un determinado rol de recurso con base en el valor de variables en el alcance de la tarea, las cuales almacenan identificadores de recursos especificados por otros recursos humanos en una tarea anterior (WRP-03), o generados por el WfMS a través de algún mecanismo para indicar los ítems de trabajo ejecutados por el recurso con anterioridad, para una o varias instancias de proceso (WRP-09).

Los Patrones WRP-05, WRP-07, WRP-15, WRP-16 y WRP-17 son soportados mediante la definición de restricciones de resolución (elemento `ResolutionConstraint`) que se aplican luego de resolver consultas parametrizadas o expresiones de asignación de recursos. Estos patrones se representan mediante la definición de restricciones que indican: la exclusión del recurso que desempeñó un determinado rol en una tarea previa (WRP-05), la selección del recurso que desempeñó un determinado rol en una tarea previa (WRP-07), la selección al azar de un recurso (WRP-15), la selección en forma cíclica de los recursos (WRP-16), o la selección de recursos con menos ítems de trabajo pendientes de ejecutar (WRP-17).

Los patrones WRP-19, WRP-20 y WRP-28 son soportados mediante la definición de eventos de ítem de trabajo. El WRP-19 es representado mediante la ausencia de eventos de ítem de trabajo. El WRP-20 se representa mediante la definición de un evento de ítem de trabajo como disparador (*trigger*) de la asignación de un rol de recurso. El WRP-28 se representa mediante la definición de un evento de ítem de trabajo de un rol de recurso, tal como el evento *escalation*.

Los patrones WRP-12, WRP-13, WRP-14, WRP-36 y WRP-37, los cuales distinguen entre estrategias de distribución tipo *push* o tipo *pull* son soportados mediante la asignación de roles con distintos significados y privilegios de tarea. Estos patrones pueden expresarse de la siguiente manera:

- WRP-12 y WRP-13: especificando la asignación de un rol que representa una estrategia de distribución de tipo *pull*, tal como el rol `PotentialOwner` definido en el RPIM de YAWL, a un único recurso (WRP-12) o a múltiples recursos (WRP-13).
- WRP-14: especificando la asignación de un rol que representa una estrategia de distribución de trabajo de tipo *push*, tal como el rol `Owner` definido en el RPIM de YAWL.
- WRP-36 y WRP-37: especificando la asignación de un rol que representa una estrategia de distribución de trabajo de tipo *push*, tal como el rol `Executor` definido en el RPIM de YAWL, el cual provee únicamente privilegios para ejecutar operaciones que permiten ejecutar el ítem de trabajo.

El patrón WRP-11 es soportado por BPMN por medio de la definición de tareas automáticas. El patrón WRP-04 es soportado por medio de la definición de privilegios de

recursos y/o tarea como parte del Modelo de Estructura de Recursos o el Modelo Extendido del Proceso.

Los patrones WRP-21 a WRP-27, WRP-29 a WRP-33 y WRP-43 representan la posibilidad de los recursos de ejecutar distintos tipos de operación de ítem de trabajo provistas con frecuencia por WfMSs. Estos patrones pueden ser representados definiendo los privilegios de recurso o tarea para autorizar a los recursos a ejecutar las operaciones correspondientes. De modo similar, los patrones WRP-38 a WRP-42 representan operaciones de lista de trabajo provistas con frecuencia por WfMSs. Los mismos pueden representarse por lo tanto mediante la definición de los respectivos privilegios de recursos.

De lo anterior se desprende que el marco de trabajo provee soporte a la mayoría de los patrones de recursos. Los distintos patrones pueden ser expresados en términos de los conceptos genéricos provistos en los metamodelos y extensiones, por medio de la definición de: distintos tipos de consulta parametrizada de recursos, expresiones de asignación de recursos, restricciones de resolución, roles de recursos, eventos de ítem de trabajo y privilegios de recurso o tarea en modelos independientes de la plataforma. En el caso de los modelos específicos de la plataforma, los patrones que pueden ser soportados se encuentran acotados por las restricciones impuestas por las plataformas de implementación. Por ejemplo, el WRP-10 puede ser soportado en un PS-EPM si el RPIM importado define implementaciones de clasificadores de recursos que permiten representar unidades o posiciones organizacionales, las cuales puedan ser empleadas en la definición de consultas parametrizadas de recursos.

Por otra parte, existen cuatro de los WRPs que no son cubiertos por el marco de trabajo propuesto: WRP-06 Manejo de Casos, WRP-18 Distribución Temprana, WRP-34 Rehacer y WRP-35 Pre-Hacer. Los mismos no son soportados dado que contradicen las restricciones especificadas por BPMN. Básicamente, que un ítem de trabajo no puede ser distribuido o completado antes que su creación, luego de la instanciación de la tarea de usuario correspondiente (para WRP-06, WRP-18 y WRP-35), y que una instancia de una tarea no puede ser reabierto luego de su finalización (para el WRP-34).

De esta manera, es posible determinar que el marco de trabajo propuesto soporta la mayoría de los patrones de recursos, salvando aquellos que contradicen supuestos del estándar BPMN y que por otra parte son de poco uso. Esto indica que el marco de trabajo propuesto provee suficiente expresividad para representar estos patrones en modelos de procesos, tanto independientes como dependientes de la plataforma. Por otra parte, el mapeo de elementos del marco de trabajo con patrones, ayuda a definir modelos específicos de la plataforma basados en estos patrones, determinando a la vez el soporte que provea la plataforma de implementación a los mismos.

Finalmente, el marco de trabajo propuesto no se restringe a la representación de los patrones de recursos. Requerimientos adicionales que no son considerados por estos patrones también pueden ser representados. Por ejemplo, es posible definir nuevas restricciones de resolución que no se encuentran en los patrones, para soportar un nuevo enfoque de asignación en recursos; o bien nuevos privilegios de recursos o de tarea para permitir la autorización de nuevas operaciones de ítems de trabajo no consideradas en los patrones, para que los recursos puedan acceder y completar los ítems de trabajo que se les asignan.

Tabla 14. Patrones de Recursos Soportados por el Marco de Trabajo.

	Elemento del Marco de Trabajo	Patrones Cubiertos
Aspecto Distribución de Trabajo	Consulta Parametrizada de Recursos	WRP-01 Asignación Directa, WRP-02 Asignación Basada en Roles, WRP-08 Distribución Basada en Capacidades, WRP-10 Distribución Organizacional
	Expresiones de Asignación de Recursos	WRP-03 Asignación Diferida, WRP-09 Distribución Basada en la Historia.
	Restricciones de Resolución	WRP-05 Separación de Responsabilidades, WRP-07 Mantener Familiaridad, WRP-15 Asignación Aleatoria, WRP-16 Asignación Round Robin, WRP-17 Recurso más Libre
	Evento de Ítem de Trabajo	WRP-19 Distribución en la habilitación, WRP-20 Distribución Tardía, WRP-28 Escalado
	Tarea Automática	WRP-11 Ejecución Automática
	Rol de Recurso	WRP-12 Distribución por Oferta (Único Recurso), WRP-13 Distribución por Oferta (Múltiples Recursos), WRP-14 Distribución por Asignación (Único Recurso), WRP-36 Comienzo en la Creación, WRP-37 Comienzo en la Asignación
Aspecto Autorización	Privilegio de Recurso o Privilegio de Tarea	WRP-04 Autorización, WRP-21 Asignación Iniciada por el Recurso, WRP-22 Asignación Iniciada por el Recurso (Ítem de Trabajo Asignado), WRP-23 Ejecución Iniciada por el Recurso (Ítem de Trabajo Ofrecido), WRP-24 Lista de Tareas Determinada por el Sistema, WRP-25 Lista de Tareas Determinada por el Recurso, WRP-26 Autonomía de Selección, WRP-27 Delegación, WRP-29 Desasignación, WRP-30 Reasignación con Estado, WRP-31 Reasignación sin Estado, WRP-32 Suspensión-Recomienzo, WRP-33 Salto, WRP-43 Recursos Adicionales

	Privilegios de Recursos	WRP-38 Ejecución Apilada, WRP-39 Ejecución en Cadena, WRP-40 Visibilidad Configurable de Ítems de Trabajo no Asignados, WRP-41 Visibilidad Configurable de Ítems de Trabajo Asignados, WRP-42 Ejecución Simultánea
--	-------------------------	--

6.4 Conclusiones

En este capítulo se evaluó la aplicabilidad y utilidad del marco de trabajo y el método propuestos de distintas maneras.

En primer lugar, mediante la definición de los Modelos de Implementación de la Perspectiva de Recursos (RPIM) (Sección 6.1), se mostró la utilidad del RPImplMeta para mapear el modelo conceptual del marco de trabajo propuesto con lenguajes de referencia para la definición de la perspectiva de recursos, propuestos en los ámbitos académico (YAWL) y de la industria (WS-HumanTask). Esto mostró la utilidad y factibilidad de la aplicación del marco de trabajo propuesto para representar las entidades que proveen la plataforma YAWL y el estándar WS-HumanTask para implementar la perspectiva de recursos de procesos.

En segundo lugar, por medio de los casos de estudio presentados en la Sección 6.2, se mostró la utilidad y aplicabilidad del método para definir e implementar la perspectiva de recursos en procesos de negocio de organizaciones gubernamentales y empresas privadas.

La aplicación de la Extensión de Recursos y Autorización Estática para definir Modelos de Estructura de Recursos Independientes de la Plataforma (PI-RSMs) permitió clasificar y caracterizar los recursos involucrados en la ejecución de procesos siguiendo tanto el *enfoque organizacional* como el *enfoque dirigido por el workflow* descritos en la Sección 4.3.1. El *enfoque organizacional* fue aplicado para representar la estructura formal del gobierno provincial, mientras que el *enfoque dirigido por el workflow* fue aplicado en la representación de la estructura de la empresa fabricante de muebles a medida, la cual no se encontraba formalmente descripta. Esto permitió a los directivos de estas organizaciones visualizar, entender y aprobar los agrupamientos de recursos definidos en el PI-RSM.

