

Universidad Central “Marta Abreu” de Las Villas
Facultad de Matemática, Física y Computación



Trabajo para optar por el Título Académico Máster en Ciencia de la Computación

Medidas para la representación visual de modelos de procesos de negocio

Autora:

Lic. Lisbet Hernández Cárdenas

Tutores:

Dra. Isel Moreno Montes de Oca

Dr. Abel Rodríguez Morffi

Diciembre, 2016

Hago constar que el presente trabajo fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la Maestría en Ciencias de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma total como parcial y que además no podrá ser presentado en eventos ni publicado sin previa autorización de la Universidad.

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del tutor

Firma del jefe del Seminario de
Tecnologías de Programación y Sistemas de
Información

DEDICATORIA

A mi mamá.

A mi familia.

AGRADECIMIENTOS

A mi mamá, por estar siempre a mi lado apoyándome y ayudándome en todo lo posible, por ser lo más grande y valioso que tengo en la vida.

A mi familia, por estar siempre pendientes y preocupados por mí.

A Arturo, por estar a mi lado ayudándome en todo lo posible, por su paciencia y apoyo incondicional.

A mis tutores, Dra. Isel Moreno Montes de Oca y Dr. Abel Rodríguez Morffi por sus valiosas recomendaciones, por su esfuerzo y dedicación.

A MSc. Frank Reyes García, por ser parte de este proyecto, por darme su apoyo y brindarme su ayuda en todo lo que estuviera a su alcance.

A todos los profesores que hicieron posible mi formación profesional.

A todos, muchas gracias.

RESUMEN

En los últimos años, los procesos de negocio han sido evaluados y medidos desde diferentes perspectivas. Algunas de las propuestas son adaptaciones de medidas utilizadas en la ingeniería del software debido a la similitud existente entre el proceso de software y el proceso de negocio. Varios autores han propuesto diversos trabajos relacionados con las medidas de calidad, dado que una cuantificación precisa de la calidad constituye un paso importante para mejorar la misma en el proceso de modelación. De acuerdo a estudios realizados, existen diversos tipos de medidas que incluyen: de tamaño, de densidad, de modularidad, de conectividad, de ciclicidad, de concurrencia, entre otros. Todas estas medidas están relacionadas principalmente con la evaluación de la complejidad estructural y general del modelo de proceso de negocio sin tener en cuenta la representación visual, a pesar de que ésta tiene una gran influencia en la comprensión y probabilidad de error de los modelos. El objetivo de este trabajo consiste en proponer medidas de calidad para la representación visual de los modelos de procesos de negocio en BPMN. Los principales resultados obtenidos son: la definición de un conjunto de medidas de calidad para la evaluación de la representación visual de los modelos de procesos de negocio, la validación teórica de las medidas propuestas y la implementación de un módulo con las medidas propuestas en la herramienta SAD_BPMN.

ABSTRACT

In recent years, business processes have been evaluated and measured from different perspectives. Some of the proposals are adaptations of measures used in software engineering due to the similarity between the software process and the business process. Several authors have proposed various works related to quality measures, since a precise quantification of the quality is an important step to improve the same in the process of modeling. According to studies carried out, there are several types of measures that include: size, density, modularity, connectivity, cyclicity, concurrency, among others. All these measures are mainly related to the evaluation of the structural and general complexity of the business process model without regard to the visual representation, although this has a great influence on the understanding and probability of error of the models. The objective of this work is to propose quality measures for the visual representation of business process models in BPMN. The main results obtained are: the definition of a set of quality measures for the evaluation of the visual representation of the business process models, the theoretical validation of the proposed measures and the implementation of a module with the measures proposed in the tool SAD_BPMN.

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
1. ACERCA DE LA EVALUACIÓN DE LA CALIDAD DE LOS MODELOS DE PROCESOS DE NEGOCIO	6
1.1 Calidad en modelos de procesos de negocio	6
1.2 Investigaciones sobre calidad de estilo de etiquetado en modelos de procesos de negocio.....	9
1.3 Investigaciones sobre calidad del diseño visual en modelos de procesos de negocio. ..	13
1.4 Investigaciones relacionadas con calidad del etiquetado y diseño visual en otras áreas.	15
1.5 Evaluación de la calidad.....	18
1.5.1 Medición cuantitativa de modelos de procesos de negocio.	18
1.5.2 Otras formas de evaluación de calidad en modelos de procesos de negocios.....	21
1.6 Método para la definición y validación de las medidas	24
1.6.1 Etapas para la definición de medidas	25
1.6.2 Marco formal de Briand	28
1.7 Consideraciones finales del capítulo	32
2. PROPUESTA DE MEDIDAS PARA LA EVALUACIÓN DE LA REPRESENTACIÓN VISUAL	34
2.1 Contexto de la propuesta	34
2.2 Definición de las medidas	35
2.2.1 Definición de medidas de etiquetado	35
2.2.2 Definición de medidas de diseño visual	37
2.3 Directrices y medidas de representación visual asociadas	39
2.3.1 Directrices prácticas relacionadas con el estilo de etiquetado y medidas asociadas	40
2.3.2 Directrices prácticas relacionadas con el diseño visual y medidas asociadas.....	47
2.3.3 Ejemplo de aplicación de las directrices y las medidas asociadas.	54
2.4 Validación teórica de las medidas propuestas	56
2.4.1 EI y EIF como medidas de tamaño	57
2.4.2 CA como medida de complejidad.....	58
2.4.3 CA como medida de acoplamiento	60
2.5 Consideraciones finales del capítulo	61
3. IMPLEMENTACIÓN DE LAS MEDIDAS PARA LA EVALUACIÓN DE LA REPRESENTACIÓN VISUAL EN SAD_BPMN	63
3.1 Módulo de representación visual en la herramienta SAD_BPMN	63
3.1.1 Diseño de la herramienta.....	65
3.1.2 Implementación de las directrices de representación visual.	68
3.2 Pruebas y evaluación del módulo de representación visual.	71

3.2.1 Pruebas unitarias	72
3.2.2 Análisis de factibilidad de la herramienta SAD_BPMN con el módulo de representación visual.....	75
3.3 Consideraciones finales del capítulo	76
CONCLUSIONES	77
RECOMENDACIONES.....	78
REFERENCIAS BIBLIOGRÁFICAS	79
ANEXO 1: CÓDIGO FUENTE DE LOS ALGORITMOS PROPUESTOS EN LA TESIS...	86

LISTA DE FIGURAS

Figura 1.1 Ejemplo de etiquetado. a) Etiquetado compuerta de decision y flujos de secuencias. b) Etiquetado de un evento de tipo tiempo.	10
Figura 1.2 Ejemplo en EPCs con y sin etiquetas descriptivas.	11
Figura 1.3 Dos formas de representar el grafo de un cubo.	16
Figura 1.4 Estrategia general para dibujos de rejillas ortogonales. (a) Grafo dado. (b) Primera fase (<i>planarization</i>). (c) Segunda fase (<i>orthogonalization</i>). (d) Tercera fase (<i>compaction</i>).	16
Figura 1.5 Uso de la taxonomía. (a) Diagrama que presenta cruces de arcos y alto número de curvas en los arcos. (b) Diagrama resultante después de aplicar algunas categorías secundarias propuestas en la taxonomía.	17
Figura 2.1 Uso de etiquetas con estilo <i>verbo-sustantivo</i>	41
Figura 2.2 Tipos de tareas no especificados.	42
Figura 2.3 Tipos de tareas especificados.	42
Figura 2.4 Modelo que presenta actividades sin etiquetar.	43
Figura 2.5 Modelo con eventos sin etiquetar.	44
Figura 2.6 Uso incorrecto del etiquetado de compuertas y flujo de secuencias.	45
Figura 2.7 Uso correcto del etiquetado de compuertas y flujo de secuencias.	46
Figura 2.8 Modelo con presencia de etiquetas cortas.	47
Figura 2.9 Modelo con presencia de cruces y curvas en los arcos.	48
Figura 2.10 Modelo con presencia de solapamiento.	50
Figura 2.11 Modelo con flujo de secuencia inconsistente evitable.	51
Figura 2.12 Modelo que no presenta problema asociado a la dirección de los flujos de secuencia.	51
Figura 2.13 Modelo con presencia de tamaño no uniforme.	53
Figura 2.14 Modelo BPMN: Proceso de denuncia.	55
Figura 2.15 Modelo BPMN para la demostración de la propiedad 4 de complejidad.	59
Figura 2.16 Relación entre la representación visual y los atributos de calidad.	61

Figura 3.1 Interfaz gráfica de SAD_BPMN.	64
Figura 3.2 Diagrama de paquetes del Módulo de representación visual	65
Figura 3.3 Diagrama del paquete Directrices	66
Figura 3.4 Diagrama del paquete <i>bpmn_data</i>	67
Figura 3.5 Diagrama del paquete <i>visual</i>	67
Figura 3.6 Interfaz visual de la herramienta SAD_BPMN con el módulo representación visual.	68
Figura 3.7 Método diseñado para la prueba de la directriz “ <i>Etiquete las actividades</i> ”.	73
Figura 3.8 Resultado satisfactorio de la prueba para la directriz “ <i>Etiquete las actividades</i> ”. .	73
Figura 3.9 Método diseñado para la prueba de la directriz “ <i>Tipo de tareas no especificados</i> ”.	74
Figura 3.10 Resultado satisfactorio de la prueba para la directriz “ <i>Tipo de tareas no especificados</i> ”.	74
Figura 3.11 Método diseñado para la prueba de la directriz “ <i>Alto número de cruces de flujos de secuencia</i> ”.	75
Figura 3.12 Resultado satisfactorio de la prueba para la directriz “ <i>Alto número de cruces de flujos de secuencia</i> ”.	75

LISTA DE TABLAS

Tabla 2.1 Medidas bases – Estilo de etiquetado.	36
Tabla 2.2 Medidas derivadas – Estilo de etiquetado.	37
Tabla 2.3 Medidas bases – Diseño visual.	38
Tabla 2.4 Medidas derivadas – Diseño visual.	39
Tabla 2.5 Valor de medidas y problema asociado – Estilo de etiquetado.	55
Tabla 2.6 Valor de medidas y problema asociado – Diseño visual.	56

INTRODUCCIÓN

El uso adecuado de las Tecnologías de la Información y la Comunicación (TIC) ha pasado a ser un elemento estratégico en términos de competitividad para las organizaciones dentro de un entorno de globalización. Esto implica acciones de análisis, diseño, implementación y una readaptación de los procesos organizacionales. Por esta razón es relevante comunicar, entender y manejar el dominio de la información y los procesos, para comprender el fin y realizar una correcta gestión de requerimientos basada en el entendimiento de las actividades de las organizaciones (Kalpic and Bernus, 2002).

Múltiples organizaciones se centran en buscar una forma de orientar la gestión de sus procesos de negocio en la actualidad. Esta idea parte del concepto de gestión de procesos de negocio (Business Process Management; BPM) que constituye una forma de controlar y gobernar los procesos de negocio para cualquier organización con el objetivo de mejorar su rendimiento empresarial. Generalmente las organizaciones necesitan controlar y mejorar sus procesos para aumentar sus ingresos totales, la satisfacción del cliente, o para asegurar el cumplimiento normativo (Dumas et al., 2013).

Para el diseño y perfeccionamiento de los sistemas de información (SI) que respaldan las organizaciones y los procesos de negocio que tienen lugar en ellas, los modelos de procesos de negocio juegan un papel importante, por lo que su aplicación es extensa en la práctica de la gestión de procesos de negocio que incluye métodos, herramientas y tecnologías utilizados para diseñar, representar, analizar y controlar procesos de negocio operacionales (Garimella and Lees, 2008). Dentro de la variedad de lenguajes para representar procesos de negocio, BPMN (Flowers and Edeki, 2013, OMG, 2011) se considera el estándar y uno de los lenguajes más relevantes (Cadavid et al., 2008). Su objetivo principal es soportar la gestión de una forma comprensible por todos los usuarios que participan en el proceso (OMG, 2011). Esta forma unificada de representar los procesos facilita que los comportamientos y decisiones puedan ser expresados de manera gráfica y simple en una organización. Por esto, la calidad de los modelos de procesos de negocio es de gran importancia (Moody, 2005).

El ciclo de vida de los procesos de negocio abarca tres fases principales (Hollingsworth, 2004): 1) definición y modelado, 2) implementación y, 3) análisis y mejora. En la actualidad, mediante

BPM se intenta proporcionar el soporte necesario a todas las fases del ciclo de vida de los procesos de negocio. Este soporte abarca tanto cambios incrementales, como cambios de nivel llevados a cabo en una reingeniería de procesos, haciéndolo en un nivel de diseño de negocios y no a un nivel de implementación técnica.

En la fase de definición y modelado del ciclo de vida de los modelos de procesos de negocio, la etapa de modelado es fundamental para el logro de los objetivos de la empresa. Además, es una de las fases más afectadas por los cambios del entorno por lo que requiere constantes adecuaciones (Smith and Fingar, 2003). Esta fase incluye la definición de medidas clave para los procesos de negocio que proporcionen el soporte adecuado a los analistas de negocios, de tal modo que puedan reestructurar rápidamente los procesos en respuesta a las presiones competitivas u oportunidades del negocio.

En los últimos años, los procesos de negocio han sido evaluados y medidos desde diferentes perspectivas. Algunas de las propuestas son adaptaciones de medidas utilizadas en la ingeniería del software debido a la similitud existente entre el proceso de software y el proceso de negocio (Rolón, 2009). Los principales objetivos planteados para la medición de los modelos de procesos de negocio en estas propuestas han sido: la definición de medidas de calidad que ayuden a los diseñadores a identificar problemas de calidad y oportunidades de rediseño, así como medir los efectos en la calidad del modelo (Weber and Reichert, 2008), medidas para medir la probabilidad de error de los modelos (Mendling and Neumann, 2007), medidas para evaluar la complejidad estructural de los procesos de negocio desde la perspectiva de flujos de control (Cardoso, 2006), entre otros.

Varios autores han propuesto diversos trabajos relacionados con las medidas de calidad, dado que una cuantificación precisa de la calidad constituye un paso importante para mejorar la misma en el proceso de modelación (Sanchez-Gonzalez et al., 2010b). El uso de la información obtenida a partir de estas medidas hace posible para las organizaciones aprender del pasado con el objeto de mejorar el desempeño y lograr mejores predicciones con el tiempo (Sanchez-Gonzalez et al., 2010a). El diseño de medidas está relacionado con las propiedades estáticas de los procesos de negocio y se definen sobre el modelo de procesos de negocio mientras se diseña.

De acuerdo a estudios realizados por varios autores (Mendling, 2008a, Gruhn and Laue, 2006, Cardoso, 2006, Reijers and Vanderfeesten, 2004), existen diversos tipos de medidas que

incluyen medidas de tamaño, de densidad, de modularidad, de conectividad, de ciclicidad, concurrencia, entre otros. Todas estas medidas están relacionadas principalmente con la evaluación de la complejidad estructural y general del modelo de proceso sin tener en cuenta la representación visual, a pesar de que estas tienen una gran influencia en la comprensión y probabilidad de error de los modelos. Por otra parte, en el laboratorio de Tecnologías de Programación y Sistemas de Información perteneciente al Centro de Investigaciones de la Informática (CII), ubicado en la Universidad Central “Marta Abreu” de Las Villas, se realizan investigaciones relacionadas con la calidad y la modelación de procesos de negocios. En dicho centro se desarrolló la herramienta SAD_BPMN para evaluar modelos de procesos de negocios tomando como base medidas de calidad relacionadas con la complejidad de los modelos (estructural y general), pero no incluye la evaluación de la representación visual. La problemática descrita anteriormente conduce a plantear que el **problema de investigación** se enmarca en la necesidad de cuantificar los problemas de calidad asociados con la representación visual de modelos de procesos de negocio.

El **objetivo general** de esta investigación consiste en proponer medidas de calidad para la representación visual de los modelos de procesos de negocio en BPMN.

Para lograr el cumplimiento del objetivo planteado se proponen los siguientes **objetivos específicos**:

1. Identificar los elementos a tener en cuenta para evaluar la representación visual de los modelos de procesos de negocio en BPMN.
2. Definir medidas de calidad para la evaluación de la representación visual de los modelos de procesos de negocio a partir de los elementos visuales identificados.
3. Validar teóricamente las medidas propuestas.
4. Implementar un módulo con las medidas propuestas en la herramienta SAD_BPMN.

Las **preguntas de investigación** planteadas son:

1. ¿Qué aspectos deben considerarse para la formulación de medidas de calidad, que permitan evaluar la representación visual de los modelos de procesos de negocio?
2. ¿Cómo lograr la integración de las medidas propuestas en la herramienta SAD_BPMN para la evaluación de la calidad de los modelos de procesos de negocio?

3. ¿Cómo realizar una validación teórica de las medidas?

El **valor práctico** del trabajo está dado por:

- El módulo de representación visual en la herramienta SAD_BPMN dispone de los métodos para evaluar los modelos y de esta forma facilita a los investigadores y docentes revisar sus modelos de procesos de negocio tomando como base las medidas de calidad propuestas.

La tesis está estructurada en tres capítulos. A continuación, se describe esta organización.

- **Capítulo 1: Acerca de la evaluación de la calidad de los modelos de procesos de negocio.** En este capítulo se ofrece una perspectiva sobre la calidad de los modelos de procesos de negocio. Además, se presentan investigaciones relacionadas con la representación visual, tanto de modelos de procesos de negocios como en otras áreas. De igual forma se tratan diferentes formas de evaluar la calidad de los modelos de procesos de negocios.
- **Capítulo 2: Propuesta de medidas para la evaluación de la representación visual.** En este capítulo se presenta el contexto y la justificación de la propuesta de medidas para la evaluación de procesos de negocio a partir del modelo que los representa en un nivel conceptual, incluyendo la definición teórica y formal de las medidas y su validación teórica.
- **Capítulo 3: Implementación de las medidas para la evaluación de la representación visual de modelos de procesos de negocio.** En este capítulo se presentan algunos aspectos de implementación de las directrices de etiquetado y diseño visual en la herramienta SAD_BPMN, así como el diagrama de paquetes y clases de diseño necesarias para el desarrollo del módulo de representación visual.

Este documento culmina con las conclusiones, recomendaciones, referencias bibliográficas y anexos.

ACERCA DE LA EVALUACIÓN DE LA CALIDAD DE LOS
MODELOS DE PROCESOS DE NEGOCIO

1. ACERCA DE LA EVALUACIÓN DE LA CALIDAD DE LOS MODELOS DE PROCESOS DE NEGOCIO

La modelación de procesos de negocio ha adquirido mayor importancia en los últimos años (Davies et al., 2006). Esta popularidad se ha visto propiciada por el surgimiento de nuevas y diferentes propuestas para la gestión de procesos, lo que ha traído como consecuencia que se haya incrementado considerablemente la cantidad y la variedad de usuarios y diseñadores de modelos y de los propósitos para los cuales son usados los modelos de procesos (Becker-Kornstaedt and Belau, 2000). Esto ha llevado a muchos autores a realizar sus trabajos de investigación en el área relacionada con la calidad de los modelos de procesos de negocio (Mendling, 2010).

En el presente capítulo se ofrece una perspectiva sobre la calidad de los modelos de procesos de negocios. Además, se presentan investigaciones relacionadas con la representación visual, tanto de modelos de procesos de negocios como en otras áreas. De igual manera se tratan diferentes formas de evaluar la calidad de los modelos de procesos de negocios.

1.1 Calidad en modelos de procesos de negocio

Los modelos de procesos de negocio son de gran importancia en la actualidad y juegan un papel fundamental no solo en el campo de la gestión de los procesos de negocio, sino también en la etapa de diseño y análisis del ciclo de vida del desarrollo del software (Gruhn and Laue, 2009). A pesar de sus múltiples beneficios, esta etapa de diseño y análisis es raramente practicada, y cuando se hace, no se realiza con la calidad requerida (Dufresne and Martin, 2003). Esto hace que sea una preocupación creciente en el mundo empresarial, ya que muchas veces se confían las iniciativas de modelación a personas con escasa experiencia en el tema. Además, aunque las primeras etapas en el desarrollo del software no reporten costos significativos, con el paso del tiempo estos crecen exponencialmente, de ahí que sea imprescindible asegurar una calidad óptima en los modelos de procesos de negocio.

Si bien existen diversas definiciones de calidad en los distintos campos de investigación, no se ha encontrado una definición consensuada respecto a qué es la calidad de los modelos conceptuales. Al respecto en (Moody, 2005) se propone como definición “la totalidad de los rasgos y características de un modelo conceptual que influyen en su habilidad de satisfacer las

necesidades implícitas o declaradas”. En este contexto, un modelo de mayor calidad dará lugar a un SI de mayor calidad. Por lo tanto, la calidad del modelo de proceso de negocio puede afectar tanto la eficiencia (tiempo, costo, esfuerzo), como la efectividad (calidad de los resultados) del desarrollo de los SI.

La evolución y mejora de los procesos es muy importante no solo desde el punto de vista empresarial, sino también desde el punto de vista de sistemas, al considerarse la fase de modelado del proceso de negocio como una parte esencial de cualquier proyecto de desarrollo de software. Durante el desarrollo del modelo de proceso de negocio, se deben tener en cuenta diversos aspectos o propiedades que en su conjunto conllevan a la obtención de un modelo de calidad.

La norma ISO/IEC 9126 (ISO/IEC, 2001) define la calidad del producto de software como la totalidad de las características relacionadas con su habilidad para satisfacer necesidades establecidas o implicadas. El modelo de calidad de este estándar categoriza los atributos de calidad en seis características: funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad. A su vez cada una de estas características se subdivide en sub-características que pueden ser medidas por métricas internas o externas.

Analizando la relación existente entre los modelos de procesos de negocio y los modelos conceptuales de producto de software, se ha considerado que el modelo de calidad de un producto de software establecido en el estándar ISO 9126, se puede utilizar para caracterizar los aspectos de calidad de los modelos de procesos de negocio (Rolón, 2009).

Diversos trabajos de investigación sobre la calidad de los modelos de procesos de negocio contribuyen a la evolución del conocimiento en el área. Al mismo tiempo, han evolucionado las investigaciones sobre la calidad del modelado conceptual. En la literatura se diferencian fundamentalmente cuatro líneas de investigación relacionadas con particularidades de calidad para la modelación conceptual de procesos (Mendling et al., 2010c):

- Marcos de trabajo de calidad: Los marcos de trabajos intentan producir orden y esclarecimiento a la representación de la calidad en los modelos conceptuales y a la calidad del proceso de modelación conceptual (Lindland et al., 1994a), definiendo los niveles de calidad que un determinado modelo debe cumplir (Pérez-Santiago, 2014).

- Medidas de calidad: las medidas de calidad representan un componente esencial para cuantificar de forma precisa la calidad de los diagramas, lo que a su vez es un paso importante hacia la mejora de calidad interna de los modelos de procesos de negocio (Sanchez-Gonzalez et al., 2010b).
- Estudios empíricos relativos a las técnicas de modelación: investigan entre otros, el uso de lenguajes de modelación como UML (Agarwal, 2003), la comprensión de los modelos de procesos en el nivel del lenguaje de modelación (Sarshar, 2005), analizan el lenguaje BPMN (Recker et al., 2007), proponen extensiones a este (Magnani and Montesi, 2007) y hacen comparaciones entre los enfoques de modelación (He et al., 2010).
- Propuestas para mejorar la práctica de modelación: las líneas de investigación mencionadas anteriormente poseen beneficios significativos, pero pueden resultar demasiado abstractos para ser aplicados tanto por académicos como por modeladores. Sin embargo, las directrices prácticas representan una guía determinante para la calidad de los diagramas porque están dirigidas a los modeladores y se encuentran a un bajo nivel de abstracción, lo que posibilita su adopción en la práctica (Moreno Montes de Oca et al., 2014, Moreno-Montes de Oca and Snoeck, 2014).

Según (Multamäki, 2002) los objetivos de la modelación de procesos de negocio son: a) mejorar la comprensión de una situación para que pueda comunicarse entre los involucrados del negocio y b) usar el proceso como una herramienta para lograr las metas de un proyecto de desarrollo de procesos. Para el logro de estos objetivos es importante que los modelos de negocio sean de alta calidad. Sin embargo, varios estudios (Mendling, 2009, Mendling et al., 2006) muestran que varios modelos de procesos de negocio contienen errores, como por ejemplo errores sintácticos. En (Nelson et al., 2012) se definen cuatro dimensiones de calidad:

- Calidad sintáctica: se refiere a cuán bien la representación corresponde a la extensión del lenguaje. Este aspecto está relacionado con aspectos puramente estructurales del modelo. Relaciona el modelo con el lenguaje de modelado describiendo las relaciones entre las estructuras del lenguaje sin considerar su significado.
- Calidad semántica: se refiere al significado intrínseco del modelo, es decir, cuán bien la representación corresponde al dominio. La calidad semántica puede ser descompuesta

en dos principios básicos: validez y completitud. La validez quiere decir que todas las declaraciones en el modelo son correctas y tienen importancia para el problema; la completitud quiere decir que el modelo contiene todas las declaraciones pertinentes que estarían en lo correcto.

- Calidad pragmática: se refiere a la comprensión de la representación física final por parte de los usuarios, es decir, qué tan bien un usuario comprende el contenido del modelo. Es la correspondencia entre la interpretación del usuario y el propio modelo. La calidad pragmática relaciona hasta qué punto los usuarios entienden completamente y con precisión las declaraciones en el modelo que son pertinentes para el propio usuario.
- Calidad empírica: es una medida de legibilidad en una representación conceptual que está en función de la representación misma. En este sentido, modelos que cuenten con un nivel de calidad empírica alto serán más sostenibles y comprensibles que una representación similar que cuenta con calidad empírica baja.

Se considera la comprensibilidad de los modelos de procesos de negocio como uno de los principales determinantes para la probabilidad del error. Esto se basa en que los modelos de procesos de negocio son contruidos por modeladores humanos y su plan está sujeto a una racionalidad de errores limitada (Gruhn and Laue, 2006). Además, la comprensibilidad de un modelo por un usuario está dada por una diversidad de factores, los cuales influyen en el entendimiento de dicho modelo por usuarios que en ocasiones no tienen un vasto conocimiento acerca del proceso. Estos factores están relacionados con la complejidad general (tamaño y morfología) y la representación visual (diseño visual y estilo de etiquetas). La representación visual está relacionada puramente con la estructura visual de los modelos y pueden realizarse modificaciones a los mismos sin cambiar su semántica. Según los expertos del dominio, una mayor calidad en cuanto a la representación visual mejora la legibilidad general (calidad empírica y pragmática) de los modelos de procesos e implica grandes mejoras para la comprensión de los mismos (Mendling, 2009, Mendling et al., 2006).

