

**Análisis Visual de Variabilidad en Modelos de Procesos SPEM 2.0: vAVISPA**



**Universidad  
del Cauca**

Monografía de Trabajo de Grado para Optar al Título de  
Ingeniero de Sistemas

**Jhonnatham Samir Salgado.**

Director:

**PhD. Julio Ariel Hurtado Alegría**

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Sistemas

Grupo IDIS – Investigación y Desarrollo en Ingeniería de Software

Línea de Investigación en Procesos Software

Popayán, Diciembre 2014

## Resumen

Las organizaciones de software han encontrado en la definición de sus procesos de software una estrategia para ser más competitivas. Esta tarea normalmente es larga y costosa, más aún cuando el proceso debe ser adaptado e instanciado a cada proyecto. Por la gran inversión del modelado, evaluar la calidad del modelo de procesos es una tarea necesaria. Este problema se agrava cuando la complejidad del modelo de proceso se ve incrementada cuando se consideran variabilidades para facilitar su adaptación a contextos específicos. Dar formalidad al modelo facilita que actividades como el modelado de procesos y su evaluación puedan ser asistidas por la computadora como lo es el caso de AVISPA, una herramienta que facilita el análisis de modelos especificados en SPEM 2.0. Sin embargo AVISPA tiene limitaciones cuando el modelo de proceso SPEM 2.0 incluye variabilidades. Esta tesis presenta a vAVISPA, una herramienta computacional que facilita el análisis de modelos de procesos SPEM 2.0 que incluyen variabilidades. VAVISPA es una extensión de AVISPA que agrega dos nuevos planos, el plano de variabilidad y el plano de elementos en uso. Adicionalmente, vAVISPA modifica toda la pila de AVISPA desde la importación a la visualización de patrones, para resolver positivos falsos que se presentan en la versión original, debido a la no resolución de la variabilidad. vAVISPA es evaluada a través del análisis de dos modelos de procesos con variabilidad mostrando un incremento significativo en la efectividad de la detección de errores.

## **Abstract**

Defining the software process to achieve competitiveness is an identified strategy by software organizations. This task normally is time consuming and expensive, more when the process model is tailored and instantiated for each project. Quality assessment of the process models become in a critical activity, because the great investing required for modeling the process model. This problem is accentuated when the complexity of the software process model is increased by the inclusion of variabilities in the process model, in order to facilitate their adaptation to specific contexts. When formality is given to process modeling, some activities such as modeling and analysis could be supported by computer, it is the AVISPA case, an analysis tool of SPEM 2.0 process models including variabilities. However, AVISPA has some limitations for identify errors in process models including variabilities. In this tesis vAVISPA, an analysis tool for analyzing SPEM 2.0 software process models including variability is presented. VAVISPA is an AVISPA extension including two new blueprints: variability blueprint and use blueprint. Also, vAVISPA has modified each layer in the stack of AVISPA, from the importer to the pattern error visualizations, in order to resolve positive falses found in the original version, because the variability resolution is not supported by the previous version. VAVISPA was evaluated using two process models with variability, where an increment in the effectiveness for identify errors was obtained.

Este trabajo quiero dedicarlo a todas aquellas personas que con su paciencia, su esfuerzo, confianza y comprensión permitieron que se llevara a cabo. En especial es dedicado a mis padres que me brindaron su apoyo incondicional frente a cada una de las adversidades que se presentaron a lo largo de mi camino, mis hermanos con los que cuento incondicionalmente y a cada uno de esos amigos que con sus palabras demostraron esa confianza que tienen en mí y en el trabajo que desarrolle. Sin el apoyo de cada una de esas personas no hubiera obtenido cada una de las metas de mi vida.

## **Agradecimientos**

Ante todo mi agradecimiento con Dios el cual me permite lograr cada una de las metas que consigo en mi vida. También el agradecimiento y mi total gratitud con el ingeniero Julio Ariel Hurtado que con su confianza, sus palabras de ánimo, su tutoría y todo ese esfuerzo que me brindo durante este trabajo, sin el cual posiblemente no hubiera sido posible culminar. A mis amigos Ricardo Gallego, Miriam Pilar, Francisco Obando, Monserrat Zepahua, y todos aquellos que siempre me brindaron su apoyo y confiaron en el trabajo que realizaba me dieron ánimo de seguir adelante y pusieron un granito de arena para culminar este proceso.

## Tabla de contenido

|  |    |
|--|----|
| Tabla de contenido.....  | 6  |
| Índice de Figuras .....  | 8  |
| Índice de Tablas .....   | 9  |
| 1. INTRODUCCIÓN .....  | 1  |
| 1.1. Motivación.....   | 1  |
| 1.2. Contexto del problema.....  | 2  |
| 1.3. Problema.....   | 3  |
| 1.4. Pregunta de investigación .....   | 3  |
| 1.5. Objetivos .....   | 4  |
| 1.5.1. General .....   | 4  |
| 1.5.2. Específicos.....  | 4  |
| 1.6. Propuesta y Justificación.....  | 4  |
| 2. ESTADO DEL ARTE .....   | 6  |
| 2.1. Marco Teórico .....   | 6  |
| 2.2. Lenguajes de ejecución de procesos .....                                | 6  |
| Little-Jil .....   | 6  |
| BPMN .....   | 7  |
| XML Lenguaje de Definición de Procesos .....                                 | 7  |
| Business Process Execution Language .....                                    | 7  |
| 2.3. Lenguajes para el modelado de procesos Software .....                   | 7  |
| PROMENADE .....  | 7  |
| E3.....  | 8  |
| FUNSOFT .....  | 8  |
| MVP-L.....   | 9  |
| UML4SPM .....  | 9  |
| SPEM 2.0.....  | 10 |
| 2.4. Correctitud a nivel de Procesos .....                                   | 13 |
| 2.5. Variabilidad de Modelos de Procesos .....                               | 14 |
| 2.6. Técnicas y enfoques para el proceso de V&V de modelos de procesos ..... | 17 |
| 3. ANÁLISIS VISUAL DE LA VARIABILIDAD EN MODELOS DE PROCESO SPEM 2.0         |    |

|        |   |    |
|--------|---|----|
| 3.1.   | Introducción al modelo de Visualización .....   | 22 |
| 3.2.   | Modelo de proceso software .....  | 23 |
| 3.3.   | Resolución de la Variabilidad .....   | 24 |
| 3.3.1. | Reutilización del contenido del método .....  | 24 |
| 3.3.2. | Adaptación del modelo de proceso por medio de constructos SPEM 2.0 ....                           | 26 |
| 3.4.   | Métricas .....  | 30 |
| 3.5.   | Blueprints de modelos de procesos .....   | 32 |
| 3.5.1. | Blueprint Variabilidad del modelo de proceso .....  | 34 |
| 3.5.2. | Blueprint Method Content vs Process .....   | 35 |
| 3.6.   | Patrones .....  | 37 |
| 3.7.   | Prototipo vAVISPA .....   | 39 |
| 3.7.1. | Diseño e implementación de vAVISPA.....   | 41 |
| 3.7.2. | Realización de los casos de uso a nivel de diseño e implementación .....                          | 45 |
| 4.     | Estudio de caso: Análisis Visual de Variabilidad en Modelos de Procesos SPEM 2.0:<br>vAVISPA..... | 51 |
| 4.1.   | Metodología .....   | 51 |
| 4.2.   | Pregunta de Investigación .....   | 54 |
| 4.3.   | Objetivo del estudio de caso.....   | 54 |
| 4.4.   | Selección del estudio de caso.....  | 54 |
| 4.5.   | Descripción del estudio de caso.....  | 55 |
| 4.5.1. | Los modelos de proceso para el análisis .....   | 55 |
| 4.5.2. | Indicadores y métricas.....   | 57 |
| 4.6.   | Ejecución del Estudio de caso .....   | 60 |
| 4.7.   | Resultados .....  | 63 |
| 4.7.1. | Resultados Cuantitativos .....  | 63 |
| 4.7.2. | Resumen Cualitativo.....  | 65 |
| 4.8.   | Análisis de los resultados .....  | 66 |
| 4.9.   | Amenazas de Validez .....   | 75 |
| 4.10.  | Síntesis y Discusión .....  | 75 |
| 5.     | CONCLUSIONES, LIMITACIONES Y TRABAJO FUTURO.....  | 77 |
| 5.1.   | Conclusiones.....   | 77 |
| 5.2.   | Limitaciones .....  | 78 |

|   |    |
|---|----|
| 5.3. Lecciones Aprendidas y problemas enfrentados ..... | 79 |
| 5.4. Trabajos Futuros .....                             | 80 |
| 6. Referencias.....                                     | 81 |

## Índice de Figuras

|  |    |
|--|----|
| Fig. 1 Estructura de SPEM 2.0 Meta-Modelo .....  | 11 |
| Fig. 2 Blueprint de tareas generado por AVISPA.....                                      | 19 |
| Fig. 3 Proceso Análisis visual AVISPA.....   | 20 |
| Fig. 4 Proceso de análisis visual vAVISPA [15].....                                      | 22 |
| Fig. 5 Estructura de Composición de los Elementos SPEM 2.0 .....                         | 23 |
| Fig. 6 Representación del modelo según SPEM 2.0 [20].....                                | 24 |
| Fig. 7 Framework de trabajo especificado por SPEM 2.0 [8].....                           | 25 |
| Fig. 8 XML openUP generado por EPF composer.....   | 26 |
| Fig. 9 Esquema reglas contribución .....   | 27 |
| Fig. 10 Esquema reglas reemplazo.....  | 28 |
| Fig. 11 Esquema reglas extensión .....   | 29 |
| Fig. 12 Esquema reglas Extensión-Reemplazo .....   | 30 |
| Fig. 13 Elemento del proceso: Tarea (Task).....  | 31 |
| Fig. 14 Elemento del proceso: Rol .....  | 31 |
| Fig. 15 Elemento del proceso: Producto del trabajo (Work Product) .....                  | 32 |
| Fig. 16 Blueprint patrón para localización de tareas con múltiple propósito AVISPA ..... | 33 |
| Fig. 17 Blueprint patrón para localización de tareas con múltiple propósito vAVISPA..... | 34 |
| Fig. 18 Esquema de representación de variabilidad .....                                  | 35 |
| Fig. 19 Task variability Blueprint caso openUP .....                                     | 35 |
| Fig. 20 Representación gráfica de las tareas utilizada por proceso.....                  | 36 |
| Fig. 21 Blueprint Method content vs Process del proceso OpenUP .....                     | 37 |
| Fig. 22 Patrón independencia sub proyectos - Task (AVISPA) .....                         | 38 |
| Fig. 23 Patrón independencia sub proyectos - Task (vAVISPA).....                         | 38 |
| Fig. 24 Patrón independencia sub proyectos - Work Product (AVISPA).....                  | 39 |
| Fig. 25 Patrón independencia sub proyectos - Work Product (vAVISPA) .....                | 39 |
| Fig. 26 Interfaz vAVISPA.....  | 42 |
| Fig. 27 Estructura de paquetes vAVISPA.....  | 43 |
| Fig. 28 Caso de Uso vAVISPA .....  | 43 |
| Fig. 29 Estructura Meta-Modelo vAVISPA.....  | 45 |
| Fig. 30 Diagrama de Secuencia importar vAVISPA .....                                     | 46 |
| Fig. 31 Descomposición por componentes del método importXMLDocument .....                | 47 |
| Fig. 32 Segmento resolveVariability - Variabilidad tipo Contribute .....                 | 48 |
| Fig. 33 Diagrama Secuencia generación Task Variability Blueprint .....                   | 49 |
| Fig. 34 Definición viewTaskVariabilityBlueprintOn .....                                  | 50 |
| Fig. 35 Metodología del estudio de caso .....  | 52 |



|   |    |
|---|----|
| Fig. 36 Flujo de trabajo REMICS[54].....  | 57 |
| Fig. 37 Evaluación del Estudio de caso.....                                       | 61 |
| Fig. 38 Plantilla para reportar errores detectados .....                          | 61 |
| Fig. 39 Prototipo vAVISPA.....  | 62 |
| Fig. 40 Patrón sub proyecto (Blueprint Tareas) AVISPA.....                        | 68 |
| Fig. 41 Patrón sub proyecto (Blueprint Tareas) vAVISPA .....                      | 68 |
| Fig. 43 Task Variability Blueprint.....   | 70 |
| Fig. 44 Task Blueprints .....   | 70 |
| Fig. 45 Blueprints Roles .....  | 71 |
| Fig. 46 Blueprints Work Product .....   | 72 |
| Fig. 47 Fragmento Blueprint Independent Subproyect Pattern.....                   | 73 |
| Fig. 48 Fragmento Task Blueprint - Presencia de múltiples objetos similares ..... | 73 |
| Fig. 49 Fragmento Task Variability Blueprint .....                                | 74 |
| Fig. 50 Fragmento Workproduct Waste Pattern.....                                  | 74 |

## Índice de Tablas

|   |    |
|---|----|
| Tabla 1 Tipos Variabilidad SPEM 2.0.....  | 16 |
| Tabla 2. Análisis Métodos de validación y Verificación .....  | 21 |
| Tabla 3 Modelos de procesos identificados para el estudio de caso .....                               | 56 |
| Tabla 4 Tabla de Métricas e Indicadores .....   | 58 |
| Tabla 5 Patrones de Error .....   | 60 |
| Tabla 6 Información variabilidad y uso del modelo de proceso .....                                    | 63 |
| Tabla 7 Costo en tiempo de las actividades de evaluación .....  | 64 |
| Tabla 8 Relación de errores detectados en OpenUP .....  | 64 |
| Tabla 9 Tabla 8 Relación de errores detectados en Remics .....  | 64 |
| Tabla 10 Valor Índices Obtenido en OpenUP .....   | 65 |
| Tabla 11 Lista de Errores detectados en el proceso de evaluación patrón multipropósito<br>tarea ..... | 69 |

## 1. INTRODUCCIÓN

En las últimas décadas las organizaciones enfocadas en el desarrollo software, han tomado en cuenta la relevancia que tiene la formalización de sus procesos. Debido a esto desde las áreas académicas de investigación y la industria se han generado lazos con el fin de afrontar los nuevos retos que trae consigo el desarrollo de un producto software. Estos retos generados por la vertiginosa evolución de las tecnologías, nuevas necesidades de los clientes, mayor competitividad en la industria y necesidad de adaptación a nuevos mercados, requieren que las organizaciones dispongan de una fuerte estructura organizacional que le facilite la adaptación a los nuevos retos, que le ayude a la promulgación (ejecución) efectiva de sus procesos, la correcta definición de tiempos y costos de un desarrollo, que además permita la medición efectiva de sus estrategias y permitan a las organizaciones generar planes de mejora que las ayuden a ser más competitivas. Con el fin de lograr todos estos beneficios las empresas de desarrollo se han puesto a la tarea de desarrollar formalmente sus modelos de procesos .

A partir de la necesidad de formalizar sus procesos se han generado diversos lenguajes que le permiten la construcción de sus modelos y la promulgación de los mismos. Pero con cada nuevo avance obtenido genera nuevos retos para la industria, uno de estos retos ha sido la generación de estrategias efectivas que le permitan la evaluación de los procesos, por lo cual los investigadores tienen como reto generar mecanismos de validación y verificación que le ayuden a generar modelos de proceso de mayor calidad, lo que permitirá una mejor eficiencia organizacional y garantizar productos de mejor calidad [2, 3]. La obtención de procesos de calidad que fortalecen la gestión de una organización lleva a las organizaciones a buscar como reutilizar todo el conocimiento obtenido en la formalización de un proceso, como generar procesos adaptables a partir de un núcleo de proceso que ha sido evaluado e implementado con éxito. Por lo que en la industria del software ha da paso al concepto de variabilidad de los modelos de procesos, cuya única finalidad está basada en generar procesos adaptables a diferentes tipos de proyectos o necesidades de mercado [4] dándole así mayor provecho al conocimiento logrado para la formalización del modelo de proceso, a los casos exitosos de gestión y ampliando las posibilidades de mercado para la organización.