La aplicación de la Extensión de Distribución de Trabajo y Autorización Dinámica permitió analizar y comprender el modo en que los recursos de la organización deben ser involucrados en la ejecución de las tareas de sus procesos y proveyó un medio para expresar los requerimientos relacionados con estos aspectos de manera estructurada.

La aplicación de las propiedades de evaluación permitió identificar el grado de soporte provisto por los WfMSs a los requerimientos de perspectiva de recursos, compararlos y conocer las limitaciones de los mismos en una etapa temprana del desarrollo de los SIOPs. Así mismo, las tablas de resumen de resultados de evaluación proporcionaron un punto de partida para definir la implementación de los requerimientos de perspectiva de recursos por medio de las entidades de implementación provistas por los WfMSs.

Las reglas de verificación de PS-RSMs y PS-EPMs permitieron determinar la adecuación de las soluciones tecnológicas definidas a las restricciones impuestas por los WfMSs empleados como plataforma de implementación. Por su parte, el procedimiento de validación permitió evaluar la consistencia de los SIOPs desarrollados con los requerimientos definidos en los modelos independientes de la plataforma y conocer las inconsistencias existentes en caso que las hubiese. Finalmente, las transformaciones de modelo a código desarrolladas contribuyeron a reducir el tiempo requerido y los errores en la definición de las especificaciones que fueron incorporadas en los WfMSs que dieron soporte a los procesos.

La aplicación del marco de trabajo y el método también presentó algunas dificultades. La más importante consistió en la definición del RPIM. Aunque las extensiones BPMN fueron fácilmente comprendidas y empleadas por las personas involucradas en el diseño de los procesos y en el desarrollo de los SIOPs, la definición del RPIM resultó compleja para las mismas. Esto se debe al conocimiento que se requiere de la plataforma para definir este modelo. Por lo tanto, los RPIMs de Bonita y YAWL fueron definidos por nosotros (el tesista y el equipo de investigación de los proyectos en el que se enmarca la tesis) y almacenado en el repositorio de modelos de la herramienta desarrollada con el propósito de permitir su reutilización.

Finalmente, el marco de trabajo propuesto fue evaluado en términos de su capacidad para expresar los requerimientos recurrentes de perspectiva de recursos identificados en los patrones de recursos de workflow. El resultado de esta evaluación fue que el marco de trabajo permite definir 39 de los 43 patrones de recursos. Sin embargo, el marco de trabajo propuesto no se limita a la representación de estos patrones. Dado el carácter genérico de sus elementos, el marco de trabajo permite representar requerimientos adicionales, por medio de la descripción en lenguaje natural de los requerimientos en modelos independientes de la plataforma, y su referencia a la implementación en WfMSs que provean soporte a los mismos.

7 Conclusiones

Los procesos de workflow son procesos de negocio cuyo resultado consiste en información. Los mismos incluyen a los procesos de servicios y administrativos. Estos procesos se caracterizan por un alto grado de participación de recursos humanos en su ejecución. El desarrollo de Sistemas de Información Orientados a Procesos (SIOPs) para soportar estos procesos requiere, por lo tanto, una definición detallada de los recursos humanos involucrados en la ejecución del proceso, la forma en que el trabajo requerido por las tareas del proceso debe ser asignado a los recursos humanos y el modo en que éstos pueden gestionar el trabajo que les fue asignado.

En esta tesis se abordaron dos problemas principales: el de la definición de los requerimientos de la perspectiva de recursos en modelos conceptuales de procesos de negocio; y el de la implementación de dichos requerimientos de manera consistente en los SIOPs. Los lenguajes de modelado de procesos actuales proveen un soporte limitado a la representación de los requerimientos de la perspectiva de recursos. Por otra parte, los Sistemas de Gestión de Workflow (WfMSs) que constituyen las plataformas de implementación de los SIOPs presentan disparidades, tanto en los conjuntos de requerimientos de la perspectiva de recursos que soportan, como en los conceptos en los que se basan para ello. Finalmente, los métodos existentes para el desarrollo de SIOPs no proporcionan lineamientos adecuados que guíen a los analistas de negocio y desarrolladores de SIOP en la definición e implementación de los requerimientos de esta perspectiva de manera consistente.

Estos problemas fueron abordados aplicando la Metodología de Investigación de Ciencias de Diseño, la cual es ampliamente utilizada para llevar a cabo investigaciones en sistemas de información. La aplicación de la misma facilitó la conducción de la investigación a través del diseño y desarrollo de metamodelos, extensiones a BPMN 2.0, métodos y herramientas; y proveyó lineamientos para la evaluación y presentación de los resultados de la investigación realizada.

7.1 Contribuciones

En respuesta a los problemas identificados, en esta tesis se propuso:

- Un *Marco de Trabajo para Definir la Perspectiva de Recursos*. Compuesto por un conjunto de metamodelos y extensiones a BPMN, este marco de trabajo posibilita la definición de los requerimientos de la perspectiva de recursos de procesos de negocio en modelos conceptuales independientes de la plataforma y en modelos de procesos específicos de la plataforma para su posterior implementación.

- Un *Método para Definir la Perspectiva de Recursos en el Desarrollo de SIOPs* método de desarrollo basado en los principios del desarrollo dirigido por modelos, el cual posibilita la aplicación de un enfoque sistemático para la definición de la perspectiva de recursos de los procesos de negocio en el desarrollo de SIOPs.
- Una *Herramienta* que soporta el marco de trabajo y las etapas del método propuestos. La herramienta implementa los metamodelos y extensiones definidos en el marco de trabajo para dar soporte a la definición de los modelos (artefactos) que describen la perspectiva de recursos de procesos de negocio, y la gestión de un repositorio de dichos modelos. También facilita y posibilita a los analistas de negocio y desarrolladores de sistemas llevar a cabo las distintas etapas del método propuesto.

7.1.1 Marco de Trabajo para Definir la Perspectiva de Recursos

El marco de trabajo propuesto posibilita la definición de la perspectiva de recursos en modelos de procesos. El mismo permite estructurar la definición de esta perspectiva con base en tres aspectos: Estructura de Recursos, Distribución de Trabajo y Autorización. Estos aspectos agrupan los distintos tipos de requerimientos de la perspectiva de recursos en términos de la información requerida por tres componentes de los Sistemas de Gestión de Workflow (WfMSs) que implementan dicha perspectiva: Repositorio Organizacional, Distribución de Trabajo y Manejador de Listas de Tareas. Estos componentes se presentan en todos los modelos de referencia y arquitecturas de WfMSs que se conocen.

El marco de trabajo se basa en tres metamodelos: el Metamodelo de BPMN, el Metamodelo de la Perspectiva de Recursos (RPMeta), y el Metamodelo de Implementación de la Perspectiva de Recursos (RPIimplMeta). Los conceptos provistos por el metamodelo del lenguaje BPMN 2.0 para la definición de la perspectiva de recursos son tomados como base en los restantes metamodelos del marco de trabajo. Por lo tanto, el marco de trabajo propuesto es compatible y consistente con el estándar BPMN 2.0, el cual es ampliamente usado en la industria. Esto permite incorporar definiciones de la perspectiva de recursos en modelos de procesos definidos con BPMN durante el desarrollo de SIOPs.

El RPMeta define el modelo conceptual de la perspectiva de recursos propuesto en esta tesis. Provee un conjunto de elementos que permiten definir los aspectos y elementos necesarios para describir en forma completa la perspectiva de recursos de procesos de negocio. Este metamodelo fue definido tomando como punto de partida los elementos del metamodelo de BPMN para representar esta perspectiva. De este modo, además de representar el modelo conceptual del marco de trabajo, el RPMeta especifica los elementos adicionales requeridos para suplir las falencias de BPMN 2.0 en la representación de la perspectiva de recursos.

El RPIimplMeta define los elementos que permiten expresar el soporte a la perspectiva de recursos provisto por los WfMSs. Cada elemento del RPIimplMeta permite representar la implementación de un elemento del RPMeta por un WfMS. De este modo, el RPIimplMeta permite mapear los modelos conceptuales adoptados por distintos WfMSs al modelo conceptual del marco de trabajo propuesto. Por medio de la instanciación de este metamodelo es posible crear Modelos de Implementación de la Perspectiva de Recursos (RPIMs), los cuales contienen las entidades de implementación provistas por WfMSs para dar soporte a esta perspectiva. De esta manera es posible describir, analizar y comparar distintas plataformas de implementación existentes o futuras en términos de un único modelo conceptual.

El marco de trabajo provee además dos extensiones a BPMN 2.0 definidas a partir de los metamodelos mencionados: la Extensión de Estructura de Recursos (RSExt) y la Extensión de Distribución de Trabajo (WDExt). Estas extensiones permiten incorporar los elementos del RPMeta para definir los aspectos identificados de la perspectiva de recursos en modelos BPMN. Mediante la aplicación de las mismas es posible definir modelos conceptuales de la perspectiva de recursos y su posterior implementación en un WfMS en dos tipos de modelos BPMN, el Modelo de Estructura de Recursos (RSM) y el Modelo de Proceso Extendido (EPM).

La RSExt soporta la definición de los aspectos Estructura de Recursos y Autorización Estática en RSMs. Con el propósito de subsanar la falta de una especificación estándar para la definición de estos aspectos que pueda ser empleada en conjunto con BPMN, esta extensión provee un enfoque para su representación reutilizando los elementos Resource y ResourceParameter provistos por BPMN. Esto simplifica la tarea de importar RSMs en modelos de proceso BPMN para soportar la definición de asignaciones de recursos. También permite representar un amplio rango de estructuras organizacionales en términos de dos conceptos genéricos: Recurso Humano y Clasificador de Recurso. Dado que la definición de estos aspectos cae fuera del alcance de BPMN, una notación también fue propuesta para permitir la visualización de RSMs.