1.2 Investigaciones sobre calidad de estilo de etiquetado en modelos de procesos de negocio.

El modelado de procesos de negocio es una parte esencial de la gestión de procesos en las grandes empresas. Varias de ellas diseñan y mantienen cientos de modelos para capturar sus

operaciones (Rosemann, 2006). La calidad de los modelos de procesos ha sido enfocada desde diferentes ángulos, incluyendo la verificación, la probabilidad de error y la comprensión (Mendling, 2008b, Mendling et al., 2007).

Para lograr una mayor calidad, se debe tener en cuenta el estilo de las etiquetas. Una etiqueta¹ es una palabra o frase descriptiva, que permite identificar de forma única, indicar o dar a conocer el contenido de un elemento. Típicamente una etiqueta trabaja sobre elementos individuales del modelo. Una etiqueta de una actividad captura una acción y un objeto del negocio, tal que la acción es realizada, por ejemplo “validar dirección” o “crear orden”. Por otra parte, la etiqueta de los *pools*² indica el nombre del proceso correspondiente a dicho *pool*. Otros elementos del modelo que también se etiquetan son las compuertas de decisión de tipo OR y XOR, así como los flujos de secuencias que salen de estas compuertas. Las etiquetas de las compuertas deben indicar claramente la decisión o condición evaluada cuando aplique. Este tipo de etiqueta debe ser un nombre compuesto por un verbo, un sustantivo y entre signos de interrogación para identificar lo que se está evaluando o se pueden utilizar preguntas para aclarar la decisión en cuestión, mientras que las etiquetas de los flujos de secuencias deben indicar la condición relacionada. De forma similar los eventos también son etiquetados, principalmente los de tiempo. Las etiquetas de los eventos de tiempo indican su duración. En la Figura 1.1 se muestra un ejemplo del etiquetado de las compuertas de decisión y los flujos de salidas asociados (Figura 1.1a), así como algunos ejemplos de eventos de tiempo etiquetados (Figura 1.1b).

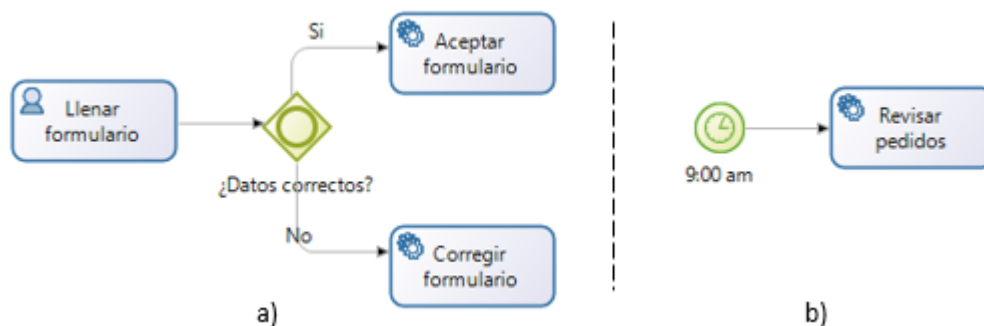


Figura 1.1 Ejemplo de etiquetado. a) Etiquetado compuerta de decision y flujos de secuencias. b) Etiquetado de un evento de tipo tiempo.

¹ Una acepción del término etiqueta, según la RAE.

² Traducido al español: contenedor

Actualmente, varios trabajos tratan la evaluación de la calidad del etiquetado como un aspecto de la calidad de los modelos de procesos de negocios (Fettke et al., 2012, Mendling et al., 2010b, Gruhn and Laue, 2011). Para obtener valiosos resultados para mejorar la calidad de los modelos en la práctica, las medidas aplicadas tienen que cubrir los aspectos multidimensionales del proceso de modelación. De esta forma en (Fettke et al., 2012), mediante un ejemplo demuestran un enfoque diferente para interpretar los resultados de otros ejemplos. En la Figura 1.2 se muestra un ejemplo en el lenguaje de modelación EPCs (Event-Driven Process Chains). Primeramente, las cadenas de caracteres usadas para los descriptores tienen una influencia significativa en lo que se entiende como un modelo conceptual. Este es el propósito intencional de las etiquetas descriptivas. En segundo lugar, las etiquetas particulares pueden ser interpretadas de una forma muy diferente dependiendo de la perspectiva. Lo que se interpreta como una orden de compra en una empresa es una orden de trabajo en una compañía. Estas consideraciones muestran que el contexto también tiene una influencia significativa en la comprensión de un modelo conceptual como objeto de investigación.

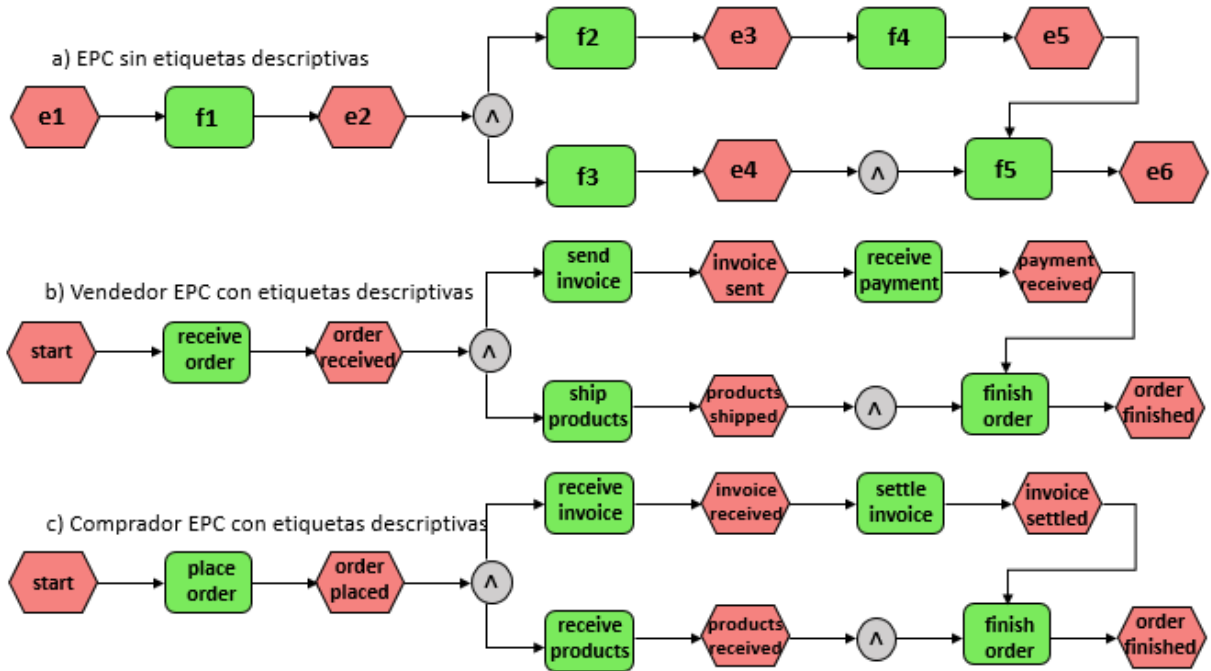


Figura 1.2 Ejemplo en EPCs con y sin etiquetas descriptivas.

En línea con (Leopold et al., 2011) existen varios enfoques para determinar la calidad del etiquetado en los modelos. Como muestran los estudios, los trabajos se enfocan principalmente

en el estilo de etiquetado. Sintácticamente, el estilo verbo-objeto ha probado ser superior en lo que se refiere a la comprensión humana (Mendling et al., 2010b).

Como se mencionó anteriormente, en (Mendling et al., 2010b) se realiza una investigación sobre el uso de etiquetas del tipo verbo-objeto y se identifican tres estilos de etiquetado que difieren en el grado de ambigüedad, los cuales se muestran a continuación:

- Etiquetas del tipo verbo-objeto: La mayoría de las etiquetas del tipo verbo-objeto parecen intuitivamente comprensibles para los modeladores y usuarios. Sin embargo, hay algunos casos que son ambiguas desde el punto de vista gramatical. Por ejemplo, en el idioma Inglés, la misma palabra puede a la vez ser un sustantivo y un verbo. Considérense, por ejemplo las siguientes etiquetas: *Measure Processing*, *Export License Check* y *Process Cost Planning*. Estas tienen en común que la primera palabra puede ser un verbo, pero la lectura como un objeto que describe una acción también es posible. En el caso de la etiqueta *Measure Processing* podría referirse al procesamiento de una medida o a la medición de un procesamiento. La misma observación es válida para las otras etiquetas. Algunas de estas ambigüedades pueden resolverse teniendo en cuenta la información del contexto. Si el estilo de verbo-objeto se utiliza constantemente como un estándar en todo un modelo de proceso, sería claro para interpretar el primer término como un verbo.
- Etiquetas del tipo acción-sustantivo: Con respecto a las etiquetas de tipo acción-sustantivo, algunas de ellas pueden ser fácilmente interpretadas, pero también pueden existir casos de ambigüedad gramatical. Considérense, por ejemplo, la siguiente etiqueta en idioma Inglés *Notification Printing*. Una vez más hay dos interpretaciones posibles: se imprime una notificación o a alguien se le notifica de un trabajo de impresión. Por otra parte el verbo podría haber sido olvidado por el modelador, esta interpretación es usada en los casos en que la etiqueta acción-sustantivo podría ser un objeto, como *Order*, que puede referirse tanto a una acción como a un objeto. Este tipo de ambigüedad es llamada como ambigüedad acción-objeto. Con el uso de una etiqueta de tipo verbo-objeto se evitaría el problema de la ambigüedad acción-objeto y la necesidad de tener que inferir un objeto para establecer el significado apropiado.

- Otros tipos de etiquetas: Las etiquetas que están en esta categoría apuntan claramente a un objeto de negocio específico, de tal manera que un verbo podría deducirse del contexto. Sin embargo, también hay etiquetas de actividades como *DEÜV* y *Jamsostek* que son difíciles de entender por completo, por ejemplo la primera se refiere a la regulación alemana para el almacenamiento y transmisión de datos y la segunda al sistema de seguridad social de Indonesia. De esta forma las etiquetas de este tipo requieren información clara del contexto, pues una inferencia de la acción a realizar podría ser una tarea problemática debido a la ocurrencia de ambigüedad verbo-inferencia, es decir, el problema de inferir a partir del contexto de la etiqueta el tipo de acción que debe realizarse como parte de la tarea del proceso considerado.

Basados en estos estilos se diseña un estudio (Mendling et al., 2010b) para probar la conexión hipotética entre el estilo de etiquetado y la calidad. Los resultados que se obtuvieron sugieren que utilizar el estilo verbo-objeto es superior a otras formas de construir las etiquetas.

1.3 Investigaciones sobre calidad del diseño visual en modelos de procesos de negocio.

El diseño visual considera posiciones relativas de los elementos (relacionan elementos) y la organización general del modelo y se refiere al formato de la estructura del grafo primario, por ejemplo, minimizar las curvas en el flujo de secuencia. Varios trabajos relacionados con la calidad y la legibilidad de los modelos de procesos de negocio se enfocan en el diseño visual de los mismos.

En (Effinger et al., 2009) se propone una herramienta que permite integrar varios conceptos que fueron desarrollados y adoptados por los autores para el uso en modelos BPMN. Los enfoques integrados comprenden las tareas de encontrar un esquema automático para un modelo en BPMN, aumentar la legibilidad de los modelos y un método para la división automática de modelos complejos. Para el desarrollo de esta herramienta se tuvieron en cuenta algunos criterios para mejorar la legibilidad de modelos BPMN, tales como:

- División: Considerar las particiones, por ejemplo (contraer/expandir) los *pools* y los *swimlanes*³.

³ Traducido al español: sendas o carriles.

- Bimodalidad: Objetos conectores de entrada y salida que entran a una compuerta por lados diferentes. Si una compuerta tiene dos objetos conectores de salida, estos deben conectarse en lados opuestos de la compuerta.
- Ortogonal: Objetos conectores que están dibujados como una secuencia de segmentos horizontales y verticales. Maximizar el número de objetos conectores de forma ortogonal.
- Flujo: Objetos conectores que están dibujados monolíticamente en una dirección prescrita. Maximizar el número de objetos conectores que respetan la dirección del flujo de trabajo.

De manera similar en (Effinger et al., 2010) se analizan las preferencias de grupos de usuarios al modelar con BPMN. Teniendo en cuenta el estándar BPMN como una notación basada en grafo se consideraron como necesarios los siguientes criterios:

- Minimizar el número de cruces de arcos.
- Minimizar el área de dibujo.
- Minimizar el número de curvas en los arcos.
- Minimizar el número de solapamientos de arcos y elementos.
- Maximizar el número de arcos dibujados ortogonalmente.
- Maximizar el número de arcos que respetan el flujo de dirección.
- Minimizar los objetos de flujo que tienen diferentes tamaños.
- Las particiones deben ser consideradas, por ejemplo, los *pools* y los *swimlanes*.

Estos criterios definidos por los autores pueden ser considerados para adaptaciones de herramientas utilizadas por diferentes usuarios de BPMN.

En varios trabajos de investigación se han propuesto directrices prácticas cuyo objetivo es aconsejar a los modeladores para lograr una mejor calidad de los modelos de procesos de negocios. De estos estudios se extrajo un amplio conjunto de directrices prácticas para la modelación de procesos de negocio. Las directrices coleccionadas pueden agruparse de acuerdo a lo que refieren en dos grupos fundamentales: directrices que tratan sobre la complejidad de los modelos y directrices que tratan sobre el diseño visual de los mismos (Moreno-Montes de Oca et al., 2015).

De manera general, para lograr una buena calidad pragmática y empírica en los modelos, primeramente se debe mejorar el diseño visual del modelo, pues este tiene una alta incidencia sobre la comprensión del mismo según (Mendling and Strembeck, 2008, Schrepfer et al., 2009). Ello permitirá comprender mejor el modelo después de este paso y detectar otros problemas. Seguidamente, se solucionan los problemas relacionados con el estilo de etiquetado, los cuales se incluyen en la dimensión de representación visual junto con los problemas de diseño.

1.4 Investigaciones relacionadas con calidad del etiquetado y diseño visual en otras áreas.

Mejorar la representación visual, no solo es un tema tratado en investigaciones relacionadas con modelos de procesos de negocios. En la literatura también se pueden encontrar diversas propuestas para la evaluación y mejora de la representación visual en otras áreas.

Varios problemas de presentación de datos involucran el dibujo de un grafo. Sin embargo, en casi todas las aplicaciones de presentación de datos, la utilidad de un dibujo de un grafo depende de su legibilidad, es decir, la capacidad de transmitir el significado del diagrama de forma clara y rápida (Johnson, 1983). Existen varios criterios relacionados con la representación visual de los grafos que permiten obtener un dibujo atractivo de un grafo general no dirigido. Entre estos criterios se encuentran:

- Evitar el cruce de arcos
- Evitar curvas en los arcos
- Mantener una longitud uniforme en los arcos
- Distribuir los vértices uniformemente

En general, los problemas de optimización asociados con estos criterios son NP-duros. Varios resultados relacionados con estos criterios y su complejidad se informan en (Johnson, 1983, Orlin, 1985).

Uno de los criterios para lograr una buena representación visual es minimizar el número de cruces entre los arcos (WADDLE, 2001, FORSTER). Debe tenerse en cuenta que el uso de estos criterios es subjetivo y pueden necesitar adaptaciones para satisfacer preferencias personales, tradiciones y cultura. Por ejemplo, el grafo de un cubo tradicionalmente es dibujado con cruces en los arcos, como se muestra en la Figura 1.3.

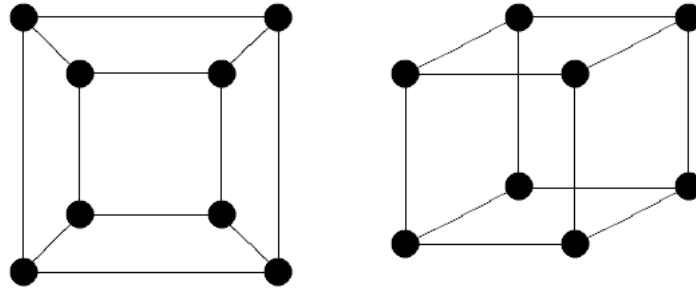


Figura 1.3 Dos formas de representar el grafo de un cubo.

Un enfoque comprensivo para la construcción de dibujos de rejillas ortogonales, basado en varios algoritmos de grafos, se presentan en (C. Batini, 1984, Tamassia et al., 1988). En este enfoque la construcción del dibujo es especificado en tres fases, como se muestra en la Figura 1.4. La primera fase (*planarization*), determina la topología del dibujo (Figura 1.4(b)). La segunda fase (*orthogonalization*), computa una forma ortogonal para el dibujo (Figura 1.4(c)). La tercera fase (*compaction*), produce el dibujo final (Figura 1.4(d)). Este enfoque permite un tratamiento homogéneo de una gama amplia de representaciones diagramáticas, criterios y restricciones.

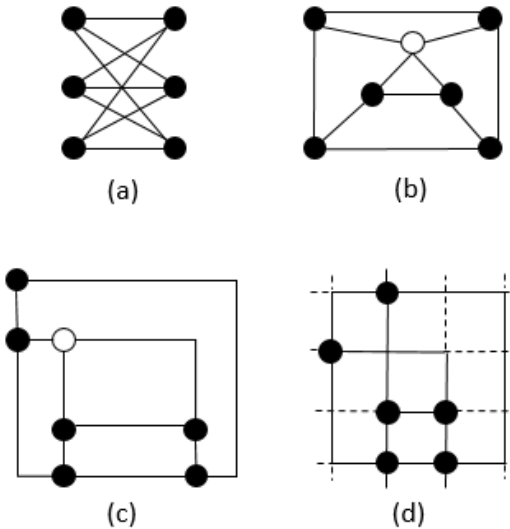


Figura 1.4 Estrategia general para dibujos de rejillas ortogonales. (a) Grafo dado. (b) Primera fase (*planarization*). (c) Segunda fase (*orthogonalization*). (d) Tercera fase (*compaction*).

En (Eichelberger, 2005) se presenta un estudio con el fin de mejorar la representación visual, principalmente el diseño visual, de modelos representados como grafos. Para ello se propone un conjunto de medidas que permiten evaluar aspectos visuales de los grafos como: cruces de

aristas, tamaño de nodos y aristas, solapamiento de los elementos del grafo, número de curvas en las aristas y flujo uniforme.

Con el creciente éxito de lenguajes de especificación visual como UML para el modelado y manejo de ingeniería de software, directrices visuales son necesarias para regular la presentación y el intercambio de modelos con respecto a la comunicación humana, la comprensión y la legibilidad. En (Eichelberger and Schmid, 2009) se introduce una nueva y abarcadora taxonomía de directrices visuales que capturan la calidad del diseño visual de diagramas de clases en UML. Estas directrices son propuestas como un marco de trabajo para mejorar la calidad del diseño visual y la comprensión de los diagramas de clases en UML. La taxonomía propuesta consiste de cuatro niveles de categorías primarios y varios niveles de categorías secundarios: reglas estructurales, reglas de diseño, reglas de esquema y reglas de aplicación de dominio específico. Entre estos cuatro niveles el más significativo para este trabajo es el nivel correspondiente a las reglas de esquema. En este nivel se encuentran las siguientes categorías secundarias:

- Los arcos deben mantener un flujo uniforme.
- Minimizar el número de curvas en los arcos.
- Evitar el solapamiento entre elementos, entre arcos, así como entre elementos y arcos.
- Minimizar el área del dibujo.
- Minimizar o evitar un alto número de cruces de arcos.
- Proporcionar una apariencia uniforme del dibujo.

La Figura 1.5 muestra un ejemplo del uso de esta taxonomía teniendo en cuenta el número de curvas en los arcos y el número de cruces de arcos.

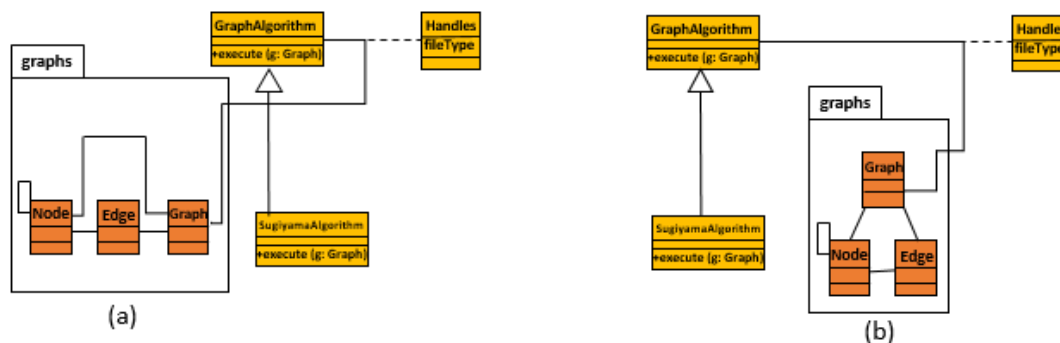


Figura 1.5 Uso de la taxonomía. (a) Diagrama que presenta cruces de arcos y alto número de curvas en los arcos. (b) Diagrama resultante después de aplicar algunas categorías secundarias propuestas en la taxonomía.

1.5 Evaluación de la calidad

El modelado y diseño de procesos de negocio ha adquirido gran atención en los últimos años tanto en el contexto de SI, como en la Reingeniería de Procesos de Negocio (Soffer and Wand, 2004). Asimismo, la comprensión de los modelos de procesos ha adquirido gran relevancia al considerarse que no sólo los expertos en modelado están involucrados en el diseño de modelos de procesos, y que, por otro lado, los modelos de procesos son usados frecuentemente para propósitos puramente organizacionales, más allá de su uso tradicional dentro de la ingeniería del software. Además, se aprecia un considerable aumento en las diferentes técnicas y herramientas para el modelado de procesos disponibles en el mercado. Considerando lo anterior y en el contexto ya mencionado de entornos de mercado competitivos y en constante cambio a los que tienen que hacer frente las organizaciones, es necesario evaluar la calidad de los modelos de procesos desde diferentes puntos de vista. A continuación se presentan las propuestas más relevantes encontradas en la bibliografía relacionadas con la evaluación y medición de los modelos de procesos de negocio.

1.5.1 Medición cuantitativa de modelos de procesos de negocio.

Debido al reconocimiento por parte de la comunidad científica sobre la importancia que representan las tareas de mantenimiento de los procesos en etapas tempranas de su desarrollo (Rolón et al., 2009), como una manera de evitar errores y costos más elevados en fases tardías, en los últimos años se han generado diversas propuestas con el interés de evaluar los procesos de negocio a nivel del modelo conceptual que los representa.

Una característica que destaca en la mayoría de las propuestas que se encuentran en la literatura, es la evaluación de la complejidad como un aspecto clave para obtener modelos de calidad, más fáciles de entender y de mantener en un futuro. La complejidad como la define (Cardoso, 2005) es el grado con el cual un proceso de negocio es difícil de analizar, entender o explicar.

De ahí el creciente interés en evaluar la complejidad de los procesos de negocio, ya que con la medición y control de los procesos se pueden obtener modelos más comprensibles. Diversas propuestas iniciales de medidas de complejidad para modelos de proceso de negocio han sido recopiladas y analizadas por algunos autores. Uno de estos estudios se presenta como reporte técnico (Latva-Koivisto, 2001). En el mismo se presenta una recopilación de medidas para la

complejidad de los procesos de negocio encontradas en la literatura, con la finalidad de encontrar una medida de complejidad estructural que cumpla los siguientes criterios: validez, confiabilidad, computabilidad, facilidad de implementación, intuitividad, independencia de otras medidas relacionadas, habilidad para medir la complejidad de procesos reiterativos, modularidad, aditividad e independencia del nivel de detalle en el modelado.

Por su parte, siguiendo la idea de que para ser capaces de predecir el desempeño de un nuevo proceso antes de su implementación, la dirección necesita medidas de efectividad estructural que se encarguen de evaluar propiedades más estáticas o menos cambiantes de los procesos de negocio, en (C. Batini, 1984) se definen tres medidas :

- a) Medida de complejidad: mide el grado en el cual el proceso se estructura en torno a muchas actividades simples, dando lugar a una gran transferencia de información entre los que realizan las actividades. Por tanto a mayor número de participantes, actividades y transferencia de información, mayor es el valor de la complejidad.
- b) Medida de integración: mide el grado en el cual la estructura de un proceso de negocio soporta la entrega de los resultados de una actividad justo cuando son necesitados por las actividades de sus usuarios.
- c) Medida de dinamismo: mide el grado en el cual la estructura del proceso de negocio soporta la necesidad de cambios en el proceso.

Otro aspecto importante que se destaca en las propuestas de medición de procesos de negocio, es que la gran mayoría parte de la adecuación o adaptación de medidas originalmente definidas para la medición de procesos software. Estas propuestas generalmente están enfocadas a la definición de medidas de calidad, que como resultado de su aplicación proporcionen las bases para obtener modelos de calidad. Un ejemplo de ello es el trabajo presentado en (Reijers and Vanderfeesten, 2004), quienes inspirados en las semejanzas entre los programas de software y los procesos de *workflow* hacen un estudio de las medidas existentes para la calidad del software y su aplicabilidad al diseño de procesos *workflow*. Como resultado definen las medidas de cohesión y acoplamiento, las cuales se centran en el contenido de las actividades (sus operaciones) y tienen el propósito de ayudar a los diseñadores en la creación de modelos con una mejor calidad de ejecución y que sean más fáciles de entender por los usuarios. Estas

medidas, posteriormente habrían de ser extendidas en diversas direcciones de acuerdo a los primeros hallazgos de carencias en la definición inicial (Vanderfeesten et al., 2008).

El análisis comparativo de medidas software y su aplicación en procesos de negocio generó otras propuestas. Entre estas figura el trabajo presentado por (Cardoso, 2005, Cardoso, 2006), que tiene como objetivo evaluar la complejidad estructural de los modelos de procesos de negocio, proponiendo un conjunto de medidas para evaluar la complejidad estructural de los procesos de negocio desde la perspectiva de sus flujos de control. Estas medidas, denominadas medidas CFC (*Control-Flow Complexity*) tienen como principal objetivo conseguir una gestión eficaz de los procesos considerando para ello el análisis de la complejidad del proceso como aspecto básico. La definición de las medidas CFC se basa en la hipótesis de que el comportamiento de los flujos de control de un proceso es afectado por constructores tales como las divisiones y uniones (*splits* y *joins* en lo sucesivo).