### 1.1. Motivación

Con el reconocimiento por parte de la industria de las ventajas que son obtenidas con el desarrollo formal de procesos vienen generándose esfuerzos con el fin de alcanzar estos beneficios. Sin embargo también se toma conciencia que, lograr estas metas, conlleva una serie de costos e involucra la necesidad de un gran esfuerzo para la definición de sus procesos [5]. Es necesario tener en cuenta que cada organización maneja una metodología, diversos tipos de proyectos y diferentes contextos de negocios. Lo que

genera que la comunicación de un proceso sea compleja. Adicional a la diversidad de elementos que intervienen a la hora de modelar el proceso dando como resultado una enorme complejidad para los diseñadores de procesos para generar modelos de calidad que fortalezcan la productividad de la organización [6, 7].

Con el propósito de formalizar la definición de los modelos de procesos, han venido desarrollándose diversos lenguajes que favorecen su definición, entre los que podemos mencionar una serie de lenguajes de propósito específico como SPEM 2.0 [8], BPMN [9], XPDL [10], LittleJil [11], vSPEM [12], PROMENADE [13], entre otros. Algunos de estos lenguajes son de propósito específico a procesos de software como SPEM2.0 y PROMENADE [5]. Todos estos esfuerzos están enfocados a conseguir cierto grado de estandarización en la industria. Pero con cada nuevo avance se generan nuevos retos para la investigación. Las organizaciones tienden a evolucionar y con ella evolucionan sus procesos lo que evidencia la necesidad de generar procesos flexibles y adaptables a los nuevos contextos que puede presentar la organización. Por lo cual los lenguajes también evolucionan e incorporan dentro de sí mecanismos que le permitan afrontar estos retos [1, 4]. Uno de estos lenguajes es SPEM el cual es un meta-modelo con gran acogida dentro de la industria que contiene una serie de constructos para adaptar los modelos a las nuevas necesidades organizacionales o al contexto del proyecto que va ser desarrollado [8]. Para entender la importancia que tiene que los lenguajes contengan mecanismos de adaptación a las necesidades organizacionales, cabe mencionar a vSPEM una adaptación del lenguaje SPEM el cual adiciona una serie de constructos que permiten especificar con mayor claridad las variabilidades de un modelo a otro, pero que a pesar de estas adiciones es aún un prototipo lo cual reduce su uso dentro de la industria [12].

## **1.2. Contexto del problema**

La gran cantidad de factores involucrados a la hora de definir un modelo de proceso dan a entender el enorme esfuerzo que es requerido por parte de las organizaciones para la definición de sus modelos de procesos con calidad [1]. Con la necesidad de las organizaciones de adaptar sus procesos a las condiciones del contexto, es necesaria la incorporación de variabilidad. Esto genera una mayor dificultad para la definición de los procesos y a su vez incrementa la complejidad para la validación y verificación del mismo [14]. Todo esto lleva a la generación tanto de estándares, metodologías y (o) herramientas que han favorecido en cierto grado al proceso de diseño y validación de modelos de proceso.

Por lo tanto uno de los grandes retos que se presentan está ligado a la necesidad de contar con mecanismos, metodologías o herramientas que le permitan la verificación y validación de modelos de proceso. Esto requiere de nuevos esfuerzos y mayor colaboración entre la industria y la academia con el fin de generar estrategias que le permitan a las organizaciones generar modelos de proceso de mayor calidad que garanticen la efectividad de los procesos que son ejecutados [5, 15]. Todo este contexto,

más la adición de variabilidad dentro de la definición de procesos, genera la necesidad de contar con estrategias, métodos o herramientas que faciliten en cierto grado la verificación de los modelos de proceso.

A pesar de contar con algunas estrategias para la evaluación de los modelos de procesos, es de notar que las limitaciones para implementar éstas estrategias, la poca adaptación al contexto de las empresas o el mismo costo que involucra su implementación, hace resaltar la necesidad de nuevas implementaciones o la generación de adaptaciones de los métodos existentes, que le permitan a las industria validar la calidad de los procesos que en ella se implementan [15].

### **1.3. Problema**

Todo este contexto demuestra que a pesar de que existen lenguajes fuertemente desarrollados y trabajados dentro de la industria como es el caso de SPEM, no garantiza que los procesos definidos bajo el mismo estén bien definidos [15, 16], más aún cuando estos procesos involucran variabilidad dentro de su definición, debido a la complejidad adicional generada al tener que determinar en qué parte del proceso resulta más efectivo o no la implementación de un punto de variación o en sí mismo el cómo se debe desarrollar la resolución de dicha variación en el diseño. Así un enfoque manual de revisión resulta costoso y poco efectivo [4]. En consecuencia, el análisis asistido juega un papel fundamental para asegurar la calidad de los modelos de proceso con variabilidad. Particularmente, indica la necesidad de contar con métodos y herramientas que permitan el mejor análisis y la retroalimentación de los modelos realizados con SPEM que incorporan variabilidad dentro de su definición. Existen algunos enfoques herramientas que permiten analizar la calidad de los modelo de procesos, sin embargo, hasta donde sabemos, no hay alguno que considere el análisis de procesos que incluyan variabilidad en su especificación.

### **1.4. Pregunta de investigación**

Con toda esta problemática y los antecedentes encontrados en cuanto a verificación y validación de modelos de proceso que incorporan variabilidad, se plantea la siguiente pregunta de investigación ¿Cómo facilitar la evaluación de la correctitud de los modelos de proceso especificados con SPEM 2.0 considerando sus mecanismos de variabilidad?

## 1.5. Objetivos

### 1.5.1. General

- Definición de un conjunto de Blueprints <sup>1</sup>(VPB- Variability Process Blueprint), para facilitar el análisis visual de la variabilidad en familias de modelos de procesos especificados en SPEM 2.0.

### 1.5.2. Específicos

- Determinar técnicas y herramientas para evaluar la correctitud<sup>2</sup> de modelos de proceso que incluyan variabilidad
- Definir un conjunto de Blueprints que faciliten el análisis visual de la correctitud en la resolución de variabilidad en modelos de procesos definidos en SPEM 2.0.
- Implementar el conjunto de Blueprints propuestos en vAVISPA que permitan identificar la correctitud de la resolución de la variabilidad de forma visual.
- Aplicar los Blueprints implementados en vAVISPA para evaluar la correctitud en dos familias de modelos de procesos existentes a través de un estudio de caso.

## 1.6. Propuesta y Justificación

El desarrollo software involucra factores como requerimientos, conocimiento del negocio, stakeholders, prácticas de desarrollo, gestión de recursos, test, buenas practicas, etc. Todos estos factores son determinantes a la hora de definir costos y tiempos de desarrollo. Las empresas dedicadas al desarrollo software poco a poco han tomado conciencia de los beneficios operacionales y organizacionales que involucran cada uno de los factores que intervienen en un desarrollo, formalizando así el conocimiento obtenido a partir experiencia de desarrollo, como resultado de estos esfuerzos surgen los PM (Process Model) [1]. Estos modelos desarrollados con el fin de condensar conocimiento y facilitar la comprensión de los diferentes tipos de usuarios involucrados en el desarrollo de

---

<sup>1</sup> Blueprint: Vista a alto nivel que pretende representar una entidad a través de un conjunto de características visuales que lo identifiquen. 17. Lanza, M. and S. Ducasse, *A categorization of classes based on the visualization of their internal structure: the class blueprint*. Proceeding OOPSLA '01 Proceedings of the 16th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, 2001. **36**(11, 11/01/2001 ): p. 300-311.

<sup>2</sup> Correctitud: Que incluye elementos correctos y las relaciones correctas entre ellos y no violen las reglas y convenciones establecidas. 18. Mohagheghi, P., V. Dehlen, and T. Neple, *Towards a Tool-Supported Quality Model for Model-Driven Engineering*. Proceedings of the 3rd Workshop on Quality in Modeling, 2009: p. 74-88.

un producto software, no son desarrollados o pensados a perpetuidad, la naturaleza variable de las organizaciones genera que los PM sean adaptados al contexto de negocio, de esta forma se incorpora el concepto de variabilidad al desarrollo de modelos de proceso [4].

El problema de formalización procesos ha venido siendo trabajado desde un tiempo atrás dando como resultado los lenguajes de definición de modelos (PMLs - Process Model Languages), algunos de estos lenguajes han incorporado estructuras para representar la variabilidad como es el caso de SPEM 2.0 [8]. La definición de modelos de proceso ha sido una temática abordada por diferentes estudios, de la misma forma se cuenta con herramientas, propuestas, metodologías y normas que buscan garantizar la calidad de los modelos de procesos. Pero al realizar un análisis de los mecanismos disponibles observamos que ninguno está enfocado en analizar el impacto que tiene la variabilidad en un modelo de proceso. Tomando en cuenta que la inclusión de dichos mecanismos de variabilidad incrementa la complejidad de la validación y verificación de calidad en los modelos que incluyen dicha característica [14]. Teniendo en cuenta que partir de un modelo de proceso del cual esta garantiza su calidad, en ningún momento es garantía que el modelo resultante al incorporar variabilidad conserve dicha característica. Con todo este contexto surgen propuestas de un enfoque visual ya que facilita al diseñador detectar de forma natural algún tipo de inconsistencia en el modelo. Teniendo como base la propuesta AVISPA la cual presentó un análisis a partir de Blueprints para la evaluación del modelo de proceso tomando los aspecto de tareas, roles y productos de trabajo [15]. Con la limitante que el análisis de la propuesta no toma en cuenta la incorporación de variabilidad en la definición del modelo de procesos, pretendiendo extender esta propuesta con el fin de analizar dichos modelos y determinan el impacto que tiene sobre la correctitud el incorporar variabilidad en la definición de modelos de procesos definidos en SPEM 2.0.

El presente proyecto pretende hacer un aporte significativo a la academia, especialmente con la línea de investigación de Ingeniería del Software del grupo IDIS (Investigación y Desarrollo en Ingeniería del Software) , puesto que su ejecución requiere de la utilización de conocimientos informáticos, así como también de temas relacionados con la Ingeniería del Software, concretamente, aspectos relacionados con los métodos de evaluación visual de procesos de software así como de estrategias de definición y evaluación de variabilidad. Teniendo en cuenta que ésta es una temática de gran importancia en la actualidad ya que permite adaptar a las organizaciones a diferentes contextos mejorando la productividad, reduciendo costos y dando mayor aprovechamiento al conocimiento adquirido por medio de las experiencias. Ya que el análisis de la variabilidad a sido poco utilizada y explorada en el ámbito académico de la Universidad del Cauca.

## 2. ESTADO DEL ARTE

### 2.1. Marco Teórico

El proceso software según Lonchamp lo define como *"Un conjunto de pasos parcialmente ordenados, conjuntos de artefactos, recursos humanos e informáticos, las estructuras organizativas y las limitaciones, agrupadas con la intención de producir y mantener los entregables de software solicitados"* [19]. Otra definición que podemos encontrar de lo que es un proceso software es como "Un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, instalar y mantener un producto software" [20]. podemos observar con estas definiciones que el proceso software involucra diversos factores que afectan directamente la efectividad y eficiencia del mismo, lo que indica la necesidad de generar una abstracción que permita representar como estos elementos de proceso interactúan para poder llevar a cabo un proceso software en la forma más adecuada posible [15]. Como mecanismo de representación de esa abstracción, aparece el concepto de modelo de proceso (PM) software que es definido como una descripción abstracta y parcial de la realidad del proceso software que se expresa a través de un lenguaje de modelado adecuado, para el caso de modelos de procesos son conocidos como Lenguaje de modelado de procesos (PMLs). La especificación de un modelo de proceso puede realizarse formal o informalmente [21]. La especificación informal es simple, sin embargo puede introducir ambigüedades, ser incompleta o inconsistente, con lo que la aplicación del modelo puede realizarse de forma errónea. La especificación formal es más difícil, pero al ser más precisa puede ser evaluada y utilizada para producir otras especificaciones en forma asistida y consistente. Por ejemplo hacer una validación sintáctica, ejecutar una simulación, realizar una adaptación o una instanciación [19]. La formalización de procesos genera la necesidad de contar con meta-modelos o lenguajes que permitan definir con más claridad y precisión los diferentes elementos que se ven involucrados en un proceso software [4].

Con el fin de avanzar en este propósito se han generado diversos lenguajes desde diferentes dominios, que pretenden facilitar constructos para representar modelos de procesos de software. Estos están agrupados en lenguajes de definición y lenguajes para la ejecución de procesos.

### 2.2. Lenguajes de ejecución de procesos

**Little-Jil** es un lenguaje de programación para la coordinación de los procesos en ejecución, es un lenguaje de alto nivel con una sintaxis formal y unas semánticas operacionales rigurosamente definidas. El desarrollo de Little-Jil está basado en cuatro principios fundamentales: simplicidad, expresividad, precisión y flexibilidad. Basado en estos principios dispone de dos estructuras para el modelado: paso y conexión entre los

pasos. La coordinación del proceso se basa en una jerarquización de los pasos que conforman el proceso, cada paso puede contener o no pre-requisitos o post-requisitos, al mismo un paso solo puede iniciarse si es ejecutado por un agente [11]. Como lenguaje para el modelado de procesos genérico y de alto nivel, sus escasas estructuras especializadas lo limitan a la hora del modelado de todos los factores que intervienen en un proceso software.

**BPMN** (Business Process Model and Notation) estándar propuesto por la OMG para el modelado de procesos de negocio, este lenguaje está basado en notación UML. El proceso se representa por medio de un gráfico que contiene elementos con un flujo, el cual corresponde a un conjunto de actividades, eventos o puntos de decisión. BPMN es un lenguaje con una gran variedad de constructos para el modelado de un proceso [9], pero está más enfocado para el modelo de procesos a nivel de gestión, lo cual lo limita para abarcar muchos de los factores técnicos que intervienen dentro de un proceso software.

**XML Process Definition Language** (XPDL) este lenguaje está desarrollado en el ámbito de gestión y control de procesos de negocio, gran parte de la definición de conceptos y terminología es tomada de BPMN. Su principal objetivo es el de intervenir como un formato estándar que puede ser compartido entre diferentes herramientas, siempre limitado por la definición de su propio meta-modelo y la capacidad de expresión de la herramienta. Su notación está basada en XML y cuenta con definición para diversos elementos propios de la gestión de negocios, Adicional cuenta con una representación gráfica con el propósito de facilitar la comunicación entre el usuario de negocio y el técnico. Sus mayores fuertes están en la gran cantidad de elementos de su definición que le permite operar entre diversas herramientas, y a la vez su mayor limitante ya que las particulares de la herramienta reducen el potencial del modelo de negocios [10].

**Business Process Execution Language** (BPEL) este lenguaje permite la gestión y control de servicios así como especificación de protocolos de negocio, está basado en XML. BPEL permite la comunicación, organización y control del flujo de datos entre diversos BPELs lo que le da la capacidad de encapsulamiento, y le da una gran fortaleza para el manejo de procesos de gran complejidad. La definición formal de un documento BPEL consiste en tres elementos que son: Asociados que corresponden a aquellos servicios que interactúan con el proceso, una serie de variables que son las que permiten definir el estado del proceso y una actividad la cual es la que determina la lógica de interacción entre el proceso y sus asociados [22].

### 2.3. Lenguajes para el modelado de procesos Software

**PROMENADE** (PROcess-oriented Modelling and ENActment of software DEvelopments) es un lenguaje basado en notación UML, tratando de aprovechar la familiaridad de los



modeladores con el contexto de la ingeniería de software. Dispone una serie de estructuras definidas para el modelado como son: documentos, mecanismos de comunicación, tareas, agentes, recursos y roles. La construcción del modelo en PROMENADE esta basada en una definición de un modelo de referencia el cual resulta como base para la generación de nuevos modelos. En el modelo de referencia están definidas las primitivas que incorpora el lenguaje, por lo que realmente es un meta-modelo. El desarrollo de otros modelos consiste en generar instancias de las clases del meta-modelo (modelo base), y asignar un valor a cada uno de los atributos de las distintas instancias que se han generado [13]. PROMENADE dispone de un conjunto de modelos de referencia, de los cuales extiende para generar nuevos modelos, siendo así mecanismo para definir la variabilidad de un modelo de proceso a otro, lo cual lo limita al no disponer de un manejo particular para el proceso de variabilidad.