La WDExt permite definir los aspectos Distribución de Trabajo y Autorización Dinámica en EPMs. La misma provee un conjunto de atributos y elementos adicionales para el elemento ResourceRole de BPMN, que permiten describir las distintas políticas de distribución de trabajo requeridas para las tareas de usuario de los procesos, sin modificar el metamodelo de BPMN mediante el agregado de nuevas subclases de este elemento. Estos atributos adicionales permiten definir autorizaciones dinámicas, restricciones de resolución, disparadores y escaladores en modelos de proceso BPMN. No se definieron artefactos de notación adicionales para esta extensión. Los atributos de extensión son representados en los diagramas de procesos de BPMN por medio de artefactos de anotación de texto provistos por BPMN, asociados con las tareas de usuario.

Una característica importante de estas extensiones es que ambas posibilitan describir modelos conceptuales independientes de la plataforma, como así también modelos específicos de la plataforma. Estos últimos refieren a modelos BPMN que referencian un RPIM, con el propósito de describir el modelo en términos de la plataforma de implementación. En estos modelos BPMN específicos de la plataforma, sus elementos, los cuales representan los aspectos de la perspectiva de recursos, son mapeados y relacionados con los elementos de un RPIM, indicando de esta manera cómo se implementan los mismos en un WfMS.

En resumen, los principales beneficios del marco de trabajo son:

- Provee un conjunto de metamodelos para definir los requerimientos de la perspectiva de recursos, y describir las entidades de implementación provistas por distintos WfMSs, con base en un único modelo conceptual.
- Permite definir la perspectiva de recursos en modelos de procesos de alto nivel de abstracción. Esto mejora la posibilidad de analistas de negocio y desarrolladores de SIOPs de tomar decisiones en conjunto respecto de la perspectiva de recursos, es decir, acerca de la forma en la que se debe distribuir el trabajo de las tareas de los procesos entre los recursos de la organización.
- Se basa en el enfoque adoptado por BPMN 2.0 para definir la asignación de recursos a las tareas de usuario definidas en modelos de proceso. Esto permite a personas familiarizadas con el lenguaje BPMN comprender con mayor facilidad los nuevos conceptos introducidos por el marco de trabajo. Esto también incrementa la posibilidad de reutilizar las herramientas existentes que soportan el lenguaje.
- Define un conjunto de elementos genéricos para expresar el aspecto Estructura de Recursos. Esto permite clasificar y caracterizar a los recursos de las organizaciones empleando cualquier concepto. También permite definir distintos tipos de estructuras de recursos, aplicando enfoques de modelado dirigidos por el workflow o dirigidos por la estructura de la organización, empleando conceptos provenientes tanto del dominio de la organización como de la tecnología de la información.
- Permite definir el aspecto Estructura de Recursos en modelos BPMN separados (RSMs). Esto posibilita la reutilización de la definición de la estructura de la organización en múltiples modelos de proceso de negocio.
- Permite definir el aspecto Estructura de Recursos reutilizando los elementos Resource y ResourceParameter de BPMN. De este modo, la información contenida por estos modelos puede ser evaluada en modelos de proceso empleando los mecanismos estándar provistos por BPMN para definir la

asignación de recursos a los roles asignados a las tarea de usuario de los procesos.

- Permite definir el aspecto Distribución de Trabajo por medio de un único elemento (ResourceRole de BPMN), el cual es de carácter genérico. A través de la definición de valores para los atributos de este elemento agregados en la WDExt, es posible describir en modelos de procesos BPMN las distintas políticas de distribución de trabajo (*pull* o *push*), los eventos de escalado y asignación tardía, los desvíos iniciados por los recursos (tales como delegación o reasignación), y las restricciones de resolución (tales como RoundRobin o separación de responsabilidades).
- Permite definir el aspecto Autorización mediante la asociación de privilegios de manera estática con recursos (en forma individual o grupal), o bien en forma dinámica mediante la definición de los privilegios otorgados por un rol de recurso o revocados a nivel de una tarea de usuario. De este modo es posible describir cualquier operación que un recurso puede llevar a cabo para organizar o ejecutar los trabajos que se le asignan. Esto también habilita a definir distintos tipos de políticas de autorización para la ejecución de tareas de usuario, agregándose de esta manera aspectos relacionados a la seguridad en el acceso a las tareas de usuario de los procesos automatizados por un SIOP.
- Permite describir las plataformas de implementación de SIOPs con base en un único modelo conceptual. Esto simplifica la tarea de comprender el soporte provisto por los distintos WfMSs a la perspectiva de recursos, sin la necesidad de lidiar con los distintos conceptos empleados por cada uno de ellos para definir esta perspectiva. También habilita la comparación y análisis de diferentes plataformas o WfMSs.
- Hace uso del mecanismo provisto por BPMN para su extensión, lo cual permite incorporar detalles de la perspectiva de recursos en modelos de procesos BPMN sin afectar la portabilidad de los mismos. De este modo, los modelos BPMN extendidos pueden ser creados, leídos y editados por herramientas BPMN que utilicen el formato de almacenamiento adoptado por BPMN para los modelos de procesos.

El marco de trabajo fue evaluado en términos de la expresividad, aplicabilidad y utilidad de sus metamodelos y extensiones a BPMN para definir e implementar la perspectiva de recursos. Por un lado se evaluó la aplicación de los mismos en casos de estudio de procesos reales, para describir las entidades de implementación provistas por distintos tipos de WfMSs. Por otro lado se evaluó la capacidad expresiva de los elementos de los metamodelos y las extensiones a BPMN para representar los Patrones de Recursos de Workflow (WRPs). El resultado de dicha evaluación puede resumirse de la siguiente manera:

- Los RSMs permitieron reducir las brechas de comunicación entre miembros de las organizaciones, analistas de negocio y desarrolladores de SIOP por medio de la representación de estructuras organizacionales en términos de conceptos familiares para sus miembros, permitiendo a la vez definir privilegios de recursos y su posterior mapeo en plataformas de implementación.
- Los EPMs permitieron integrar la información de RSMs y RPIMs en modelos de procesos BPMN para definir las políticas de distribución de trabajo requeridas y su implementación en un WfMS.
- Los RPIMs permitieron representar las entidades de implementación provistas por distintos tipos de WfMSs, tales como YAWL de uso académico, Bonita de de uso comercial, y aquellos basados en los estándares BPEL4People y WS-HumanTask para procesos de workflow.
- La evaluación de los metamodelos y las extensiones a BPMN en términos de su capacidad para expresar los WRPs mostró que la mayoría de los patrones (39 de un total de 43) pueden ser soportados. Esto implica que el marco de trabajo propuesto provee una gran expresividad para representar la perspectiva de recursos de procesos de negocio. Cabe destacar que su expresividad no se limita a la representación de los WRPs. Dado que el metamodelo de la perspectiva de recursos propuesto se basa en conceptos genéricos, permite expresar otros requerimientos no considerados por los WRPs, y que están presentes en otros modelos de referencia o WfMSs, tales como WS-HumanTask.

7.1.2 Método para Definir la Perspectiva de Recursos en el Desarrollo de SIOPs

El método propuesto para definir la perspectiva de recursos en el desarrollo de SIOPs tiene como propósito permitir la representación, de manera conceptual e independiente de la tecnología de implementación, de los requerimientos relacionados con esta perspectiva, como así también su implementación consistente en una plataforma de implementación adecuada. Para ello, provee un enfoque basado en los principios del desarrollo dirigido por modelos. Esto implica comenzar con la definición de modelos de procesos independientes de la plataforma de implementación; luego derivar de los mismos los modelos específicos de la plataforma, es decir, descritos en términos de los elementos del WfMS seleccionado; y finalmente, generar a partir de estos modelos las especificaciones ejecutables de procesos (código), las cuales serán interpretadas por el WfMS en que se implementa el SIOP.

El método emplea el marco de trabajo propuesto para partiendo de un modelo de proceso BPMN, llevar a cabo: (i) la definición de los requerimientos de la perspectiva de recursos en un Modelo de Estructura de Recursos Independiente de la Plataforma (PI-RSM) y un Modelo de Proceso Extendido Independiente de la Plataforma (PI-EPM); (ii) la evaluación y selección de un WfMS o plataforma de implementación representada por un

Modelo de Implementación de Perspectiva de Recursos (RPIM); (iii) la implementación de los requerimientos de la perspectiva de recursos en un Modelo de Estructura de Recursos Específico de la Plataforma (PS-RSM) y un Modelo de Proceso Extendido Específico de la Plataforma (PS-EPM), ambos con base en el RPIM seleccionado; (iv) la verificación y validación de los requerimientos de la perspectiva de recursos definidos en modelos específicos de la plataforma contra las restricciones impuestas por el RPIM y los requerimientos definidos en los modelos independientes de la plataforma; (v) la generación automática de especificaciones ejecutables de workflow implementando la perspectiva de recursos en el WfMS seleccionado.