Otra propuesta es la de Mendling y Neuman (Mendling and Neumann, 2007), quienes motivados por el supuesto de que los modelos de procesos de negocio son construidos por modeladores humanos y que su diseño está sujeto a una racionalidad limitada, consideran que la comprensibilidad de cualquier modelo por una persona es influenciado por una variedad de factores incluyendo factores relacionados con el modelo (tamaño), factores personales (aptitud de modelado), conocimiento del dominio, lenguaje de modelado o diseño gráfico del modelo. En (Mendling, 2008a) se describe un conjunto de medidas que capturan diversos aspectos relacionados a la estructura del modelo de proceso discutiendo su impacto en la probabilidad de error. El conjunto de medidas definidas en (Mendling, 2008b) se subdividen en seis categorías que son: tamaño, densidad, particionalidad, interacción de conectores, ciclicidad y concurrencia. Otra propuesta más reciente de medidas para modelos de procesos de negocio se puede encontrar en (Huang and Kumar, 2009), donde se presenta una propuesta sistemática para desarrollar medidas de calidad para modelos de procesos estructurados por bloques, que ofrecen menos poder expresivo que las redes de Petri, pero que tiene una semántica más fácil (Huang and Kumar, 2009). Esta propuesta se basa en la noción de un mal puntaje que es usado para calcular la calidad y debe ser igual para modelos equivalentes.

Por su parte, (Jung, 2008) introduce el concepto de entropía para la medición de los modelos de procesos de negocio. Jung considera que la medida de entropía para modelos de procesos puede

ser usada para cuantificar su incertidumbre o variabilidad en base a la probabilidad de los flujos de control y al porcentaje de ejecución del proceso.

En (Rolón, 2009) se ha presentado un conjunto representativo de medidas para modelos de proceso de negocio, cuya finalidad es evaluar la complejidad estructural del modelo. El objetivo final es la utilización de las medidas como un medio para identificar modelos de procesos más fáciles de entender y modificar que por tanto faciliten las tareas de mantenimiento, proporcionando a la vez soporte a la gestión de procesos de negocio en su fase de definición y modelado.

El análisis de los modelos de procesos de negocio generalmente se usa para investigar nuevas maneras de mejorar los procesos, como por ejemplo reducir los costos (Rolón, 2009). Sin embargo, derivado de los constantes cambios que sufren los modelos de procesos durante su ciclo de vida, es necesario redirigir su evaluación hacia un nivel conceptual que proporcione datos objetivos acerca de los modelos, facilitando de este modo las tareas de mantenimiento de los mismos (Rolón, 2009).

1.5.2 Otras formas de evaluación de calidad en modelos de procesos de negocios.

En la literatura se pueden encontrar diversas propuestas para la evaluación y mejora de los procesos de negocio, que en su mayoría están enfocadas a evaluar aspectos tales como los resultados obtenidos, el tiempo y costo de ejecución de los procesos, entre otros. Estos análisis se hacen en la etapa de ejecución de los procesos.

En (Serrano et al., 2002) se abarcan los temas de la medición y el control de los procesos de negocio, para identificar mecanismos de control que resulten efectivos en diferentes tipos de entornos. Este trabajo se basa en la idea de que un proceso organizacional bien diseñado incluye un mecanismo de control mediante el cual la administración puede decidir qué aspectos de la ejecución del proceso deben ser medidos, y de qué manera estas mediciones pueden ser usadas para cambiar el nivel de recursos utilizados en los procesos. En este análisis se resaltan las interacciones entre: a) las demandas del entorno de los procesos, que podrían ser aleatorias o estáticas, b) la señal de control elegida, por ejemplo el tiempo del ciclo o los retrasos y, c) el tipo de control usado (proporcional o diferencial). Los resultados que se obtienen sugieren que a pesar de la demanda del entorno, un proceso de control basado en retrasos del sistema es

generalmente más robusto que las otras alternativas, pues un desempeño adecuado se logra sobre un amplio rango de parámetros de control. Además, resumen que en la mayoría de los casos el control proporcional por sí mismo es inadecuado para ofrecer un desempeño efectivo y el control diferencial es un complemento necesario.

Otra propuesta se presenta por (R. Tamassia, 1988), donde se propone una nueva metodología para definir medidas de procesos de negocio en base a un metamodelo de medición. Su estudio parte de la problemática que surge de la diversidad de metodologías para la gestión de procesos de negocio, las cuales usan métodos numéricos para calcular las debilidades y fortalezas de un negocio. Estas metodologías a su vez están soportadas por diversas herramientas que aunque proporcionan lo mejor de la metodología para un área reducida, tienen la desventaja de que no pueden soportar simultáneamente diversas metodologías. Este trabajo presenta un conjunto de medidas y reglas relacionadas a conceptos de negocios, usando el Lenguaje de Modelado Unificado (UML) (OMG, 2005). Asimismo utiliza una propuesta de metamodelo acorde a *Meta Object Facility* (MOF) (WADDLE, 2001) para analizar la teoría y las metodologías, proporcionando nuevas posibilidades para la medición de procesos de negocio y reduciendo el vacío entre las soluciones técnicas y las metodologías para la gestión de recursos.

El marco de trabajo propuesto por (FORSTER), está basado en la distinción entre los aspectos sintácticos, semánticos y pragmáticos de los modelos empresariales. Además, incluye cualidades intrínsecas como medidas absolutas que pueden ser computadas para un modelo específico, y cualidades comparativas, es decir, medidas relativas que comparan modelos. Los tres aspectos evaluados en este marco son:

- a) La sintaxis, que se refiere al tipo de constructores y manera legal de combinarlos, este aspecto está relacionado con aspectos puramente estructurales del modelo.
- b) La semántica, que se refiere al significado intrínseco del modelo, o bien, está enfocado en esclarecer el significado del nombre usado para un elemento específico del modelo. En otras palabras, está relacionado con la correspondencia entre el modelo y el dominio subyacente.
- c) El pragmatismo, que está relacionado al contexto, es decir, con aquellas medidas y criterios que no pueden ser evaluados en base a la información contenida dentro del

modelo, pero que requiere la consideración de información en cuanto al uso, entorno o contexto del modelo.

En (Hollingsworth, 2004) se presenta un marco de decisión para evaluar proyectos de mejora de procesos de negocio (*Business Process Improvement*, BPI) basándose en la idea de que estas mejoras están vinculadas a los objetivos estratégicos de la organización. Para lograr su objetivo, el marco que se propone involucra la síntesis del proceso de redes analíticas y el análisis de desarrollo de datos, partiendo de la base de que cuando estas dos técnicas son usadas de manera conjunta, pueden proporcionar evaluaciones subjetivas y objetivas para la toma de decisiones ejecutivas. Se considera además que para la evolución de cualquier proyecto estratégico, son necesarias medidas cualitativas, cuantitativas, estratégicas y operativas. Esta propuesta se enfoca en proporcionar apoyo a la toma de decisiones al llevar a cabo la mejora de los procesos de negocio.

En (Aversano et al., 2004) se propone un marco para la medición basado en el paradigma GQM (*Goal-Question-Metric*), que puede ser aplicado de forma general a cualquier proceso de negocio y al sistema de software que lo soporta tras su instanciación. El objetivo de esta propuesta es ayudar a los analistas de negocios a encontrar indicadores útiles relativos al desempeño del proceso, sus elementos críticos, los servicios resultantes que pueden ser mejorados, los nuevos servicios requeridos, el impacto de los cambios elegidos, las mejoras esperadas y el futuro de las herramientas de soporte. Con este marco se crea una base para la colección de datos cuantitativos y cualitativos de un proceso y sus actividades. Además no sólo incluye el marco de medición, sino también un conjunto de tablas críticas y un entorno software denominado WebEv+, el cual puede ser usado tanto en la etapa de planeación, como en la etapa de ejecución de la evaluación.

Por otra parte, una de las razones por la que los procesos de negocio han adquirido gran popularidad en los últimos años, es el surgimiento de una gran diversidad de técnicas y herramientas para su modelado. Esta situación también conlleva a una problemática adicional al elegir la técnica de modelado adecuada, que además de satisfacer los requisitos de la empresa que los adopte, tengan un nivel de calidad deseado. Como consecuencia surgen nuevas líneas de investigación que se enfocan en la evaluación de la calidad de las diversas técnicas utilizadas para el modelado de procesos de negocio. En el estudio realizado por (Hommes and van

Reijswoud, 2000, Hommes et al., 2000) se propone un marco para la descripción y evaluación de las técnicas de modelado, el cual dividen en dos partes: en la manera de modelar y en la manera de trabajar una técnica de modelado. El objetivo de dicho estudio es el de proporcionar un conjunto de propiedades bien definidas, así como de una serie de procedimientos para hacer una medición objetiva de las mismas. Las propiedades de calidad analizadas en el marco propuesto son: conveniencia, integridad, coherencia, expresividad, comprensibilidad, arbitrariedad, efectividad y eficiencia.

El marco propuesto por Hommes, permite la evaluación tanto de la calidad del producto, como de la calidad de las técnicas de modelado, y el beneficio más importante que aporta es que proporciona un conjunto de categorías y propiedades que permiten una descripción uniforme y formal de los elementos dentro de un tipo de modelo, o bien de los diferentes tipos de modelos usados dentro de una técnica de modelado. Sin embargo, el método no incluye medidas cuantificables para expresar la calidad de una técnica de modelado.

1.6 Método para la definición y validación de las medidas

Para definir un conjunto de medidas para procesos de negocio se ha seguido el método de trabajo elaborado por el Grupo Alarcos en proyectos de medición, para la obtención de medidas para software de una manera estructurada que sean válidas teóricamente y útiles en la práctica (Serrano et al., 2002).

La definición de las medidas debe basarse en objetivos de medición claros y siguiendo las necesidades de la organización. En la literatura relativa a las medidas para software se comenta la necesidad de la validación, por lo que han surgido propuestas que integran todos los aspectos necesarios a tener en cuenta en dicha validación. Una de estas propuestas y en la cual se basa el proceso de definición de medidas de esta tesis, es el proceso basado en el método utilizado por (Calero et al., 2001) y en el método MMLC (*Measure Model Life Cycle*) (Cantone and Donzelli, 2000). Estos métodos constan de diversas fases que van desde la identificación de los objetivos y las hipótesis de trabajo hasta la aplicación y posterior retirada de una medida. A continuación, se describen en detalle cada una de las etapas del proceso para la obtención de medidas válidas.

1.6.1 Etapas para la definición de medidas

1. **Identificación:** Es la etapa donde se definen los objetivos que se persigue con la medida creada y se plantean las hipótesis de cómo se llevará a cabo la medición. Sobre los elementos de esta etapa (objetivos e hipótesis) se basarán todas las etapas siguientes. Como resultado de esta etapa se generan los requisitos que debe cumplir la medida. Los objetivos indican lo que se pretende conseguir con la utilización del proceso de medición y representan la razón por la que se llevará el proceso de medición (el “porqué”). Las hipótesis son la forma en la que se pretende llevar a cabo la medición (el “cómo”), identificando la información que se debe manejar para conseguir alcanzar los objetivos deseados. Este proceso suele estar basado en la experiencia y el conocimiento de los expertos y puede utilizar mecanismos basados en GQM (Basili and Rombach, 1988, Basili and Weiss, 1984, Hommes and Van Reijswoud, 1999). Como resultado de esta fase se deben obtener los requisitos que debe cumplir la medida, los cuales serán utilizados en la etapa de creación.
2. **Creación:** En esta etapa se realiza la definición de la medida y su validación teórica y empírica. Esta es una de las etapas más importantes, puesto que abarca un proceso iterativo del que debe salir una medida válida. El proceso de creación de las medidas es evolutivo e iterativo y se subdivide en varias etapas intermedias. Al final de la etapa de creación, las medidas se considerarán válidas y aquellas que no sean válidas, serán descartadas. Las etapas en las que se subdivide la creación son:
 - **Definición:** Es el primer paso de esta fase que debe realizarse considerando las características del producto que se va a medir y la experiencia de los profesionales. En la definición se deben considerar objetivos claros y es deseable que la definición de las medidas se realice de manera formal para evitar ambigüedades.
 - **Validación teórica.** El objetivo principal de la validación teórica es demostrar que la medida mide el atributo que pretende medir, es decir, comprobar si la idea intuitiva acerca del atributo que está siendo medido se refleja en la medida. Además, la validación teórica proporciona información relacionada con las escalas de las medidas y así se puede determinar qué tipo de operaciones

matemáticas y pruebas estadísticas aplicar para analizar los valores de las medidas en estudios empíricos.

Hay dos tendencias principales para la validación teórica de medidas: los marcos basados en propiedades (Briand et al., 1996, Weyukr, 1988) y los que se basan en la teoría de la medida (Poels and Dedene, 2000, Whitmire, 1997) cuyo objetivo es obtener la escala matemática a la que pertenece una medida, y por tanto sus transformaciones admisibles, estadísticos y pruebas aplicables especifican un marco general en el que las medidas deben ser definidas.

- Validación empírica: El objetivo de esta etapa es probar la utilidad práctica de las medidas propuestas. El saber general, la intuición o la especulación, no son fuentes fiables de conocimiento (Basili et al., 1999), por lo que es necesario realizar validaciones empíricas con las medidas. La validación empírica se utiliza para obtener información objetiva sobre la utilidad de las medidas propuestas ya que puede que una medida sea correcta desde un punto de vista formal, pero no tener relevancia práctica para un problema determinado.
 - Explicación psicológica: Idealmente es necesario tener la capacidad de explicar la influencia de los valores de las medidas desde un punto de vista psicológico. Algunos autores, como (Siau, 1999), proponen el uso de la psicología cognitiva como una disciplina de referencia. De esta manera, las teorías como ACT (*Adaptive Control of Thought*) (Anderson, 1983), pueden justificar la influencia de ciertas medidas en la comprensión de los sistemas.
3. Aceptación: Una vez obtenida una medida válida, suele ser necesario pasar por una etapa de aceptación de la medida en la que se harán pruebas en entornos reales, de manera que se pueda comprobar si la medida cumple con los objetivos deseados dentro del campo de aplicación real. Suele ser necesaria la existencia de una fase de pruebas en laboratorio en la que se realice una experimentación sistemática en entornos reales y con usuarios reales para verificar si cumple los objetivos buscados dentro de un entorno de trabajo real. Esta etapa se diferencia de los casos de estudio en que en éstos últimos se suele trabajar en el entorno final de aplicación. En esta etapa se determina si las medidas “válidas” que se consiguieron al final de la fase de creación son aceptables en entornos

de aplicación reales, teniendo en cuenta los objetivos definidos en la etapa de identificación.

Esta etapa debe ser realizada con proyectos no críticos y con riesgos controlados. Idealmente debería usarse en proyectos piloto de manera que el fracaso de aceptación de la medida no suponga un fracaso en un proyecto importante. Si se consigue demostrar que la medida sigue cumpliendo los objetivos, es posible pasar a la etapa de aplicación, y si no es así, es necesario volver a la etapa de creación.

4. Aplicación: En esta etapa, una vez que se tiene la medida aceptada, la utilizamos dentro del campo de la aplicación para la que fue diseñada. Esta fase se lleva a cabo en paralelo con la fase de acreditación.
5. Acreditación: Es la última etapa del proceso, que discurre en paralelo con la fase de aplicación y tiene como objetivo el mantenimiento de la medida, de manera que pueda adaptarse al entorno cambiante de aplicación. Como consecuencia de esta etapa, puede que una medida sea retirada, porque ya no sea útil en el entorno en el que se aplica o que sea reutilizada para iniciar de nuevo el proceso. Esta última fase del proceso es una etapa dinámica que persigue el aseguramiento de la medida y la mejora continua de la misma, en función de cómo evoluciona el entorno de aplicación, de manera que se puedan seguir cumpliendo los objetivos que se perseguían al principio del método.

Teniendo en cuenta que el objeto de investigación de la tesis es definir medidas de la calidad para la representación visual (RV) de los modelos de procesos de negocio, se ha considerado la aplicación del método de trabajo específico para medidas de software descrito anteriormente. Para la propuesta de medidas se han seguido las fases de identificación y creación. En la fase de creación se ha realizado la definición y validación teórica. Respecto a la validación teórica se ha seguido el método propuesto en el marco formal de Briand (Briand et al., 1996). La validación teórica de las medidas ayuda a los diseñadores a saber cuándo aplicar las medidas y de qué manera. Varios autores en sus propuestas de medidas (Serrano et al., 2002, Rajnish, 2014, Rajnish, 2013, Sharma et al., 2015) han utilizado el marco de Briand para validar dichas propuestas. El resultado de estas validaciones determinó que todas las medidas son teóricamente validadas y, por lo tanto, son útiles (Sharma et al., 2015).

Para proporcionar rigor matemático y una base axiomática a las medidas, las propiedades necesarias han sido establecidas en el marco formal de Briand, mediante el cual se pueden validar las medidas propuestas. La validación de las medidas con los conceptos de medición de Briand brinda información sobre las características y defectos de las mismas. Además, sirve como una herramienta para la clasificación y comprensión. La validación basada en las propiedades propuestas en este marco también puede conducir potencialmente a la formulación de nuevas medidas (Rajnish, 2014).

1.6.2 Marco formal de Briand

El principal objetivo de la validación teórica es comprobar si la idea intuitiva acerca del atributo que se está midiendo se refleja en la medida tomada. Esto se hace analizando los requisitos o condiciones teóricos que deben satisfacerse al medir. Por lo tanto la validación teórica se basa en el análisis de las propiedades de los atributos a medir, y el primer requisito es que el diseñador tenga conocimiento intuitivo del concepto que está siendo medido.

El marco formal de Briand define varios conceptos relacionados con la medición, como son el tamaño, la longitud, la complejidad, la cohesión y el acoplamiento. Además un conjunto de propiedades que debe satisfacer cada concepto con el objetivo de clasificar las medidas.

En primer lugar se presentan las definiciones básicas relacionadas con los objetos de estudio a los que pueden ser aplicados los conceptos (tamaño, longitud, complejidad, acoplamiento y cohesión), con el fin de determinar la entidad sobre la que se van a medir. Finalmente se introducen las propiedades necesarias de cada uno de los conceptos definidos.

Las principales definiciones a considerar en el contexto de este marco son sistemas y módulos. En este marco un sistema queda caracterizado por sus elementos y las relaciones entre ellos. Es decir, un sistema S será representado como un par $\langle E, R \rangle$, donde E representa el conjunto de elementos de S , y R es una relación binaria en E ($R \subseteq E \times E$) representando las relaciones entre elementos de S . Luego, dado un sistema $S = \langle E, R \rangle$, un sistema $m = \langle E_m, R_m \rangle$ es un módulo de S si y sólo si $E_m \subseteq E$, $R_m \subseteq E \times E$ y $R_m \subseteq R$.

Los autores definen una serie de conceptos que son subjetivos, pero que están caracterizados por propiedades intuitivas y no ambiguas (Briand et al., 1996). Dichas propiedades son

consideradas necesarias, pero no suficientes porque no garantizan que las medidas que las cumplen sean útiles o incluso que tengan sentido.

Los principales conceptos definidos y sus propiedades son:

- **Tamaño:** el tamaño de un sistema $S = \langle E, R \rangle$ es una función $Size(S)$ que se caracteriza por las siguientes propiedades:

Propiedad 1: No negatividad

El tamaño de un sistema $S = \langle E, R \rangle$ es no negativo

$$Size(S) \geq 0$$

Propiedad 2: Valor nulo

El tamaño de un sistema $S = \langle E, R \rangle$ es nulo si E es vacío.

$$E = \emptyset \Rightarrow Size(S) = 0$$

Propiedad 3: Aditividad de módulos

El tamaño de un sistema $S = \langle E, R \rangle$ es igual a la suma de los tamaños de sus módulos, tal que no tienen elementos en común.

$$Size(S) = Size(m1) + Size(m2)$$

- **Longitud:** la longitud de un sistema $S = \langle E, R \rangle$ es una función $Length(S)$ que se caracteriza por las siguientes propiedades:

Propiedad 1: No negatividad

La longitud de un sistema $S = \langle E, R \rangle$ es no negativa

$$Length(S) \geq 0$$

Propiedad 2: Valor nulo

La longitud de un sistema $S = \langle E, R \rangle$ es nulo si E es vacío.

$$E = \emptyset \Rightarrow Length(S) = 0$$

Propiedad 3: Monotonidad no creciente para componentes conectados

Sea un sistema $S = \langle E, R \rangle$ y m un módulo de S , talque m es representado por una componente conectada del grafo que representa a S . Si se introduce una nueva relación entre dos elementos que pertenecen al mismo componente conexo (se dice que dos elementos de un sistema S pertenecen al mismo componente conexo si existe un camino de uno a otro en el grafo no dirigido obtenido del grafo que representa S eliminando las

direcciones en los arcos) del grafo que representa el sistema, la longitud del nuevo sistema no es mayor que la del sistema original.

Propiedad 4: Monotonidad no decreciente para componentes no conectados

Sea $S = \langle E, R \rangle$ un sistema y $m1, m2$ dos módulos de S , tal que $m1$ y $m2$ están representados por dos componentes conectadas separadas del grafo que representa a S . Si se introduce una nueva relación entre dos elementos que pertenecen a dos componentes conectados diferentes, la longitud del sistema no es menor que la longitud del sistema original.

Propiedad 5: Módulos Disjuntos

La longitud de un sistema $S = \langle E, R \rangle$ de dos módulos disjuntos $m1$ y $m2$ es igual a la longitud máxima de $m1$ y $m2$.

$$Length(S) = \max\{Length(m1), Length(m2)\}$$

- **Complejidad:** La complejidad es una de las características que se consideran más relevantes dentro de un sistema. Para los autores la complejidad solo puede ser aportada por las relaciones y no constituye una propiedad de los elementos aislados. La complejidad de un sistema $S = \langle E, R \rangle$ es una función $Complexity(S)$ que se caracteriza por las siguientes propiedades:

Propiedad 1: No negatividad

La complejidad de un sistema $S = \langle E, R \rangle$ es no negativo

$$Complexity(S) \geq 0$$

Propiedad 2: Valor nulo

La complejidad de un sistema $S = \langle E, R \rangle$ es nulo si E es vacío.

$$E = \emptyset \Rightarrow Complexity(S) = 0$$

Propiedad 3: Simetría

La complejidad de un sistema $S = \langle E, R \rangle$ no debería ser sensible a las convenciones de representación utilizadas respecto a la dirección de los arcos que representan las relaciones en el sistema.

$$(S = \langle E, R \rangle \text{ y } S^{-1} = \langle E, R^{-1} \rangle) \Rightarrow Complexity(S) = Complexity(S^{-1})$$

Propiedad 4: Monotonidad de módulos

La complejidad de un sistema $S = \langle E, R \rangle$ no es menor que la suma de las complejidades de cualquier colección de sus módulos, tal que no existe relaciones compartidas entre ellos, aunque puedan compartir elementos.

Propiedad 5: Aditividad de módulos disjuntos

La complejidad de un sistema $S = \langle E, R \rangle$ compuesto de dos módulos disjuntos $m1$, $m2$ es igual a la suma de las complejidades de los módulos.

$$Complexity(S) = Complexity(m1) + Complexity(m2)$$

- **Cohesión:** El concepto de cohesión valora en qué grado las características relacionadas de un programa están agrupadas en sistemas o módulos.

Propiedad 1: No negatividad y normalización

La cohesión de un sistema modular o un módulo es no negativa y debe estar especificada en un intervalo.

$$Cohesion(m) \in [0, Max]$$

Propiedad 2: Valor nulo

La cohesión de un sistema modular o módulo $m = \langle E_m, R_m \rangle$ es nulo si R_m es vacío.

$$R_m = \emptyset \Rightarrow Cohesion(m) = 0$$

Propiedad 3: Monotonidad

La adición de relaciones internas en los módulos no puede disminuir la cohesión, ya que se supone que son pruebas que vienen a justificar el agrupamiento de esos elementos del sistema.

Propiedad 4: Módulos cohesivos

Cuando dos o más módulos que no muestran relaciones entre ellos se fusionan, la cohesión no puede aumentar porque elementos aparentemente no relacionados son encapsulados juntos.

- **Acoplamiento:** El acoplamiento recoge el grado de relación entre elementos que pertenecen a diferentes módulos de un sistema. Dado un módulo m , es posible definir dos tipos de acoplamiento: entrante y saliente. El primero recoge el número de relaciones desde elementos externos a m hacia elementos dentro de m ; el último recoge la cantidad de relación desde elementos de m a elementos fuera de m . El acoplamiento de un módulo es una función $Coupling(S)$ caracterizada por las siguientes propiedades.

Propiedad 1: No negatividad

El acoplamiento de un sistema modular o módulo $m = \langle E_m, R_m \rangle$ es no negativo

$$Coupling(m) \geq 0$$

Propiedad 2: Valor nulo

El acoplamiento de un sistema modular o módulo $m = \langle E_m, R_m \rangle$ es nulo si no existen relaciones entre módulos.

Propiedad 3: Monotonicidad

Cuando se crean relaciones adicionales entre módulos, se espera que el acoplamiento no decrezca, dado que esos módulos pasan a tener una mayor interdependencia.

Propiedad 4: Fusión de módulos

El acoplamiento de un módulo obtenido por la fusión de dos módulos no es mayor que la suma de los acoplamientos de los dos módulos originales, dado que pueden existir relaciones entre ellos y cuando se fusionan, estas relaciones desaparecen.

Propiedad 5: Aditividad de módulos disjuntos

El acoplamiento de un módulo obtenido por la fusión de dos módulos disjuntos es igual a la suma de los acoplamientos de los dos módulos originales.

1.7 Consideraciones finales del capítulo

La calidad de los modelos de procesos de negocios es un aspecto importante de los mismos que proporciona soporte a la Gestión de Procesos de Negocio, mediante la evaluación temprana en una de sus primeras etapas (definición y modelación). Diversos estudios para la evaluación de procesos de negocio, presentan propuestas para evaluar la calidad relacionada con el etiquetado y el diseño visual. Estos incluyen el uso de *frameworks* de calidad, directrices prácticas y criterios relacionados con el estilo de etiquetado y el diseño visual. En este capítulo se presentaron investigaciones relacionadas con la representación visual, tanto de modelos de procesos de negocios como en otras áreas, concluyendo que ninguno de los trabajos encontrados ofrece medidas para la evaluación de la representación visual de los modelos de procesos de negocio.

PROPUESTA DE MEDIDAS PARA LA EVALUACIÓN DE LA
REPRESENTACIÓN VISUAL

2. PROPUESTA DE MEDIDAS PARA LA EVALUACIÓN DE LA REPRESENTACIÓN VISUAL

Teniendo en cuenta el interés en evaluar los procesos de negocio a un nivel conceptual con el objetivo de obtener evidencia empírica sobre la influencia que la representación visual de los modelos de procesos de negocio tiene en su comprensibilidad y modificabilidad, se ha definido un conjunto de medidas en base a los elementos que componen la notación BPMN (Business Process Modeling Notation) (OMG, 2011). En este capítulo se presenta el contexto y la justificación de la propuesta de medidas para la evaluación de procesos de negocio a partir del modelo que los representa en un nivel conceptual, incluyendo la definición teórica y formal de las medidas.