**E3** Propuesta académica desarrollada a partir de un enfoque orientado a objetos (OO), ha generado una sintaxis formal para la definición de modelos de procesos, la cual proporciona una representación gráfica que facilita la representación y comunicación del modelo entre los diferentes involucrados. E3 está basado en tres niveles conceptuales: creación, definición y el nivel de instanciación. En el nivel de creación se desarrollan las clases y asociaciones involucradas en el modelo de procesos. Para el nivel de definición las clases y asociaciones creadas son ensambladas entre si lo cual genera un plantilla. Por ultimo en el nivel de instanciación las clases y asociaciones pertenecientes a las plantillas son instanciadas en objetos y referenciadas en el modelo de procesos. Entre las características de la OO en E3 podemos encontrar la de herencia y agregación la cual le permiten la reutilización de componentes ya desarrollados, Las clases heredan sus características de la clase E3Object, mientras que las asociaciones heredan desde asociaciones llamadas *binary* o *ternary*. También es de resaltar que este lenguaje cuenta con una herramienta para el desarrollo de modelos llamada E<sup>3</sup>p-draw [23].

**FUNSOFT** nets consiste en una red petri de alto nivel, su semántica está definida en términos de una red Pr/T lo que permite usar técnicas estándar de análisis de redes petri para la examinación de propiedades de modelos de procesos. Las redes petri no están desarrolladas para el modelado de proceso software, por lo que muchas de las características que debe poseer un lenguaje software no pueden llegar a ser valoradas. Para definir FUNSOFT nets se basa en una tupla de la forma (S,T,F,O,P,J,E,C,A,Mo) donde (S,T;F) denota la red, los elementos de S son llamados canales y los elementos de T llamados instancias. Los demás elementos están definidos de la siguiente forma:

- O: define una serie de tipos de objetos.
- P: define una serie de predicados de activación.
- J: corresponde al conjunto de trabajos o actividades, cada instancia define un trabajo.
- E: los arcos que controlan el flujo de información corresponden a este elemento.

- C: define dos funciones una asignar atributos a cada canal y la segunda definen el orden a ser consumidos.
- A: las instancias (trabajos), pueden ser asignadas de 4 formas que se definen en este elemento.
- Mo: Corresponde a una primera valoración de los canales para el orden a ser consumidos.

FUNSOFT nets también cuenta con una representación gráfica que facilita la creación de modelos de proceso. También es de resaltar la suite GRAS que contiene diferentes herramientas que permiten la definición de la red, el análisis y la simulación lo cual la convierten en una herramienta muy completa para desarrollar redes con FUNSOFT [24].

**MVP-L** Consiste en un prototipo de lenguaje de modelado de procesos desarrollado a partir del proyecto MVP una iniciativa impulsada por la Universidad de Maryland y la Universidad de Kaiserslautern. MVP-L distingue tres tipos de elementos: procesos, productos y recursos.

- Procesos: Representan las actividades que se ejecutan durante un proyecto
- Productos: Corresponden a todos los entregables de un proyecto (producto, subproductos, artefactos y documentación del producto)
- Recursos: Son todas las entidades necesarias para ejecutar el proceso

Cada uno de los elementos cuenta con una serie de atributos que definen el modelo de proceso [25]. Para coordinar la interacción entre las actividades debe realizarse un control del flujo de los procesos que deben ser ejecutadas, adicionalmente, existe un flujo de control de los recursos que se deben consumir. Este lenguaje al ser un prototipo cuenta con muchos limitantes para modelar procesos y aún no cuenta con una herramienta que apoye el desarrollo y análisis del modelo de proceso [26].

**UML4SPM** Corresponde a una propuesta desarrollada con el fin de superar algunas de las limitaciones detectadas en la definición de la OMG SPEM 1.1, para lo cual toma las ventajas en cuanto a expresividad y familiaridad presentadas con el UML 2.0 y las extendieron con un sub-conjunto de elementos que resultaban favorables para la representación de modelos de procesos. El UML4SPM se encuentra definido a partir de una jerarquía de paquetes que en su nivel más exterior está basado en dos paquetes: el *SPEM\_Foundation* y el paquete *SPEM\_Extension*. El primero contiene todas las características del UML 2.0 requeridas para el modelado de procesos, principalmente las que corresponden a actividades, acciones y comportamiento de un proceso adicional a esto también encontramos el paquete Kernel. Mientras que el paquete *SPEM\_Extension* incorpora los paquetes *ProcessStructure* que da soporte para la gestión, control y representación de los estados del proceso, y el paquete *WorkProduct* que define nuevas

características a la clase WorkProduct que le dan mayor flexibilidad y control dentro del proceso [27].

**SPEM 2.0** desarrollado por la OMG (Object Management Group). SPEM 2.0 está basado en el meta-modelo MOF<sup>3</sup> una especificación generada por la OMG. SPEM 2.0 es un meta-modelo que esta centrado en proporcionar las estructuras de información necesarias para el modelado de actividades y la gestión de un proceso. Proporciona los conceptos necesarios para modelar, documentar, presentar, gestionar, intercambiar y promulgar los métodos y procesos involucrados en un desarrollo. SPEM 2.0 consta de siete paquetes, los cuales se combinan entre si dependiendo de la dependencia y perfil buscado. Estos paquetes son: *Core*, *Process Structure*, *Process Behavior*, *Managed Content*, *Method Content*, *Process With Methods* y *Method Plugin* cada uno de los cuales dispone de constructos para la definición de un modelo de proceso.

---

<sup>3</sup> MOF: Meta-objeto facilitador, es la base de la estandarización de la OMG, para la exportación e importación de sus modelo.28. OMG, O.M.G. *OMG's MetaObject Facility*. 2012 [cited 2012; Available from: <http://www.omg.org/mof/>.

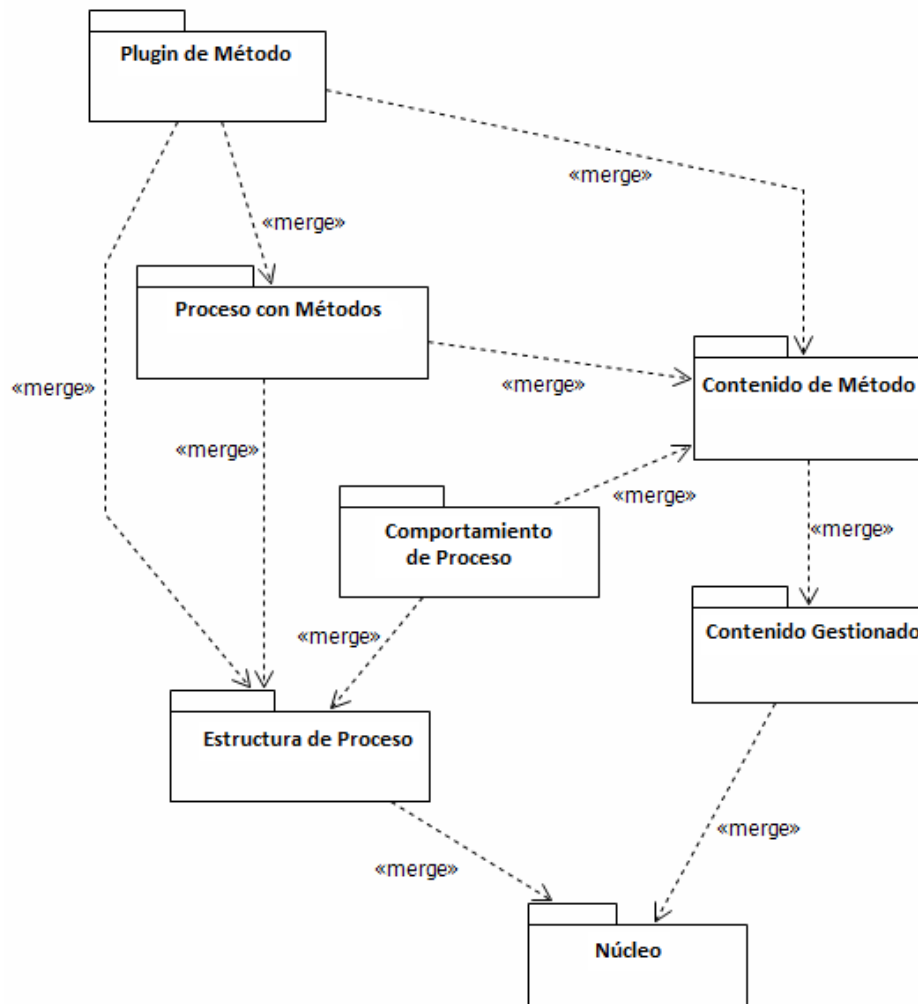


Fig. 1 Estructura de SPEM 2.0 Meta-Modelo

- **Núcleo:** Contiene la definición de las clases y abstracciones fundamentales del meta-modelo que componen la base de todas las clases de los demás paquetes del meta-modelo.
- **Estructura de Proceso:** Define la base para los modelos de proceso, apoyando la creación simple y flexible de los mismos. En su estructura básica se encuentra la descomposición de las actividades anidadas, las cuales contienen un listado de referencia a las actividades a llevar a cabo, los roles involucrados y los productos de trabajo de entrada y salida de cada actividad. Adicionalmente proporciona mecanismos para la reutilización de procesos.
- **Comportamiento de Proceso:** Define la estructura básica para definir el modelo de proceso, pero no alcanza a definir el comportamiento de los elementos. Este paquete permite extender las estructuras del paquete **process structure** con modelos de comportamiento. Este paquete no define sus propios modelos de

comportamiento, si no que proporciona links a modelos de comportamiento definidos externamente (pe. UML).

- **Contenido Gestionado:** Contiene los conceptos para la gestión textual de los contenidos incorporados en el proceso, está diseñado para soportar esas descripciones que le permiten al modelo salir de la definición formal y dar pautas en lenguaje natural para facilitar la adopción del modelo de proceso. Estos conceptos pueden usarse de forma independiente o enlazándolos a los conceptos del paquete **Process Structure**.
- **Contenido de Método:** Almacena los conceptos que permiten generar una base de conocimiento para los usuarios y las organizaciones independiente del tipo de proyecto o el ciclo de vida de desarrollo. Aquí está registrada la explicación paso a paso de las mejores prácticas que conformaran la base de conocimiento de la organización, las cuales pueden ser adaptadas a diversos proyectos.
- **Procesos con Métodos:** Define nuevas estructuras y redefinen otros conceptos del paquete **process structure** con instancias de conceptos del paquete **method content**. Son incorporadas las cualidades descritas por el paquete **method content** con las estructuras que pueden ser descritas a través del paquete **process structure**.
- **Plugin de Método:** Introduce los conceptos para el diseño y gestión de la mantenibilidad, reusabilidad y configuración de librerías o repositorios de contenidos de métodos y procesos. Con los conceptos introducidos con los plugins de métodos, componentes de procesos y variabilidad pueden definirse nuevos procesos con más capacidades.

Se observa que SPEM 2.0 no solo dispone los constructos necesarios para la definición del proceso, sino que proporciona conceptos y mecanismos para gestionar las lecciones aprendidas dentro del proceso y mecanismos para la gestión de repositorios que le permiten generar procesos con más capacidades [8]. Estos y muchos otros factores demuestran porque SPEM 2.0 cuenta con gran apoyo y acogida por la industria de desarrollo software para especificar sus procesos. Cabe de resaltar que para el desarrollo de modelos de procesos contamos con diversas herramientas que facilitan su construcción y validación, como es el caso de Eclipse Process Framework Composer (EPFC) una herramienta desarrollada dentro el entorno Eclipse la cual sirve para editar fragmentos de procesos y metodologías, a la vez que facilita su promulgación generando la documentación adecuada en formatos para la web los cuales pueden ser utilizados por otras herramientas case. EPFC utiliza el **Unified Method Architecture (UMA)** que a su vez está basado en el meta-modelo SPEM 2.0. Entre otros factores que impulsa el uso de EPFC es la comunidad que da soporte y apoyo a la evolución de EPFC, adicional que la herramienta incluye las implementaciones públicas, abiertas y gratuitas de varias metodologías [20].

## 2.4. Correctitud a nivel de Procesos

Al igual a lo que se puede notar a lo largo de esta investigación en cuanto a técnicas, estrategias o metodologías a nivel de modelos de procesos la información y trabajos de investigación de correctitud y variabilidad han sido muy escasos. Los esfuerzos generados en cuanto a esta temática refiere han sido desarrollados más a nivel de modelos de procesos de negocio (BPM) y procesos de desarrollo, aunque es de resaltar que estos esfuerzos están enfocados a nivel de definición del proceso y técnicas de configuración para el desarrollo de las variabilidades, teniendo como principal dificultad la evaluación de los modelos de procesos generados, debido a la gran cantidad de modelos que pueden surgir a partir de la incorporación de variabilidad. Algunos de estos esfuerzos los presentamos a continuación. Para analizar la correctitud se han definido dos enfoques de evaluación, el primero de ellos es una evaluación de la correctitud sintáctica donde el objetivo de la evaluación está basado en que la definición del modelo de proceso sea consistente con el meta modelo garantizando que cumpla con la restricciones preestablecida desde su definición. El segundo enfoque es una evaluación semántica de la correctitud la cuál busca analizar el modelo de procesos a nivel del contexto, es desarrollado el análisis basado en una verificación de que tan efectivo resulta el modelo para las restricciones y contexto organizacional [29]

Entre las técnicas de verificación encontradas esta la propuesta de MAM nets (Modeling, Analysis, and Management nets) la cual consiste en una técnica derivada de las redes petri, proponiendo un mecanismo para el modelado de procesos a partir de una tupla similar a como es trabajada en las redes Petri. La validación del modelo está desarrollada en base a dos puntos de vista, el primero consiste en un análisis estructural que es realizado en la etapa de modelamiento del proceso; basado en el estándar de redes Petri este análisis busca detectar anomalías en el modelo como puntos muertos o sifones en la red. La segunda parte esta especializada en analizar las ayudas gerenciales para la fase de gestión del proceso centrándose en la ejecución del proceso respecto a las limitaciones de recursos humanos y actividades propias del proceso, con base a los resultados de este segundo análisis es planteada la retroalimentación del modelo para que sea más efectivo al contexto organizacional [30].

Otra de las propuestas que podemos encontrar también está desarrollada para modelos de negocios pero a diferencia del primer caso basa su análisis en la configuración de modelos de procesos, para este propósito utiliza un tipo especial de redes llamadas redes de flujo de trabajo (Work Flow nets), el proceso parte de un modelo de proceso correctamente definido sintáctica y semánticamente, el cual es tomado como núcleo para generar diversos modelos de procesos a partir de ciertas reglas de configuración, se determina una configuración de un modelo a partir del valor que le es asignado a cierto nodo configurable, estos nodos pueden ser ocultados o removidos de la red, reestructurando el modelo. Para garantizar la correctitud de la configuración del modelo de proceso de negocio proponen una serie de restricciones de lógica proposicional que

garantizan el descarte de configuraciones incorrectas de esta manera generar un modelo de proceso de negocio sintáctica y semánticamente correcto a partir de un modelo de proceso configurable [31].