Los principales beneficios del método son:

- Permite incorporar definiciones de la perspectiva de recursos en modelos de procesos definidos con BPMN. Dado que BPMN es un lenguaje ampliamente adoptado por analistas de negocio y desarrolladores de SIOPs para definir la lógica de los procesos de negocio.
- El método propuesto puede ser utilizado como un método complementario a aquellos existentes para modelar los procesos de una organización empleando BPMN. Los enfoques o métodos existentes para definir modelos de procesos se centran en representar con detalle sólo las perspectivas de control y de datos de los procesos, pero dejan de lado los aspectos de la perspectiva de recursos. Partiendo de estos modelos que describan el flujo de control y de datos de los procesos, luego se puede aplicar el método propuesto en esta tesis para adicionar los detalles de la perspectiva de recursos y guiar el desarrollo de los SIOPs.
- Es de carácter prescriptivo. Provee guías para la utilización de los elementos provistos por el marco de trabajo con el propósito de representar los aspectos de la perspectiva de recursos. Estas guías son acompañadas por un conjunto de procedimientos de inicialización de modelos; y técnicas de evaluación, verificación, validación y generación de especificaciones descriptas con un nivel de detalle que permite su automatización en una herramienta. De este modo, se provee una indicación clara a analistas de negocio y desarrolladores respecto de cómo obtener el resultado de cada etapa a partir de la información o los modelos que constituyen la entrada de la misma.
- Provee un conjunto de propiedades de evaluación que permiten calcular tasas de soporte de los distintos WfMSs a los requerimientos de perspectiva de recursos definidos en modelos conceptuales. Esto posibilita contar con una base cuantitativa para la comparación y selección del WfMS más adecuado como plataforma de implementación de un proceso, de acuerdo a los requerimientos

relacionados con los distintos aspectos de la perspectiva de recursos del proceso en consideración.

- Provee guías para la creación de modelos específicos de la plataforma (PS-RSM y PS-EPM) por medio del cálculo del conjunto de entidades de implementación candidatas para cada elemento Resource y ResourceRole de los modelos conceptuales (PI-RSM y PI-EPM). Provee también lineamientos para lidiar con las limitaciones impuestas por los WfMSs. Aún en el caso que se concluya que el WfMS seleccionado no es adecuado, esto puede determinarse en las primeras etapas del desarrollo del SIOP sobre la base de modelos, lo cual tiene un costo significativamente menor a que sea determinado en las etapas finales de implementación del SIOP sobre la base de especificaciones ejecutables de procesos.
- Provee reglas de verificación para constatar la adecuación de los modelos específicos de la plataforma a las restricciones impuestas por el WfMS seleccionado. Esto permite constatar que los modelos específicos de la plataforma a ser empleados son adecuados para ser usados como entrada para las transformaciones de modelo a código que generan las especificaciones ejecutables de procesos para el WfMS seleccionado.
- Provee mecanismos de validación para determinar la consistencia de la perspectiva de recursos definida en los modelos específicos de la plataforma respecto de la definida en los modelos independientes de la plataforma. Esto resulta de particular importancia dado que con frecuencia los modelos específicos de la plataforma requieren la introducción de modificaciones en forma manual para acomodarse a las restricciones impuestas por el WfMS. Además, y de principal importancia, es que esto contribuye a mantener la consistencia entre los requerimientos de la perspectiva de recursos definidos en el nivel de negocio u organizacional y la implementación de dichos requerimientos según el WfMS seleccionado.
- Provee un enfoque para automatizar la generación de especificaciones ejecutables requeridas por el WfMS, las cuales son construidas a partir de los modelos específicos de la plataforma y de los RPIMs definidos para cada plataforma de implementación. Esto permite reducir el costo y esfuerzo requerido para implementar SIOPs con base en un determinado WfMS, especialmente mediante la reutilización de los RPIMs y transformaciones de modelos a especificaciones de procesos ya predefinidas.

La aplicabilidad y utilidad del método propuesto fueron evaluadas por medio de su aplicación en casos de estudio reales. Como resultado de la aplicación del método se observó que:

- El carácter prescriptivo del método contribuyó con el entendimiento de los elementos del marco de trabajo y permitió una definición más rica de los requerimientos relacionados con la perspectiva de recursos al guiar a analistas de negocio y desarrolladores en el uso de dichos elementos.
- La aplicación del método permitió reducir el costo y esfuerzo requeridos para desarrollar SIOPs con base en WfMSs, que implementen los aspectos de la perspectiva de recursos en forma consistente con los requerimientos definidos en modelos conceptuales que constituyen las soluciones de negocio.

7.1.3 Herramienta

Una herramienta que soporta el marco de trabajo y las etapas del método fue desarrollada con base en la plataforma Oryx. El propósito de la misma es mostrar la factibilidad de implementar el enfoque propuesto en herramientas BPMN existentes, así como tener una herramienta que permita aplicar el marco de trabajo y el método en casos de estudio. Los principales beneficios de la herramienta son:

- Muestra la factibilidad de implementar el marco de trabajo y el método propuestos reutilizando y extendiendo herramientas existentes para definir procesos de negocio que soportan BPMN 2.0. En este caso se extendió la plataforma Oryx.
- Provee editores que permiten definir los distintos tipos de modelos definidos en el marco de trabajo, así como asistentes que permiten almacenar y recuperar dichos modelos de un repositorio e integrarlos para cumplimentar las distintas etapas del método propuesto.
- Soporta la automatización de las tareas de inicialización de modelos, de las técnicas de cálculo de tasas de soporte de RPIMs, la verificación y validación de modelos, y la generación de especificaciones ejecutables de procesos.
- Es extensible para soportar la generación de especificaciones ejecutables requeridas por WfMSs existentes y nuevos, por medio de la incorporación en el repositorio de modelos de RPIMs adicionales y sus transformaciones modelo a código asociadas como complementos del lado del servidor.

La herramienta desarrollada fue empleada para desarrollar los distintos casos de estudio presentados en esta tesis.

7.2 Limitaciones y Trabajos Futuros

Aunque el marco de trabajo y el método soportan la definición de un amplio rango de requerimientos de la perspectiva de recursos en modelos conceptuales, la implementación de los mismos se encuentra restringida por las funcionalidades provistas por los WfMSs existentes. Aún en el caso que se definiera un RPIM por cada WfMS existente, no es

posible asegurar que exista un WfMS que satisfaga todos los requerimientos de la perspectiva de recursos definidos en un PI-RSM o PI-EPM.

Con el propósito de resolver este problema, un trabajo en curso consiste en el desarrollo de un método para definir nuevas plataformas de implementación de la perspectiva de recursos, las cuales puedan ser integradas en WfMSs para mejorar su capacidad de gestionar esta perspectiva en tiempo de ejecución. Este método deberá proveer:

- Un enfoque sistemático para definir, empleando modelos de procesos BPMN, el comportamiento de un servicio Web de orquestación que implemente las políticas de distribución de trabajo soportadas por las entidades de implementación definidas en un RPIM, por medio de la invocación de operaciones primitivas provistas por una arquitectura orientada a servicios desarrollada a tal fin.
- Un enfoque basado en los principios del desarrollo dirigido por modelos para definir la integración de las plataformas de implementación de la perspectiva de recursos resultantes con los distintos tipos de WfMSs.

Aunque el marco de trabajo y el método propuestos proveen mecanismos de verificación para determinar la adecuación de los modelos específicos de la plataforma a las restricciones definidas en el RPIM correspondiente, no provee mecanismos para verificar propiedades relacionadas con la correcta definición de la perspectiva de recursos. Por ejemplo, determinar el uso de políticas de distribución de trabajo incompatibles en una tarea, determinar si el uso de políticas de delegación de tareas pueden llevar a bloqueos o ciclos infinitos que impidan la ejecución de una tarea y la finalización del proceso, si el uso incorrecto de restricciones de acceso impiden la autorización de un usuario para ejecutar una tarea que tiene asignada, entre otros. Con este propósito, deberían identificarse propiedades deseadas o esperadas de esta perspectiva, las cuales puedan ser evaluadas del mismo modo en que propiedades como *soundness* son evaluadas para la perspectiva de control de flujo. Además, el análisis de las implicaciones en cuanto al desempeño de las distintas políticas de distribución de trabajo definidas puede ser adecuado para su validación. Propuestas en este sentido también se consideran como trabajos futuros.

8 Anexo I Definición y Validación de Modelos BPMN+X

En este anexo se presentan los pasos para definir un modelo BPMN+X a partir de un modelo conceptual de la extensión (Sección 8.1) y para verificar que el modelo BPMN+X se adecúa a las restricciones impuestas por el mecanismo de extensión de BPMN (Sección 8.2).

8.1 Definición de un Modelo BPMN+X a partir de un Modelo Conceptual de Extensión

El procedimiento que permite definir un modelo BPMN+X con base en el modelo conceptual de la extensión definido en UML consiste en los siguientes pasos:

- Crear un modelo de extensión BPMN+X con el mismo nombre que el modelo conceptual UML y poblarlo con un BPMNElement y un BPMNEnum para cada clase y enumeración del modelo conceptual caracterizado como Concepto de BPMN. Luego, agregar un elemento ExtensionEnum por cada enumeración caracterizada como Concepto de Extensión.
- Analizar las propiedades de cada clase del modelo conceptual. Sea c una clase del modelo conceptual y p una propiedad de tipo t , la cual es un atributo de c o que es navegable a partir de c a través de una asociación con t . La representación de c , p y t en el modelo BPMN+X depende de:
 - si c fue caracterizada como Concepto de BPMN o Concepto de Extensión;
 - si p es una propiedad de un Concepto BPMN definida en la especificación de BPMN (Original) o una propiedad de un Concepto BPMN o un Concepto de Extensión definida como parte del modelo conceptual de la extensión (Nueva)
 - si t es un Concepto de BPMN, un Concepto de Extensión o un Tipo Primitivo, donde Tipo Primitivo incluye enumeraciones definidas tanto por BPMN como en el modelo de la extensión y tipos de dato simples tales como cadenas, números o booleanos.

La Tabla 15 resume las reglas que definen las representaciones posibles de p en un modelo BPMN+X con base en los parámetros introducidos arriba. La representación de c y t es luego inferida con base en la representación de p .

La regla 1 se aplica cuando p es una propiedad original de un *Concepto de BPMN*. Si t es un Tipo Primitivo (regla 1a), p se representa como un atributo del elemento BPMNElement que corresponde con c en el modelo BPMN+X. En cambio, si p es de un tipo definido por una instancia de BPMNElement, p se representa como una asociación navegable desde la instancia de BPMNElement que corresponde con c a la que corresponde con t .