2.1 Contexto de la propuesta

En los últimos años han surgido varias propuestas para evaluar diversos aspectos de los procesos de negocio a partir del modelo que los representa gráficamente. Estas propuestas se basan en la noción de que al evaluar el proceso en fases tempranas de desarrollo, es posible obtener información sobre la complejidad del proceso, detectar y corregir errores, predecir las probabilidades de error, etc., facilitando de esta manera la reingeniería de proceso.

Los procesos de negocio pueden ser representados mediante diversas notaciones o lenguajes de modelado, algunos de los cuales han sido diseñados específicamente para tal efecto. Entre ellos se pueden citar UML (acrónimo de Unified Modeling Language) (Eichelberger and Schmid, 2009), IDEF3 (Integrated Definition for Process Description Capture Method) (He et al., 2010), EPC (Event-driven Process Chain) (Mendling, 2008b), el Lenguaje de Definición de Procesos XML (XML Process Definition Language; XPDL) (WFMC, 2002), BPMN (OMG, 2011) entre otros. Esta tesis se centra en el uso de BPMN debido a que el creciente interés en BPMN lo convierte en un importante estándar respecto al modelado de procesos en las organizaciones (Recker, 2008, Zur Muehlen and Recker, 2008). Por tanto, con las medidas propuestas es posible evaluar la representación visual de los modelos de proceso de negocio expresados con BPMN, abarcando de esta manera todos los elementos centrales de un proceso de negocio.

Se considera que con una adecuada medición a nivel conceptual, se pueden obtener modelos más fáciles de entender, lo que a su vez facilita las tareas de mantenimiento y la selección de

alternativas de modelado. De esta manera, al analizar estructuralmente el modelo también puede ser evaluada su calidad. Por otra parte, y de acuerdo con (Lindland et al., 1994b), la calidad de un producto final depende en gran medida de la precisión en la especificación de requisitos y está más centrada en la mejora de las fases tempranas de desarrollo. Con esta filosofía, se estaría proporcionando soporte a la Gestión de Procesos de Negocio, mediante la evaluación temprana de su calidad en sus primeras etapas de definición y modelado.

Desde el punto de vista meramente conceptual del modelo, (Nelson et al., 2012) propone un marco que distingue los objetivos para la mejora de la calidad y los medios para lograr esos objetivos. Para ello se definen cuatro criterios de calidad para modelos conceptuales, que son: la calidad semántica, la calidad sintáctica, la calidad pragmática y la calidad empírica. En base a este marco propuesto, se considera que utilizando las medidas definidas en esta tesis, sería posible evaluar y mejorar la calidad de los modelos conceptuales de procesos de negocio.

2.2 Definición de las medidas

Las medidas propuestas en esta tesis para evaluar la representación visual de modelos de proceso de negocio representados con BPMN han sido agrupadas en dos categorías: medidas de etiquetado y medidas de diseño visual (Moreno-Montes de Oca and Snoeck, 2015).

2.2.1 Definición de medidas de etiquetado

Este conjunto de medidas trata aspectos de etiquetado y de información incluida en los elementos en los modelos de procesos BPMN, como por ejemplo, sobre el uso de iconos provenientes del estándar BPMN en las actividades para facilitar la comprensión del modelo. Estas medidas trabajan sobre elementos individuales y pueden subdividirse en estilo de etiquetado de elementos, estilo de etiquetado de actividades, etiquetado de eventos, etiquetado de compuertas, etiquetado de flujo de secuencia y etiquetado de *pools*.

De este modo, se han definido un total de 22 medidas relacionadas con el estilo de etiquetado las cuales se agruparon en dos categorías: medidas base y medidas derivadas. En la Tabla 2.1 se muestran las medidas base que se definieron relacionadas con el estilo de etiquetado.

A partir de las medidas base se ha definido un conjunto de medidas derivadas, con las cuales es posible conocer proporciones existentes entre elementos relevantes del modelo. En la

Tabla 2.2 se presentan las medidas derivadas propuestas, relacionadas con el estilo de etiquetado.

Tabla 2.1 Medidas bases – Estilo de etiquetado.

Subconjunto	Nombre Medida	Definición de la medida base
Etiquetado de actividades	<i>EI</i>	Número de actividades con etiquetas incorrectas
	<i>FST</i>	Número de actividades con tipo no especificado
	<i>F</i>	Total de actividades
	<i>FSE</i>	Número de actividades sin etiquetar
	<i>ERI</i>	Número de actividades etiquetadas como enviar/recibir que no son de este tipo
Etiquetado de elementos	<i>l</i>	Cantidad de palabras de las etiquetas sin incluir artículos, preposiciones, ni conjunciones.
	<i>Etiquetas</i>	Conjunto de etiquetas del modelos
	<i>N</i>	Número de elementos del modelo
Etiquetado de compuertas	<i>CSE</i>	Número de compuertas de división de tipo <i>or</i> o <i>xor</i> sin etiquetar
	<i>S_{or}</i>	Número de compuertas de división de tipo <i>or</i>
	<i>S_{xor}</i>	Número de compuertas de división de tipo <i>xor</i>
Etiquetado de flujo de secuencias	<i>SFSE</i>	Número de flujos de secuencias sin etiquetar
Etiquetado de eventos	<i>ESE</i>	Número de eventos sin etiquetar
	<i>E</i>	Cantidad de eventos del modelo

Tabla 2.2 Medidas derivadas – Estilo de etiquetado.

Nombre medida	Definición	Fórmula
EI_F	Razón entre el número de actividades con etiquetas incorrectas y el total de actividades.	$EI_F = \frac{ EI }{ F }$
FST_F	Razón entre el número de actividades que no poseen tipo especificado y el número total de actividades.	$FST_F = \frac{ FST }{ F }$
ESE_E	Razón entre el número de eventos sin etiquetar y la cantidad de eventos en el modelo.	$ESE_E = \frac{ ESE }{ E }$
FSE_F	Razón entre el número de actividades sin etiquetar y la cantidad de actividades.	$FSE_F = \frac{ FSE }{ F }$
CSE_C	Razón entre el número de compuertas de división de tipo or y xor sin etiquetar y la cantidad de compuertas de este tipo.	$CSE_C = \frac{ CSE }{ S_{or} + S_{xor} }$
$SFSE_S$	Razón entre el número de flujos de secuencia sin etiquetar y el número de flujos de secuencia de salida de compuertas de tipo or o xor.	$SFSE_S = \frac{ SFSE }{\sum_{c \in S_{or} \cup S_{xor}} d_{out}(c)}$
$\bar{l}_{Etiquetas}$	Longitud promedio en términos de cantidades de palabras de las etiquetas del modelo de proceso.	$\bar{l}_{Etiquetas} = \frac{1}{ N } \sum_{e \in Etiquetas} l(e)$
$\hat{l}_{Etiquetas}$	Largo máximo de una etiqueta del modelo de proceso.	$\hat{l}_{Etiquetas} = \max\{l(e) e \in Etiquetas\}$

2.2.2 Definición de medidas de diseño visual

Los aspectos relacionados con el diseño de los modelos son importantes para una mejor comprensión y legibilidad de estos. El conjunto de medidas propuestas para el diseño visual se refiere principalmente al formato de la estructura del grafo primario. Por ejemplo, minimizar las curvas en el flujo de secuencia. Estas medidas consideran posiciones relativas de los elementos (relacionan elementos) y la organización general del modelo. Una subdivisión posterior de este conjunto puede contener tres subconjuntos: organización general, abarca las medidas relacionadas con la organización del modelo; flujo, incluye medidas dirigidas específicamente

a la organización del flujo de secuencia del modelo; y otros elementos, comprende medidas de diseño relacionadas con posiciones relativas de los distintos elementos del modelo.

De este modo, se han definido un total de 17 medidas relacionadas con el diseño visual. De forma similar al grupo de medidas propuestas anteriormente, estas medidas también han sido agrupadas en: medidas base y medidas derivadas. La Tabla 2.3 y Tabla 2.4 presentan las medidas base y derivadas, relacionadas con el diseño visual del modelo.

Tabla 2.3 Medidas bases – Diseño visual.

Subconjunto	Nombre Medida	Definición de la medida base
Flujos de secuencias	<i>CA</i>	Número de cruces de flujos de secuencia
	<i>A</i>	Número de flujos de secuencia del modelo
	<i>SCurvas</i>	Número de curvas en los flujos de secuencia
	<i>SAA</i>	Número de solapamientos de flujos de secuencia que no tienen intención de unirse
	<i>FSI</i>	Número de flujos de secuencia con dirección inconsistente
	<i>LD</i>	Número de flujos de secuencia de longitud no uniforme
Organización general	<i>NTI</i>	Número de elementos con tamaño no uniforme
	<i>FSL</i>	Número de flujos de secuencia largos en el modelo
	<i>SAN</i>	Número de solapamiento de flujos de secuencia con elementos
Otros elementos	<i>AL</i>	Razón entre el alto y el largo del contenedor del proceso. $AL = \frac{ap}{lp} \quad (\text{Dónde: } lp: \text{largo y } ap: \text{alto})$
	<i>AD</i>	Área sin utilizar. $AD = ap \times lp - \sum_{n \in N} ar(n)$ (Dónde: <i>ap</i> : alto del <i>pool</i> , <i>lp</i> : largo del <i>pool</i> y <i>ar</i> (<i>n</i>): área del elemento <i>n</i>)

Tabla 2.4 Medidas derivadas – Diseño visual.

Nombre medida	Definición	Fórmula
CA_A	Razón entre el número de cruces de flujos de secuencia y el número de flujos de secuencia en el modelo.	$CA_A = \frac{ CA }{ A }$
$SCurvas_A$	Razón entre el número de curvas en los flujos de secuencia y el número de flujos de secuencia.	$SCurvas_A = \frac{ SCurvas }{ A }$
SAA_A	Razón entre el número de solapamientos de flujos de secuencia y el número de flujos de secuencia.	$SAA_A = \frac{ SAA }{ A }$
FSI_A	Razón de flujos de secuencia con dirección inconsistente y el número de flujos de secuencia.	$FSI_A = \frac{ FSI }{ A }$
LD_A	Razón de flujos de secuencia no uniformes y el número de flujos de secuencia.	$LD_A = \frac{ LD }{ A }$
NTI_N	Razón entre el número de elementos con tamaño no uniforme y el número de elementos.	$NTI_N = \frac{ NTI }{ N }$

2.3 Directrices y medidas de representación visual asociadas

En línea con la importancia de modelos de procesos de alta calidad, varios estudios recientes han investigado diferentes factores que influyen en la misma, como por ejemplo, cómo ciertos atributos estructurales de los modelos de procesos afectan su calidad (Mendling et al., 2010a). En varios trabajos de investigación se han propuesto directrices prácticas cuyo objetivo es aconsejar a los modeladores para lograr una mejor calidad de los modelos de procesos de negocios. De estos estudios se extrajo un amplio conjunto de directrices prácticas para la modelación de procesos de negocio (Moreno-Montes de Oca and Snoeck, 2015). El principal objetivo de estas directrices es aconsejar a los modeladores para lograr una mejor calidad de los modelos de procesos de negocios y de esta forma facilitar la comunicación entre los expertos del negocio involucrados (Reijers et al., 2011, Arkilic et al., 2013). Las directrices coleccionadas pueden agruparse de acuerdo a lo que refieren en dos grupos fundamentales: directrices que tratan sobre la complejidad de los modelos y directrices que tratan sobre la representación visual de los mismos (Moreno-Montes de Oca and Snoeck, 2015).

En la literatura existe un conjunto de medidas estructurales asociadas a las directrices de complejidad general que permiten cuantificar el efecto de aplicar dichas directrices a un modelo. De ahí, la necesidad de asociar las medidas propuestas en esta tesis al conjunto de directrices asociadas a la representación visual de los modelos, dado por la ausencia en la literatura de

medidas estructurales que permitan cuantificar algunas directrices de este grupo, como por ejemplo aquellas relacionadas con las etiquetas utilizadas en los modelos.

Para establecer la relación entre las medidas propuestas en esta tesis y las directrices relacionadas con la representación visual, se deben tener en cuenta algunos elementos del patrón propuesto por (Moreno-Montes de Oca and Snoeck, 2015) como:

- Problema: características cuya ocurrencia denota la infracción de la directriz.
- Directriz unificada: constituye una buena práctica o consejo explícito que ofrece una guía sobre cómo un modelo de proceso puede mejorarse (Mendling et al., 2010c).
- Medidas propuestas: listan las medidas propuestas en este trabajo como resultado del análisis de las directrices para cuantificar la ocurrencia de los problemas en los modelos.

2.3.1 Directrices prácticas relacionadas con el estilo de etiquetado y medidas asociadas

Algunas directrices tratan aspectos relacionados con el etiquetado y con la información incluida en los elementos en los modelos de procesos BPMN. A continuación se presentan cada una de las medidas asociadas a las directrices relacionadas con el estilo de etiquetado.

Problema 1 (Estilo verbo-sustantivo): Las etiquetas de las actividades no siguen el estilo verbo-sustantivo.

Directriz unificada: Use etiquetas del tipo verbo-sustantivo para las actividades.

Medidas propuestas:

Definición:	Número de actividades con etiquetas incorrectas.	
Medida:	$S_{EI}(G) = EI $	(1)
Definición:	Razón entre el número de actividades con etiquetas incorrectas y el total de actividades.	
Medida:	$EI_F(G) = \frac{ EI }{ F }$	(2)

Se ha demostrado a través de estudios empíricos que el uso de etiquetas de actividades que siguen el estilo verbo-sustantivo contribuye a la calidad de los modelos en términos de menor ambigüedad percibida y mayor utilidad percibida, así como mayor comprensión (Mendling and Reijers, 2008, Mendling et al., 2010b). Esta directriz puede influenciar positivamente la directriz sobre el uso de etiquetas cortas pues este estilo sugiere de forma implícita la abreviación de las etiquetas. La Figura 2.1 muestra un ejemplo del uso de etiquetas que no siguen el estilo verbo-

sustantivo y un ejemplo de cómo quedaría aplicando la directriz. El uso de esta directriz y las medidas asociadas permiten mejorar la calidad empírica, la calidad pragmática y la calidad semántica percibida de los modelos de procesos de negocio.

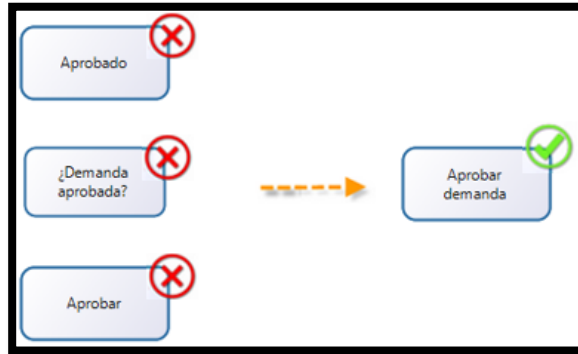


Figura 2.1 Uso de etiquetas con estilo verbo-sustantivo.

Problema 2 (Tipos de tareas no especificados): No se especifican los tipos de actividades usando los iconos destinados para ello.

Directriz unificada: Especifique los tipos de actividades.

Medidas propuestas:

Definición: Número de actividades con tipo no especificado.

Medida: $S_{FST}(G) = |FST|$ (3)

Definición: Razón entre el número de actividades que no poseen tipo especificado y el número de actividades.

Medida: $FST_F(G) = \frac{|FST|}{|F|}$ (4)

Uno de los atributos BPMN sin representación normalizada en el diagrama es el tipo de tarea. Existen tareas que no especifican de que tipo son (humano, automático). La especificación define varios tipos de tareas, pero hay dos que son muy importante distinguir: usuario (tarea humana) y el servicio (tarea automatizada). Afortunadamente, la mayoría de las herramientas BPMN distinguen los tipos de tareas con iconos dentro de la figura de actividad. Especificar los tipos de tareas en el modelo permite mejorar la calidad pragmática y empírica de los mismos.

Por ejemplo la Figura 2.2 muestra un modelo donde no se especifican los tipos de tareas. En este caso el valor obtenido para la medida $S_{FST}(G) = 4$ y para la medida $FST_F(G) = 1$. Sin embargo en la Figura 2.3 se muestra un ejemplo donde los tipos de tareas son especificados

correctamente. De forma similar el valor obtenido para la medida $S_{FST}(G) = 0$ y para la medida $FST_F(G) = 0$.

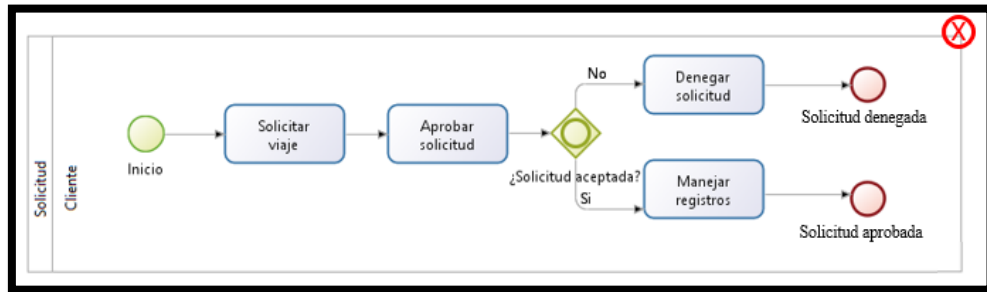


Figura 2.2 Tipos de tareas no especificados.

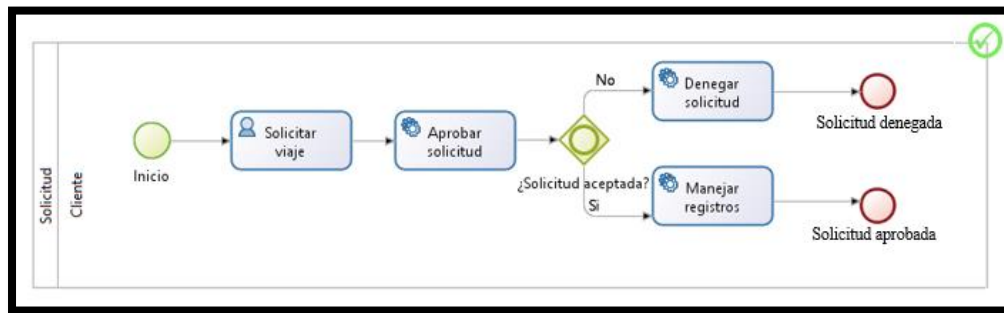


Figura 2.3 Tipos de tareas especificados.

Problema 3 (*Etiquetas enviar/recibir no están reservadas para estos elementos*): Existen actividades o eventos de distintos tipos etiquetados con etiquetas *enviar/recibir*.

Directriz unificada: Reserve las palabras *enviar/recibir* para actividades y eventos de este tipo.

Medidas propuestas:

Definición: Número de elementos etiquetados como enviar/recibir que no son de este tipo.

Medida: $S_{ERI}(G) = |ERI|$ (5)

El nombramiento correcto de los diferentes elementos de los diagramas es fundamental para una comprensión fácil y correcta de los procesos. Al revisar los *logs*, es útil saber cómo se ejecutó el proceso. Cuando no se nombra alguna forma en el proceso, los *logs* se muestran en blanco, lo que hace que sea difícil de entender. A continuación se muestran varios problemas relacionados con el etiquetado de algunos elementos de BPMN.

Problema 4 (*Actividades sin etiquetar*): Hay actividades sin etiquetar.

Directriz unificada: Etiquete las actividades.

Medidas propuestas:

Definición: Número de actividades sin etiquetar.

Medida: $S_{FSE}(G) = |FSE|$ (6)

Definición: Razón entre el número de actividades sin etiquetar y la cantidad de actividades.

Medida: $FSE_F(G) = \frac{|FSE|}{|F|}$ (7)

Etiquetar las actividades permite a los lectores entender con claridad el objetivo de una tarea. Además, permite mejorar la calidad pragmática y empírica de los modelos. La Figura 2.4 muestra un modelo donde hay presencia de actividades sin etiquetar. El resultado obtenido para las medidas asociadas a este problema son: $S_{FSE}(G) = 1$ y $FSE_F(G) = 0.33$.

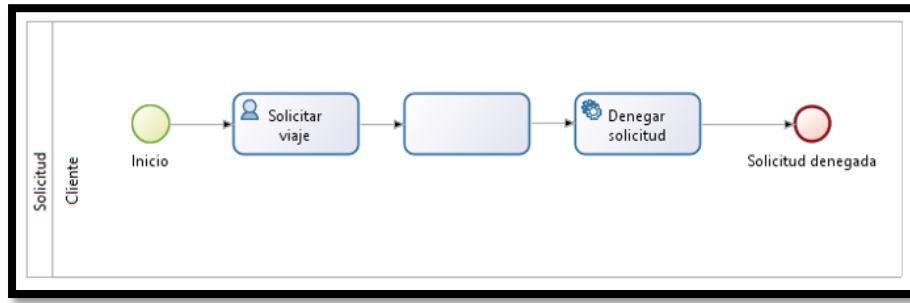


Figura 2.4 Modelo que presenta actividades sin etiquetar.

Problema 5 (Eventos sin etiquetar): El modelo contiene eventos sin etiquetar, por ejemplo, eventos de tiempo sin etiquetas de tipo fecha/hora o duración del evento.

Directriz unificada: Etiquete los eventos.

Medidas propuestas:

Definición: Número de eventos sin etiquetar.

Medida: $S_{ESE}(G) = |ESE|$ (8)

Definición: Razón entre el número de eventos sin etiquetar y la cantidad de eventos en el modelo.

Medida: $ESE_E(G) = \frac{|ESE|}{|E|}$ (9)

Etiquetar los eventos facilita que el diagrama pueda explicarse por sí mismo, principalmente cuando se utilizan múltiples eventos de inicio y fin o eventos de tiempo. Permitiéndole al usuario una mejor concepción de cómo funciona y termina el proceso. El uso de etiquetas en los eventos permite mejorar la calidad pragmática y empírica de los modelos. En la Figura 2.5 se muestra un modelo con eventos sin etiquetar. Al evaluar este modelo tenemos que $S_{ESE}(G) = 2$ y $ESE_E(G) = 0.67$.

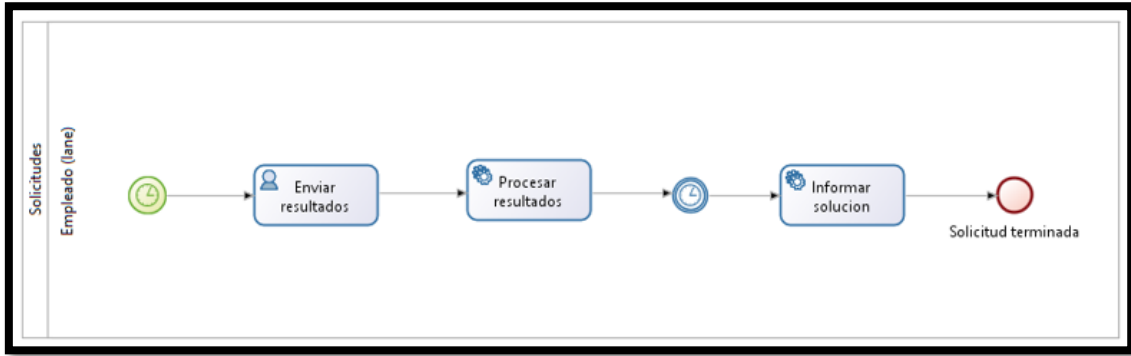


Figura 2.5 Modelo con eventos sin etiquetar.

Problema 6 (Compuertas sin etiquetar): Hay compuertas de división de tipo *or* o de tipo *xor* que carecen de etiquetas.

Directriz unificada: Etiquete las compuertas de división de tipo *or* y de tipo *xor*.

Medidas propuestas:

Definición: Número de compuertas de división de tipo *or* o *xor* sin etiquetar.

Medida: $S_{CSE}(G) = |CSE|$ (10)

Definición: Razón entre el número de compuertas de división de tipo *or* y *xor* sin etiquetar y la cantidad de compuertas de este tipo.

Medida: $CSE_C(G) = \frac{|CSE|}{|S_{or}| + |S_{xor}|}$ (11)

Problema 7 (Flujos de secuencia de salida sin etiquetar): Existen flujos de secuencia provenientes de compuertas de división de tipo *or* o *xor* sin etiquetar.

Directriz unificada: Etiquete todos los flujos de secuencia de salida de las compuertas de división de tipo *or* y *xor*.

Medidas propuestas:

Definición: Número de flujos de secuencia sin etiquetar.

Medida: $S_{SFSE}(G) = |SFSE|$ (12)

Definición: Razón entre el número de flujos de secuencia sin etiquetar y el número de flujos de secuencia de salida de compuertas de tipo *or* o *xor*.

Medida: $SFSE_S(G) = \frac{|SFSE|}{\sum_{c \in S_{or} \cup S_{xor}} d_{out}(c)}$ (13)

Las compuertas de división de tipo *or* o *xor* deben tener un nombre que indique claramente la decisión o condición evaluada cuando aplique. Es útil utilizar un nombre compuesto por un verbo, un objeto, y un signo de interrogación para identificar lo que se está evaluando, incluso

se puede utilizar preguntas para aclarar la decisión en cuestión. De forma similar el nombre de las transiciones debe indicar la condición relacionada. Por ejemplo la Figura 2.6 muestra un modelo donde hay un uso incorrecto de las etiquetas en la compuerta de tipo *or* y los flujos de secuencias provenientes de dicha compuerta. Al evaluar las medidas relacionadas con el etiquetado de compuertas en este modelo se obtuvo como resultado que el número de compuertas con etiquetas incorrectas es uno ($S_{CSE}(G) = 1$) y la razón entre el número de compuertas con etiquetas incorrectas y la cantidad total de compuertas también es uno ($CSE_C(G) = 1$). De forma similar se evaluaron las medidas relacionadas con el etiquetado de flujos de secuencia de salida y se obtuvo que $S_{SFSE}(G) = 2$ y $SFSE_S(G) = 1$.

Sin embargo en la Figura 2.7 se muestra un uso correcto de estas etiquetas. Al evaluar las medidas relacionadas con el etiquetado de compuertas en este modelo se obtuvo como resultado que el número de compuertas con etiquetas incorrectas es cero ($S_{CSE}(G) = 0$) y la razón entre el número de compuertas con etiquetas incorrectas y la cantidad total de compuertas también es cero ($CSE_C(G) = 0$). De forma similar se evaluaron las medidas relacionadas con el etiquetado de flujos de secuencia de salida y se obtuvo que $S_{SFSE}(G) = 0$ y $SFSE_S(G) = 0$.

Etiquetar las compuertas de división de tipo *or* y de tipo *xor*, así como los flujos de secuencia de salida de este tipo de compuertas permite mejorar la calidad pragmática y empírica de los modelos.