Entre las escasas propuestas que podemos encontrar para analizar modelos de proceso encontramos AVISPA, un enfoque basado en el análisis visual de modelos de procesos. Para el desarrollo del análisis propone el uso de tres Blueprints (Roles, Tareas y Productos de Trabajo) que básicamente corresponde a una representación gráfica de las interacciones dentro del modelo y sus características. Como resultado de esta investigación detectaron una serie de patrones de error frecuentes en la especificación de modelos de procesos, los cuales permiten identificar pérdida de información en la especificación o un error conceptual en la definición. Para desarrollar este análisis parte de una definición de modelo de proceso SPEM 2.0 del cual obtienen sus interacciones entre los objetos de análisis (Roles, Tareas y Productos de trabajo) para generar los Blueprints, de estas vistas generadas son resaltados posibles falencias en la definición del modelo para que sean evaluadas por el diseñador del modelo.[15]

## **2.5. Variabilidad de Modelos de Procesos**

Los lenguajes de definición de modelos de procesos fueron desarrollados con el propósito de estandarizar en cierto grado el modelado de procesos, en busca de este mismo objetivo han desarrollado una serie de modelos referencia como son CMMI [32], ISO 12207 [33], SCAMPI [34], SCRUM[35], ISO 15504 [36], entre otros, e incluso procesos definidos como el Proceso Unificado [37]. Dentro de las organizaciones, son definidos sus procesos a partir de estándares, en el marco de sus proyectos de mejoramiento. Los procesos resultantes tienen la dificultad que es presentada con los modelos genéricos, donde existe poca relación entre un proceso y el contexto que será aplicado, el tipo de individuos involucrados, las actividades y las propias características de la organización. Esto implica un problema al momento de diseñar el modelo de referencia adaptable, y al adaptarlo a su contexto. Toda esta problemática lleva a incorporar constructos que faciliten modelar variabilidades que permitan al procesos genéricos adaptarse al contexto organizacional [12].

Simmonds et al. [5] analiza lenguajes de modelado de variabilidad de procesos, que están especializados en el modelado de líneas de productos. Uno de estos lenguajes es el Feature Model, el cual consiste en un enfoque basado en la definición de características, con dichas características pueden determinar variaciones, que le permitan definir un producto. El conjunto de características básicas de todo producto es convertido en la raíz (root). Desde esta raíz empieza a diversificarse nodos que corresponden a características (padres) y sub-características (hijos) que desprenden de los nodos padres. Un padre puede poseer cierta cardinalidad de hijos que determinan posibles variaciones de características con respecto al padre. Desde Feature Model se permite representar de una manera gráfica las variaciones que están presentes. Otro lenguaje propuesto por los

autores es el Orthogonal Variability Model (OVM) propuesto por Klaus Pohl, Günter Böckle y Frank J. van der Linden [38] basado en puntos de variación y variaciones. Para el modelado de procesos, un punto de variación identifica una parte en el proceso, que puede ser opcional o que puede ser reemplazada con otra que cumplirá similar función. Mientras que una variante, consiste en el fragmento de proceso que es aplicado sobre un punto de variación. Como desventaja más notoria de esta técnica es su generalidad [14, 39]. Ambos lenguajes (Feature Models y OVM) solo están desarrollados a partir de una línea de productos, lo cual no necesariamente corresponde con las necesidades del modelo de procesos pero brindan fuertes bases para el modelado de variabilidad [5].

Es de notar los grandes esfuerzos para mejorar el entendimiento de las líneas de productos que incorporan variabilidad, esfuerzos no tan desarrollados como los anteriormente expuestos han sido desarrollados del lado de modelado de procesos. Uno de estos esfuerzos lo podemos observar dentro de la definición de SPEM 2.0 en la cual han incorporado un conjunto especializado de constructos cuya finalidad es la de expresar la variabilidad del modelo de procesos. La relación de variabilidad presentada entre dos elementos, para lo cual SPEM está basado en tres principios: contribución, reemplazo, extensión. De los cuales desprenden cuatro posibles constructos para expresar la variabilidad que son: contribución, reemplazo, extensión y extensión-reemplazo. Cada uno de ellos configura una determinada asociación entre las entradas y salidas así como de los atributos que dispone cada una de las entidades relacionadas, estas relaciones de variabilidad están representadas entre un elemento variable y un elemento base [4, 8].

| Tipo de variabilidad | Descripción   |
|----------------------|---|
| na                   | No presenta Variabilidad en el elemento   |
| Contribución         | Presentada cuando instancias de un elemento variable realiza contribución a un elemento base sin que estos aportes modifiquen las propiedades ya existentes, la contribución se realiza con valores de atributos y asociaciones instanciadas. Al realizarse la contribución el elemento base variable modifica su lógica aumentándola con los valores de atributo recibido y las asociaciones adquiridas.                                     |
| Reemplazo            | Instancias de un elemento variable reemplazan a un elemento base variable afectando directamente las propiedades que dispone. Las propiedades reemplazadas corresponden a valores de atributos y asociaciones instanciadas. Al realizarse el reemplazo toda la lógica del elemento base son reemplazadas con la lógica del elemento variable, y las asociaciones entrantes que disponía son reasignadas al elemento que realizó el reemplazo. |
| Extensión            | Permite la fácil reutilización desde un elemento base, permitiendo un tipo de herencia desde el elemento base hacia el elemento variable. Los valores de atributos y asociaciones instanciadas son heredadas por el   |



|                     |  |
|---------------------|--|
|                     | elemento variable. En esencia se puede decir que el elemento variable dispone de las mismas propiedades que el elemento base, pero las propiedades heredadas podrían ser reemplazadas con sus propios valores. Por esto se podría decir que la extensión no consiste en un método para modificar el contenido, si no es más bien un método que dispone de las habilidades de un elemento y le permite redefinirla a partir de sus propios valores. |
| Extensión-Reemplazo | Combina los efectos de la variabilidad extensión y reemplazo. A diferencia de la variabilidad de reemplazo que substituye los atributos y las asociaciones de salida, la variabilidad tipo extensión-reemplazo solo permite que se reemplace los valores obtenidos por la extensión (heredados), en otras palabras se permite seleccionar qué elementos serán reemplazados.  |

Tabla 1 Tipos Variabilidad SPEM 2.0

Otro de los esfuerzos que podemos mencionar en el modelado de variabilidad corresponde a vSPEM. Está notación es una extensión de SPEM, que es desarrollada como propuesta por T. Martínez Ruiz et al. [12] tomando en cuenta las limitaciones que disponen los mecanismos de SPEM en cuanto a comprensibilidad al modelar la variabilidad en procesos. Para ello incorpora nuevos elementos centrados en la descripción de modelos de procesos que involucran variabilidad, para lo cual incluye un nuevo paquete llamado **ProcessLineComponents**. Este nuevo paquete está basado en puntos de variación y variaciones. Un punto de variación puede presentarse en cualquiera de los elementos que están definidos en SPEM 2.0. Para la construcción de modelos, SPEM dispone de una serie de clases abstractas que representan: actividades, tareas, roles, productos de trabajo y herramientas. Los puntos de variación y variantes heredan desde los elementos de SPEM los cuales poseen la capacidad de presentar algún tipo de variabilidad [5, 12]. A pesar de la capacidad que dispone vSPEM para representar más claramente la variación de los procesos, no ha sido incorporado a la industria, por lo cual carece de procesos reales para su estudio.

Cada uno de los lenguajes denota la importancia y relevancia que tiene el manejo de variabilidad, por lo que dispone de un enfoque y diversos constructos para afrontar el desarrollo de variabilidad dentro del proceso software. Pese a estos esfuerzos el determinar un punto de variación o un cambio de un proceso a otro, resulta en un gran esfuerzo para el diseñador [6]. El determinar la fase del proceso desde la cual es desarrollada la variación, incurre en un gran desgaste a la hora de desarrollar el modelado de proceso. La variabilidad no es más que el aprovechamiento del conocimiento adquirido al desarrollar procesos que le permiten determinar mejores formas de desarrollar una actividad. Pero el disponer o no de dicha experiencia no garantiza el éxito en la toma de decisión al definir la implementación de dicha variabilidad, pero la correcta definición de los puntos de variación y las fases para desarrollar la variación pueden determinar el éxito o fracaso de un proyecto [40]. Por lo cual es requerido el desarrollo de enfoques, métodos de evaluación y herramientas que permitan a una organización determinar a cierto nivel

que tan efectivo resulta la implementación de una o varias variabilidades dentro de las actividades que desarrolladas en un proceso [4].

## **2.6. Técnicas y enfoques para el proceso de V&V de modelos de procesos**

Las ventajas de promulgación, mejora, valoración, facilidad de adaptación, entre otras, que son obtenidas al definir correctamente los procesos que desarrolla una organización ha llevado a que las organizaciones dediquen mayores esfuerzos para fortalecer sus procesos [2, 3, 41]. Esto se ha convertido en un reto para los ingenieros de procesos por la cantidad de elementos involucrados a la hora de diseñar el modelo de proceso, y aún mayores esfuerzos cuando los modelos incorporan variabilidad dentro de su definición. La dificultad generada radica en definir un punto que pueda presentar una variabilidad, así como en la forma que debe ser la resolución de la misma [4]. Dada esta complejidad adicional, es necesario disponer de métodos o estrategias que permitan la validación y verificación (V&V) de los modelos de procesos que incluyan variabilidad. Dentro de las posibilidades para evaluar un modelo de proceso disponemos de diferentes enfoques: métricas, pruebas piloto, simulación, chequeo formal y análisis visual. Estos enfoques incorporan una serie de metodologías o herramientas que facilitan su aplicación [15].

El análisis por medio de métricas radica en la identificación de una serie de atributos dentro del proceso los cuales son utilizados para determinar o valorar cierta propiedad dentro del modelo de proceso software. Un ejemplo de este tipo de valoración es el estudio realizado por un grupo de investigadores en la industria brasileña cuyo objetivo era determinar la mantenibilidad de un modelo con el fin de establecer qué tan susceptible es un modelo a cambios que pueden presentarse según el contexto de la organización [42]. En este enfoque presenta como retos, el determinar qué desea cuantificar o cuales atributos medir para evaluar una propiedad del modelo de procesos. Aunque existen marcos teóricos que definen una serie de métricas que guían el proceso de evaluación, es requerido un nivel de experticia que permita que sea efectivo dicho proceso. Adicional requiere de la aplicación del modelo de proceso en un proceso real que permita la recolección de las métricas necesarias para la valoración del modelo, lo cual incurre en costos y tiempo consumidos para la organización [15, 42].

Las pruebas piloto consisten en un enfoque orientado a aplicar el modelo de proceso en un proyecto piloto, evaluar los resultados y de acuerdo a su análisis refinar el modelo y los activos de procesos. Este enfoque de V&V es implementado por los modelos de mejora basado en modelos como el CMMI o el estándar de la ISO/IEC 15504. Para el estándar 15504 cuenta con seis niveles de clasificación, que van desde el nivel 0 Incompleto hasta el nivel 6 optimizado. La evaluación del proceso es realizada a través de los atributos de proceso que consisten en métricas que determinar el nivel de cumplimiento que dispone la organización de un nivel específico [36]. Este tipo de evaluación es altamente deseado por la industria por el plus competitivo que le representa a la organización el obtener una

certificación, pero debido a la complejidad y activos requeridos en los niveles superiores resulta poco efectivo en pequeñas organizaciones.

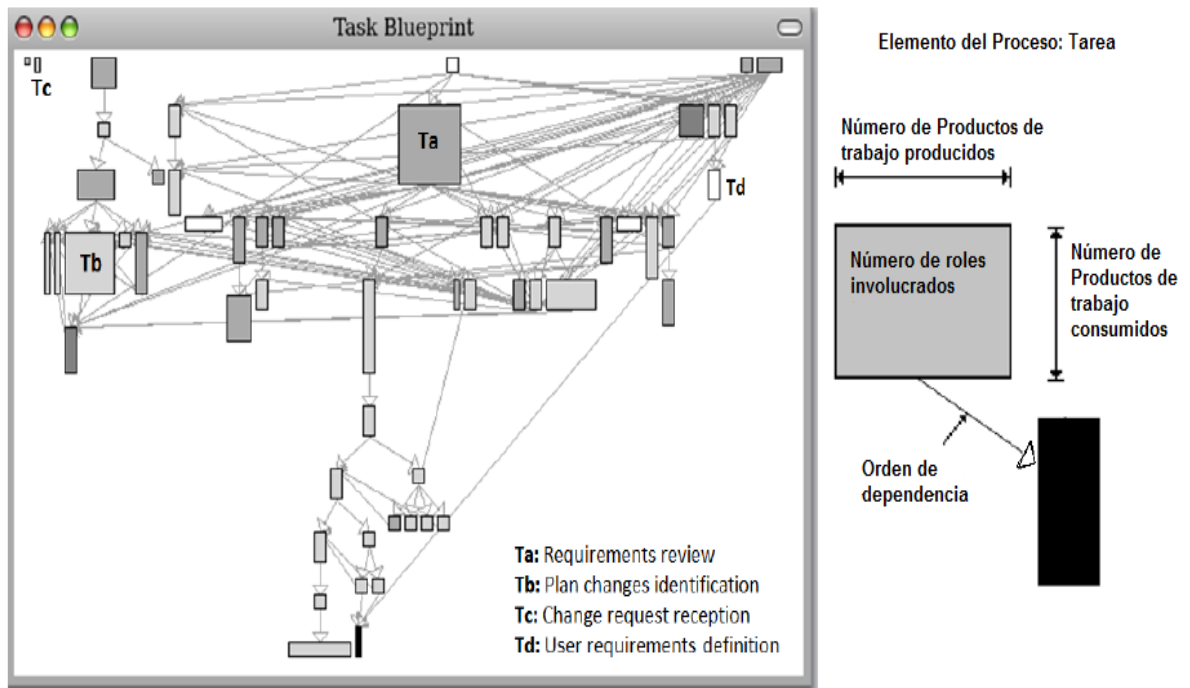
Frente a la dificultad de aplicarlo en pequeñas organizaciones, diversos esfuerzos a nivel de investigación se han emprendido para adecuar un modelo optimizado para las necesidades de este mercado, entre las propuestas encontramos algunas como: RAPID, SPINI, FAME, TOPS, MARES entre otros basadas en la adecuación y el manejo de cierto grado de personalización que le permiten adaptarse a las características de las pequeñas organizaciones [43]. Este método de validación requiere de un proceso previo de valoración para el cual es requerido un nivel histórico de los procesos desarrollados por la organización y un proceso de mejora dentro de la misma así como la participación de un experto que guíe el desarrollo del proceso de mejora, lo cual genera un costo y esfuerzo adicional para su implantación [15, 43]. Sin embargo contar con un modelo de proceso acorde a modelos y estándares internacionales no garantiza que el proceso exhiba ciertas características de calidad como mantenibilidad, correctitud, ni garantiza que sea adecuado para el tipo de organización o el tipo de proyecto.

El análisis por medio de simulación corresponde a la configuración de un escenario, por el cual es puesto a prueba el modelo de proceso, con el fin de determinar su comportamiento, así como identificar posibles falencias o posibles procesos de mejora. Para el análisis por medio de simulación pueden identificarse 3 etapas. La etapa de validación que analiza que tan exacto es el modelo de proceso propuesto con respecto a las necesidades organizacionales. La etapa de verificación donde analiza que las propiedades que están especificadas son efectivas para el objetivo que está planteado y por último la etapa de post-evaluación donde son estudiados y analizados los resultados obtenidos en busca de posibles fallas o mejoras para el modelo de procesos. La simulación corresponde a un método muy efectivo a la hora de validar y verificar el modelo de proceso, pero esta efectividad está estrechamente relacionada a la configuración del ámbito de la simulación. Dependiendo del objetivo o las variables que se deseen analizar, deben configurarse una serie de parámetros que reflejen la realidad por medio de la simulación y así realizar la correcta evaluación del comportamiento que puede presentar el modelo de proceso [44, 45].

Es ampliamente aceptada por la industria las ventajas de contar con un modelo de proceso, pero aunque muchas organizaciones disponen de una definición formal de sus procesos, estas divergen sustancialmente de lo que es la ejecución como tal del modelo de proceso. Tomando en cuenta este contexto Cook y Wolf proponen un método de chequeo formal el cual consiste en contraponer la definición del modelo de proceso y la ejecución del proceso, para medir las discrepancias que existe entre uno y otro. La detección de las discrepancias consiste solo en el primer paso de la validación, el problema radica en cómo medirlas, para lo que están definidas dos técnicas, una lineal que consiste en medir en términos de inserción y eliminación de eventos. Y la no lineal

que está basada en medir el tamaño de la discrepancia. Esta técnica es eficiente para determinar el comportamiento de un modelo de proceso y la asimilación o formalismo con que es aplicado el modelo de procesos [46].