La regla 2 es válida cuando p es una propiedad nueva de un Concepto de BPMN, lo cual corresponde con un elemento `ExtensionAttributeDefinition` en el mecanismo de extensión de BPMN. En esta etapa, un elemento `ExtensionDefinition` asociado con la por medio de un elemento `ExtensionRelationship` con la instancia de `BPMNElement` que corresponde con c debe ser creado con nombre adecuado. Luego de ello, si t es un Tipo Primitivo, (regla 2a), p se representa como un atributo del elemento `ExtensionDefinition` creado. Si t es un Concepto de BPMN (regla 2b), p se especifica como una asociación navegable desde el elemento `ExtensionDefinition` creado a una instancia de `BPMNElement` que corresponde con t . En caso que t corresponda con un Concepto de Extensión concreto, en el sentido que t defina un nuevo tipo de elemento que requiere instanciarse en un modelo BPMN, debe crearse una instancia de `ExtensionElement` para representar este concepto y p se representa como una asociación navegable desde la instancia creada de `ExtensionDefinition` a la instancia de `ExtensionElement` que representa t (regla 2c).

La regla 3 es aplicable cuando p es una nueva propiedad de un Concepto de BPMN y t es un Concepto de Extensión abstracto, definido únicamente con el propósito de agrupar un conjunto de propiedades a ser agregadas a c . En este caso, t se representa como una instancia de `ExtensionDefinition` y p se representa como una instancia de `ExtensionRelationship` desde el elemento `BPMNElement` que representa c al elemento `ExtensionDefinition` que representa t .

La regla 4 es apropiada cuando p es una propiedad de un Concepto de Extensión. Un requisito para la aplicación de esta regla es la clasificación previa de c como `ExtensionDefinition` o como `ExtensionElement`. Esto es realizado por medio de la aplicación de las reglas 2c, 3 o 4c a las propiedades del tipo definido por c ; y la aplicación de las reglas 7 y 8 a las relaciones de generalización que involucren a c . Si t es un Tipo Primitivo (regla 4a), p se representa como un atributo de c . En cambio, si t es un Concepto de BPMN (regla 4b), p se define como una asociación navegable desde c a t . Finalmente, si t es un Concepto de Extensión (regla 4c), t se especifica como una instancia de `ExtensionElement` y p por medio de una asociación navegable desde c a t .

Tabla 15. Posibles representaciones de una propiedad de un modelo conceptual en un modelo BPMN+X.

Regla	Clase (c)	Propiedad (p)	Tipo (t)	Representación BPMN+X de p
1a	Concepto de BPMN	Original	Tipo Primitivo	<code>BPMNElement Property</code>
1b	Concepto de BPMN	Original	Concepto de BPMN	<code>BPMNElement Property</code>
2a	Concepto de BPMN	Nueva	Tipo Primitivo	<code>ExtensionAttributeDefinition</code>

2b	Concepto de BPMN	Nueva	Concepto de BPMN	ExtensionAttributeDefinition
2c	Concepto de BPMN	Nueva	Concepto de Extensión	ExtensionAttributeDefinition
3b	Concepto de BPMN	Nueva	Concepto de Extensión	ExtensionRelationship
4a	Concepto de Extensión	Nueva	Tipo Primitivo	ExtensionElement Property / ExtensionAttributeDefinition
4b	Concepto de Extensión	Nueva	Concepto de BPMN	ExtensionElement Property / ExtensionAttributeDefinition
4c	Concepto de Extensión	Nueva	Concepto de Extensión	ExtensionElement Property / ExtensionAttributeDefinition

Además de las propiedades definidas en el modelo conceptual de la extensión, es necesario analizar las relaciones de generalización entre los elementos del mismo. Sea g una relación de generalización desde una clase c a una súper clase s , la representación de g , c y s en el modelo BPMN+X depende de la caracterización de c y s y de si la relación g es original o nueva. La Tabla 16 resume las reglas para define la representación de g en el modelo BPMN+X con base en estos parámetros. La representación de c y s puede inferirse con base en la representación de g .

La regla 5 es válida cuando g es una relación de generalización original definida entre dos Conceptos de BPMN. En este caso, g se representa como una relación de generalización entre las instancias de BPMNElement que representan c y s .

La regla 6 se aplica cuando g es una relación de generalización nueva definida entre Conceptos BPMN, la cual no se encuentra definida en el metamodelo de BPMN. En este caso, g se considera como inválida y no es representada en el modelo BPMN+X.

La regla 7 es adecuada cuando g es una relación de generalización desde un Concepto de Extensión a un Concepto de BPMN. En este caso debe decidirse si g debe representarse como una instancia de ExtensionRelationship o considerada como inválida. El mecanismo de extensión de BPMN soporta el enfoque de extensión por adición dado que únicamente permite agregar elementos específicos de un dominio a los elementos originales de BPMN. Este mecanismo no permite crear un nuevo tipo de elemento con base en un elemento original del lenguaje. Por lo tanto, en sentido estricto, g debería considerarse como inválido (regla 7b). Sin embargo, como una solución de compromiso, g puede representarse como una instancia de ExtensionRelationship y c como una instancia de ExtensionDefinition (regla 7a). De este modo, las propiedades adicionales definidas por c pueden ser agregadas a s .

Finalmente, la regla 8 es válida cuando g es una relación de generalización desde un Concepto de Extensión a otro Concepto de Extensión. En este caso, g se representa como

una relación de generalización desde c a s ; y c y s se representan por medio de instancias de `ExtensionElement` (regla 8a) o `ExtensionDefinition` (regla 8b).

Tabla 16. Posibles representaciones de una generalización de un modelo conceptual en un modelo BPMN+X.

Regla	Clase (c)	Generalización (g)	Super Clase (s)	Representación BPMN+X de g
5	Concepto de BPMN	Original	Concepto de BPMN	Generalización entre elementos BPMNElement
6a	Concepto de BPMN	Nueva	Concepto de BPMN	Invalid
6b	Concepto de BPMN	Nueva	Concepto de Extensión	Invalid
7a	Concepto de Extensión	Nueva	Concepto de BPMN	ExtensionRelationship
7b	Concepto de Extensión	Nueva	Concepto de BPMN	Invalid
8a	Concepto de Extensión	Nueva	Concepto de Extensión	Generalización entre elementos ExtensionElement
8b	Concepto de Extensión	Nueva	Concepto de Extensión	Generalización entre elementos ExtensionDefinition

8.2 Verificación de Modelos BPMN+X

Para adecuarse a la especificación de BPMN, un modelo BPMN+X debe observar un conjunto de restricciones. Estas restricciones fueron definidas empleando el lenguaje OCL e incorporadas en la definición del perfil BPMN+X. Las mismas se muestran a continuación:

```

context ExtensionModel
inv define_extensions:self.base_Package.packagedElement->exists(c | not
c.extension_ExtensionDefinition.oclIsUndefined())
inv only_has_stereotyped_classes:
self.base_Package.packagedElement->forall(p |
p.oclIsTypeOf(uml::Class) implies not (
p.extension_ExtensionDefinition.oclIsUndefined() and
p.extension_BPMNElement.oclIsUndefined() and
p.extension_BPMNElement.oclIsUndefined()))
inv only_has_stereotyped_enumerations:
self.base_Package.packagedElement->forall(p |
p.oclIsTypeOf(uml::Enumeration) implies not (
p.extension_ExtensionEnum.oclIsUndefined() and
p.extension_BPMNEnum.oclIsUndefined()))
inv not_navigable_reference_extension_definition_elements:
self.base_Package.packagedElement->select(o |
o.oclIsTypeOf(uml::Association))->forall(a |
a.oclAsType(uml::Association).navigableOwnedEnd->forall(e |
e.type.extension_ExtensionDefinition.oclIsUndefined()))
inv not_reference_extension_definition_element:
self.base_Package.packagedElement->select(o |
o.oclIsTypeOf(uml::Association))->forall(a |
a.oclAsType(uml::Association).navigableOwnedEnd->isEmpty() implies

```

```

    a.oclasType(uml::Association).relatedElement->forall(c |
    c.extension_ExtensionDefinition.oclisUndefined()))
inv bpmn_elements_only_reference_bpmn_elements:
self.base_Package.packagedElement->select(o |
o.oclisTypeOf(uml::Association))->forall(a |
    a.oclasType(uml::Association).relatedElement->exists(c |
    not c.extension_BPMNElement.oclisUndefined()
    ) implies
    a.oclasType(uml::Association).navigableOwnedEnd->forall(e |
    not e.type.extension_BPMNElement.oclisUndefined()) or
    ( a.oclasType(uml::Association).navigableOwnedEnd->isEmpty() and
    a.oclasType(uml::Association).relatedElement->forall(c | not
    c.extension_BPMNElement.oclisUndefined()))))
inv only_single_inheritance:
self.base_Package.packagedElement->forall(p |
    p.oclisTypeOf(uml::Class) implies
    p.oclasType(uml::Class).generalization->size() <= 1)
inv no_extension_definition_attributes:
self.base_Package.packagedElement->forall(p |
    p.oclisTypeOf(uml::Class) implies
    p.oclasType(uml::Class).attribute->forall(a |
    a.type.extension_ExtensionDefinition.oclisUndefined()))

context ExtensionDefinition
inv only_inherits_from_extension_definitions:
self.base_Class.allParents()->forall(c |
    not c.extension_ExtensionDefinition.oclisUndefined())
inv is_abstract: self.base_Class.isAbstract = true

context ExtensionElement:
inv only_inherits_from_extension_elements:
self.base_Class.allParents()->forall(c |
    not c.extension_BPMNElement.oclisUndefined())

context BPMNElement
inv only_inherits_from_bpmn_elements:
self.base_Class.allParents()->forall(c |
    not c.extension_BPMNElement.oclisUndefined())
inv only_reference_bpmn_elements:
self.base_Class.ownedAttribute->select(at |
    at.type.oclisTypeOf(uml::Class))->forall(a |
    not a.type.extension_BPMNElement.oclisUndefined())

context ExtensionRelationship
inv client_must_be_bpmn_element: self.base_Dependency.client->forall(c |
    not c.extension_BPMNElement.oclisUndefined())
inv supplier_must_be_extension_definition:
self.base_Dependency.supplier->forall(s |
    not s.extension_ExtensionDefinition.oclisUndefined())