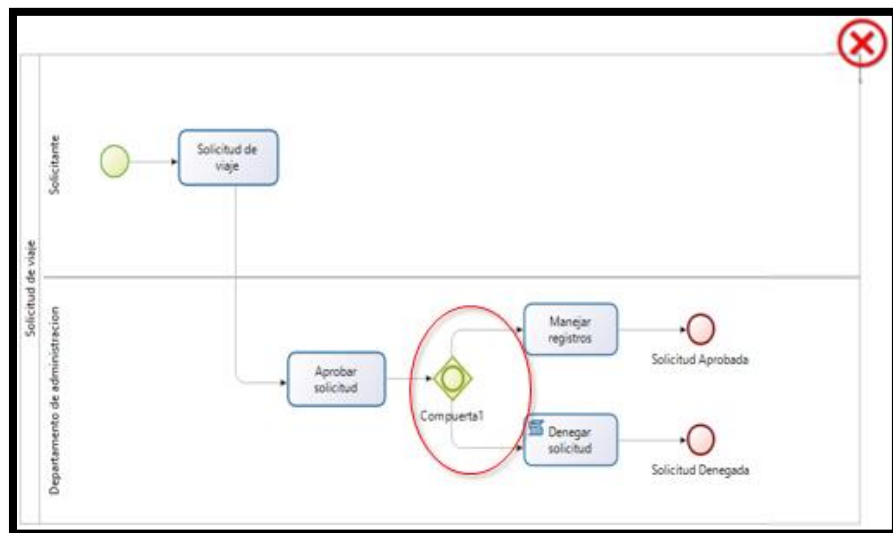


Figura 2.6 Uso incorrecto del etiquetado de compuertas y flujo de secuencias.

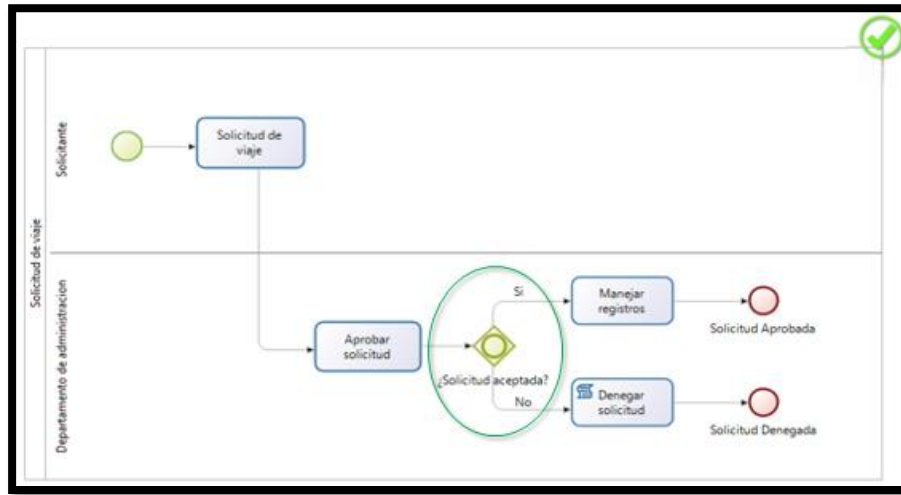


Figura 2.7 Uso correcto del etiquetado de compuertas y flujo de secuencias.

Problema 8 (Pool sin etiquetar): Los contenedores del proceso no se encuentran etiquetados.

Directriz unificada: Etiquete los contenedores de procesos.

Medidas propuestas:

Definición: Falta etiqueta en el *pool*.

Medida: $PSE(G) = \begin{cases} 1 & : \text{si el pool no tiene etiqueta} \\ 0 & : \text{en otro caso} \end{cases} \quad (14)$

Los nombres de los contenedores deben describir claramente el propósito principal del proceso. Además, debe asegurarse de no utilizar abreviaturas para etiquetar este tipo de elemento. El hecho de etiquetar los contenedores de procesos permite mejorar la calidad empírica de los modelos.

Problema 9 (Etiquetas largas): Las etiquetas de los elementos son largas.

Directriz unificada: Use etiquetas cortas.

Medidas propuestas:

Se proponen dos medidas inspiradas en grado promedio de las compuertas y en grado máximo de las compuertas (Mendling, 2008b), para la cuantificación relacionada con el largo de las etiquetas.

Definición: Longitud promedio en términos de cantidades de palabras de las etiquetas del modelo de proceso.

Medida: $\bar{l}_{Etiquetas}(G) = \frac{1}{|N|} \sum_{e \in Etiquetas} l(e) \quad (15)$

Definición: Largo máximo de una etiqueta del modelo de proceso.

Medida: $\hat{l}_{Etiquetas}(G) = \max\{l(e) | e \in Etiquetas\}$ (16)

La longitud de las etiquetas de los elementos está negativamente relacionada con la comprensión de los modelos. El uso de etiquetas cortas permite el diseño de un modelo menos complejo, lo cual contribuye a una mejor comprensión. El uso de esta directriz facilita la mejora de la calidad empírica, la calidad pragmática y la calidad semántica percibida de los modelos de procesos de negocio. La Figura 2.8 muestra un modelo con etiquetas cortas, el resultado de evaluar las medidas relacionadas a este problema es: $\bar{l}_{Etiquetas}(G) = 1.6$ y $\hat{l}_{Etiquetas}(G) = 2$.

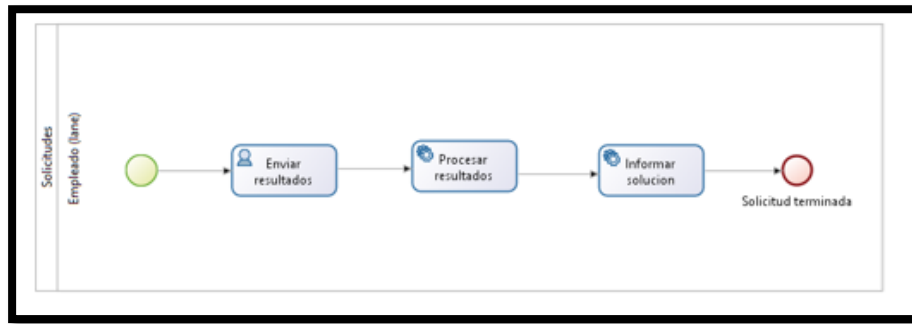


Figura 2.8 Modelo con presencia de etiquetas cortas.

2.3.2 Directrices prácticas relacionadas con el diseño visual y medidas asociadas

Los aspectos relacionados con el diseño de los modelos son importantes para una mejor comprensión y legibilidad de estos. A continuación se presentan cada una de las medidas asociadas a las directrices relacionadas con el diseño visual.

Problema 10 (Alto número de cruces de flujos de secuencia): El modelo contiene un alto número de cruces de flujos de secuencia.

Directriz unificada: Minimice el número de cruces de flujos de secuencia.

Medidas asociadas:

La directriz sobre el número de cruces carece de una medida asociada en la literatura que permita tener una noción sobre la ocurrencia del problema en los modelos de procesos de negocio. Por esto se proponen las siguientes medidas:

Definición: Número cruces de flujos de secuencia en el modelo

Medida: $S_{CA}(G) = |CA|$ (17)

Definición: Razón entre el número de cruces de flujos de secuencia y el número de flujos de secuencia en el modelo.

Medida: $CA_A(G) = \frac{|CA|}{|A|}$ (18)

Un mayor número de cruces de flujos de secuencia en un modelo de procesos de negocio se traduce en una mayor dificultad para el ojo humano para encontrar a cuáles elementos están conectados. Estos aspectos, aunque no están relacionados con la semántica del modelo, influyen decisivamente en la comprensión que pueden tener del mismo los involucrados en la documentación e interpretación de los procesos de negocio. No se cuenta con un umbral en la literatura relacionado con el número de cruces de flujos de secuencia que un modelo puede contener sin que se afecte su legibilidad. Disminuir el número de cruces de flujo de secuencia en los modelos influye positivamente en una mayor calidad empírica y en una mayor calidad pragmática de los mismos. La Figura 2.9 muestra un modelo donde hay presencia de cruces de arcos. Al evaluar las medidas relacionadas a este problema los resultados son: $S_{CA}(G) = 2$ y $CA_A(G) = 0.25$.

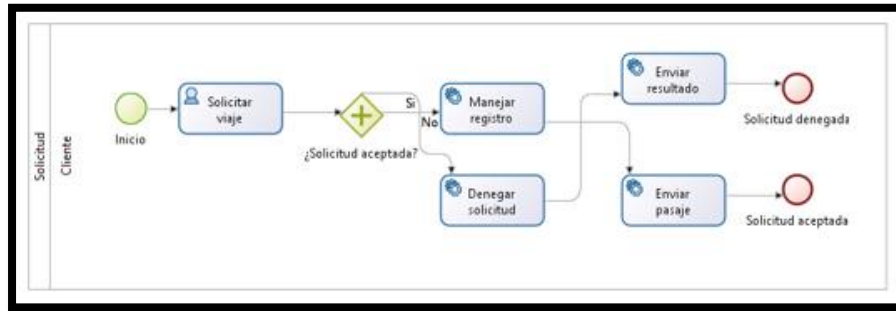


Figura 2.9 Modelo con presencia de cruces y curvas en los arcos.

Problema 11 (Curvas en los flujos de secuencia): Los flujos de secuencia contienen alto número de curvas.

Directriz unificada: Minimice las curvas en los flujos de secuencia.

Medidas propuestas:

Definición: Número de curvas en los flujos de secuencia del modelo.

Medida: $S_{SCurvas}(G) = |SCurvas|$ (19)

Definición: Razón entre el número de curvas en los flujos de secuencia y el número de flujos de secuencia.

Medida: $SCurvas_A(G) = \frac{|SCurvas|}{|A|}$ (20)

El número de curvas en los flujos de secuencias afectan negativamente la compresión del modelo de proceso. Minimizar el número de curvas en los arcos o por lo menos evitar las curvas innecesarias marcan una pauta relevante para el diseño general de un diagrama. De forma general, los autores en (Basili and Rombach, 1988) concluyen que un número bajo de curvas en los arcos juegan un importante papel en la percepción y el entendimiento de los modelos. Disminuir el número de curvas en los flujos de secuencia en los modelos, influye positivamente en una mayor calidad empírica y en una mayor calidad pragmática de los mismos. La Figura 2.9 muestra un modelo donde hay presencia de curvas en los arcos. Al evaluar las medidas relacionadas a este problema en dicho modelo los resultados son: $S_{SCurvas}(G) = 6$ y $SCurvas_A(G) = 0.75$.

Problema 12 (Solapamiento de flujos de secuencia): Los flujos de secuencia se solapan y carecen de unión intencional.

Directriz unificada: Evite solapamientos de flujos de secuencia.

Medidas propuestas:

Definición: Número de solapamientos de flujos de secuencia.

Medida: $S_{SAA}(G) = |SAA|$ (21)

Definición: Razón entre el número de solapamientos de flujos de secuencia y el número de flujos de secuencia.

Medida: $SAA_A(G) = \frac{|SAA|}{|A|}$ (22)

En esta tesis, al referir el solapamiento de arcos se deben analizar dos casos principalmente. El primero consiste en aquellos arcos que se solapan, pero con la intención de unirse, en este caso no se tiene en cuenta el solapamiento. Por otra parte, en el segundo caso el modelo contiene arcos que tienen más de un segmento del camino en común. En este último caso sí se viola la directriz, pues los segmentos solapados de los arcos no son visibles como caminos individuales, ocasionando ilegibilidad en el diagrama. Evitar el solapamiento de arcos permite mejorar la calidad empírica del modelo. La Figura 2.10 muestra un modelo donde hay presencia de solapamiento de arcos. Al evaluar las medidas relacionadas a este problema en dicho modelo los resultados son: $S_{SAA}(G) = 1$ y $SAA_A(G) = 0.12$.

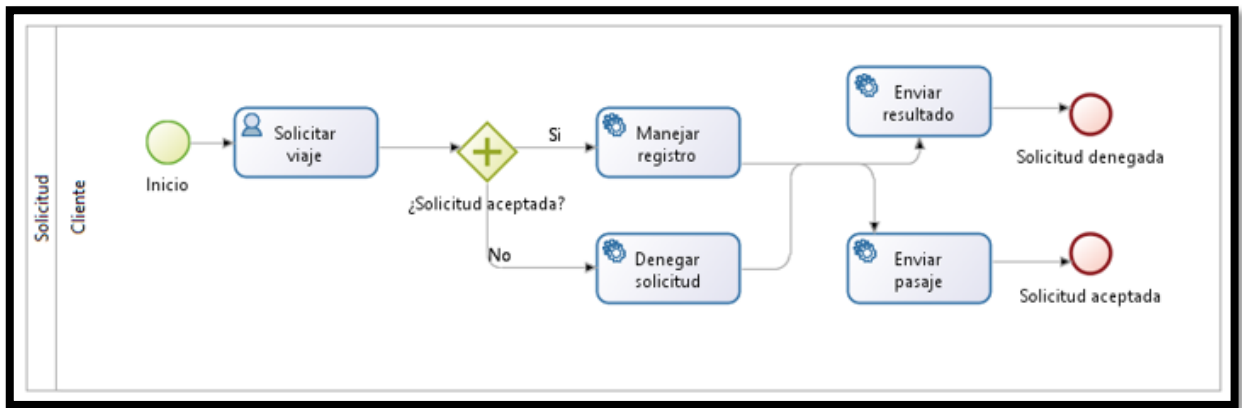


Figura 2.10 Modelo con presencia de solapamiento.

Problema 13 (Flujo de secuencia inconsistente): La dirección de los flujos de secuencia del modelo es inconsistente⁴.

Directriz unificada: Mantenga consistente la dirección de los flujos de secuencia.

Medidas propuestas:

Definición: Número de flujos de secuencia con dirección inconsistente.

Medida: $S_{FSI}(G) = |FSI|$ (23)

Definición: Razón de flujos de secuencia con dirección inconsistente y el número de flujos de secuencia.

Medida: $FSI_A(G) = \frac{|FSI|}{|A|}$ (24)

Las construcciones gráficas y el diseño estructural del modelo influyen en la visibilidad lógica del proceso en el diagrama, así como en el modo en que el modelo de proceso es comprendido por el usuario. La dirección de los flujos de secuencia individuales debe ser consistente con la dirección general del diagrama, se debe hacer corresponder la dirección de los flujos de secuencia del modelo con la dirección general del mismo, es decir, de izquierda a derecha o de arriba hacia abajo. Mantener una secuencia de tiempo y una dirección de flujo constante facilita la lectura y la comunicación eficiente. Además, permite mejorar la calidad pragmática y empírica del modelo.

Por ejemplo la Figura 2.11 muestra un modelo donde hay presencia de flujo de secuencia inconsistente evitable. En este caso el valor obtenido para la medida $S_{FSI}(G) = 2$ y para la

⁴ Según la RAE, una de las acepciones del término inconsistente es falta de coherencia entre los elementos de un conjunto.

medida $FSI_A(G) = 0.5$. Sin embargo en la Figura 2.12 se muestra un modelo que no presenta problema asociado a la dirección de los flujos de secuencia. De forma similar el valor obtenido para la medida $S_{FSI}(G) = 0$ y para la medida $FSI_A(G) = 0$.

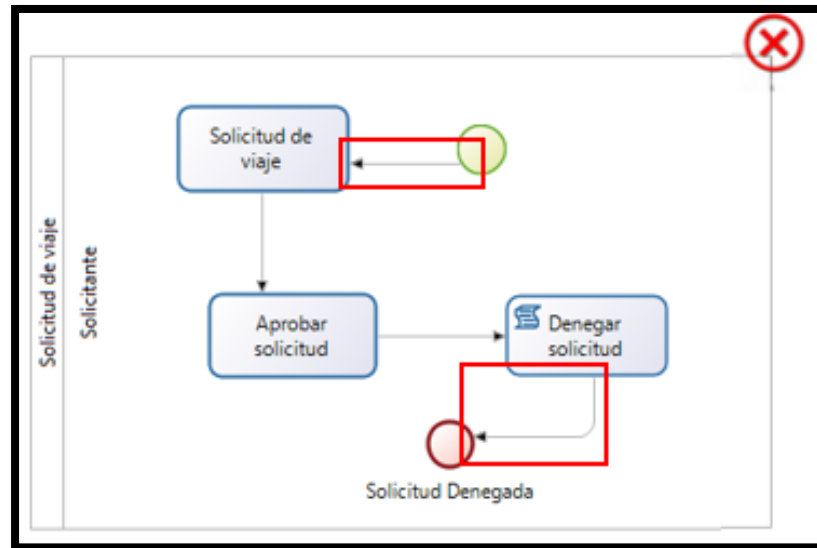


Figura 2.11 Modelo con flujo de secuencia inconsistente evitable.

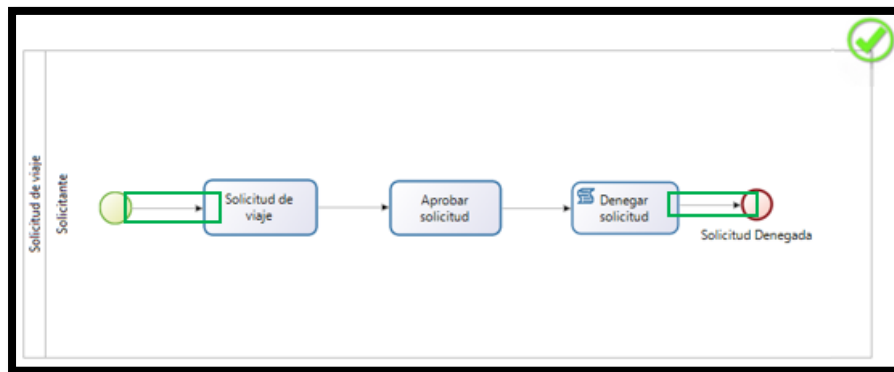


Figura 2.12 Modelo que no presenta problema asociado a la dirección de los flujos de secuencia.

Existe una mayor posibilidad de que un modelo con problemas de cruces de flujos de secuencia y solapamiento sea ilegible, que un modelo que no presente estos problemas. Sin embargo, un modelo puede presentar alguno de ellos y aun así, ser considerado ordenado o uniforme por los usuarios.

Problema 14 (Solapamiento de flujos de secuencia y elementos): Algún flujo de secuencia o segmento de flujo de secuencia solapa algún elemento.

Directriz unificada: Evite solapamientos entre flujos de secuencia y otros elementos del modelo.

Medida propuesta:

Definición: Número de solapamientos de flujos de secuencia con elementos.

Medida: $S_{SAN}(G) = |SAN|$ (25)

Problema 15 (Longitud variada de flujos de secuencia): La longitud de los flujos de secuencia del modelo no es uniforme⁵.

Directriz unificada: Mantenga uniforme la longitud de los flujos de secuencia.

Medidas propuestas:

Definición: Número de flujos de secuencia de longitud no uniforme.

Medida: $S_{LD}(G) = |LD|$ (26)

Definición: Razón de flujos de secuencia no uniformes y el número de flujos de secuencia.

Medida: $LD_A(G) = \frac{|LD|}{|A|}$ (27)

Al evaluar la medida asociada al número de solapamientos de flujos de secuencia con elementos en el modelo de la Figura 2.12, tenemos que el valor de $S_{SAN}(G) = 0$, pues no hay presencia de solapamiento entre arcos y elementos. De forma similar ocurre al evaluar las medidas relacionadas con la longitud variada de flujos de secuencias, pues en dicho modelo se puede apreciar una uniformidad en la longitud de los arcos y por tanto $S_{LD}(G) = 0$ y $LD_A(G) = 0$.

Problema 16 (Tamaño no uniforme de los elementos): El tamaño de los elementos del modelo no es uniforme.

Directriz unificada: Use un tamaño uniforme para sus elementos.

Medidas propuestas:

Definición: Número de elementos con tamaño no uniforme.

Medida: $S_{NTI}(G) = |NTI|$ (28)

Definición: Razón entre el número de elementos con tamaño no uniforme y el número de elementos.

Medida: $NTI_N(G) = \frac{|NTI|}{|N|}$ (29)

Mantener un formato único a lo largo de los diagramas y enfocarse en una apariencia limpia y agradable, suele ser una buena práctica de modelación. El uso de diferentes tamaños de fuente,

⁵ Según la RAE, dicho de dos o más cosas que tienen igual forma ||2. Igual, conforme, semejante.

colores, dimensiones de cajas o etiquetas superpuestas podrían hacer que la lectura de los diagramas sea un desafío. Por eso mantener un tamaño uniforme de los elementos del modelo permite mejorar la calidad pragmática y empírica de los mismos.

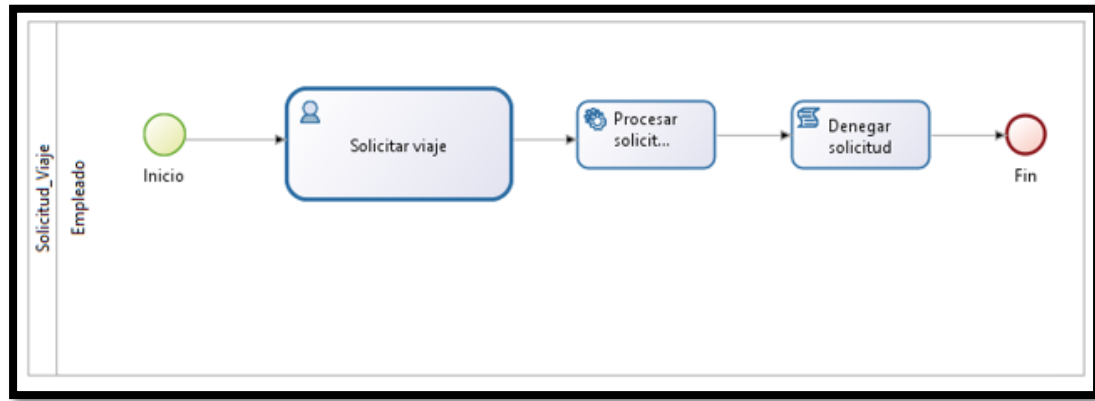


Figura 2.13 Modelo con presencia de tamaño no uniforme.

La Figura 2.13 muestra un modelo donde hay presencia de tamaño no uniforme. Al evaluar las medidas relacionadas a este problema en dicho modelo los resultados son: $S_{NTI}(G) = 2$ y $NTI_N(G) = 0.67$.

Problema 17 (Lejanía de elementos relacionados): Existen flujos de secuencia largos entre elementos relacionados directamente⁶.

Directriz unificada: Coloque los elementos relacionados próximos entre sí.

Medida propuesta:

Definición: Número de flujos de secuencia largos.

Medida: $S_{FSL}(G) = |FSL|$ (30)

Donde FSL es el número de flujos de secuencia largos en el modelo.

Problema 18 (Modelo cuadrado): La forma del modelo es cuadrada en lugar de rectangular.

Directriz unificada: Haga sus modelos largos y delgados, en lugar de cuadrados.

Medida propuesta:

Definición: Razón entre el alto y el largo del contenedor del proceso.

Medida: $AL(G) = \frac{ap}{lp}$ (31)

⁶ Un elemento se relaciona de forma directa con elementos pertenecientes a su anterioridad o posterioridad.

Problema 19 (Área de dibujo extensa): El área de dibujo contiene espacios en blanco que pudieran contener elementos.

Directriz unificada: Minimice el área de dibujo que contiene el modelo.

Medida propuesta:

Definición: Área sin utilizar.

Medida: $AD(G) = ap \times lp - \sum_{n \in N} ar(n)$ (32)

Para evaluar las medidas relacionadas al **Problema 18** y **Problema 19**, se usó el modelo de la Figura 2.13. El resultado para la razón entre el alto y el largo del contenedor del proceso es $AL(G) = 0.22$. Mientras que el valor de calcular el área sin utilizar es $AD(G) = 256578.0$.

Las directrices propuestas pueden resumirse en dos directrices más generales:

Problema 20 (Apariencia no uniforme): La apariencia del modelo no es uniforme para respaldar la homogeneidad, por ejemplo, elementos distribuidos sin simetría, dibujados con distintos tamaños.

Directriz unificada: Use un estilo uniforme para la distribución de elementos y flujos de secuencia.

Medidas asociadas: Se asocian las medidas (26)-(29) que están relacionadas con la uniformidad de los elementos del modelo.

Problema 21 (Diagrama desordenado): El modelo no está ordenado⁷.

Directriz unificada: Mantenga el diagrama tan ordenado como sea posible.

Medidas asociadas: Se asocian las medidas (17)-(32) que están relacionadas con la organización de los elementos del modelo.

2.3.3 Ejemplo de aplicación de las directrices y las medidas asociadas.

Para ilustrar el uso de las directrices y el cálculo de las medidas definidas para modelos de proceso de negocio, se proporciona un modelo diseñado en el software de modelación Bonita, el cual describe el proceso de denuncia en una unidad de la policía, como se muestra en la Figura 2.14.

⁷ Una acepción del término orden, según la RAE, es que guarda buena disposición de las cosas entre sí.

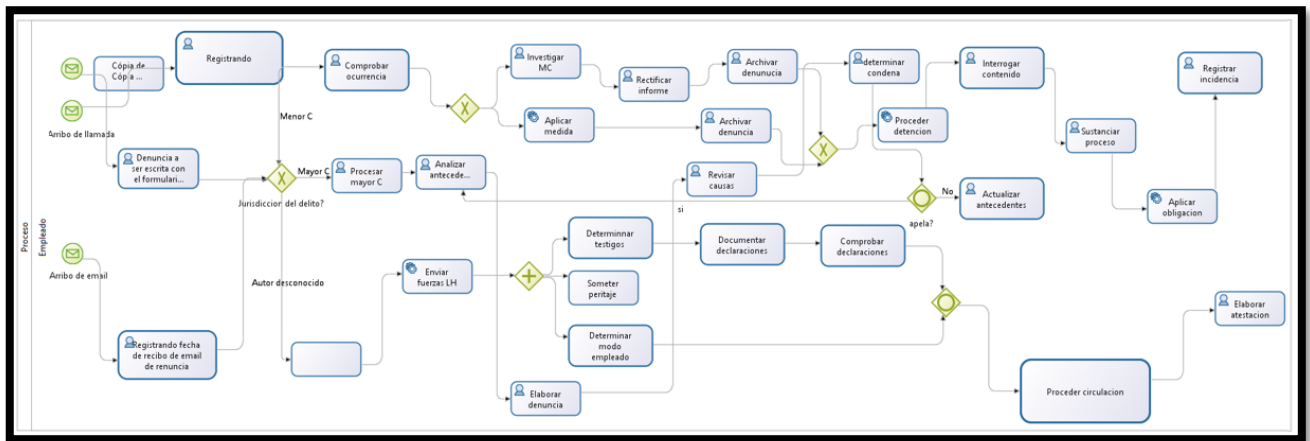


Figura 2.14 Modelo BPMN: Proceso de denuncia.