El análisis visual corresponde a una técnica de evaluación que está basada en el aprovechamiento de la capacidad visual de las personas para identificar y localizar problemas en un artefacto. Dado que el análisis es una actividad intensa y que requiere de un gran conocimiento, lo que busca el enfoque visual es facilitar el análisis de la calidad de los procesos en lugar de reemplazar con una automatización en sí misma. Las facilidades están orientadas a las tareas más complejas del análisis como son la identificación y localización de errores, permitiendo a las personas centrarse en las partes más críticas del modelo de proceso. Este enfoque soporta en la representación de cierta característica del modelo de forma visual permitiendo que sea de forma más explícita la detección de anomalías por parte del diseñador del proceso. Un ejemplo de este tipo de enfoque visual de análisis es la propuesta AVISPA, una herramienta que interpreta un modelo de proceso de un formato XML a una representación gráfica denominada Blueprints donde especifican características de los elementos del proceso por medio de representaciones graficas como se observa con las tareas en la Fig.2.



**Fig. 2** Blueprint de tareas generado por AVISPA

Para la implementación de este enfoque requiere generar una correcta visualización de los datos con el fin de permitir de manera más eficiente, la evaluación de gran cantidad de volúmenes de información [47]. Para generar una representación correcta del modelo ha

diseñado una representación enfocada en tres aspectos principales de un modelo de procesos que son: Roles, Tareas y Productos de trabajo. Estas vistas del modelo de procesos son obtenidas a partir de un modelo de proceso desarrollado en SPEM 2.0 y la obtención de una serie de métricas que permiten generar un diseño por medio de nodos y aristas con las cuales reflejan las relaciones entre cada una de las entidades. Cada nodo está generado con una serie de características determinadas por las métricas obtenidas, con las que busca facilitar el análisis visual. Este enfoque ha permitido la detección de una serie de patrones de error que son detectados frecuentemente a la hora de diseñar un modelo de proceso. Todo esto centrado únicamente en la capacidad visual del evaluador [16]. A diferencia de algunos enfoques, éste en particular cuenta con una herramienta que permite automatizar en cierto grado a la hora de efectuar el análisis del modelo de procesos. Esta herramienta es AVISPA (Analysis and Visualization for Software Process Assessment), la cual nos permite visualizar de manera más efectiva los Blueprints que son necesarios para realizar el análisis, también nos facilita el desarrollo del análisis sin la necesidad de estar implementado en un proceso real lo cual resulta un filtro que permite a las organizaciones ahorrar en costos frente a la implementación de un proceso. Esta herramienta está centrada en resaltar ya sea por la forma, el color, o simples enlaces, posibles fallas o anomalías que pueden encontrarse en una definición de un modelo [15]. Es de resaltar que AVISPA es una herramienta que efectúa análisis de modelos de procesos SPEM 2.0, sin embargo, la herramienta posee limitaciones en cuanto al análisis de modelos de procesos que incorporen variabilidad en su definición.

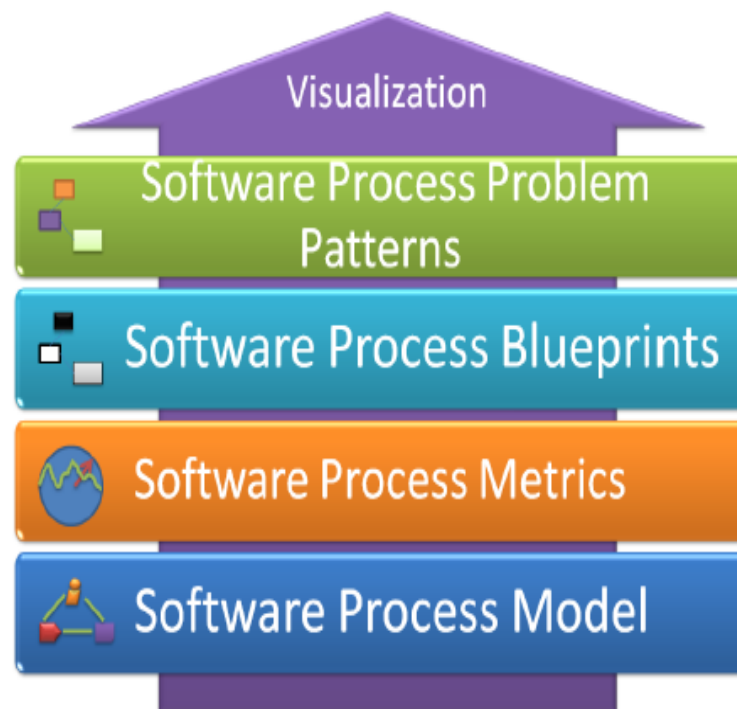


Fig. 3 Proceso Análisis visual AVISPA

Existen diferentes técnicas o enfoques que facilitan la verificación y validación de los modelos de procesos, desarrollados para organizaciones de diferentes contextos o tamaños. Aunque definen diferentes métodos para analizar modelos de procesos, ninguno de ellos resulta excluyente de los demás. Los métodos se complementan entre sí, lo que permite mejorar la calidad de los procesos y productos que generan en las organizaciones.

| Método o técnica para V&V de modelos de procesos | Esfuerzo Aplicación practica | Complejidad de interpretación resultados | Modelos de Proceso con Variabilidad | Basado en SPEM |
|--|------------------------------|--|-------------------------------------|----------------|
| Métricas [42]                                    | Alto                         | Media                                    | No                                  | SPEM 1.1       |
| Pruebas Piloto [36]                              | Alto                         | Alta                                     | No                                  | Independiente  |
| Simulación [44, 45]                              | Media                        | Baja                                     | Si                                  | No             |
| Chequeo Formal [46]                              | Muy Alto                     | Alta                                     | Si                                  | No             |
| Análisis Visual [15]                             | Bajo                         | Baja                                     | No                                  | SPEM 2.0       |
| vAVISPA  | Bajo                         | Baja                                     | SI                                  | SPEM 2.0       |

**Tabla 2. Análisis Métodos de validación y Verificación**

Para definir productos y procesos adaptables, son utilizadas técnicas de análisis de variabilidad. Estas técnicas buscan extraer qué es lo común y lo variable dentro de un proceso, normalmente para definir una línea de procesos similar a como es efectuado en el análisis de líneas de productos [14, 48]. La propuesta de ésta tesis vAVISPA, difiere de dichas técnicas en que el modelo de proceso a analizar ya se encuentra formalizado en SPEM 2.0 e incorporada alguno de los diversos tipos de variabilidad de SPEM 2.0. Es precisamente donde toma valor el enfoque al ayudar a determinar la calidad de éste tipo de modelos de proceso. Por lo tanto el análisis de ésta tesis es una técnica de verificación y validación en lugar de ser una técnica de extracción de conocimiento o formulación de modelos de procesos.

### 3. ANÁLISIS VISUAL DE LA VARIABILIDAD EN MODELOS DE PROCESO SPEM 2.0

#### 3.1. Introducción al modelo de Visualización

El modelo arquitectónico de las diferentes abstracciones que aplican sobre el modelo de procesos para el análisis visual por parte de vAVISPA está dividido en 5 niveles como se observa en la Fig. 4, conservando como base la arquitectura utilizada en AVISPA presentada en la Fig.3 e incorpora un módulo encargado de la resolución de variabilidad presente en él modelo. Como primer nivel está la especificación del modelo de proceso obtenido de EPFC basada en SPEM 2.0 tomando como base la definición del modelo de proceso se realiza la importación a vAVISPA recuperando las interacciones entre los roles, tareas y productos del trabajo del modelo de proceso, con la información recuperada del importe el segundo nivel de abstracción de vAVISPA toma las reglas definidas en SPEM 2.0 para los cinco tipos de variabilidad (no variable, contribución, reemplazo, extensión y extensión-reemplazo) y aplica el algoritmo de resolución para cada uno de los casos sobre las diferentes interacciones recuperadas en el proceso de importe, resueltas las relaciones de variabilidad el tercer nivel del modelo aplica una serie de métricas que determinan las características relacionadas al proceso para poderlo describir gráficamente. Con la obtención del modelo a partir de una medida cuantificable el cuarta nivel del modelo procede al desarrollo de los Blueprints los cuales consisten en una serie de vistas polimétrica del modelo y con las cuales permiten iniciar la última parte del análisis del modelo visual, identificando y resaltando de manera visual una serie de patrones de error que pueden representar pérdida de especificación, definición o un error conceptual en la definición del modelo de proceso.



Fig. 4 Proceso de análisis visual vAVISPA [15]

### 3.2. Modelo de proceso software

vAVISPA usa como entrada modelos SPEM2.0 de manera similar a AVISPA, los modelos de proceso generados por la herramienta Eclipse Process Framework Composer – EPFC aprovechando su cualidad de generar la documentación necesaria en formato web para nuestro caso XML. Sin embargo, vAVISPA extiende la capacidad de análisis de AVISPA recuperando del modelo las relaciones de variabilidad entre los elementos que lo componen. La Fig. 5, presenta una estructura de descomposición de una librería de método SPEM 2.0 deducida de la estructuración de un modelo de proceso en SPEM2.0 definido con EPFC. En esta estructura de descomposición se observa que la forma de modularizar un proceso SPEM 2.0 es través de paquetes, bien sea de contenido de método o de proceso.

La taxonomía de paquetes de SPEM 2.0 permite distinguir claramente entre lo que corresponde a la definición de elementos base recopilados en el paquete contenido del método (package method content ) y la definición del proceso que corresponde al paquete de proceso (package process) la cual básicamente es la utilización de algunos de los componentes definidos en el contenido del método.

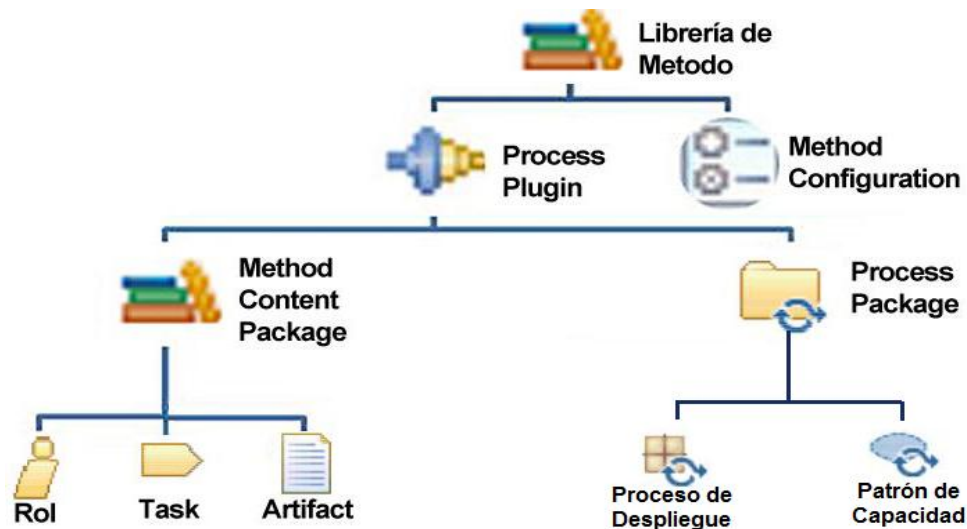


Fig. 5 Estructura de Composición de los Elementos SPEM 2.0

Como último nivel en la representación de elementos del contenido del método en SPEM 2.0 encontramos los elementos que basan la idea central para la representación de un proceso según el estándar que son: roles, tareas y productos de trabajo. Donde las tareas representan el esfuerzo a ser realizado, los roles representan quien realiza el esfuerzo y por último los productos de trabajo que representan la materia prima para la realización de una tarea y a su vez el elemento generado [20]. Básicamente SPEM 2.0 propone para la representación del modelo por medio de una relación de ejecución de un esfuerzo que



involucra y es iniciado por un responsable en la cual se consumirá una serie de recursos para la generación de nuevos productos de consumo.

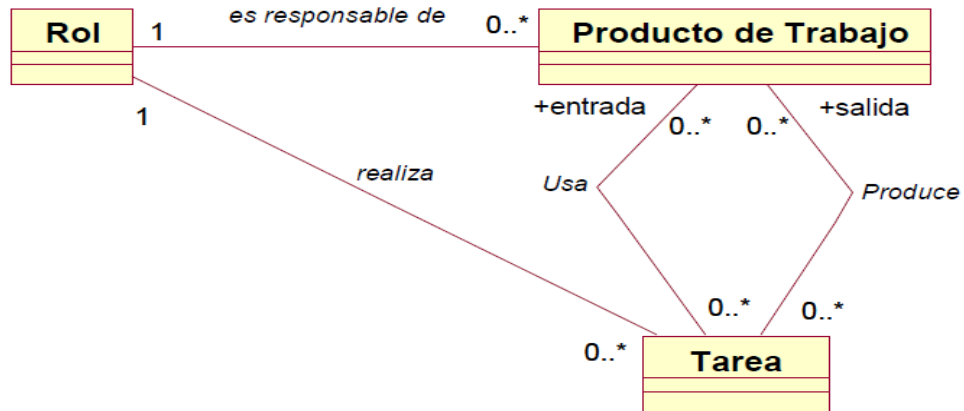


Fig. 6 Representación del modelo según SPEM 2.0 [20]

Los componentes bases para la definición del proceso son la fuente de entrada utilizada por vAVISPA para desarrollar su análisis extraídos directamente desde la definición de modelo de proceso generado por EPFC. vAVISPA recupera las propiedades de los componentes y su relaciones con otros elementos lo que define al modelo de proceso.

### 3.3. Resolución de la Variabilidad

La estructura por medio de paquetes del modelo de proceso permite que sean reutilizados tantos elementos definidos dentro del contenido del método así como del paquete de proceso. Por lo que SPEM 2.0 permite dos tipos de reutilización para la definición de sus procesos. La primera forma de reutilización consiste en crear un nuevo plugin referenciando directamente a un plugin ya definido en el paquete de contenido del método. Mientras que en el segunda mecanismo de reutilización es modificado de forma directa el contenido de un plugin a partir de un plugin diferente, este tipo de reutilización es generada cuando son utilizados los mecanismos propuestos en la definición del meta-modelo SPEM 2.0.

#### 3.3.1. Reutilización del contenido del método

En el primer caso de reutilización está basado en un enfoque más centrado en el aprovechamiento del conocimiento obtenido por la organización ya sea de experiencias pasadas o conocimientos de prototipos o estándar de procesos. Se utiliza tomando en cuenta la definición de paquetes propuesta por SPEM 2.0 con el paquete contenido del método el cual establece un diccionario de elementos primarios o elementos básicos que pueden ser combinados o reutilizados por uno o varios procesos. La definición de una única fuente de información para todos los procesos permite el extender las capacidades

y experiencias de la organización a diferentes contextos según sea la necesidad del proceso.