```


9 Anexo II Reglas de Verificación de PS-RSM y PS-EPM

En este anexo se presentan las restricciones OCL definidas con el propósito de formalizar las reglas de verificación de PS-RSMs y PS-EPMs respecto de RPIMs descriptas en la Sección 4.6.

```
R1)
context Resource
inv resourceClassifications:
self.resourceParameters->forall(rc:ResourceParameter |
    rc.extensionDefinitions->forall(ed:ExtensionDefinition |
        ed.name = 'ResourceClassification'
        implies
        rc.extensionValues->select(ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name = 'target'
        )->asSequence()->first() .valueRef.oclAsType(Resource) .
        extensionValues->select(ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name = 'implementation'
        )->asSequence()->first() .valueRef
        =
        rc.extensionValues->select(ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name = 'implementation'
        )->asSequence()->first() .valueRef.oclAsType
        (ClassificationImpl) .target
        and
        self.extensionValues->select(ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name = 'implementation'
        )->asSequence()->first() .valueRef
        =
        rc.extensionValues->select(ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name = 'implementation'
        )->asSequence()->first() .valueRef .
        oclAsType(ClassificationImpl) .source
    )
)
```

```
R2)
context Resource
inv resourceReferences:
self.resourceParameters->forall(rr:ResourceParameter |
    rr.extensionDefinitions->forall(ed:ExtensionDefinition |
        ed.name = 'ResourceReference'
        implies
        rr.extensionValues->select(ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name = 'target'
        )->asSequence()->first() .valueRef.oclAsType(Resource) .
        extensionValues->select(ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name = 'implementation'
        )->asSequence()->first() .valueRef
        =
        rr.extensionValues->select(ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name =
            'implementation'
        )->asSequence()->first() .valueRef .
        oclAsType(ResourceReferenceImpl) .target
    )
)
```

```

        and
        self.extensionValues->select(ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name =
                'implementation'
        )->asSequence()->first().valueRef
    =
    rr.extensionValues->select(ev:ExtensionAttributeValue |
        ev.extensionAttributeDefinition.name = 'implementation'
    )->asSequence()->first().valueRef.
    oclAsType(ResourceReferenceImpl).source
    )
)

```

R3)

```

context ResourceParameter
inv subsumptions:
self.extensionDefinitions->forAll(ed:ExtensionDefinition |
    ed.name = 'ResourceReference'
    implies
    self.extensionValues->select(ev:ExtensionAttributeValue |
        ev.extensionAttributeDefinition.name = 'isSubsumption'
    )->asSequence()->first().value
    =
    self.extensionValues->select(ev:ExtensionAttributeValue |
        ev.extensionAttributeDefinition.name = 'implementation'
    )->asSequence()->first().valueRef.
    oclAsType(ResourceReferenceImpl).isSubsumption
)
)

```

R4)

```

context Resource
inv valuedParameters:
self.resourceParameters->forAll(rp:ResourceParameter |
    rp.extensionDefinitions->forAll(ed:ExtensionDefinition |
        ed.name = 'ValuedParameter'
        implies
        self.extensionValues->select(ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name = 'parameterValues' and
            ev.value.parameter = rp
        )->asSequence()->first().value.
        oclAsType(ResourceParameterValue).implementation
        =
        rp.extensionValues->select(ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name = 'implementation'
        )->asSequence()->first().valueRef.
        oclAsType(ResourceParameterImpl).type
        and
        self.extensionValues->select(ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name = 'implementation'
        )->asSequence()->first().valueRef
        =
        rp.extensionValues->select(ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name = 'implementation'
        )->asSequence()->first().valueRef.
        oclAsType(ResourceParameterImpl).resource
    )
)
)

```

```

    )
)
R5)
context Resource
inv resourcePrivileges:
self.extensionDefinitions->forAll(ed:ExtensionDefinition |
    ed.name = 'HumanResource' or ed.name = 'ResourceClassifier'
implies
self.extensionValues->select(ev:ExtensionAttributeValue |
    ev.extensionAttributeDefinition.name = 'resourcePrivileges'
)->forAll(ev:ExtensionAttributeValue |
    ev.value.oclAsType(ResourcePrivilege).implementation.
oclAsType(ResourcePrivilegeImpl).resource
    =
self.extensionValues->select(ev:ExtensionAttributeValue |
    ev.extensionAttributeDefinition.name = 'implementation'
)->asSequence()->first() .valueRef
    )
)
)

```

```

R6)
context ResourceRole
inv resourceParameterBinding:
not self.resourceRef.oclIsUndefined()
implies
self.resourceRef.extensionDefinitions->forAll(
    ed:ExtensionDefinition |
    ed.name = 'HumanResource' or ed.name = 'ResourceClassifier'
implies
self.resourceRef.extensionValues->select(
    ev:ExtensionAttributeValue |
    ev.extensionAttributeDefinition.name = 'implementation'
)->asSequence()->first() .valueRef.
oclAsType(ResourceImpl).isReferenceable
)
and
self.resourceParameterBindings->forAll(
    rp:ResourceParameterBinding |
    rp.parameterRef.extensionDefinitions->forAll(
        ed:ExtensionDefinition |
        ed.name = 'ValuedParameter'
implies
        rp.extensionValues->select(ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name = 'implementation'
        )->asSequence()->first() .valueRef.
oclAsType(ResourceParameterImpl).type.isBindable
    )
)
)

```

```

R7)
context Definitions
inv resourceAssignmentExpression:
self.rootElements->select(p |
    p.oclIsTypeOf(Process) and

```

```

    p.oclAsType(Process).flowElements->select(u |
        u.oclIsTypeOf(UserTask) and
        u.oclAsType(UserTask).resources->select(r:ResourceRole |
            not r.resourceAssignmentExpression.oclIsUndefined()
        )->size() > 0
    )->size() > 0
)->size() > 0
implies
self.rootElements->select(r |
    r.oclIsTypeOf(Resource))->forall(r : Resource |
        r.extensionValues->select(ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name = 'implementation'
        )->asSequence()->first().valueRef.
        oclAsType(ResourceImpl).isBindable
    )
)

```

R8)

```

context ResourceRole
inv resolutionConstraint :
self.extensionValues->select(ev:ExtensionAttributeValue |
    ev.extensionAttributeDefinition.name = 'constraints'
)->forall(c:ResolutionConstraint |
    (
        not c.expression.oclIsUndefined()
        implies
        c.implementation.oclAsType(ResolutionConstraintImpl).hasExpression
    )
    and
    (
        c.isSoft
        =
        c.implementation.oclAsType(ResolutionConstraintImpl).isSoft
    )
)
)

```

R9)

```

context ResourceRole
inv trigger :
self.extensionValues->select(
    ev:ExtensionAttributeValue |
    ev.extensionAttributeDefinition.name = 'trigger'
)->forall(ev: ExtensionAttributeValue |
    ev.value.oclAsType(WorkItemEvent).implementation.
    triggerableOperations->select(
        o:WorkItemOperationImpl |
        o.triggers->includes(
            self.extensionValues->select(
                ev1:ExtensionAttributeValue |
                ev1.extensionAttributeDefinition.name =
                'implementation'
            )->asSequence()->first().valueRef
        )
    )->size() > 0
)
)

```

```

R10)
context ResourceRole
inv escalation :
self.extensionValues->select(ev:ExtensionAttributeValue |
    ev.extensionAttributeDefinition.name = 'escalation'
)->forall(ev: ExtensionAttributeValue |
    ev.value.oclAsType(WorkItemEvent).implementation.
    triggerableOperations->select(
        o: WorkItemOperationImpl |
        o.escalates->includes(
            self.extensionValues->select(
                ev1:ExtensionAttributeValue |
                ev1.extensionAttributeDefinition.name =
                    'implementation'
            )->asSequence()->first() .valueRef
        )
    )->size() > 0
)

R11)
context UserTask
inv revokedPrivileges :
self.extensionValues->select(ev:ExtensionAttributeValue |
    ev.extensionAttributeDefinition.name = 'revokedPrivileges'
)->forall(p:TaskPrivilege |
    p.implementation.oclAsType(TaskPrivilegeImpl).isRevokable
)

R12)
context Definitions
inv classificationsCardinality:
self.extensionValues->select(ev:ExtensionAttributeValue |
    ev.extensionAttributeDefinition.name = 'RPIM'
)->asSequence()->first() .value.classifications->forall(
    ci:ClassificationImpl |
    self.rootElements->select(r |
        r.oclIsTypeOf(Resource)
        and
        r.oclAsType(Resource).extensionValues->select(
            ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name = 'implementation'
        )->asSequence()->first() .valueRef = ci.target
    )->forall(t:Resource |
        self.rootElements->select(r |
            r.oclIsTypeOf(Resource)
            and
            r.oclAsType(Resource).extensionValues->select(
                ev:ExtensionAttributeValue |
                ev.extensionAttributeDefinition.name =
                    'implementation'
            )->asSequence()->first() .valueRef = ci.source
        ) and
        r.resourceParameters->select(rp:ResourceParameter |
            rp.extensionDefinitions->select(
                ed:ExtensionDefinition |
                ed.name = 'ResourceClassification'
            )
        )
    )
)