Los valores obtenidos de las distintas medidas definidas en este trabajo se pueden observar en la Tabla 2.5 y Tabla 2.6.

Tabla 2.5 Valor de medidas y problema asociado – Estilo de etiquetado.

Problema asociado a las medidas	Nombre Medida	Valor
Etiquetas no siguen el estilo verbo-sustantivo	EI	5
	EI_F	0.17
Tipos de tareas no especificados	FST	8
	FST_F	0.27
Etiquetas enviar/recibir no están reservadas para estos elementos	ERI	1
Actividades sin etiquetar	FSE	2
	FSE_F	0.07
Eventos sin etiquetar	ESE	1
	ESE_F	0.33
Compuertas sin etiquetar	CSE	3
	CSE_C	0.6
Flujos de secuencia de salida sin etiquetar	$SFSE$	4
	$SFSE_s$	0.44
Etiquetas largas	$\bar{l}_{Etiquetas}$	2.02
	$\hat{l}_{Etiquetas}$	5

Tabla 2.6 Valor de medidas y problema asociado – Diseño visual.

Problema asociado a las medidas	Nombre Medida	Valor
Alto número de cruces de flujos de secuencia	CA	10
	CA_A	0.25
Curvas en los flujos de secuencia	$SCurvas$	56
	$SCurvas_A$	1.4
Solapamiento de flujos de secuencia	SAA	1
	SAA_A	0.025
Flujo de secuencia inconsistente	FSI	15
	FSI_A	0.37
Solapamiento de flujos de secuencia y elementos	SAN	3
Tamaño no uniforme de los elementos	NTI	14
	NTI_N	0.47
Lejanía de elementos relacionados	FSL	4
Modelo cuadrado	AL	0.32
Área de dibujo extensa	AD	939424
Longitud variada de flujos de secuencia	$S_{LD}(G)$	0
	$LD_A(G)$	0

2.4 Validación teórica de las medidas propuestas

En este epígrafe se describe el resultado de la validación teórica realizada sobre las medidas de acuerdo al marco de Briand (Briand et al., 1996).

Para realizar la definición de sistema, módulo, elemento y relación de acuerdo a este marco es importante considerar que las medidas se definen sobre modelos de procesos de negocio compuestos por actividades, eventos, nodos de decisión, así como de flujos de secuencia entre estos elementos. En función de las características de cada medida y de los atributos del modelo de procesos de negocio sobre los que se aplican, se definirá para su validación de acuerdo a este marco lo que se entiende por sistema, módulo, elemento y relación.

Al llevar a cabo la validación teórica bajo este marco se encontró que las medidas propuestas se pueden caracterizar en tres atributos: de tamaño, de acoplamiento y de complejidad.

2.4.1 EI y EI_F como medidas de tamaño

De acuerdo al objetivo de estas medidas y en concordancia con el marco de Briand, se considera que el modelo de proceso de negocio (sistema) está compuesto por actividades (elementos) y flujos de secuencia (relaciones). Un módulo está compuesto por un subconjunto de actividades y los flujos de secuencia entre ellas.

Se demuestra primeramente que la medida EI (número de actividades con etiquetas incorrectas) cumple con los axiomas que caracterizan a las medidas de tamaño:

Propiedad 1: No negatividad

El número de actividades con etiquetas incorrectas es siempre mayor o igual a cero, por lo que EI nunca puede ser negativa ($EI \geq 0$).

Propiedad 2: Valor Nulo

Si no hay actividades con etiquetas incorrectas, entonces $EI = 0$.

Propiedad 3: Aditividad de módulos

Si consideramos que un modelo de proceso de negocio está compuesto por módulos que no tienen actividades en común, el número de actividades siempre será la suma del número de actividades de sus módulos. Por tanto, el número de actividades con etiquetas incorrectas de un modelo es la suma del número de actividades con etiquetas incorrectas de sus módulos.

De forma similar, se demuestra que la medida EI_F (razón entre el número de actividades con etiquetas incorrectas y el total de actividades) también cumple con los axiomas que caracterizan a las medidas de tamaño.

Propiedad 1: No negatividad

Como se demostró anteriormente EI cumple con la propiedad de no negatividad al igual que F (total de actividades), por tanto $\frac{EI}{F}$ siempre es mayor o igual a cero. Luego, EI_F nunca puede ser negativa ($EI_F \geq 0$).

Propiedad 2: Valor nulo

Pueden ocurrir dos casos:

- 1er Caso: $EI = 0$ y $F \neq 0$

En este caso se cumple que $\frac{0}{F} = 0$, por tanto $EI_F = 0$.

- 2do Caso: $F = 0$

Se debe tener en cuenta que la división por cero no está definida, pero en este caso se asume que si $F = 0$, entonces EI también es cero. Esto está dado por el hecho de que si $F = 0$ no existen actividades en el modelo y por tanto se asume que EI_F también es cero. Luego se puede afirmar que EI_F también cumple esta propiedad.

Propiedad 3: Aditividad de módulos

Si se considera que un modelo está compuesto por módulos que no tienen actividades en común, el número de actividades siempre será la suma del número de actividades de sus módulos. De esta forma, la razón entre el número de actividades con etiquetas incorrectas de un módulo y el total de actividades del modelo sería $EI_{F1} = \frac{EI_1}{F}$. De forma similar la razón entre el número de actividades con etiquetas incorrectas de otro módulo y el total de actividades del modelo sería $EI_{F2} = \frac{EI_2}{F}$. Luego, se tiene que:

$$EI_{F1} + EI_{F2} = \frac{EI_1}{F} + \frac{EI_2}{F} = \frac{EI_1 + EI_2}{F} = \frac{EI}{F} = EI_F$$

Por tanto EI_F también cumple con esta propiedad. Entonces se puede afirmar que EI_F es una medida de tamaño.

Siguiendo el razonamiento análogo al realizado para la medida base EI , se demuestra que las medidas base: $FST, F, FSE, ERI, l, Etiquetas, CSE, Sor, Sxor, ESE, E, NTI$ también son medidas de tamaño. De forma similar, se demuestra también que las medidas derivadas: $FST_F, ESE_E, FSE_F, CSE_C, NTI_N$ también son medidas de tamaño.

2.4.2 CA como medida de complejidad

Se demuestra que la medida CA (número de cruces de flujos de secuencias) cumple con los axiomas que caracterizan a las medidas de complejidad:

Propiedad 1: No negatividad

El número de cruces de flujos de secuencias es siempre mayor o igual a cero, por lo que CA nunca puede ser negativa ($CA \geq 0$).

Propiedad 2: Valor nulo

Si no hay cruces entre los flujos de secuencias, entonces $CA = 0$.

Propiedad 3: Simetría

El número de cruces entre los flujos de secuencias no depende de la convención usada para representar dichos cruces.

Propiedad 4: Monotonicidad de módulos

De acuerdo a la definición de esta propiedad, se demuestra que siendo M1 y M2 dos módulos cualquiera de un modelo que no tienen cruces de flujos en común, entonces se cumple que la complejidad del modelo no es menor que la suma de las complejidades de los módulos M1 y M2, es decir $CA(M12) \geq CA(M1) + CA(M2)$. Considérese el ejemplo de la Figura 2.15 en el que $CA(M1) = 1$, $CA(M2) = 1$ y $CA(M12) = 2$ lo que satisface este axioma.

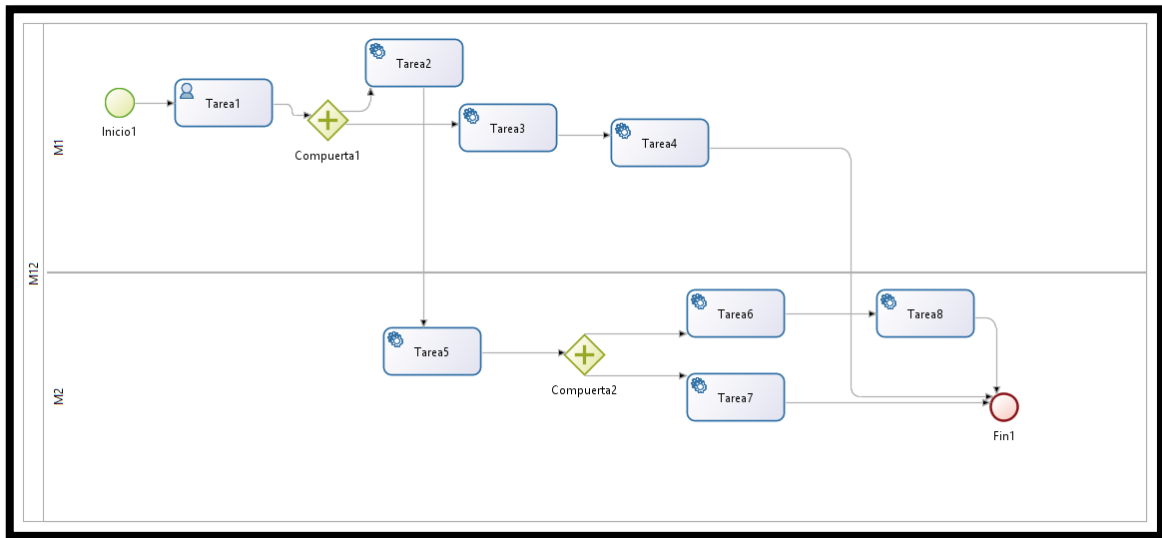


Figura 2.15 Modelo BPMN para la demostración de la propiedad 4 de complejidad.

Propiedad 5: Aditividad de módulos disjuntos

El número de cruces de flujos de secuencias en un módulo obtenido mediante la unión de dos módulos disjuntos, es igual a la suma del número de cruces de flujos de los dos módulos por separados. El cumplimiento de este axioma también se ilustra en la Figura 2.15, donde se dispone de los módulos disjuntos M1 y M2, siendo $M1=1$ y $M2=1$. Al unir estos dos módulos se genera un nuevo módulo M12 cuyo valor de CA es igual a 2, es decir, a la suma de $CA(M1)$ y $CA(M2)$.

Siguiendo el razonamiento análogo al realizado para la medida AC , se ha demostrado que las medidas: $SFSE$, SAN , $SFSE_s$, $SCurvas$, CA_A , $SCurvas_A$ también son medidas de complejidad.

2.4.3 *CA* como medida de acoplamiento

Para que *CA* sea una medida de acoplamiento debe cumplir con las cinco propiedades descritas en el epígrafe 1.6.2. En el epígrafe anterior se demostró que la medida *CA* cumple con las siguientes propiedades: No negatividad, valor nulo y aditividad de módulos disjuntos. Estas propiedades son las tres primeras propiedades que debe cumplir una medida para ser de acoplamiento. A continuación, se demuestran las otras dos propiedades.

Propiedad 4: Monotonicidad

Adicionar flujos de secuencias o cruces entre arcos de módulos distintos nunca decrementa el valor de *CA*.

Propiedad 5: Fusión de módulos

El número de cruces de arcos de un módulo obtenido como la unión de otros dos módulos no es mayor que la suma del número de cruces de arcos en cada módulo.

Siguiendo el mismo razonamiento, se ha demostrado que las medidas: *SFSE*, *SFSE_s*, *SCurvas*, *SAA*, *FSI*, *LD_A*, *FSL*, *SAN*, *CA_A*, *SCurvas_A*, *SAA_A*, *FSI_A*, *LD_A* también son medidas de complejidad.

La definición de las medidas permitió investigar la relación entre la representación visual y la comprensión y modificabilidad como atributos externos de los modelos de procesos de negocio representados con BPMN. Al efectuar la validación teórica de las medidas definidas fue posible agruparlas de acuerdo a tres propiedades de la calidad interna del modelo, como tamaño, acoplamiento y complejidad, como se muestra en la Figura 2.16.

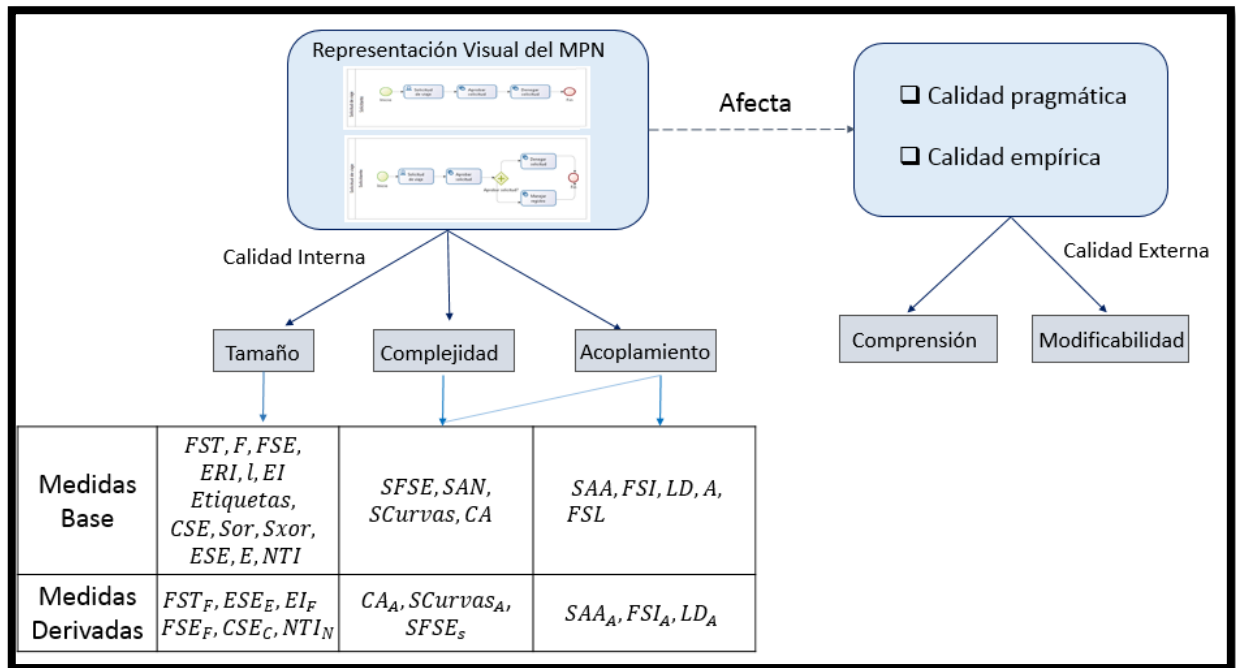


Figura 2.16 Relación entre la representación visual y los atributos de calidad.

Con los resultados de la validación teórica realizada de acuerdo al marco formal de Briand (Briand et al., 1996), se obtuvo que un total de 19 medidas (base y derivadas) cumplen con los axiomas de la propiedad de tamaño, cuatro medidas base y tres medidas derivadas, cumplen con los axiomas que caracterizan a las medidas de complejidad y tres medidas derivadas y cinco medidas base cumplen con los axiomas de la propiedad de acoplamiento.

2.5 Consideraciones finales del capítulo

En este capítulo se definieron un conjunto de medidas de calidad que ofrecen una noción cuantitativa sobre problemas de calidad en los modelos. De estas, 21 están relacionadas con el estilo de etiquetado y 17 con el diseño visual. Estas medidas se validaron teóricamente con el marco formal de Briand y a su vez, este permitió clasificarlas de la siguiente forma: 19 fueron clasificadas como medidas de tamaño, siete como medidas de complejidad y ocho de acoplamiento.

IMPLEMENTACIÓN DE LAS MEDIDAS PARA LA
EVALUACIÓN DE LA REPRESENTACIÓN VISUAL EN
SAD_BPMN

3. IMPLEMENTACIÓN DE LAS MEDIDAS PARA LA EVALUACIÓN DE LA REPRESENTACIÓN VISUAL EN SAD_BPMN

El software SAD_BPMN proporciona una herramienta para la evaluación de modelos de procesos de negocio en BPMN, tomando como base directrices prácticas de calidad relacionadas con el tamaño y la morfología de los modelos. En este capítulo se presentan algunos aspectos de implementación de las directrices de etiquetado y diseño visual en la herramienta, así como el diagrama de paquetes y clases de diseño necesarias para el desarrollo del módulo de representación visual.

3.1 Módulo de representación visual en la herramienta SAD_BPMN

SAD_BPMN es una herramienta para la evaluación de modelos BPMN que ha sido concebida para recomendar a los usuarios cómo mejorar la calidad de sus modelos de procesos de negocio mediante directrices prácticas de calidad. La herramienta analiza un modelo de procesos de negocio permitiéndole al usuario obtener una información sobre los problemas relacionados con las mismas. Para ello interpreta los archivos de salida de las herramientas de modelación Bonita y Bizagi con extensiones *bpmn* y *xpdl* respectivamente. Estos archivos de salida se basan en XML, por lo que el trabajo en la herramienta se dirige principalmente a la interpretación de este tipo de archivo. En el sistema se encuentra implementado un módulo para el análisis de los modelos de procesos de negocio tomando como base directrices de complejidad general.

Con el objetivo de ofrecer las directrices a los modeladores a partir del análisis de los diagramas, se reutiliza el cálculo de medidas de calidad. Para las medidas, se comparan los valores obtenidos tras su aplicación con los valores de umbrales recolectados de la literatura. De no existir dichos umbrales, se ofrece al usuario la información referente a las medidas.

La interfaz visual de la aplicación, como se muestra en la Figura 3.1, se puede dividir en cuatro partes fundamentales:

1. Carga del modelo: es donde se selecciona el archivo que contiene el modelo de proceso de negocio que se pretende evaluar.

2. Contenedor de directrices a evaluar en el modelo: incluye las directrices de complejidad general que pueden sugerirse a partir de la evaluación de los modelos de procesos de negocio.
3. Panel que muestra resultados de la evaluación: sugiere las directrices prácticas que el modelador debe aplicar a sus modelos de procesos de negocio evaluados para incrementar su calidad.
4. Sección de exportación de los resultados hacia un fichero Excel y visualización en un fichero Word de la descripción detallada de las directrices prácticas como ayuda al modelador.

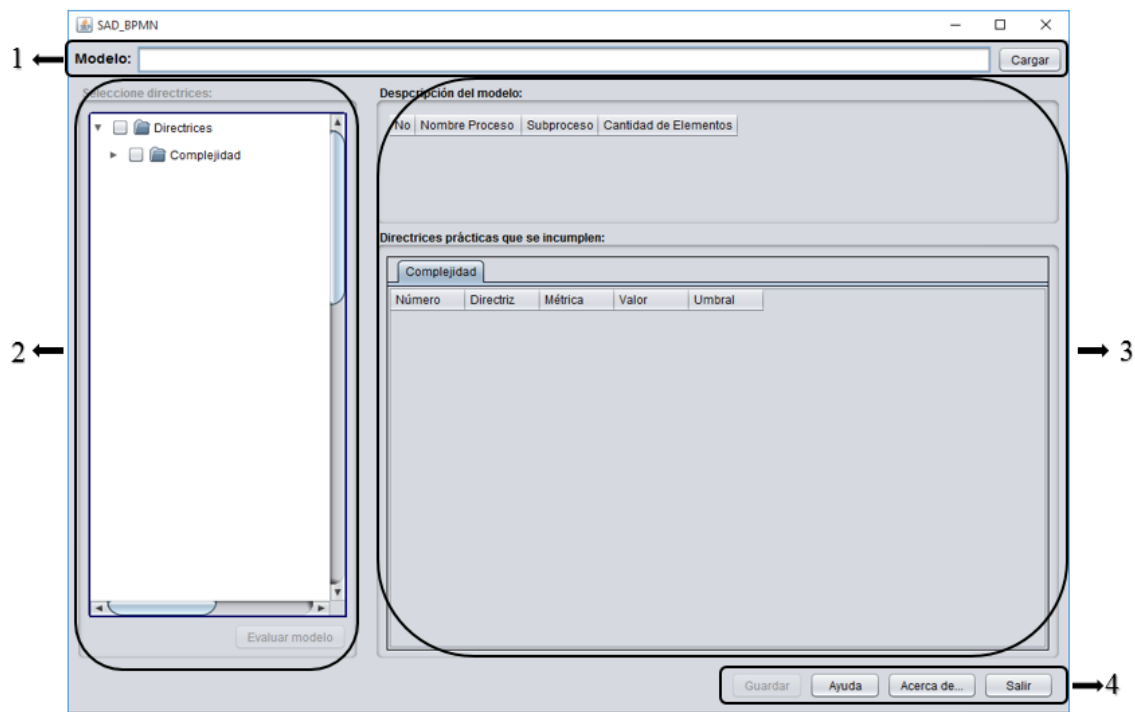


Figura 3.1 Interfaz gráfica de SAD_BPMN.

Con el fin de lograr una evaluación más completa de los modelos se decidió adicionar un nuevo módulo que permita la evaluación de los mismos a través de las directrices de representación visual. Para ello se utilizan las medidas propuestas anteriormente, lo cual permite tener una noción de la existencia de cada problema de representación visual descrito, relacionado a las directrices en el modelo.

3.1.1 Diseño de la herramienta.

Un diagrama de paquetes en el Lenguaje Unificado de Modelado (UML) representa las dependencias entre los paquetes que componen un modelo. Es decir, muestra cómo un sistema está dividido en agrupaciones lógicas y las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema. Se puede decir que existe dependencia entre varios paquetes cuando un elemento de un paquete requiere de otro elemento que pertenece a un paquete distinto. En la Figura 3.2 Diagrama de paquetes del Módulo de representación visual se muestra el diagrama de paquetes correspondiente al módulo de representación visual. El diagrama contiene solamente los paquetes que fueron necesarios para la realización de dicho módulo y las relaciones existentes entre los diferentes paquetes.

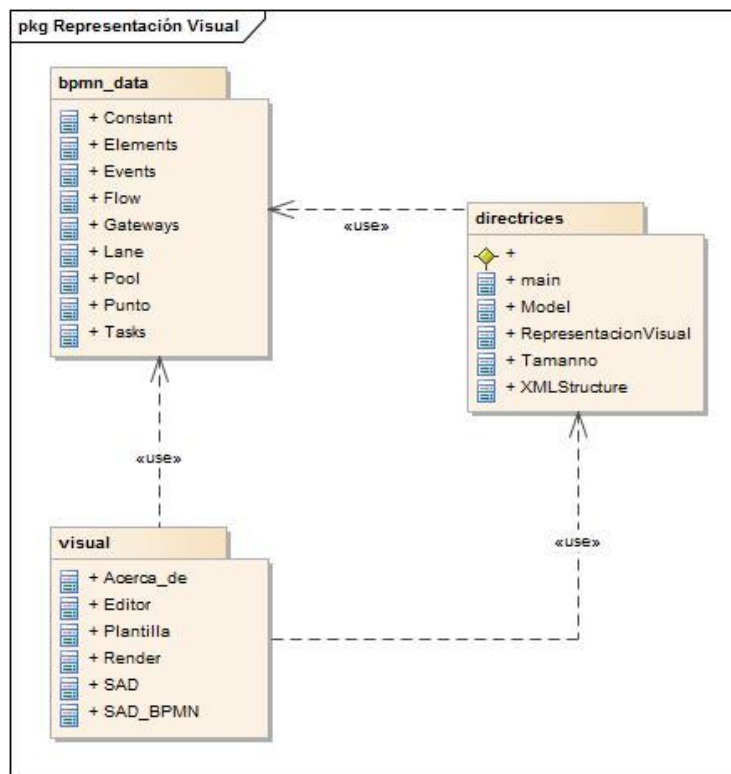


Figura 3.2 Diagrama de paquetes del Módulo de representación visual

Por otra parte, los diagramas de clases se utilizan para modelar la vista de diseño estática de un sistema. Un diagrama de clases es gráficamente una colección de nodos y arcos, que muestra un conjunto de clases, interfaces, colaboraciones y sus relaciones que pueden ser de dependencia,

generalización o asociación (OMG, 2005). Además pueden contener paquetes o subsistemas para agrupar elementos de un modelo en partes más grandes. A continuación se describe una selección de los paquetes que conforman el módulo de representación visual.

El paquete *Directrices* contiene las clases *Tamanno*, *XMLStructure*, *Model* y *RepresentacionVisual*. La clase *Tamanno* es la encargada de implementar las directrices relacionadas con el módulo de Complejidad General que ya estaba implementado en la herramienta SAD_BPMN. Las clases *XMLStructure* y *Model* brindan un conjunto de métodos necesarios para la extracción de los datos del modelo. Por otra parte, la clase *RepresentacionVisual* es una nueva clase que se adicionó al paquete para el desarrollo del módulo de representación visual. Esta clase es la encargada de implementar las directrices de calidad relacionadas con el estilo de etiquetado y de diseño visual, además de calcular todas las mediadas asociadas a estas directrices. En la Figura 3.3 se muestra el paquete *Directrices* con el diagrama de clases asociado al mismo.

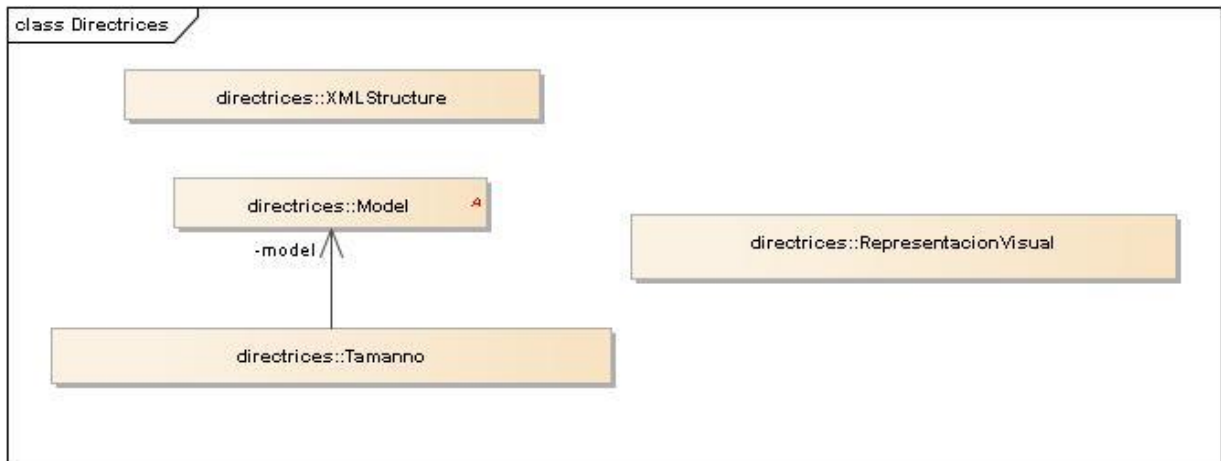


Figura 3.3 Diagrama del paquete *Directrices*

Otros paquetes utilizados en el desarrollo de este módulo son: el paquete *bpmn_data* y el paquete *visual*. En el paquete *bpmn_data* se implementa un conjunto de clases encargadas de almacenar la información de todos los componentes gráficos del modelo: *Pool*, *Gateways*, *Event*, *Task* y *Lane*. Mientras que en el paquete *visual* se implementan las clases relacionadas con el diseño visual de la herramienta. En la Figura 3.4 y Figura 3.5 se muestran las clases de los paquetes *bpmn_data* y *visual*, respectivamente.