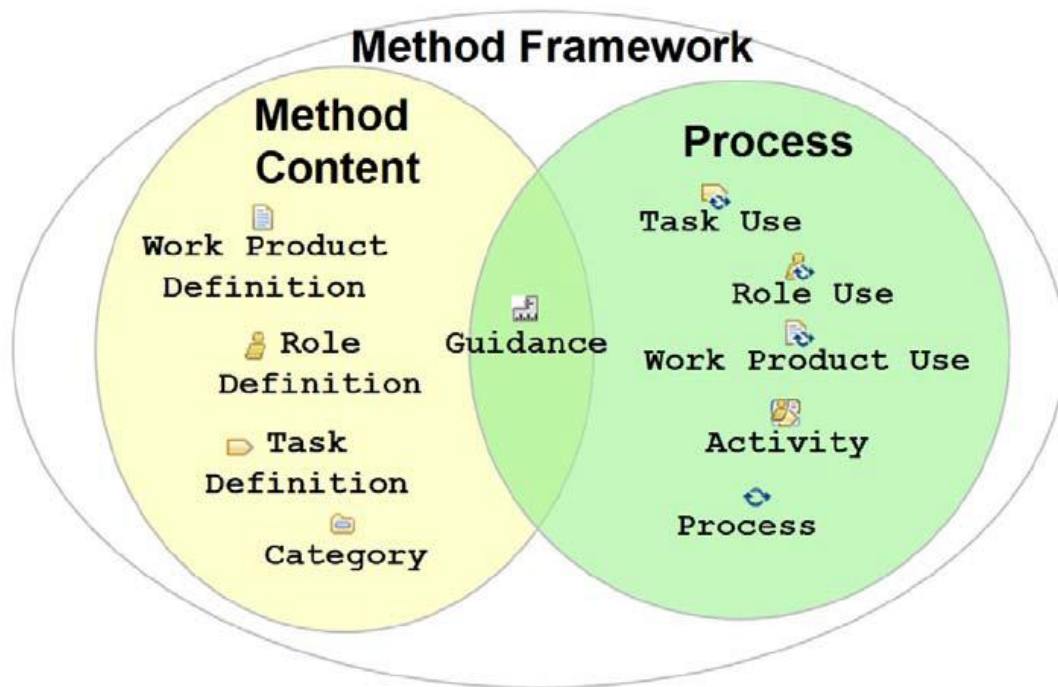


Fig. 7 Framework de trabajo especificado por SPEM 2.0 [8]

Como resultado de este tipo de reutilización la documentación generada del proceso retiene la definición tanto de elementos propios del proceso como de elementos que hacen parte del contenido del método y no son requeridos por el proceso, causando una sobrecarga de información no necesaria para el análisis del proceso. Al trabajar el modelo de proceso SPEM 2.0 utilizando la herramienta de edición EPF composer se puede visualizar al exportar el modelo en XML la categorización de elementos con la definición de un descriptor para todos los elementos definidos en el contenido del método (uma:Task, uma:Artifact, uma:Role,etc.) y un descriptor específico para los elementos que conforman el proceso (uma:TaskDescription, uma:ArtifactDescription, uma:RoleDescription, etc).

```

<ContentElement xsi:type="uma:Task" name="dsdm_run_tests" briefDescription=""
  <Presentation xsi:type="uma:TaskDescription" name="dsdm_run_tests,_Ji8x4Fqx
    <MainDescription></MainDescription>
    <KeyConsiderations></KeyConsiderations>
    <Alternatives></Alternatives>
    <Purpose></Purpose>
  </Presentation>
  <AdditionallyPerformedBy>_avOA0FkVEdul8L-IGeA7TA</AdditionallyPerformedBy>
  <AdditionallyPerformedBy>_ctbZ8FkVEdul8L-IGeA7TA</AdditionallyPerformedBy>
</ContentElement>
<ContentElement xsi:type="uma:Task" name="dsdm_outline_the_architecture" brie
  <AdditionallyPerformedBy>_ctbZ8FkVEdul8L-IGeA7TA</AdditionallyPerformedBy>
  <AdditionallyPerformedBy>_hQJ-QFkVEdul8L-IGeA7TA</AdditionallyPerformedBy>
</ContentElement>

```

Fig. 8 XML openUP generado por EPF composer

### 3.3.2. Adaptación del modelo de proceso por medio de los mecanismos especificados en SPEM 2.0

En ciertos casos es requerido solo reutilizar el contenido de cierto plugin con algún tipo de adaptación, por lo que SPEM 2.0 dispone de mecanismo de variabilidad para la modificación de elementos del método, lo que permite reestructurar las propiedades de un plugin a partir de otro. Los mecanismos de variabilidad permiten precisar diferencias con el elemento original. Este tipo de mecanismo permite modificar propiedades de los elementos definidos como son sus atributos y asociaciones con otros elementos. Como mencionamos antes SPEM 2.0 contempla cinco mecanismos de variabilidad en su meta-modelo que son: na – no se presenta algún tipo de variabilidad, Contribute – presenta contribución de un elemento o múltiples elementos hacia otro, Replace – presenta el reemplazo de un elemento por otro, Extend – un elemento extendiendo de otro y Extend-Replace – que combina tanto los efectos de la extensión como los del reemplazo.

#### 3.3.2.1. Contribución

Por medio de este constructo un elemento contribuye a otro sin modificar directamente sus propiedades. Está definida la contribución en SPEM 2.0 de la siguiente forma: Un elemento E que contribuye a un elemento base EB añade sus valores de atributo y sus asociaciones al elemento E (ver Fig. 9). Al publicar el modelo de proceso SPEM 2.0 por ejemplo en versión web el elemento base EB aparece con las adiciones aportadas, mientras que el elemento E permanece oculto. Para el desarrollo de vAVISPA es tomada en cuenta solo las asociaciones de cada elemento ya que consiste en un enfoque visual. Para ello tomamos como base las siguientes normas o reglas establecidas por SPEM 2.0:

- La asociación de entrada o salida con cardinalidad “1” definidas en E serán ignoradas si se encuentran definidas en EB, en caso contrario serán añadidas.

- Las asociaciones de entrada o salida con cardinalidad “muchos” definidas en E serán añadidas al elemento EB.
- Un elemento base puede recibir múltiples contribuciones.

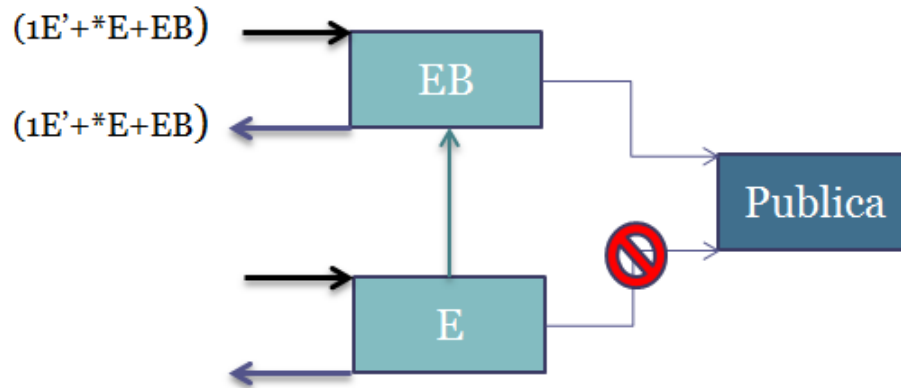


Fig. 9 Esquema reglas contribución

### 3.3.2.2. Reemplazo

Por medio de este constructo un elemento E puede reemplazar los atributos y asociaciones de salidas de un elemento EB, sin modificar directamente las propiedades del elemento EB (ver Fig. 10). Como resultado de la sustitución en la documentación generada el elemento reemplazante (E) aparecerá, mientras que el elemento reemplazado (EB) no es visualizado. Las reglas que establecidas para este tipo de variabilidad por SPEM 2.0 son:

- Las asociaciones de salida del elemento EB son reemplazados por las instancia de salida del elemento E
- Las asociaciones de entrada con cardinalidad “muchos” del elemento reemplazado (EB) son preservadas y ampliadas con las instancias de las asociaciones de entrada del elemento reemplazante (E).
- Las asociaciones de entrada con cardinalidad “1” del elemento reemplazado (EB) son reemplazadas con las asociaciones de entrada del elemento reemplazante (E). Si el elemento reemplazante no define asociaciones de entrada se conservaran las asociaciones del elemento reemplazado (EB).
- Un elemento base (EB) solo puede ser reemplazado por otro elemento en una única configuración, si un elemento base posee más de un reemplazo ninguno de los reemplazos se realizara.

- La contribución precede al reemplazo. Por tanto primero se evalúa la contribución para proceder a evaluar el reemplazo.

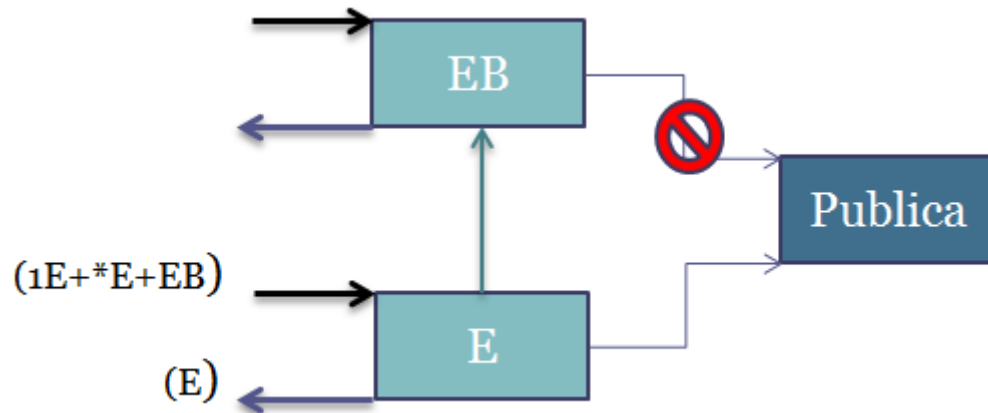


Fig. 10 Esquema reglas reemplazo

### 3.3.2.3. Extensión

En este caso el elemento E hereda las propiedades del elemento base EB ampliando los atributos del elemento E, y dejando completamente intacto al elemento base EB (ver Fig. 11). El elemento E incluirá las propiedades heredadas del elemento base y sus propias propiedades. Al final tanto el elemento base EB y el elemento E aparecerán en la vista. Las reglas establecidas para la extensión en SPEM 2.0 son:

- Las instancias de asociación de salida con cardinalidad “muchos” del elemento base (EB) serán añadidas al elemento E, si este no define sus propias asociaciones.
- Las asociaciones de salida con cardinalidad “1” del elemento base (EB) serán heredadas por elemento E si este no define sus propias asociaciones.
- Las asociaciones de entrada del elemento base (EB) son ignoradas en la extensión.
- La contribución precede a la extensión, la contribución es evaluada primero y el elemento que extiende hereda después del elemento base (incluyendo las contribuciones).
- Reemplazo precede a la extensión, Si el elemento base es reemplazado el elemento extenderá (heredará) del elemento reemplazante.

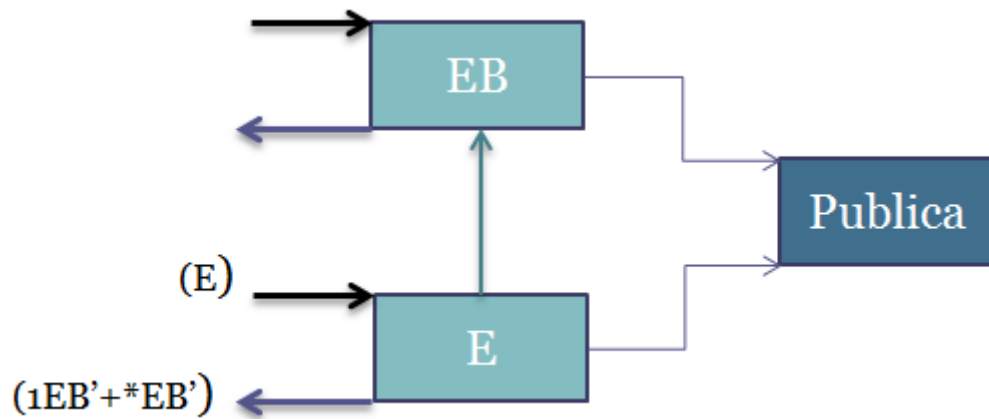


Fig. 11 Esquema reglas extensión

#### 3.3.2.4. Extensión-Reemplazo

Para este tipo de variabilidad son combinados los efectos de la extensión y el reemplazo. Por lo cual las propiedades definidas en el elemento E son extendidas y reemplazadas en las propiedades del elemento base EB (ver Fig.12). Para aplicar este tipo de variabilidad debe tenerse en cuenta que se debe resolver primero los efectos de la extensión y luego aplicar el reemplazo. El resultado final será que el elemento E aparecerá en la vista final mientras que el elemento EB estará oculto. Son establecidas las siguientes normas para aplicar esta tipo de variabilidad:

- Si el elemento que extiende y reemplaza define asociaciones de salida, estas reemplazarán las asociaciones de salida del elemento base. Si el elemento que extiende y reemplaza no define ninguna nueva asociación, el elemento resultante conservará las asociaciones del elemento base.
- Las asociaciones de entrada del elemento base son adheridas al elemento de reemplazo
- Un elemento base con relación de reemplazo o relación de extensión-reemplazo solo puede ser reemplazado o extendido y reemplazado para una configuración
- La relación de extensión-reemplazo es transitivo y evaluado de arriba hacia abajo relativo a la dirección de la sustitución. Si el elemento de sustitución también es reemplazado, el elemento de sustitución final prevalece.
- La relación de contribución se resuelve antes que la de reemplazo y que de extensión-reemplazo. La relación de extensión es resuelta al final. La relación de variabilidad siempre debe resolverse de arriba hacia abajo desde la base al elemento variable. En el mismo nivel contribución es resuelta primero, reemplazo y extensión-reemplazo son resueltas después.

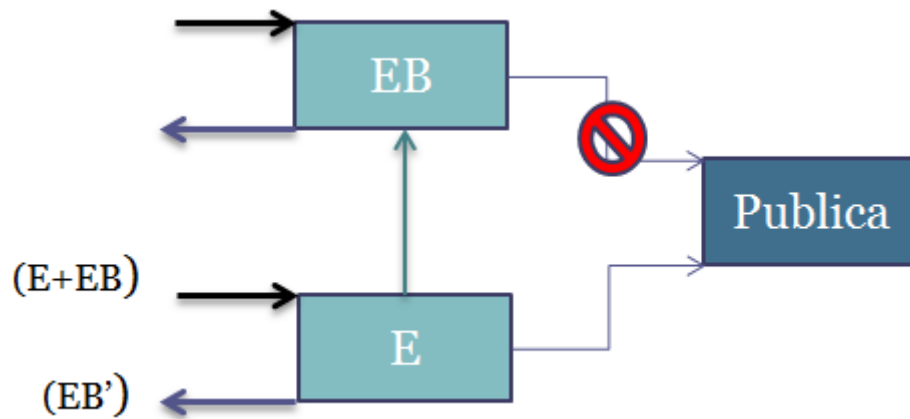


Fig. 12 Esquema reglas Extensión-Reemplazo

La propuesta vAVISPA es planteada en base al segundo tipo de adaptación, para lo cual toma en cuenta las reglas definidas por cada uno de los mecanismos de variabilidad realiza una extensión del análisis visual presentado por AVISPA que permita validar de forma más efectiva modelos de procesos que contengan variabilidad en su especificación con el fin de concentrar el análisis en problemas de perdida conceptual dentro del modelo.

### 3.4. Métricas

La propuesta vAVISPA reutiliza las métricas definidas por el enfoque de análisis visual AVISPA con el fin de determinar fallas en la definición del proceso dado que busca analizar el proceso utilizando un enfoque visual tal como es realizado con AVISPA pero incorporando el factor variabilidad en la definición del proceso. Lo cual no requiere de la implementación de nuevas métricas si no del análisis y evaluación del impacto que tiene la variabilidad en dichos modelos. Las métricas adoptadas de AVISPA y utilizadas por vAVISPA corresponden a aquellas que son definidas para la representación de cada uno de los componentes (Roles, tareas y productos de trabajo) de forma visual [49].

Para el caso del Blueprint de tareas es especificada su estructura a partir de los productos de trabajo que son consumidos y producidos por una tarea, mientras que las aristas representan la relación de dependencia entre una tarea y otra ver Fig. 13. La configuración del Blueprint de roles basa la descripción de los nodos a partir del número de tareas donde este involucrado el rol y los producto de trabajo en que participa, en el caso de las aristas describen una colaboración entre distintos roles ver Fig. 14. En el caso del Blueprint de productos de trabajo los nodos describen el número de tareas donde el nodo es consumido o producido, mientras que las aristas representan un tipo de dependencia entre un producto de trabajo y otro ver Fig. 15.