```

```

        )->size() = 1
        and
        rp.extensionValues->select(
            ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name =
                'target'
        )->asSequence()->first().valueRef.
        oclAsType(Resource).extensionValues->select(
            ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name =
                'implementation'
        )->asSequence()->first().valueRef = ci.target
    )->size() > 0
) ->size() >= ci.lower
and
self.rootElements->select(r |
    r.oclIsTypeOf(Resource)
    and
    r.oclAsType(Resource).extensionValues->select(
        ev:ExtensionAttributeValue |
        ev.extensionAttributeDefinition.name =
            'implementation'
    )->asSequence()->first().valueRef = ci.source
    and
    r.resourceParameters->select(
        rp:ResourceParameter |
        rp.extensionDefinitions->select(
            ed:ExtensionDefinition |
            ed.name = 'ResourceClassification'
        )->size() = 1
        and
        rp.extensionValues->select(
            ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name =
                'target'
        )->asSequence()->first().valueRef.
        oclAsType(Resource).extensionValues->select(
            ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name =
                'implementation'
        )->asSequence()->first().valueRef = ci.target
    )->size() > 0
) ->size() <= ci.upper
)
)

```

R13)

context Definitions

inv referencesCardinality:

```

self.extensionValues->select(ev:ExtensionAttributeValue |
    ev.extensionAttributeDefinition.name = 'RPIM'
) ->asSequence()->first().value.references->forAll(
    rri:ResourceReferenceImpl |
    self.rootElements->select(r |
        r.oclIsTypeOf(Resource)
    )
)

```

```

and
r.oclAsType(Resource).extensionValues->select(
    ev:ExtensionAttributeValue |
    ev.extensionAttributeDefinition.name = 'implementation'
)->asSequence()->first().valueRef = rri.target
)->forAll(t:Resource |
self.rootElements->select(r |
    r.oclIsTypeOf(Resource)
and
    r.oclAsType(Resource).extensionValues->select(
        ev:ExtensionAttributeValue |
        ev.extensionAttributeDefinition.name =
            'implementation'
    )->asSequence()->first().valueRef = rri.source
and
    r.resourceParameters->select(rp:ResourceParameter |
        rp.extensionDefinitions->select(
            ed:ExtensionDefinition |
            ed.name = 'ResourceReference'
        )->size() = 1
and
        rp.extensionValues->select(
            ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name =
                'target'
        )->asSequence()->first().valueRef.
        oclAsType(Resource).extensionValues->select(
            ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name =
                'implementation'
        )->asSequence()->first().valueRef = rri.target
    )->size() > 0
)->size() >= rri.lower
and
self.rootElements->select(r |
    r.oclIsTypeOf(Resource)
and
    r.oclAsType(Resource).extensionValues->select(
        ev:ExtensionAttributeValue |
        ev.extensionAttributeDefinition.name =
            'implementation'
    )->asSequence()->first().valueRef = rri.source
and
    r.resourceParameters->select(rp:ResourceParameter |
        rp.extensionDefinitions->select(
            ed:ExtensionDefinition |
            ed.name = 'ResourceReference'
        )->size() = 1
and
        rp.extensionValues->select(
            ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name =
                'target'
        )->asSequence()->first().valueRef.
        oclAsType(Resource).extensionValues->select(
            ev:ExtensionAttributeValue |

```

```

        ev.extensionAttributeDefinition.name =
            'implementation'
    )->asSequence()->first().valueRef = rri.target
    )->size() > 0
)->size() <= rri.upper
)
)

```

R14)

```

context Definitions
inv parametersCardinality:
self.extensionValues->select(ev:ExtensionAttributeValue |
    ev.extensionAttributeDefinition.name = 'RPIM'
)->asSequence()->first().value.resourceParameters->forall(
    rpi:ResourceParameterImpl |
    self.rootElements->select(r |
        r.ocllIsTypeOf(Resource)
        and
        r.ocllAsType(Resource).extensionValues->select(
            ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name = 'implementation'
        )->asSequence()->first().valueRef = rpi.resource
    )->forall(r:Resource |
        r.resourceParameters->select(rp:ResourceParameter |
            rp.extensionDefinitions->select(
                ed:ExtensionDefinition |
                ed.name = 'ValuedParameter'
            )->size() = 1
            and
            rp.extensionValues->select(
                ev:ExtensionAttributeValue |
                ev.extensionAttributeDefinition.name =
                    'implementation'
            )->asSequence()->first().valueRef = rpi
        )->size() >= rpi.lower
        and
        r.resourceParameters->select(rp:ResourceParameter |
            rp.extensionDefinitions->select(
                ed:ExtensionDefinition |
                ed.name = 'ValuedParameter'
            )->size() = 1
            and
            rp.extensionValues->select(
                ev:ExtensionAttributeValue |
                ev.extensionAttributeDefinition.name =
                    'implementation'
            )->asSequence()->first().valueRef = rpi
        )->size() <= rpi.upper
    )
)
)

```

R15)

```

context Definitions
inv mandatoryResourcePrivileges:

```

```

self.extensionValues->select(ev:ExtensionAttributeValue |
    ev.extensionAttributeDefinition.name = 'RPIM'
)->asSequence()->first() .value.resources->forall(ri:ResourceImpl |
    self.rootElements->select(r |
        r.oclIsTypeOf(Resource)
        and
        r.oclAsType(Resource).extensionValues->select(
            ev:ExtensionAttributeValue |
            ev.extensionAttributeDefinition.name = 'implementation'
        )->asSequence()->first() .valueRef = ri
    )->forall(r:Resource |
        ri.resourcePrivileges->select(mrpi:ResourcePrivilegeImpl |
            mrpi.mandatory
        )->forall(mrpi:ResourcePrivilegeImpl |
            r.extensionValues->select(ev:ExtensionAttributeValue |
                ev.extensionAttributeDefinition.name =
                    'resourcePrivileges'
                ev.valueRef = mrpi
            )->size() = 1
        )
    )
)
)
)

```

R16)

```

context ResourceRole
inv constraintCategoryCardinality:
self.extensionValues->select(ev:ExtensionAttributeValue |
    ev.extensionAttributeDefinition.name = 'implementation'
)->asSequence()->first() .valueRef.
oclAsType(ResourceRoleImpl).constraintCategories->forall(
    cci:ConstraintCategoryImpl |
    self.extensionValues->select(ev:ExtensionAttributeValue |
        ev.extensionAttributeDefinition.name = 'constraints'
        and
        cci.category = ev.value.
        oclAsType(ResolutionConstraint).implementation.category
    )->size() >= ci.lower
    and
    self.extensionValues->select(ev:ExtensionAttributeValue |
        ev.extensionAttributeDefinition.name = 'constraints'
        and
        cci.category = ev.value.oclAsType(ResolutionConstraint).
            implementation.category
    )->size() <= ci.upper
)
)

```

R 17)

```

context Definitions
inv requiredRoles:
self.extensionValues->select(ev:ExtensionAttributeValue |
    ev.extensionAttributeDefinition.name = 'RPIM'
)->asSequence()->first() .value.roles->select(rri:ResourceRoleImpl |
    rri.required
)->forall(rri:ResourceRoleImpl |

```

```

self.rootElements->select(p |
  p.oclIsTypeOf(Process)
)->forAll(p:Process |
  p.flowElements->select(u |
    u.oclIsTypeOf(UserTask)
  )->forAll(u:UserTask |
    u.oclAsType(UserTask).resources->select(
      r:ResourceRole |
      r.extensionValues->select(
        ev:ExtensionAttributeValue |
        ev.extensionAttributeDefinition.name =
          'implementation'
      )->asSequence()->first().valueRef.oclAsType(ResourceRoleImpl)
      =
      rri
    )->size() = 1
  )
)
)

```


10 Bibliografía

Abbott, K. R. & Sunil, K. S., 1994. *Experiences with workflow management: issues for the next generation*. s.l., s.n.

Agrawal, A. y otros, 2010. *Web Services HumanTask (WS-HumanTask), Version 1.1*, s.l.: s.n.

Altuhhova, O., Matulevicius, R. & Ahmed, N., 2012. Towards definition of secure business processes. *Advanced Information Systems Engineering Workshops*, pp. 1-15.

Atluri, V. & Huang, W.-k., 1996. An authorization model for workflows. *Computer Security—ESORICS*, pp. 44-64.

Atluri, V. & Huang, W.-K., 1997. Enforcing mandatory and discretionary security in workflow management systems. *Journal of Computer Security*, pp. 303-339.

Awad, A., Grosskopf, A., Meyer, A. & Weske, M., 2009. *Enabling resource assignment constraints in BPMN*, s.l.: Business Process Technology-Hasso Plattner Institute.

Bertino, E., Ferrari, E. & Atluri, V., 1999. The specification and enforcement of authorization constraints in workflow management systems. *ACM Transactions on Information and System Security (TISSEC)*.

Brucker, A., Hang, I., Luckemeyer, G. & Ruparel, R., 2012. SecureBPMN: Modeling and enforcing access control requirements in business processes. *Proceedings of the 17th ACM symposium on Access Control Models and Technologies*, pp. 123-126.

Cabanillas, C., Resinas, M. & Ruiz-Cortés, A., 2011. Defining and analysing resource assignments in business processes with ral. *Service-Oriented Computing*, pp. 477-486.

Cabanillas, C., Resinas, M. & Ruiz-Cortés, A., 2012. RAL: A high-level user-oriented resource assignment language for business processes. *Business Process Management Workshops*, pp. 50-61.

CBOK, B., 2009. *Guide to the Business Process Management Common Body of Knowledge*. s.l.:s.n.

Clark, J., 1999. Xsl transformations (xslt). *World Wide Web Consortium (W3C)*. URL <http://www.w3.org/TR/xslt>.

Crampton, J. & Khambhammettu, H., 2008. Delegation and satisfiability in workflow systems. *Proceedings of the 13th ACM symposium on Access control models and technologies*, pp. 31-40.