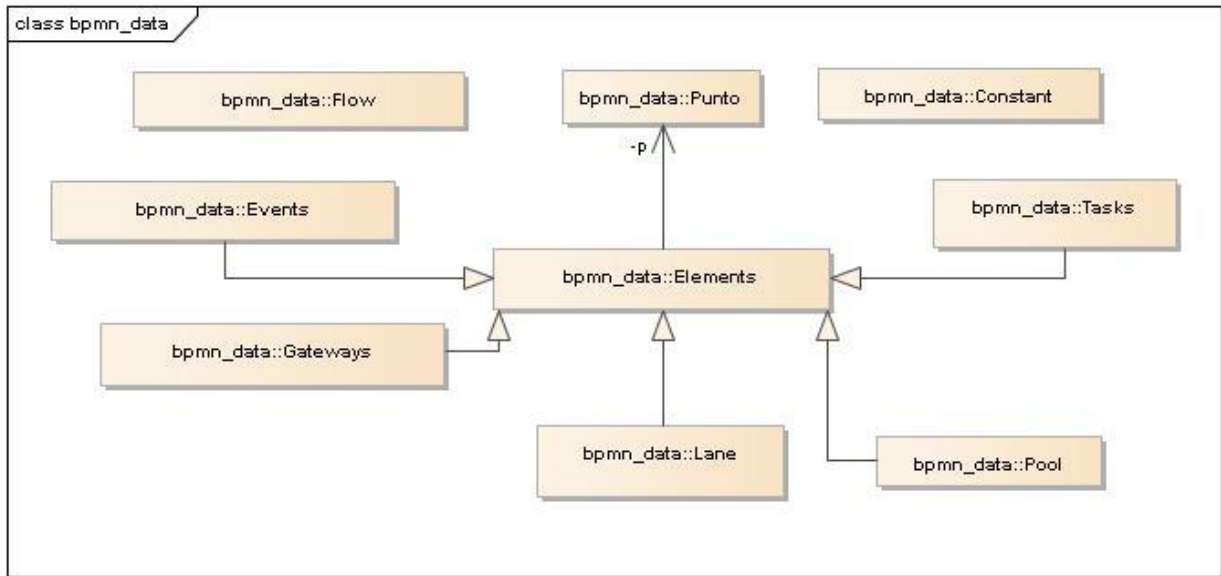


Figura 3.4 Diagrama del paquete *bpmn_data*.

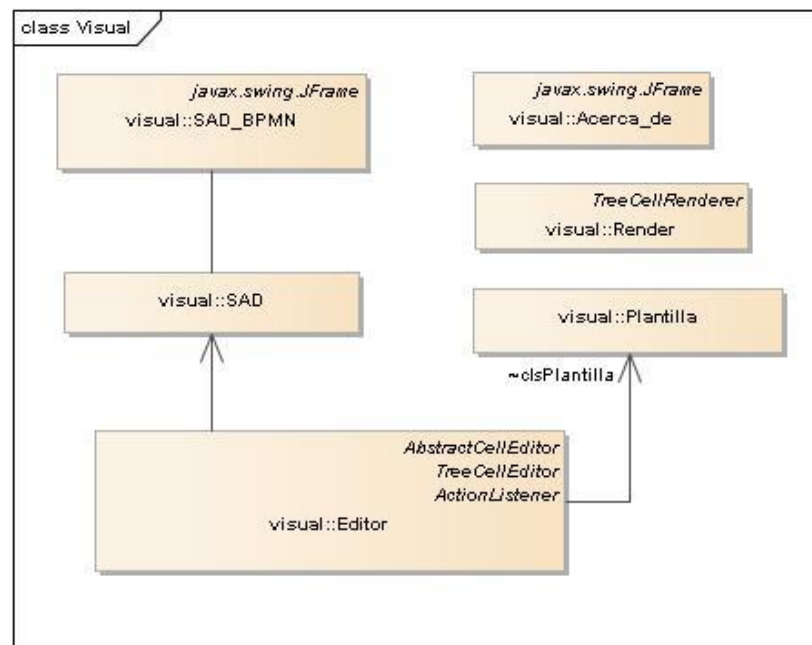


Figura 3.5 Diagrama del paquete *visual*

Finalmente, la Figura 3.6 muestra la interfaz visual de la herramienta SAD_BPMN con el módulo de Complejidad General (ya existente) y el nuevo módulo de representación visual. Este último módulo está dividido en dos grupos. El primer grupo incluye las directrices relacionadas con el estilo de etiquetado, las cuales tienen una gran importancia en la comprensión de los modelos. Estas directrices les brindan a los usuarios un conjunto de consejos relacionados con

la longitud de las etiquetas y la uniformidad de las mismas. Mientras que el segundo grupo incluye las directrices relacionadas con el diseño visual, las cuales también juegan un rol importante en la mejora de la comprensión de los modelos de procesos de negocio. Estas directrices prácticas se refieren a la generalización y conceptualización de mecanismos existentes para cambiar la representación visual de un modelo de procesos.

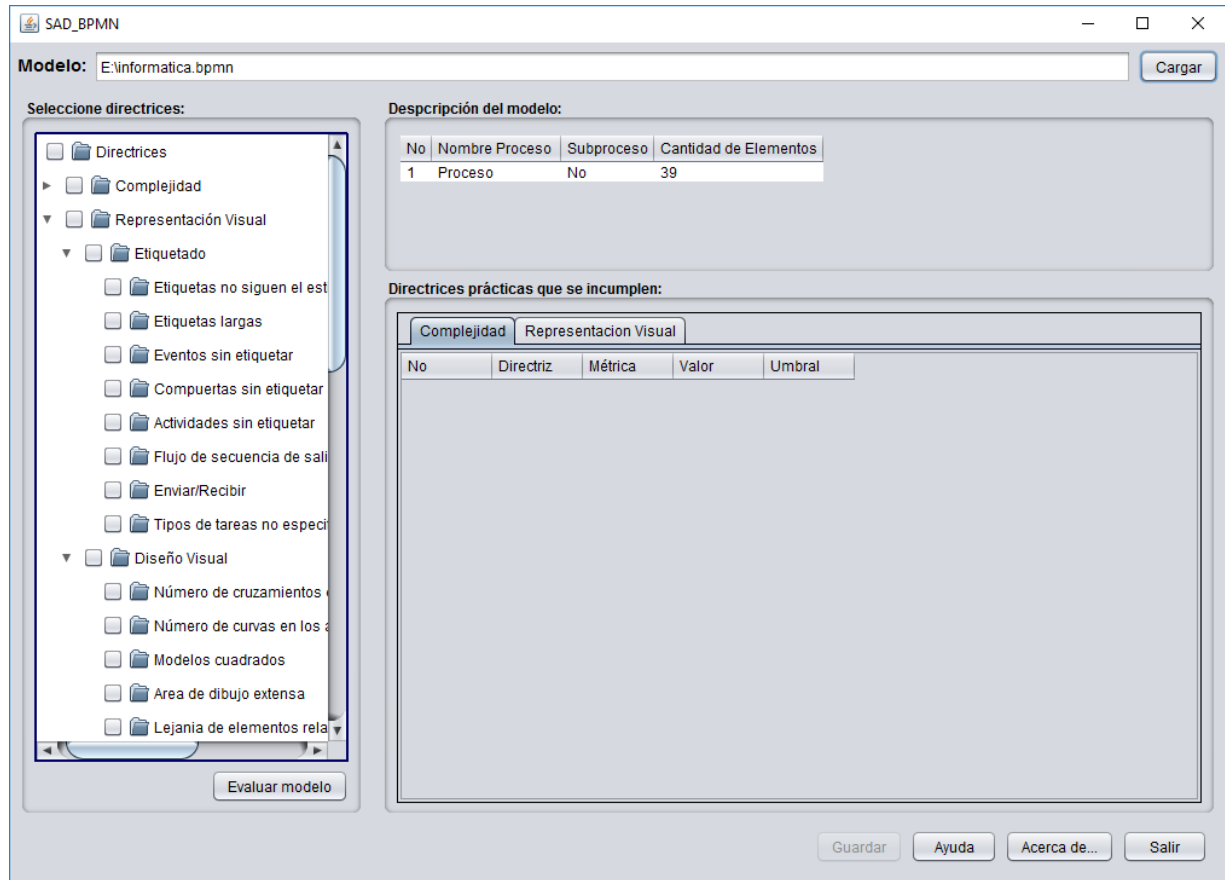


Figura 3.6 Interfaz visual de la herramienta SAD_BPMN con el módulo representación visual.

3.1.2 Implementación de las directrices de representación visual.

Para el desarrollo del módulo de representación visual se implementaron todas las medidas relacionadas con las directrices de etiquetado y de diseño visual. Como se expuso en el Capítulo 2, la mayoría de las directrices tienen dos medidas asociadas, una básica y una derivada. A continuación se proponen dos algoritmos que retornan el valor de la medida básica y el de la medida derivada. El primer algoritmo está relacionado con la directriz “*Alto número de cruces de flujos de secuencia*”, la cual es una directriz de diseño, mientras que el segundo algoritmo

corresponde a la directriz “*Eventos sin etiquetar*”, que se corresponde con una directriz de etiquetado. Ambas directrices son de representación visual de acuerdo a la clasificación taxonómica propuesta en (Moreno-Montes de Oca et al., 2015).

Algoritmo 1: ctd_cruces_arcos (Pool p), calcula la cantidad de cruces de arcos del modelo, recibe como parámetro el *pool* que se desea analizar. Este algoritmo devuelve un número entero que indica la cantidad de cruces y un número real que indica la proporción entre el número de cruces y la cantidad de arcos del modelo.

Los pasos del algoritmo son:

```
Paso 1. ctdCruces<-0
Paso 2. ArrayList<Flow> arcos<-p.getFlow()
Paso 3. para i<-0 hasta totalArcos incrementar i
    hacer
        para j<-i+1 hasta totalArcos incrementar j
            hacer
                si (arcos[i] es perpendicular a arcos[j]) entonces
                    si (arcos[i] y arcos[j] tienen un punto en común) entonces
                        incrementar ctdCruces
                    fin si
                fin si
            fin para
        fin para
Paso 4. razon<-ctdCruces/totalArcos
Paso 5. Retornar ctdCruces, razón
```

A continuación se detallan los pasos del algoritmo:

```
Paso 1. Se inicializa la variable ctdCruces en cero. La variable ctdCruces es la encargada de contar la cantidad de cruces que hay en el modelo.
Paso 2. La variable arcos recibe la lista de todos los arcos del modelo.
Paso 3. Se recorre la lista de arcos y se compara cada arco con los restantes, verificando si existen arcos perpendiculares. En caso afirmativo se verifica si tienen algún punto en común y se incrementa la variable ctdCruces.
Paso 4. Se calcula la razón entre la cantidad de cruces y el total de arcos del modelo.
```

Algoritmo 2: eventos_sin_etiquetar (Pool p), calcula la cantidad de eventos del modelo sin etiquetar, recibe como parámetro el *pool* que se desea analizar. Este algoritmo devuelve un número entero que indica la cantidad de eventos sin etiquetar y un número real que indica la proporción entre el número de eventos sin etiquetar y la cantidad de eventos del modelo.

Los pasos del algoritmo son:

```
Paso 1. ctdEventosSE<-0
Paso 2. ArrayList<Event> eventos<-p.getEvent()
Paso 3. para i<-0 hasta totalEventos incrementar i
    hacer
        si (eventos[i].nombre es vacío ||
            eventos[i].nombre contiene palabras_reservadas()) entonces
            incrementar ctdEventosSE
```

```
        fin si
    fin para
Paso 4. razon<-ctdEventosSE/totalEventos
Paso 5. Retornar ctdEventosSE, razón
```

A continuación se detallan los pasos del algoritmo:

Paso 1. Se inicializa la variable **ctdEventosSE** en cero. La variable **ctdEventosSE** es la encargada de contar la cantidad de eventos que no están etiquetados en el modelo.

Paso 2. La variable **eventos** recibe la lista de todos los eventos del modelo.

Paso 3. Se recorre la lista de eventos verificando si existe algún evento sin etiquetar. En caso afirmativo se incrementa la variable **ctdEventosSE**.

Paso 4. Se calcula la razón entre la cantidad de eventos sin etiquetar y el total de eventos del modelo.

El **Algoritmo 1** presenta en el peor de los casos una complejidad cuadrática $O(n^2)$, debido a que realiza una cantidad de **nxn** comparaciones. Esto se debe a la necesidad de comparar cada elemento de la lista con los restantes. Sin embargo, el **Algoritmo 2** presenta una complejidad lineal debido a que solo es necesario recorrer una vez la lista de eventos. Es decir, solo se ejecutarán **n** comparaciones, por lo que la complejidad es un $O(n)$. Los algoritmos correspondientes a las directrices “*Compuertas sin etiquetar*” y “*Actividades sin etiquetar*” tienen la misma estructura que el **Algoritmo 2** y por tanto la misma complejidad.

Una de las directrices que asocia una sola medida es “*Área de dibujo extensa*”. A continuación se muestra el algoritmo correspondiente a esta directriz.

Algoritmo 3: área_dibujo_extensa (Pool p), calcula el área del dibujo que contiene espacios en blanco que pudieran contener elementos, recibe como parámetro el *pool* que se desea analizar. Este algoritmo devuelve un número entero que indica el área libre que contiene el modelo.

Los pasos del algoritmo son:

```
Paso 1. areaTotalE<-0
Paso 2. areaTotalT<-0
Paso 3. areaTotalG<-0
Paso 4. ArrayList<Event> eventos<-p.getEvent()
Paso 5. ArrayList<Task> tareas<-p.getTask()
Paso 6. ArrayList<Gateway> compuertas<-p.getGateway()
Paso 7. para i<-0 hasta totalEventos incrementar i
        hacer
            areaE<-eventos[i].alto * eventos[i].largo
            areaTotalE<-areaTotalE + areaE
        fin para
Paso 8. para i<-0 hasta totalTareas incrementar i
        hacer
            areaT<-tareas[i].alto * tareas[i].largo
            areaTotalT<-areaTotalT + areaT
        fin para
```

```
Paso 9. para i<-0 hasta totalCompuertas incrementar i
      hacer
        areaG<-compuertas[i].alto * compuertas[i].largo
        areaTotalG<-areaTotalG + areaG
      fin para
```

```
Paso 10. areaTotal<-p.getAlto * p.getLargo
```

```
Paso 11. Retornar areaTotal - (areaTotalE + areaTotalT + areaTotalG)
```

A continuación se detallan los pasos del algoritmo:

Paso 1. Se inicializa la variable **areaTotalE** en cero. La variable **areaTotalE** es la encargada de acumular el área de cada evento.

Paso 2. Se inicializa la variable **areaTotalT** en cero. La variable **areaTotalT** es la encargada de acumular el área de cada actividad o tarea.

Paso 3. Se inicializa la variable **areaTotalG** en cero. La variable **areaTotalG** es la encargada de acumular el área de cada compuerta.

Paso 4. La variable **eventos** recibe la lista de todos los eventos del modelo.

Paso 5. La variable **tareas** recibe la lista de todas las tareas del modelo.

Paso 6. La variable **compuertas** recibe la lista de todas las compuertas del modelo.

Paso 7. Se recorre la lista de los eventos calculando el área de cada uno y luego se almacena en la variable **areaTotalE**.

Paso 8. Se recorre la lista de las tareas calculando el área de cada una y luego se almacena en la variable **areaTotalT**.

Paso 9. Se recorre la lista de las compuertas calculando el área de cada una y luego se almacena en la variable **areaTotalG**.

Paso 10. Se calcula **el área total del pool**

Paso 11. Se calcula la diferencia entre el área total del pool y la suma de las áreas de los elementos del modelo y se retorna ese valor.

El **Algoritmo 3** presenta una complejidad lineal, pues a pesar de que en el algoritmo están involucrados tres ciclos, cada uno de ellos se repite **n** veces de forma independiente, es decir, **n + n + n**. Por tanto la complejidad es de un $O(3n)=O(n)$. En el Anexo 1 se muestra el código fuente de estos tres algoritmos.

3.2 Pruebas y evaluación del módulo de representación visual.

En el desarrollo de un software específico, el proceso de prueba es clave para detectar errores o fallas. Conceptos como estabilidad, escalabilidad, eficiencia y seguridad se relacionan a la calidad de un producto bien desarrollado. Las pruebas son de gran importancia en la garantía del software, una selección cuidadosa de los datos de prueba puede ofrecer mucha confianza en cuanto al desempeño que posee el programa (Gómez et al., 2013).

Para probar el correcto funcionamiento, tanto interno como externo de una aplicación existen diferentes métodos de prueba. Los métodos de prueba de software tienen el objetivo de diseñar

pruebas que descubran diferentes tipos de errores con menor tiempo y esfuerzo. Uno de estos métodos de prueba es el de caja negra que se enfoca directamente en la interfaz, sin importar el código. Las pruebas se realizan mediante una serie de entradas de datos para encontrar fallas. Otro de los métodos de prueba que existe es el de caja blanca que se centra en los detalles de la implementación del software, por lo que su diseño está fuertemente ligado al código fuente de la aplicación. Las pruebas de caja blanca se basan en el conocimiento de la lógica interna del código del sistema. Al estar basadas en una implementación concreta, si ésta se modifica, por regla general las pruebas también deberán rediseñarse (Pressman, 2010).

3.2.1 Pruebas unitarias

Las pruebas unitarias se diseñan para probar pequeñas partes del código, como por ejemplo una función o un método. De acuerdo a (Gómez et al., 2013), estas pruebas están centradas en el procesamiento lógico interno y en las estructuras de datos dentro de un componente. Es aplicable a componentes representados en el modelo de implementación para verificar que los flujos de control y de datos estén cubiertos, y que ellos funcionen como se espera. Dichas pruebas requieren del conocimiento de la estructura interna del programa y son derivadas a partir de las especificaciones internas del diseño o del código programado.

Los casos de prueba son clases o módulos que deben ser configurados para probar los métodos de una clase. Así, para cada clase que se desea probar se define su correspondiente caso de prueba. De esta forma, se construye un programa que sirve para probar los diferentes módulos de la herramienta. Existen diferentes técnicas para el diseño de pruebas como son el camino básico y las pruebas de decisión-condición. Según lo planteado en (Pressman, 2010) el método del camino básico permite al diseñador de casos de pruebas obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución.

En esta tesis, para establecer los casos de prueba se toma el modelo presentado en la Figura 2.14, que representa un proceso de denuncia en el MININT. A continuación se diseñan los casos de prueba para un conjunto de métodos del módulo representación visual.

Método 1: Actividades sin etiquetar

La directriz “*Etiquete las actividades*” evalúa la cantidad de actividades o tareas que no estén etiquetadas en el modelo. Dicha directriz tiene asociada las medidas *número de actividades sin etiquetar* ($S_{FSE}(G) = |FSE|$) y *razón entre el número de actividades sin etiquetar y la cantidad de actividades* ($FSE_F(G) = \frac{|FSE|}{|F|}$). Se espera que el resultado de las medidas sean 2 y 0.07 respectivamente, por tanto el resultado de la prueba devuelve un valor *booleano* (*true*). En la Figura 3.7 se presentan los métodos de prueba para esta directriz y las medidas asociadas a la misma.

```
@Test
public void testActividadesSinEtiquetar() {
    System.out.println("actividadesSinEtiquetar");
    double[] expectedResult = new double[2];
    expectedResult[0] = 2;
    expectedResult[1] = 0.07;
    double[] result = RepresentacionVisual.actividadesSinEtiquetar(p);
    assertEquals(expectedResult, result, 0.0);
}
```

Figura 3.7 Método diseñado para la prueba de la directriz “*Etiquete las actividades*”.

Al ejecutar la prueba el valor esperado coincide con el resultado por lo que la prueba es exitosa. Este resultado se muestra en la Figura 3.8.

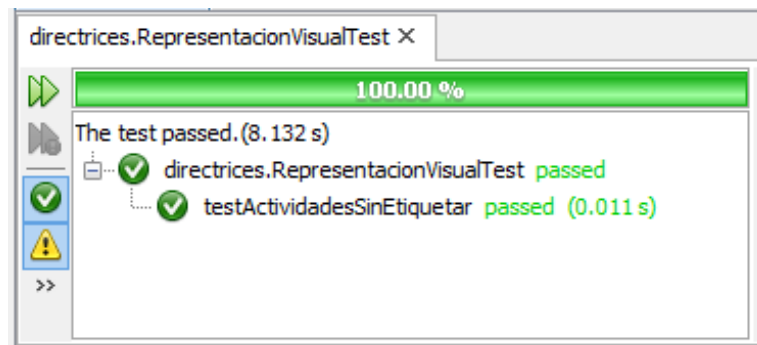


Figura 3.8 Resultado satisfactorio de la prueba para la directriz “*Etiquete las actividades*”.

Método 2: Tareas sin especificar

La directriz “*Tipos de tareas no especificados*” evalúa la cantidad de actividades o tareas del modelo que no tengan el tipo especificado. Dicha directriz tiene asociada las medidas *número de actividades con tipo no especificado* ($S_{FST}(G) = |FST|$) y *razón entre el número de actividades que no poseen tipo especificado y el número de actividades* ($FST_F(G) = \frac{|FST|}{|F|}$). Se

espera que el resultado de las medidas sean 8 y 0.27 respectivamente, por tanto el resultado de la prueba devuelve un valor *booleano* (*true*). En la Figura 3.9 se presentan los métodos de prueba para esta directriz y las medidas asociadas a la misma.

```
public void testTareaSinEspecificar() {
    System.out.println("tareaSinEspecificar");
    double[] expectedResult = new double[2];
    expectedResult[0] = 8;
    expectedResult[1] = 0.27;
    double[] result = RepresentacionVisual.tareaSinEspecificar(p);
    assertArrayEquals(expectedResult, result, 100);
    System.out.println("Tipos de tareas sin especificar");
}
```

Figura 3.9 Método diseñado para la prueba de la directriz “Tipo de tareas no especificados”.

Al ejecutar la prueba el valor esperado coincide con el resultado por lo que la prueba es exitosa. Este resultado se muestra en la Figura 3.10.

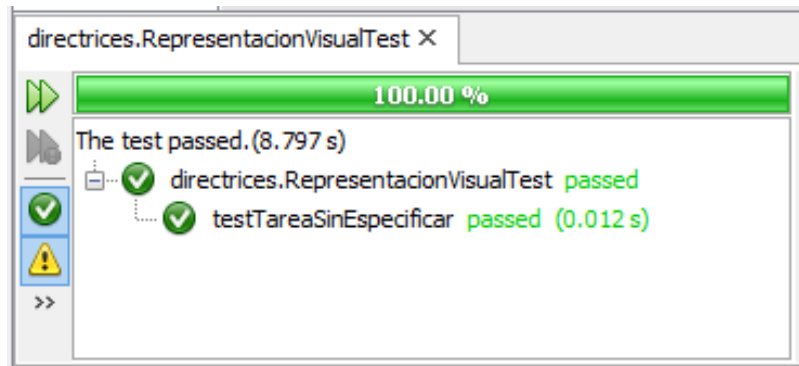


Figura 3.10 Resultado satisfactorio de la prueba para la directriz “Tipo de tareas no especificados”.

De forma similar se realizó una prueba para verificar que ocurre si el *pool* que se pasa como parámetro no tiene ningún elemento. Para este caso se probó con el método *cantidad de cruces de arcos*.

Método 3: Cantidad de cruces de arcos

La directriz “Alto número de cruces de flujos de secuencia” evalúa la cantidad de cruces de arcos existentes en el modelo. Dicha directriz tiene asociada las medidas *número de cruces de flujos de secuencia en el modelo* ($S_{CA}(G) = |CA|$) y *razón entre el número de cruces de flujos de secuencia y el número de flujos de secuencia en el modelo* ($CA_A(G) = \frac{|CA|}{|A|}$). Dado que el *pool* que se pasa como parámetro no tiene elementos se espera que el resultado de las medidas sean 0 y 0 respectivamente, por tanto el resultado de la prueba devuelve un valor *booleano* (*true*). En

la Figura 3.11 se presentan los métodos de prueba para esta directriz y las medidas asociadas a la misma.

```
@Test
public void testCtdCrucesArcos() {
    System.out.println("ctdCrucesArcos");
    Pool p = new Pool();
    double[] expResult = new double[2];
    expResult[0] = 0;
    expResult[1] = 0;
    double[] result = RepresentacionVisual.ctdCrucesArcos(p);
    assertEquals(expResult, result, 0.0);
}
```

Figura 3.11 Método diseñado para la prueba de la directriz “Alto número de cruces de flujos de secuencia”.

Al ejecutar la prueba el valor esperado coincide con el resultado por lo que la prueba es exitosa. Este resultado se muestra en la Figura 3.12.

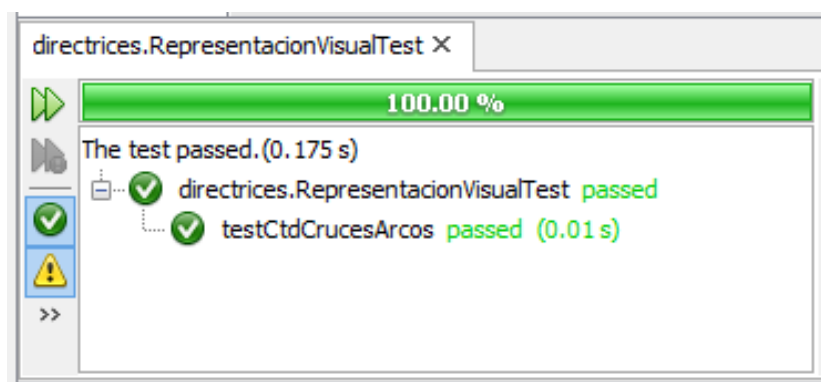


Figura 3.12 Resultado satisfactorio de la prueba para la directriz “Alto número de cruces de flujos de secuencia”.

3.2.2 Análisis de factibilidad de la herramienta SAD_BPMN con el módulo de representación visual.

Después de haber incluido el módulo de representación visual en la herramienta SAD_BPMN se valoró la realización de un estudio de factibilidad (Ivar Jacobson). A partir de esta valoración se decidió prescindir del análisis de factibilidad, pues este es ineludible solamente cuando el desarrollo del sistema no tiene una justificación económica sólida, existe un riesgo considerable, ya sea tecnológico, operativo o jurídico, o no se cuenta con una alternativa clara de implementación. No obstante, se desarrollaron algunas ideas básicas alrededor de las categorías más importantes de la factibilidad.