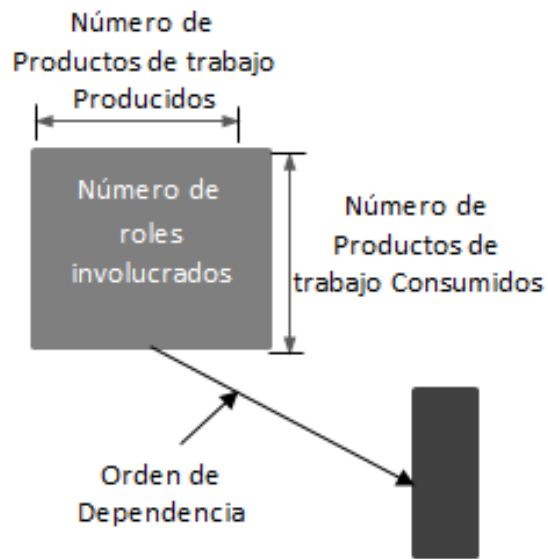


Fig. 13 Elemento del proceso: Tarea (Task)

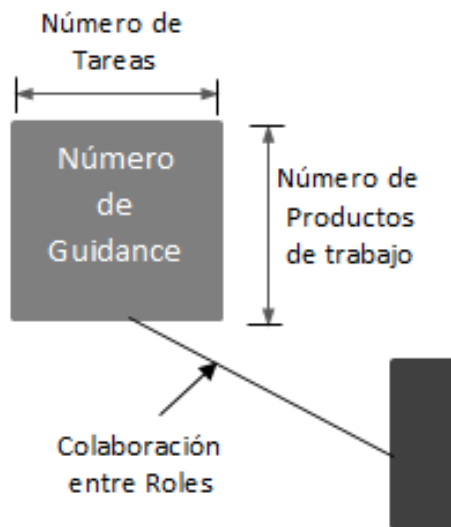


Fig. 14 Elemento del proceso: Rol



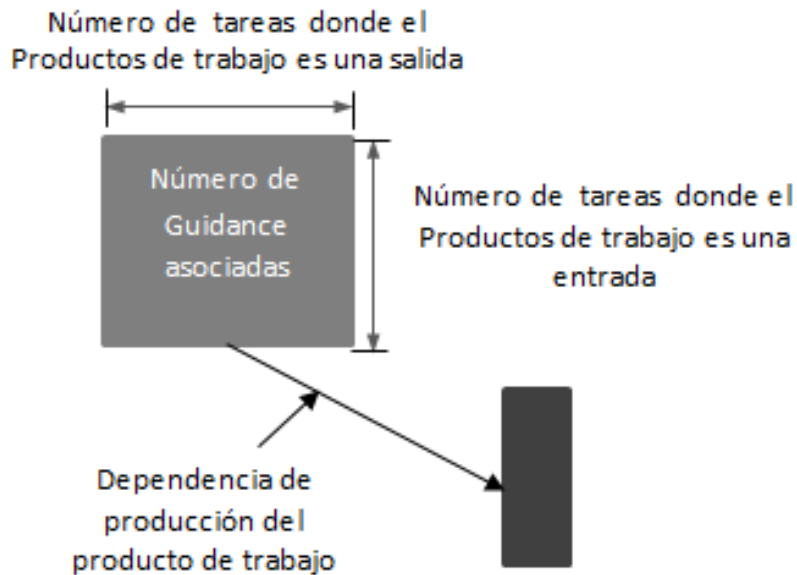


Fig. 15 Elemento del proceso: Producto del trabajo (Work Product)

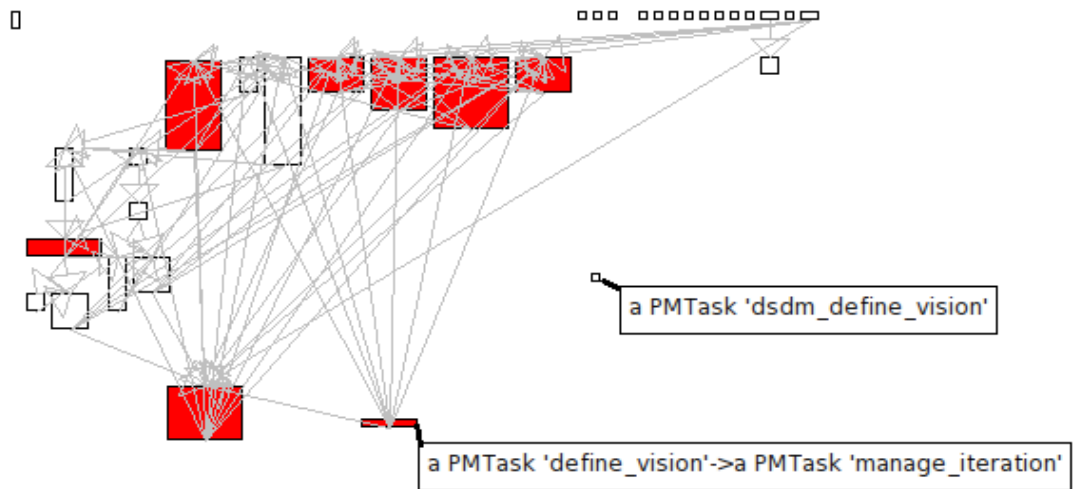
### 3.5. Blueprints de modelos de procesos

Los Blueprints corresponden al mecanismo de evaluación introducido en la propuesta AVISPA con el fin de detectar anomalías en el modelo de proceso utilizando un enfoque visual de validación el cuál busca resaltar de forma más intuitiva posibles errores en la definición del proceso sea por perdida conceptual o contextual de la información. Cada uno de los Blueprints generados toma como fuente de información una especificación del modelo proceso genera por EPF composer y procesa las propiedades y asociaciones de cada elemento definido para realizar un representación gráfica que permita describir al elemento.

La propuesta vAVISPA está centrada en depurar los Blueprints ya trabajados en AVISPA de manera que reflejen de forma más exacta las propiedades y asociaciones de cada uno de los elementos teniendo en cuenta si incorporan o no algún tipo de mecanismo de variabilidad de los dispuestos por SPEM 2.0. Utilizando las reglas establecidas para cada uno de los casos de variabilidad que puede presentar un elemento del modelo de proceso, realizar una actualización en el cálculo de las métricas que determinan la estructura visual de cada elemento del proceso así como de sus asociaciones. Esta reestructuración en la definición de las propiedades de los elementos del proceso permite identificar posibles impactos negativos en los elementos que cuentan con algún tipo de variabilidad en su definición como por ejemplo: una tarea que en la definición normal podía ser catalogada de complejidad media, puede convertirse en una actividad de carácter critico por cuenta de contribuciones realizadas por diversas tareas, lo que resultaría por cuenta de un mal uso o mala identificación de generación de un punto de variación dentro del modelo de proceso. Como ha sido recalcado a lo largo de este

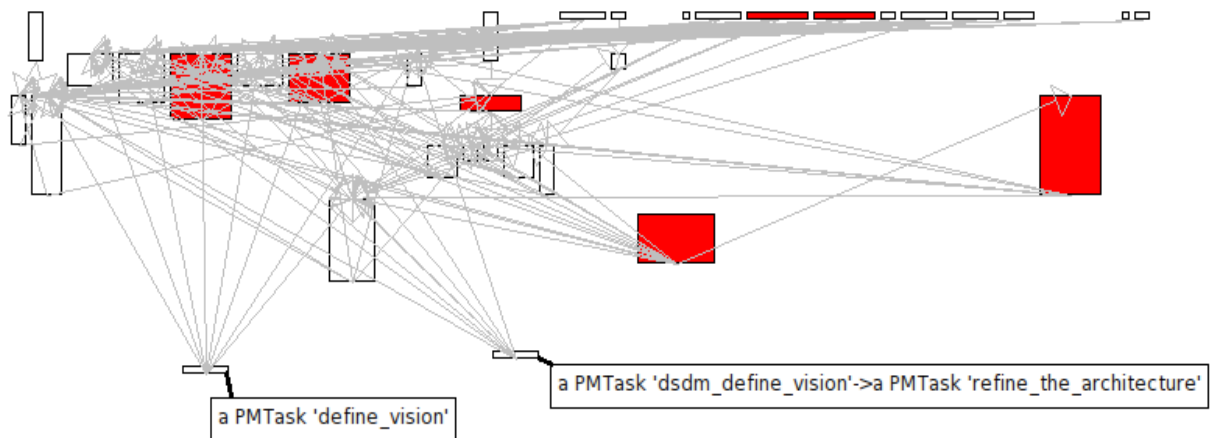
estudio la generación de puntos variables dentro del proceso genera un alto riesgo para el modelo en sí mismo, ya que pueden generar sobrecargas a tareas que antes solían ser sencillas o pérdidas de información del contexto del modelo de proceso.

Que tan importante resulta para un modelo, el impacto debido a la incorporación de variabilidad dependerá del contexto de la situación, pero debe tenerse en cuenta que un modelo consiste en una colección de elementos entrelazados entre sí para llevar a cabo una finalidad por lo que la variación de un elemento puede afectar positiva o negativamente a la totalidad del proceso. Por ejemplo utilizando la especificación de openUP (fuente abierta) analizamos el patrón propuesto por AVISPA de detección de tareas con múltiple propósito, el Blueprint resultante (ver Fig. 16), permite la identificación de tareas con múltiple propósito, como por ejemplo la tarea “define\_vision” es una tarea que representa múltiples propósitos, mientras que la tarea “dsdm\_define\_vision” se encuentra aislada del resto del proceso y representa una tarea sin ningún tipo de propósito para el modelo.



**Fig. 16 Blueprint patrón para localización de tareas con múltiple propósito AVISPA**

Al realizar el mismo proceso de análisis teniendo en cuenta la variabilidad del proceso propuesto en vAVISPA, obtenemos un Blueprint diferente al resultante en AVISPA (ver Fig. 17), el comportamiento del blueprint varia debido a la resolución de variabilidad, se detecta una relación de variabilidad (Extensión-Reemplazo) que conecta la actividad “dsdem\_define\_vision” con el resto del modelo y le brinda más granularidad a la tarea “define\_vision”.



**Fig. 17 Blueprint patrón para localización de tareas con múltiple propósito vAVISPA**

La actualización de los Blueprints que son afectados por la variabilidad del modelo del proceso corresponde al primer aporte en busca de fortalecer la efectividad del enfoque visual para el análisis del proceso. Adicional se generan dos nuevos Blueprints para mejorar la efectividad del seguimiento, control y comprensión de la variabilidad en el modelo de proceso que son:

- Blueprint Variabilidad del modelo de proceso
- Blueprint Method Content vs Process

### 3.5.1. Blueprint Variabilidad del modelo de proceso

El desarrollo del análisis del proceso representa una tarea ardua que genera un alto nivel de desgaste para el evaluador dada su alta correlación entre el modelo de proceso y diversos factores como: el contexto organizacional, el diseñador del proceso, requerimientos organizacionales, intereses particulares, etc. Teniendo en cuenta todos estos factores que determinan lo particular de cada modelo, el hacer seguimiento de que factores generan cambios en el modelo representa un riesgo más para el desarrollo de la evaluación. Dado esta problemática vAVISPA incorpora un nuevo Blueprint diseñado con el propósito de hacer seguimiento a los elementos involucrados en algún tipo de variabilidad con la finalidad de realizar un seguimiento más efectivo de los factores que generan cambios al modelo del proceso e identificar de manera más efectiva los causales de dichos cambios en el proceso.

El nuevo Blueprint denominado “Task Variability Blueprint” contiene todos los elementos incorporados en el contenido del método que son importados desde el archivo generado por EPF composer, los cuales le es adicionado un identificador de color que especifica la relación de variabilidad entre un par de elementos, la configuración establecida puede ser observada en la Fig.18. Este Blueprint facilita realizar seguimiento de la influencia de la variabilidad influencia dentro del modelo de proceso de manera más práctica a tener que

realizar la revisión completa del xml, a la vez que permite determinar qué factores pueden afectar la estructura del modelo de manera positiva o negativa según sea el caso de variabilidad.

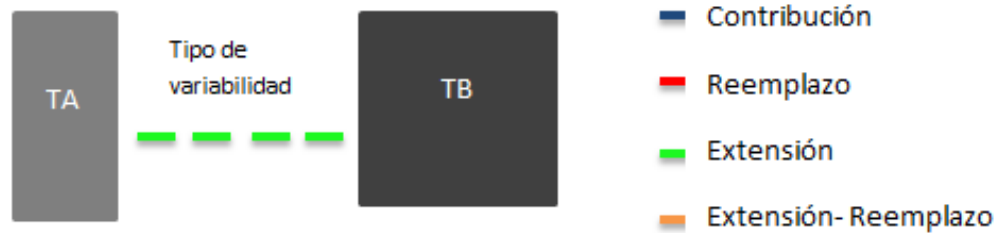


Fig. 18 Esquema de representación de variabilidad

La facilidad de identificación de relaciones de variabilidad la podemos observar por ejemplo en el Blueprint obtenido del modelo de proceso OpenUP que presenta múltiples casos de extensión y un caso de extensión-reemplazo como se observa en la Fig. 19.

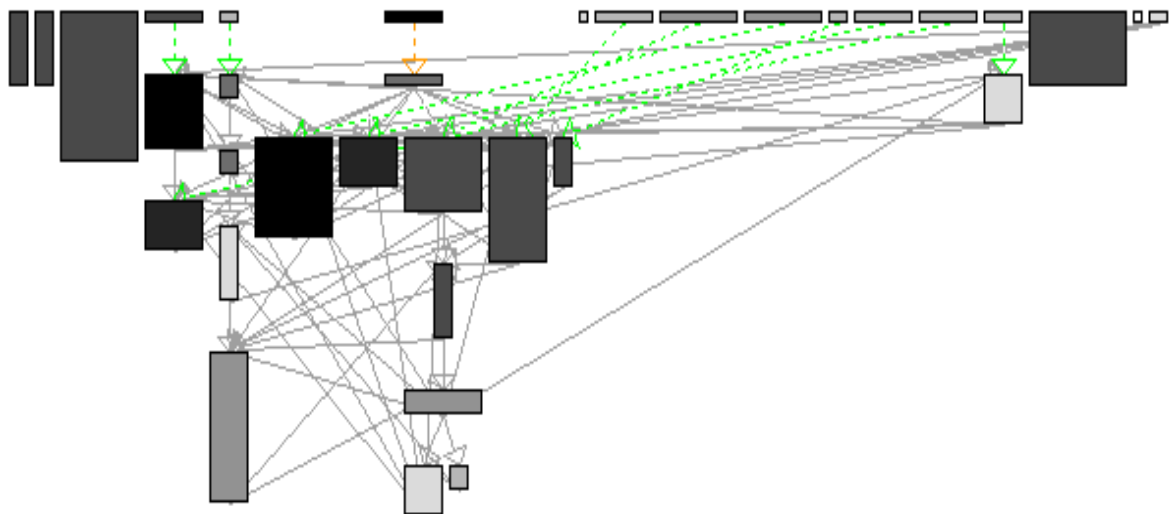


Fig. 19 Task variability Blueprint caso openUP

### 3.5.2. Blueprint Method Content vs Process

El conocimiento obtenido corresponde a uno de los activos más valiosos para una organización, hace parte del pilar fundamental para la generación de modelos de procesos de calidad. El contenido del método de SPEM 2.0 como fuente de los elementos base para la descripción del proceso tiene gran relevancia para la estructuración del proceso. Debido a la importancia que representa para el proceso el contenido del método la posibilidad de contar con un Blueprint que permita contrastar la definición del modelo de

proceso contra la base de conocimiento sobre la cual se ha generado tiene gran relevancia. Con el propósito de lograr este objetivo, un primer acercamiento a nivel de tareas es la implementación del Blueprint “Method Content vs Process”, un Blueprint diseñado para representar de forma visual que parte del contenido del método son utilizados para la definición del proceso, y a su vez permitirle al evaluador analizar que partes del contenido del método han sido descartado permitiendo verificar su impacto en el modelo de proceso final.