Decker, G., Overdick, H. & Weske, M., 2008. *Oryx - An open modeling platform for the BPM community*. s.l.:Springer.

Dijkman, R., Dumas, M. & Ouyang, C., 2008. Semantics and analysis of business process models in BPMN. *Information and Software Technology*, pp. 1281-1294.

Domingos, D., Rito-Silva, A. & Veiga, P., 2003. Authorization and access control in adaptive workflows. *Computer Security-ESORICS 2003*, pp. 23-38.

Dumas, M., Van der Aalst, W. M. & Ter Hofstede, A. H., 2005. *Process-aware Information Systems: Bridging People and Software through Process Technology*. s.l.:Wiley-Interscience.

Gerhards, M. y otros, 2011. Towards a Security Framework for a WS-HumanTask processor. *Proceedings of the 7th International Conference on Network and Services Management*, pp. 484-488.

Grosskopf, A., 2007. An extended resource information layer for bpmn. *Hasso-Plattner-Institute for IT Systems Engineering, University of Potsdam*.

Harmon, P., 2003. 4. Harmon, P. (2003). *Business Process Change: A Manager's Guide to Improving, Redesigning and Automating Processes*. s.l.:Morgan Kaufmann.

Hollingsworth, D. & Hampshire, U. K., 1993. *Workflow Management Coalition the Workflow Reference Model*. s.l.:s.n.

Holmes, T., Vasko, M. & Dustdar, S., 2008. Viebop: Extending bpel engines with bpel4people. *Parallel, Distributed and Network-Based Processing*, pp. 547-555.

Hummer, W. y otros, 2013. Enforcement of entailment constraints in distributed service-based business processes. *Information and Software Technology*.

Jordan, D. y otros, 2007. Web services business process execution language version 2.0. *OASIS*, Volumen 11, p. 11.

Jordan, D. y otros, 2007. Web services business process execution language version 2.0. *OASIS standard*, Volumen 11, p. 11.

Jouault, F. y otros, 2006. ATL: a QVT-like transformation language. *Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications*, pp. 719-720.

Kloppmann, M. y otros, 2005. *WS-BPEL Extension for People BPEL4People*, s.l.: s.n.

Kumar, A., Van Der Aalst, W. M. & Verbeek, E. M., 2002. Dynamic work distribution in workflow management systems: How to balance quality and performance. *Journal of Management Information Systems*.

Leitner, M., Mangler, J. & Rinderle-Ma, S., 2011. SPRINT-Responsibilities: design and development of security policies in process-aware information systems. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, pp. 4-26.

Leitner, M. & Rinderle-Ma, S., 2014. A systematic review on security in Process-Aware Information Systems--Constitution, challenges, and future directions. *Information and Software Technology*, pp. 273-293.

Link, S., Hoyer, P. & Schuster, T., 2008. Model-driven development of human tasks for workflows. *Software Engineering Advances*, pp. 329-335.

Neuman, B., Clifford & Ts'o, T., 1994. Kerberos: An authentication service for computer networks. *Communications Magazine, IEEE*, 32(9), pp. 33--38.

OMG, 2003. *MDA, MDA Guide v1.0*, s.l.: s.n.

OMG, 2003. *MDA, MDA Guide v1.0*, s.l.: s.n.

OMG, 2005. Meta object facility (mof) 2.0 query/view/transformation specification. *Final Adopted Specification*.

OMG, 2009. Unified Modeling Language (UML) Infrastructure V.2.2 <http://www.omg.org/spec/UML/2.2/Infrastructure>.

OMG, 2009. *Unified Modeling Language (UML) Superstructure, V.2.2.*, <http://www.omg.org/spec/UML/2.2/Infrastructure/>, s.l.: s.n.

OMG, 2013. *Business Process Model and Notation (BPMN), V.2.0.2*, s.l.: s.n.

Peffer, K., Tuunanen, T., Rothenberger, M. & Chatterjee, S., 2007. A design science research methodology for information systems research. *Journal of management information systems*, pp. 45-77.

Pesic, M. & van der Aalst, W. M., 2005. Towards a reference model for work distribution in workflow management systems. *Business Process Reference Models*.

Pesic, M. & van der Aalst, W. M., 2007. Modelling work distribution mechanisms using Colored Petri Nets. *International Journal on Software Tools for Technology Transfer*.

Reijers, H. A., 2003. *Design and Control of Workflow Processes: Business Process Management for the Service Industry*. s.l.:Springer-Verlag.

Reijers, H. A. & van der Aalst, W. M., 2005. The Effectiveness of Workflow Management Systems: Predictions and Lessons Learned. *International Journal of Information Management*.

Rodriguez, A., Fernández-Medina, E. & Piattini, M., 2007. A BPMN extension for the modeling of security requirements in business processes. *IEICE transactions on information and systems*, pp. 745-752.

Russell, N. & van der Aalst, W. M., 2007. Evaluation of the BPEL4People and WS-HumanTask extensions to WS-BPEL 2.0 using the workflow resource patterns. *Bpm center report, Department of Technology Management, Eindhoven University of Technology GPO Box*.

Russell, N., van der Aalst, W. M., ter Hofstede, A. H. & Edmond, D., 2005. Workflow resource patterns: Identification, representation and tool support. *Advanced Information Systems Engineering*.

Russell, N., van der Aalst, W. M., Ter Hofstede, A. H. & Wohed, P., 2006. *On the suitability of UML 2.0 activity diagrams for business process modelling*. s.l., s.n.

Sandhu, R., Coyne, E., Feinstein, H. & Youman, C., 1996. Role-based access control models. *IEEE Computer Society*, pp. 38-47.

Schefer-Wenzl, S., Strembeck, M. & Baumgrass, A., 2012. An approach for consistent delegation in process-aware information systems. *Business Information Systems*.

Selic, B., 2003. The pragmatics of model-driven development. *IEEE software*, pp. 19--25.

Silver, B., 2011. *BPMN Method And Style, With BPMN Implementer's Guide: A Structured Approach For Business Process Modeling*. s.l.:Cody-Cassidy Press.

Stefansen, C., Rajamani, S. & Seshan, P., 2008. *SoftAlloc: A Work Allocation Language with Soft Constraints*. s.l., s.n.

Stroppi, L. J., Chiotti, O. & Villarreal, P. D., 2011. *A BPMN 2.0 Extension to Define the Resource Perspective of Business Process Models*. s.l., s.n.

Stroppi, L. J., Chiotti, O. & Villarreal, P. D., 2011. Extending BPMN 2.0: Method and Tool Support. *Lecture Notes in Business Information Processing*.

Tan, H. & Aalst, W. M., 2006. Implementation of a YAWL work-list handler based on the resource patterns. *International Conference on Computer Supported Cooperative Work in Design*, 1(1), pp. 1-6.

Ter Hofstede, A., 2010. *Modern Business Process Automation: YAWL and its Support Environment*. s.l.:s.n.

Thomas, J., Paci, F., Bertino, E. & Eugster, P., 2007. User tasks and access control over web services. *IEEE International Conference on Web Services*, pp. 60-69.

Thomas, J., Paci, F., Bertino, E. & Eugster, P., 2007. User tasks and access control over web services. *IEEE International Conference on Web Services*, pp. 60-69.

Thomas, R. & Sandhu, R., 1994. Conceptual foundations for a model of task-based authorizations. *Proceedings of the VII Computer Security Foundations Workshop*, pp. 66-79.

Tran, H., Zdun, U. & Dustdar, S., 2008. Modeling human aspects of business processes--A view-based, model-driven approach. *Model Driven Architecture--Foundations and Applications*, pp. 246-261.

van der Aalst, W. M., 1999. Formalization and verification of event-driven process chains. *Information and Software technology*, 41(10), pp. 639-650.

van der Aalst, W. M., Kumar, A. & Verbeek, H., 2003. *Organizational modeling in UML and XML in the context of workflow systems*. s.l., ACM, pp. 603--608.

van der Aalst, W. M., Rosemann, M. & Dumas, M., 2007. Deadline-based escalation in process-aware information systems. *Decision Support Systems*.

Van der Aalst, W. M. & Ter Hofstede, A. H., 2005. YAWL: yet another workflow language. *Information systems*, 30(4), pp. 245-275.

van Der Aalst, W. M., Ter Hofstede, A. H., Kiepuszewski, B. & Barros, A. P., 2003. Workflow patterns. *Distributed and parallel databases*.

Van Der Aalst, W. & Van Hee, K. M., 2004. *Workflow management: models, methods, and systems*. s.l.:MIT Press.

Wainer, J., Barthelmeß, P. & Kumar, A., 2003. W-RBAC—A workflow security model incorporating controlled overriding of constraints. *International Journal of Cooperative Information Systems*, pp. 455--485.

Wainer, J., Kumar, A. & Barthelmeß, P., 2007. DW-RBAC: A formal security model of delegation and revocation in workflow systems. *Information Systems*, pp. 365-384.

Weske, M., 2012. *Business process management: concepts, languages, architectures*. s.l.:Springer.

White, S., 2008. *BPMN modeling and reference guide: understanding and using BPMN*. s.l.:Future Strategies Inc..

Wohed, P. y otros, 2009. Patterns-based evaluation of open source BPM systems: The cases of jBPM, OpenWFE, and Enhydra Shark. *Information and Software Technology*.

Wohed, P. y otros, 2006. *On the suitability of BPMN for business process modelling*. s.l., s.n.

Wolter, C. & Schaad, A., 2007. Modeling of task-based authorization constraints in BPMN. *Business Process Management*, pp. 64-79.

Zeilenga, K., 2006. Lightweight directory access protocol (ldap): Technical specification road map. *IETF*.

Zur Muehlen, M., 1999. *Resource modeling in workflow applications*. s.l., s.n.

Zur Muehlen, M., 2004. Organizational Management in Workflow Applications – Issues and Perspectives. *Information Technology and Management*, pp. 271-291.