- Factibilidad económica

Es importante destacar que al asumir el desarrollo del sistema no se incurrirían en gastos por conceptos de equipamiento (se cuenta con el necesario) ni de licencias de software (se utilizaría software no comercial).

Al mismo tiempo se obtendrían beneficios como los siguientes:

- Disminución de errores al automatizar procesos que se desarrollan manualmente.
- Disminución del tiempo necesario para realizar los procesos del negocio debido a su automatización.

- Factibilidad técnica

- No existen soluciones disponibles en el mercado que cumplan con las necesidades del cliente.
- Se cuenta con la tecnología y el conocimiento técnico necesarios para el desarrollo del proyecto.

- Factibilidad operacional

Existe el personal capacitado requerido para llevar a cabo el proyecto, y se cuenta además con el respaldo y compromiso verdadero, en términos de actitud y asignación de recursos, por parte del cliente, quien es a la vez el usuario final.

3.3 Consideraciones finales del capítulo

En este capítulo se presentó una descripción del módulo de representación visual de la herramienta SAD_BPMN. Se mostró una perspectiva de este módulo, haciendo uso de diagramas de paquetes y diagramas de clases. Además como parte de su verificación y validación se realizó un conjunto de pruebas unitarias con el objetivo de evaluar el funcionamiento de las directrices de calidad y las medidas asociadas a ellas.

CONCLUSIONES

Como resultado de esta investigación, se ha arribado a las conclusiones siguientes:

1. Se identificaron como elementos a tener en cuenta para evaluar la representación visual de los modelos de procesos de negocio las etiquetas y las posiciones relativas de los elementos en el modelo, debido a que influyen en la comprensión y modificabilidad de los mismos.
2. Se definieron 38 medidas para obtener una noción de problemas relacionados con la representación visual de modelos de procesos de negocios.
3. Se validaron las medidas propuestas de acuerdo al marco formal de Briand lo cual permitió clasificarlas de la siguiente forma: 19 fueron clasificadas como medidas de tamaño, siete como medidas de complejidad y ocho de acoplamiento.
4. Se implementó un módulo con las medidas propuestas para la evaluación de la representación visual de los modelos de procesos de negocio en la herramienta SAD_BPMN.

RECOMENDACIONES

Derivadas del estudio realizado, así como de las conclusiones generales emanadas del mismo, se recomienda:

- Extender la herramienta SAD_BPMN para que permita la interpretación de archivos de salida exportados por otras herramientas de modelado de proceso de negocio.
- Realizar una validación empírica de las medidas propuestas y proponer umbrales para las mismas.

REFERENCIAS BIBLIOGRÁFICAS

- AGARWAL, R. 2003. Object-oriented modeling with UML: a study of developers' perceptions. *Communications of the ACM*, 46, 248.
- ANDERSON, J. R. 1983. The Architecture of Cognition. *Cambridge, Harvard University Press*.
- ARKILIC, I. G., REIJERS, H. A. & GOVERDE, R. R. H. M. J. 2013. How good is an as-is model really? In: ROSA, M. & SOFFER, P. (eds.) *Business Process Management Workshops. BPM 2012 International Workshops*. Tallinn, Estonia: Springer-Verlag Berlin Heidelberg.
- AVERSANO, L., BODHUIN, T., CANFORA, G. & TORTORELLA, M. A framework for measuring business processes based on GQM. 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 1, 2004 Washington, DC. IEEE Computer Society, 10012.1.
- BASILI, V. & ROMBACH, H. 1988. The TAME Project: Towards Improvement-Oriented Software Environments. *IEEE Transactions on Software Engineering*, 14(6), 728-738.
- BASILI, V., SHULL, F. & LANUBILE, F. 1999. Building knowledge through families of experiments. *IEEE Transactions on Software Engineering*, 25(4), 435-437.
- BASILI, V. & WEISS, D. 1984. A Methodology for Collecting Valid Software Engineering Data. *IEEE Transactions on Software Engineering*, 10, 728-738.
- BECKER-KORNSTAEDT, U. & BELAU, W. 2000. Descriptive process modeling in an industrial environment: Experience and guidelines. In: CONRADI, R. (ed.) *Software Process Technology*.
- BRIAND, L., MORASCA, S. & BASILI, V. 1996. Property-Based Software Engineering Measurement. *IEEE Transactions on Software Engineering*, 22(1), 68-86.
- C. BATINI, E. N., M. TALAMO, AND R. TAMASSIA. A Graphtheoretic Approach to Aesthetic Layout of Information Systems Diagrams. 10th Int. Workshop on Graphtheoretic Concepts in Computer Science, June 1984 1984 Berlin. 9-18.
- CADAVID, J. J., OSPINA, C. & QUINTERO, J. B. Estudio Comparativo de Técnicas de Modelado de Negocio. CIBSE, 2008. 309-314.
- CALERO, C., PIATTINI, M. & GENERO, M. 2001. Method for obtaining correct Metrics. 3rd. *International Conference on Enterprise and Information Systems (ICEIS'2001)*. Setúbal, Portugal.
- CANTONE, G. & DONZELLI, P. 2000. Production and Maintenance of Goal-oriented Measurement Models. *International Journal of Software Engineering & Knowledge Engineering*, 10(5), 605-626.
- CARDOSO, J. 2005. How to Measure the Control-flow Complexity of Web Processes and Workflows. In: FISCHER, L. (ed.) *Workflow Handbook*.
- CARDOSO, J. Process control-flow complexity metric: an empirical validation. *International Conference on Services Computing*, 2006 Chicago, IL. IEEE, 167-73.

- DAVIES, I., GREEN, P., ROSEMAN, M., INDULSKA, M. & GALLO, S. 2006. How do practitioners use conceptual modeling in practice? *Data & Knowledge Engineering*, 58, 358-380.
- DUFRESNE, T. & MARTIN, J. 2003. Process modeling for e-business. *INF 770 – Methods for Informations Systems Engineering: Knowledge Management and E-Business*. Fairfax, VA.: George Mason University.
- DUMAS, M., ROSA, M. L., MENDLING, J. & REIJERS, H. A. 2013. *Fundamentals of business process management*, Springer-Verlag Berlin Heidelberg.
- EFFINGER, P., JOGSCH, N. & SEIZ, S. 2010. On a Study of Layout Aesthetics for Business Process Models Using BPMN. In: MENDLING, J., WEIDLICH, M. & WESKE, M. (eds.) *Business Process Modeling Notation*. Springer-Verlag Berlin Heidelberg.
- EFFINGER, P., SIEBENHALLER, M. & KAUFMANN, M. An Interactive Layout Tool for BPMN. Commerce and Enterprise Computing, 2009. CEC '09. IEEE Conference on, 20-23 July 2009 2009. 399-406.
- EICHELBERGER, H. 2005. *Aesthetics and Automatic Layout of UML Class Diagrams*. PhD Thesis. Doktorgrades, Bayerischen Julius-Maximilians-Universität Würzburg.
- EICHELBERGER, H. & SCHMID, K. 2009. Guidelines on the aesthetic quality of UML class diagrams. *Information and Software Technology*, 51, 1686-1698.
- FETTKE, P., VELLA, A. L. & LOOS, P. 2012. From measuring the quality of labels in process models to a discourse on process model quality: a case study. *2012 45th Hawaii International Conference on System Sciences (HICSS)*, 197-206.
- FLOWERS, R. & EDEKI, C. 2013. Business Process Modeling Notation. *International Journal of Computer Science and Mobile Computing*, 2, 35-40.
- FORSTER, M. Applying Crossing Reduction Strategies to Layered Compound Graphs. In: SPRINGER-VERLAG INC., N. Y., NY, USA (ed.) *Graph Drawing: 10th International Symposium, GD 2002*. Irvine, CA, USA.
- GARIMELLA, K. & LEES, M. 2008. Introducción a BPM. *New York: Jacques Boussard*.
- GÓMEZ, D., JÚSTIZ, D. & DELGADO, M. 2013. Unit Tests of Software in a University Environment. *Computación y Sistemas*, 17, 69-77
- GRUHN, V. & LAUE, R. Complexity metrics for business process models. 9th International Conference on Business Information Systems (BIS 2006), 2006 Klagenfurt. Lecture notes in informatics.
- GRUHN, V. & LAUE, R. Reducing the Cognitive Complexity of Business Process Models. In: BACIU, G., WANG, Y. X., YAO, Y. Y., KINSNER, W., CHAN, K. & ZADEH, L. A., eds. 8th Ieee International Conference on Cognitive Informatics (ICCI'09), 2009. IEEE, 339-345.
- GRUHN, V. & LAUE, R. 2011. Detecting Common Errors in Event-Driven Process Chains by Label Analysis. *Enterprise Modelling and Information Systems Architectures*, 6, 3-15.
- HE, G., XUE, G., YAO, S. & WU, Z. Business Process Modeling: A Survey. In: YANG, Y. X., ed. *Proceedings of Annual Conference of China Institute of Communications*, 2010. 172-178.
- HOLLINGSWORTH, D. 2004. The Workflow Reference Model 10 Years

- HOMMES, B.-J. & VAN REIJSWOUD, V. The Quality of Business Process Modelling Techniques: The application of a framework for understanding the quality of UML. . Proceedings of the International Conference on Information Systems Concepts: An Integrated Discipline Emerging (ISCO), 1999 Leiden, The Netherlands., 117-136.
- HOMMES, B. J. & VAN REIJSWOUD, V. Assessing the Quality of Business Process Modelling Techniques. 33rd Hawaii International Conference on System Sciences, 2000 Hawaii. IEEE.
- HOMMES, B. J., VAN REIJSWOUD, V. & BOTS, P. W. G. 2000. The quality of business process modelling methods: illustration of a framework for understanding modelling quality. *Information System Concepts: An Integrated Discipline Emerging. IFIP TC8/WG8.1 International Conference on Information System Concepts: An Integrated Discipline Emerging (ISCO-4)*, 117-39.
- HUANG, Z. & KUMAR, A. New Quality Metrics for Evaluating Process Models. In: ARDAGNA, D. & AL., E., eds. BPM 2008 Workshops, 2009. Springer-Verlag Berlin Heidelberg, 164–170.
- ISO/IEC 2001. Software engineering - product quality - Part 1: Quality Model (9126).
- IVAR JACOBSON, G. B., JAMES RUMBAUGH 2000. El proceso unificado de desarrollo de software.
- JOHNSON, M. R. G. A. D. S. 1983. Crossing Number is NP-Complete. *SIAM J. Algebraic and Discrete Methods*, 4, 312-316.
- JUNG, J. Y. Measuring entropy in business process models. International Conference on Innovative Computing, Information and Control, 2008. 246.
- KALPIC, B. & BERNUS, P. 2002. Business process modelling in industry—the powerful tool in enterprise management. *Computers in Industry*, 47, 299-318.
- LATVA-KOIVISTO, A. 2001. *Finding a complexity measure for business process models*. Technical Report Technical Report, Helsinki University of Technology.
- LEOPOLD, H., SMIRNOV, S. & MENDLING, J. 2011. Recognising Activity Labeling Styles in Business Process Models. *Enterprise Modelling and Information Systems Architectures*, 6, 16-29.
- LINDLAND, O. I., SINDRE, G. & SOLVBERG, A. 1994a. Understanding quality in conceptual modeling. *IEEE Software* 11, 42-49.
- LINDLAND, O. I., SINDRE, G. & SØLVBERG, A. 1994b. Understanding quality in conceptual modeling. *IEEE Software* 11, 42-49.
- MAGNANI, M. & MONTESI, D. 2007. BPMN: How much does it cost? An incremental approach. In: ALONSO, G. D., P. ROSEMAN, M. (ed.) *Business Process Management, Proceedings*.
- MENDLING, J. 2008a. *Metrics for process models: empirical foundations of verification, error prediction, and guidelines for correctness*, Springer Science & Business Media.
- MENDLING, J. 2008b. Metrics for process models: empirical foundations of verification, error prediction, and guidelines for correctness. Springer-Verlag Berlin Heidelberg.
- MENDLING, J. 2009. Empirical studies in process model verification. In: JENSEN, K. & VAN DER AALST, W. M. P. (eds.) *Transactions on Petri Nets and Other Models of*

- Concurrency II, Special Issue on Concurrency in Process-Aware Information Systems.* Springer-Verlag Berlin Heidelberg.
- MENDLING, J., MOSER, M., NEUMANN, G., VERBEEK, H. M. W., VAN DONGEN, B. F. & VAN DER AALST, W. M. P. 2006. Faulty EPCs in the SAP Reference Model. In: DUSTDAR, S., FIADÉIRO, J. L. & SHETH, A. P. (eds.) *Business Process Management*. Vienna, Austria: Springer-Verlag Berlin Heidelberg.
- MENDLING, J. & NEUMANN, G. 2007. Error metrics for business process models. *CAiSE Forum*.
- MENDLING, J., RECKER, J. & REIJERS, H. A. 2010a. On the usage of labels and icons in business process modeling. *International Journal of Information System Modeling and Design*, 1, 40-58.
- MENDLING, J. & REIJERS, H. A. The impact of activity labeling styles on process model quality. In: HESSE, W. & OBERWEIS, A., eds. *European Symposium on Analysis, Design, Use and Societal Impact of Information Systems*, 2008 Marburg, Germany. 117-128.
- MENDLING, J., REIJERS, H. A. & CARDOSO, J. 2007. What makes process models understandable? In: ALONSO, G. D. P. R. M. (ed.) *Business Process Management, Proceedings*.
- MENDLING, J., REIJERS, H. A. & RECKER, J. 2010b. Activity labeling in process modeling: Empirical insights and recommendations. *Information Systems*, 35, 467-482.
- MENDLING, J., REIJERS, H. A. & VAN DER AALST, W. M. P. 2010c. Seven process modeling guidelines (7PMG). *Information and Software Technology*, 52, 127-136.
- MENDLING, J., REIJERS, H. A., RECKER, J. 2010. Activity labeling in process modeling: Empirical insights and recommendations. *Information Systems*, 35, 467-482.
- MENDLING, J. & STREMBECK, M. 2008. Influence factors of understanding business process models. In: ABRAMOWICZ, W. F. D. (ed.) *Business Information Systems*. Innsbruck, Austria: Springer.
- MOODY, D. L. 2005. Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowledge Engineering*, 55, 243-276.
- MORENO-MONTES DE OCA, I. & SNOECK, M. 2014. A look into business process modeling guidelines through the lens of the technology acceptance model. *Submitted to QMMQ-14 Workshop*. Atlanta, GA USA: UCLV-KULeuven.
- MORENO-MONTES DE OCA, I. & SNOECK, M. 2015. Pragmatic Guidelines for Business Process Modeling. Management Information Systems Group, Faculty of Business and Economics, KU Leuven, Leuven, Belgium.
- MORENO-MONTES DE OCA, I., SNOECK, M., REIJERS, H. A. & RODRÍGUEZ-MORFFI, A. 2015. A systematic literature review of studies on business process modeling quality. *Information and Software Technology*, 58, 187-205.
- MORENO MONTES DE OCA, I., RODRÍGUEZ MORFFI, A., SNOECK, M., MORENO RODRÍGUEZ, R., CASAS CARDOSO, G. & GONZÁLEZ GONZÁLEZ, L. 2014.

- Directrices prácticas y métricas de calidad en la modelación de procesos de negocio: un caso de estudio. *Revista Cubana de Ciencias Informáticas*, 8, 1-18.
- MULTAMÄKI, M. 2002. Objective-driven planning of business process modeling. Department of Industrial Engineering and Management, Helsinki University of Technology, Finland.
- NELSON, H. J., POELS, G., GENERO, M. & PIATTINI, M. 2012. A conceptual modeling quality framework. *Software Quality Journal*, 20, 201-228.
- OMG 2005. Unified Modeling Language: Superstructure. *UML Superstructure Specification v2.0, formal/05-07-04, Object Management Group*.
- OMG 2011. Business Process Model and Notation (BPMN) version 2.0. OMG.
- ORLIN, Z. M. A. J. B. 1985. NP-Completeness for Minimizing Maximum Edge Length in Grid Embeddings. *J. Algorithms*, 6, 10-16.
- PÉREZ-SANTIAGO, J. 2014. *Directrices Prácticas para la Modelación Conceptual de Procesos del Negocio*. Tesis de Diploma, Universidad Central "Marta Abreu" de las Villas.
- POELS, G. & DEDENE, G. 2000. Distance-based software measurement: necessary and sufficient properties for software measures. *Information and Software Technology*, 42(1), 35-46.
- PRESSMAN, R. S. 2010. *Software Engineering. A Practitioner's Approach*.
- R. TAMASSIA, G. D. B., AND C. BATINI 1988. Automatic Graph Drawing and Readability of Diagrams. *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-18, 61-79.
- RAJNISH, K. 2014. Theoretical Validation of Inheritance Metrics for Object-Oriented Design against Briand's Property. *I.J. Information Engineering and Electronic Business*, 3, 28-33.
- RAJNISH, S. M. A. K. 2013. New Quality Inheritance Metrics for Object-Oriented Design. *International Journal of Software Engineering and Its Applications*, 7, 185-200.
- RECKER, J. 2008. BPMN Modeling - Who, Where, How and Why. *BPTrends*, 5, 1-8.
- RECKER, J., INDULSKA, M. & GREEN, P. 2007. Extending representational analysis: BPMN user and developer perspectives. In: ALONSO, G. D., P. ROSEMAN, M. (ed.) *Business Process Management, Proceedings*.
- REIJERS, H. A., FREYTAG, T., MENDLING, J. & ECKLEDER, A. 2011. Syntax highlighting in business process models. *Decision Support Systems*, 51, 339-349.
- REIJERS, H. A. & VANDERFEESTEN, I. 2004. Cohesion and coupling metrics for workflow process design. In: DESEL, J., PERNICI, B. & WESKE, M. (eds.) *BPM 2004*. Berlin, Heidelberg Springer-Verlag 2004.
- ROLÓN, E. 2009. *Medidas para asegurar la calidad de los modelos de procesos de negocio*. Tesis doctoral. dir. F.O. García Rubio y F. Ruíz González, Universidad de Castilla-La Mancha, Spain.
- ROLÓN, E., CARDOSO, J., GARCÍA, F., RUIZ, F. & PIATTINI, M. 2009. Analysis and Validation of Control-Flow Complexity Measures with BPMN Process Models. In: HALPIN, T., KROGSTIE, J., NURCAN, S., PROPER, E., SCHMIDT, R., SOFFER, P.

- & UKOR, R. (eds.) *Enterprise, Business-Process and Information Systems Modeling*. Springer Berlin Heidelberg.
- ROSEMANN, M. 2006. Potential pitfalls of process modeling: part A. *Business Process Management Journal*, 12, 249-254.
- SANCHEZ-GONZALEZ, L., GARCIA-RUBIO, F., RUIZ-GONZALEZ, F. & PIATTINI-VELTHUIS, M. 2010a. Measurement in business processes: a systematic review. *Business Process Management Journal*, 16, 114-134.
- SANCHEZ-GONZALEZ, L., GARCIA, F., MENDLING, J., RUIZ, F. & PIATTINI, M. 2010b. Prediction of Business Process Model Quality Based on Structural Metrics. In: PARSONS, J., SAEKI, M., SHOVAL, P., WOO, C. & WAND, Y. (eds.) *Conceptual Modeling - Er 2010*.
- SARSHAR, K. 2005. Comparing the control-flow of epc and petri net from the end-user perspective. *Business Process Management*, 3649, 434.
- SCHREPFFER, M., WOLF, J., MENDLING, J. & REIJERS, H. A. 2009. The impact of secondary notation on process model understanding. In: PERSSON, A. & STIRNA, J. (eds.) *Proceedings of the 2nd IFIP WG8.1 Working Conference on The Practice of Enterprise Modeling PoEM 2009*. Stockholm, Sweden: IFIP International Federation for Information Processing.
- SERRANO, M., PIATTINI, M., CALERO, C., GENERO, M. & MIRANDA, D. 2002. Un Método para la Definición de Métricas de Software. *1er. Workshop en Métodos de Investigación y Fundamentos Filosóficos en Ingeniería del Software y Sistemas de Información (MIFISIS'2002)*. Madrid.
- SHARMA, R., GAUR, H. & KUMAR, M. 2015. Advances in Intelligent Informatics. In: SPRINGER (ed.).
- SIAU, K. 1999. "Information Modeling and Method Engineering: A Psychological Perspective.". *Journal of Database Management*, 10(4), 44-50.
- SMITH, H. & FINGAR, P. 2003. *Business Proces Management: The Third Wave*.
- SOFFER, P. & WAND, Y. 2004. Goal-driven analysis of process model validity. In: PERSSON, A. & STIRNA, J. (eds.) *Advanced Information Systems Engineering, Proceedings*.
- TAMASSIA, R., DI BATTISTA, G. & BATINI, C. 1988. Automatic Graph Drawing and Readability of Diagrams. *IEEE Transactions on Systems, Man and Cybernetics* 18, 61-79.
- VANDERFEESTEN, I., REIJERS, H. A. & VAN DER AALST, W. M. P. 2008. Evaluating workflow process designs using cohesion and coupling metrics. *Computers in Industry*, 59, 420-437.
- WADDLE, V. 2001. Graph Layout for Displaying Data Structures. In: SPRINGER-VERLAG INC., N. Y., NY, USA (ed.) *Graph Drawing: 8th International Symposium GD 2000*. Colonial Williamsburg, Va, USA.
- WEBER, B. & REICHERT, M. 2008. Refactoring Process Models in Large Process Repositories. In: BELLAHSÈNE, Z. & LÉONARD, M. (eds.) *Advanced Information Systems Engineering*. Springer Berlin Heidelberg.

- WEYUKR, E. 1988. Evaluating software complexity measures. *Transactions Software Engineering*, 14(9), 1357-1365.
- WFMC 2002. Workflow Management Coalition Workflow Standard: Workflow Process Definition Interface - XML Process Definition Language (XPDL). *Workflow Management coalition*. Florida, USA: Lighthouse Point.
- WHITMIRE, S. 1997. *Object Oriented Design Measurement*, John Wiley & Sons, Inc.
- ZUR MUEHLEN, M. & RECKER, J. 2008. How Much Language is Enough? Theoretical and Practical Use of the Business Process Modeling Notation. In: BELLAHSÈNE, Z. & LÉONARD, M. (eds.) *Proceedings of the 20th International Conference on Advanced Information Systems Engineering CAiSE 2008: Advanced Information Systems Engineering*. Montpellier, France: Springer-Verlag Berlin Heidelberg.

Anexo 1: Código fuente de los algoritmos propuestos en la tesis.

- Código fuente correspondiente al Algoritmo 1: ctd_cruces_arcos

```

public static double[] ctdCrucesArcos(Pool p) {
    double[] crucesArcos = new double[2];
    ArrayList<Flow> flow = p.getFlow();
    double numeroCruces = 0;
    ArrayList<ArrayList> rectasFlujos = new ArrayList<ArrayList>();
    for (Flow flow1 : flow)
        rectasFlujos.add(rectasUnFlujo(flow1));
    for (int i = 0; i < flow.size(); i++) {
        ArrayList<Recta> r = rectasUnFlujo(flow.get(i));
        for (int j = i + 1; j < flow.size(); j++) {
            ArrayList<Recta> r1 = rectasUnFlujo(flow.get(j));
            for (int k = 0; k < r.size(); k++) {
                Recta recta1 = r.get(k);
                double m1 = recta1.pendiente();
                double n1 = recta1.valorN();
                for (int l = 0; l < r1.size(); l++) {
                    Recta recta2 = r1.get(l);
                    double m2 = recta2.pendiente();
                    double n2 = recta2.valorN();
                    if (rectasSeCortan(m1, n1, m2, n2)) {
                        double X1 = recta1.getP1().getX();
                        double Y1 = recta1.getP1().getY();
                        double X2 = recta1.getP2().getX();
                        double Y2 = recta1.getP2().getY();
                        double X3 = recta2.getP1().getX();
                        double Y3 = recta2.getP1().getY();
                        double X4 = recta2.getP2().getX();
                        double Y4 = recta2.getP2().getY();
                        if (X1 == X2 && ((Y1 < Y3 && Y3 < Y2) || (Y2 < Y3 && Y3 < Y1)))
                            if ((X3 < X1 && X1 < X4) || (X4 < X1 && X1 < X3))
                                numeroCruces++;
                        if (Y1 == Y2 && ((X1 < X3 && X3 < X2) || (X2 < X3 && X3 < X1)))
                            if ((Y3 < Y1 && Y1 < Y4) || (Y4 < Y1 && Y1 < Y3))
                                numeroCruces++;
                    }
                }
            }
        }
        crucesArcos[0] = numeroCruces;
        crucesArcos[1] = (numeroCruces / flow.size());
    }
    return crucesArcos;
}

```

- Código fuente correspondiente al Algoritmo 2: eventos_sin_etiquetar

```
public static final double[] eventoSinEtiquetar(Pool p) {
    double[] result = new double[2];
    double totalEventT = p.getEvent().size();
    double ctdEven = 0;
    for (int i = 0; i < p.getEvent().size(); i++)
        for (int j = 0; j < nombreEventos().size(); j++)
            if (p.getEvent().get(i).getName().contains(nombreEventos().get(j))
                || p.getEvent().get(i).getName().isEmpty())
                ctdEven++;
    result[0] = ctdEven;
    result[1] = ctdEven / totalEventT;
    return result;
}
```

- Código fuente correspondiente al Algoritmo 3: area_dibujo_extensa

```
public static double areaDibujo(Pool p) {
    ArrayList<Tasks> task = p.getTask();
    ArrayList<Events> event = p.getEvent();
    ArrayList<Gateways> gateway = p.getGateway();
    double areaTotalP = p.getAlto() * p.getAncho();
    double areaTotalT = 0;
    double areaE = 0;
    double areaG = 0;
    int countE = 0;
    int countG = 0;
    for (Tasks task1 : task)
        areaTotalT += task1.getAlto() * task1.getAncho();
    for (Gateways gateway1 : gateway) {
        if (gateway1.getAlto() * gateway1.getAncho() > 1849) {
            areaG += gateway1.getAlto() * gateway1.getAncho();
            countG++;
        }
    }
    for (Events event1 : event) {
        if (event1.getAlto() * event1.getAncho() > 900) {
            areaE += event1.getAlto() * event1.getAncho();
            countE++;
        }
    }
    double areaTotalG = 1849 * (gateway.size() - countG) + areaG;
    double areaTotalE = 900 * (event.size() - countE) + areaE;
    return areaTotalP - (areaTotalE + areaTotalG + areaTotalT);
}
```