El Blueprint contiene la representación visual de todos los elementos que conforman el universo de conocimiento para el desarrollo del proceso resaltando aquellos elementos que son utilizados para la definición del proceso con un borde verde denominadas tareas en uso y aquellos elementos que por decisiones de diseñador del proceso han resultado descartados para la definición del proceso resaltados en un borde rojo como se observa en la Fig. 20.



**Fig. 20 Representación gráfica de las tareas utilizada por proceso**

Con la información de este Blueprint el análisis haciendo uso del enfoque visual de que parte del conocimiento a resultado realmente aprovechado en la definición del proceso así como que impacto surge al descartar ciertos elementos definidos en el contenido del método. Otro de los posibles usos que permite la información representada se basa en la propiedad adaptativa del modelo del proceso desarrollándolo según el contexto o condiciones del negocio de la organización. Como por ejemplo un modelo de proceso desarrollado para una adaptación no debería requerir igual porcentaje de elementos para su definición que un modelo de proceso diseñado para un desarrollo total, ya que la cantidad de esfuerzo requerido es significativamente inferior.

Un ejemplo de análisis obtenido por medio de esté Blueprint es el caso de OpenUP como se observa en el blueprint de la Fig. 21, donde existen una serie de tareas que no son usadas por parte del proceso final, algunas de estas tareas producen múltiples productos de trabajo a varias tareas que si hacen parte del proceso, generando la posibilidad de un análisis que determine el impacto que puede tener su omisión para el proceso en general.

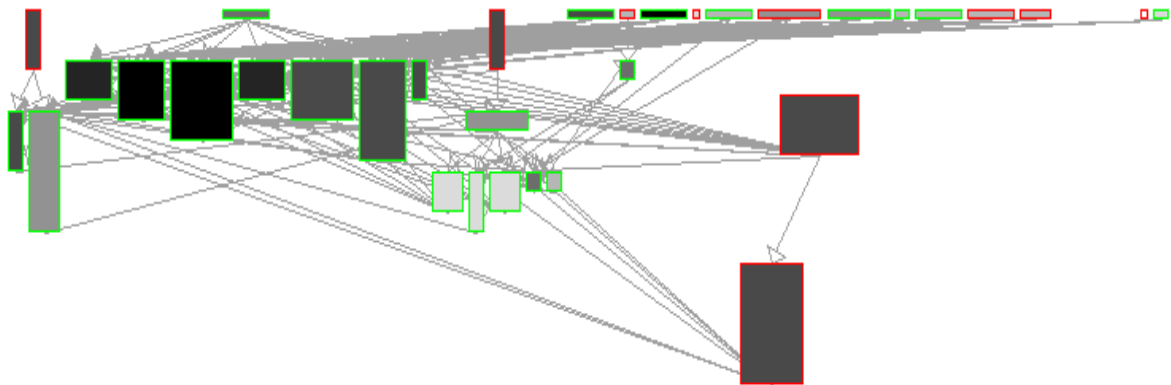


Fig. 21 Blueprint Method content vs Process del proceso OpenUP

### 3.6. Patrones

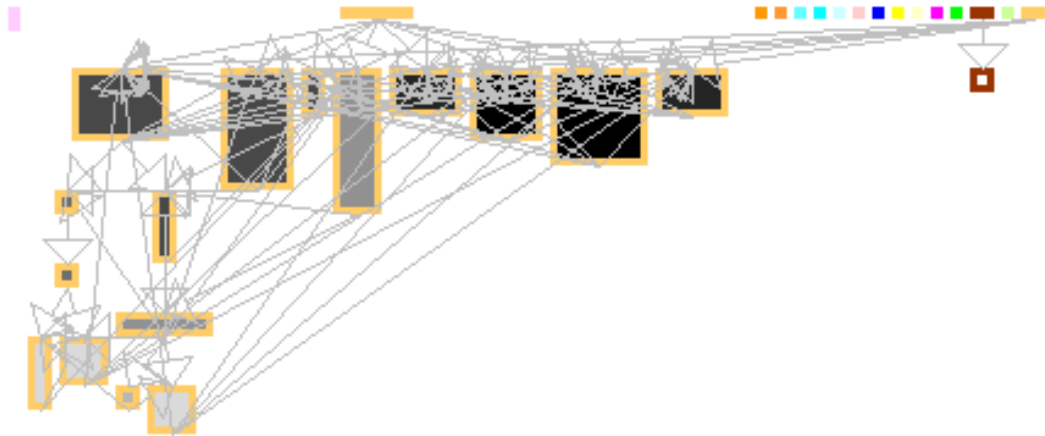
La resolución de variabilidad afecta en cierto grado a cada proceso, los Blueprints que se presentaban con AVISPA tienen algunas incongruencias entre el modelo y los Blueprints debido a los mecanismos de variabilidad de SPEM 2.0. La nueva generación de Blueprints obtenidos por parte de vAVISPA representan de manera más efectiva al modelo de proceso usando el enfoque visual, depurando posibles indicadores de error generados debido a la incorporación de mecanismos de variabilidad.

La propuesta de resolución de variabilidad no solo afecta a los Blueprints sino que también es reflejada en los patrones propuestos por AVISPA para la detección de posibles errores en la definición del modelo. Entre los patrones que son afectados debido a la resolución de la variabilidad tenemos:

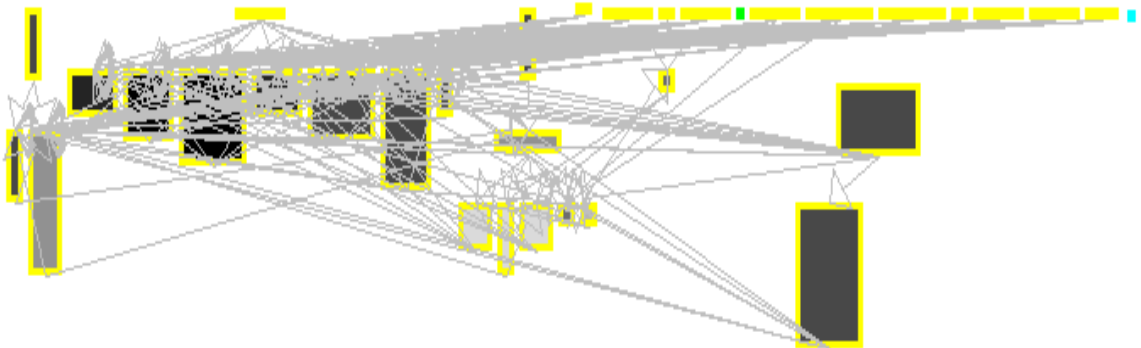
**Tareas con múltiple propósito:** Una tarea que en el Blueprint de tareas puede observarse demasiado ancha representa una tarea responsable de la producción de múltiples productos de trabajo, lo cual puede significar un cuello de botella para el proceso. Por lo que es resaltada de color rojo para llamar la atención del ingeniero del proceso donde la tarea puede presentar pérdida de especificación al no tener una meta clara y puede ser sub-dividida en tareas más específicas. Como se observó en la Fig. 16 y la Fig. 17 la incorporación de variabilidad reduce las responsabilidades de la tarea “define\_vision” depurando un posible error definido por AVISPA.

**Independencia de Sub proyectos:** La detección de proyectos independientes representa una pérdida en la especificación del modelo de proceso. Debido a que todas las tareas y productos de trabajo deberían interactuar entre sí con el fin de lograr la meta del proceso. AVISPA y vAVISPA cuentan con dos Blueprints que identifican sub proyectos a nivel de tareas (task) y productos de trabajo (work product) en los que es resaltado cada uno de los proyectos con un determinado color estableciendo la relación entre los elementos.

El impacto de la adecuación de resolución de variabilidad al proceso se observa al contrastar el Blueprint del patrón a nivel de tareas generado por AVISPA donde detectamos múltiples subprocesos (ver Fig. 22), comparado con los procesos identificados sobre el mismo patrón en el caso de vAVISPA (ver Fig. 23).



**Fig. 22 Patrón independencia sub proyectos - Task (AVISPA)**



**Fig. 23 Patrón independencia sub proyectos - Task (vAVISPA)**

Realizando un análisis superficial identificamos como a nivel de tareas la incorporación de resolución de variabilidad por vAVISPA depura falsos errores en el patrón de independencia de sub proyectos que eran detectados por AVISPA, los cuales podían representar un falso error debido a no tener en cuenta el aspecto de variabilidad e impacto en el modelo de proceso para el enfoque de validación visual.

En el caso de identificación de sub proyectos a nivel de productos de trabajo los resultados obtenidos por AVISPA detectan 12 posibles sub procesos como se observa en

la Fig. 24, mientras que el análisis sobre el mismo proceso realizado utilizando vAVISPA detecta solo 10 sub procesos que puede identificarse en la Fig. 25.

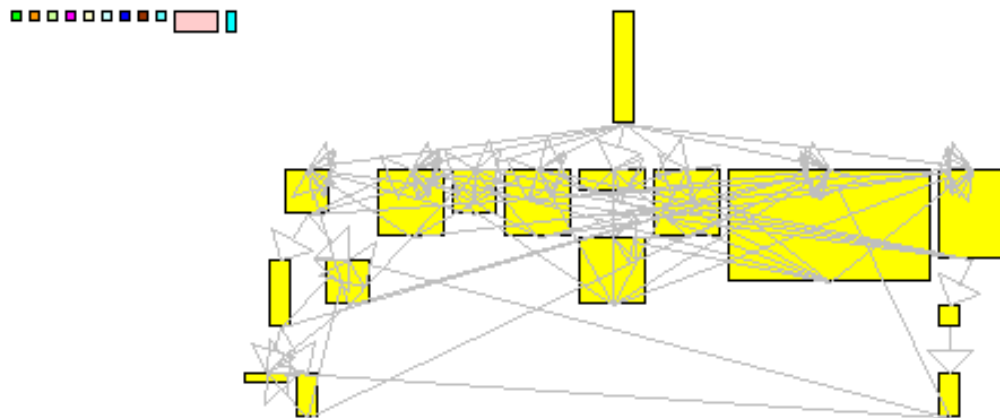


Fig. 24 Patrón independencia sub proyectos - Work Product (AVISPA)

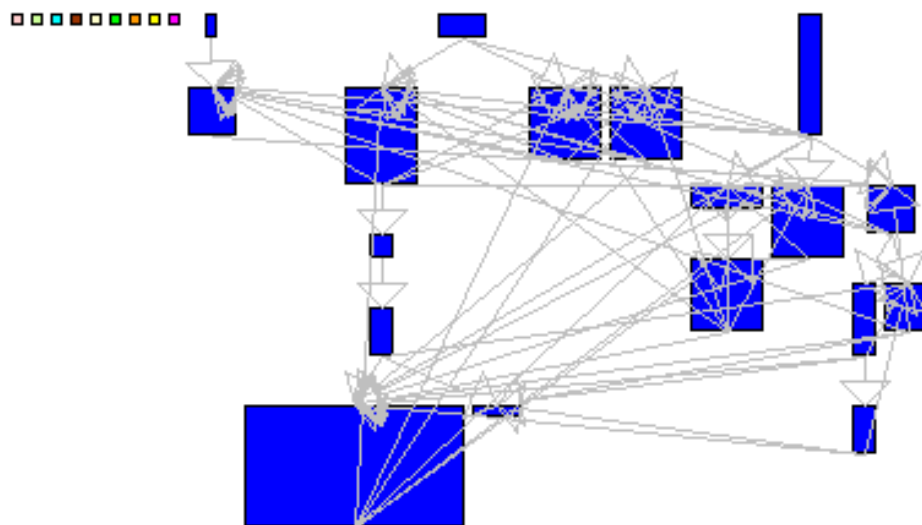


Fig. 25 Patrón independencia sub proyectos - Work Product (vAVISPA)

Con el Blueprint de independencia de sub proyectos a nivel de producto de trabajo resultaron identificados un mayor número de sub proyectos que a nivel de tareas, a la vez que se detectan dos productos de trabajo que identificados como sub proyectos por AVISPA pero con la resolución de variabilidad implementada por vAVISPA son incorporados al proceso principal.

### 3.7. Prototipo vAVISPA

En la fase exploración del estado del arte se identificaron pocos recursos en cuanto a variabilidad de procesos, entre las pocas herramientas detectadas para el análisis de



variabilidad encontramos AVISPA. La cual permite la importación de un proceso en formato XML especificados en SPEM 2.0 y desarrollados utilizando EPF composer para posteriormente procesarlo en un esquema visual que representa del modelo de proceso denominado Blueprints pero que carece de un manejo efectivo de la variabilidad presente en el modelo de proceso. Considerándolo como el candidato ideal para el desarrollo del análisis visual propuesto en este trabajo de grado. AVISPA, dado que es una herramienta con un entorno ágil y ligero que permite recuperar y procesar información del modelo de proceso para la construcción de Blueprints que permitan realizar el análisis y evaluación del modelo de proceso especificado en SPEM 2.0.

AVISPA es un proyecto de código abierto que está implementada sobre la plataforma de código abierto Moose, por lo que permite la realización de una extensión de las funcionalidades de análisis visual que le permita importar, procesar y analizar las variabilidades presentes en la definición del modelo de proceso. Como se observa el extender AVISPA requiere de tres configuraciones:

- Importar de la definición del modelo de proceso los elementos necesarios para poder realizar la identificación de la variabilidad en el modelo de proceso
- Procesar la información que obtenida del modelo de manera que se pueda realizar una representación visual para el desarrollo del enfoque.
- Generar o actualizar los Blueprints necesarios que permitan al ingeniero de procesos analizar de forma efectiva el modelo de proceso

El desarrollo de la primera brecha a superar para lograr la extensión del análisis visual presentado por AVISPA estará centrado en el obtener la información necesaria para describir una representación de la variabilidad en el modelo de proceso con la **importación** de la definición del mismo. El desarrollo de esta primera actividad requirió un mayor esfuerzo por parte del investigador, dado alguna serie de factores como: El poco conocimiento de conceptos alrededor de modelos de proceso, poco conocimiento de Smalltalk lenguaje sobre el que se desarrolló AVISPA, la escasa información encontrada del lenguaje y la lenta curva de aprendizaje terminaron en generar un gran reto para alcanzar la meta propuesta. Lo cual hizo necesario una mayor interacción entre el investigador y el director de proyecto quien tenía experiencia previa tanto en modelos de procesos, smalltalk y AVISPA al ser uno de los desarrolladores de la propuesta, también haciendo uso de las escasas fuentes de información sobre el lenguaje samlltalk que estén disponibles [50, 51].

Superado el primer hito para el desarrollo de vAVISPA, la fase de **procesamiento** puede efectuarse, para esta fase incorpora la información recuperada a partir del importe al modelo visual propuesto por AVISPA que permita expresar de forma clara el modelo de proceso con los factores de variabilidad encontrados. Planteando adaptar la nueva información obtenida al proceso ya establecido por AVISPA adicionando al proceso actual

un módulo especializado en la resolución de variabilidad de acuerdo a lo establecido por SPEM 2.0 para cada tipo de variabilidad disponible [8].

La última fase del proceso de extensión está enfocada en plantear el mecanismo más efectivo para presentar la información del modelo de proceso al diseñador del proceso para el **análisis** y verificación del modelo. Como estaba planteada en la anterior fase la reutilización del mecanismo propuesto por AVISPA para la representación del modelo de proceso representa la modificación de algunos de los Blueprints que presenta AVISPA y la generación de nuevos Blueprints que permitan que el proceso de validación propuesto por vAVISPA sea más efectivo en comparación al análisis actualmente logrado por AVISPA.

En esta sección enfatiza en el funcionamiento interno del prototipo vAVISPA, dando una breve descripción de algunos de los componentes que han sido desarrollados y otros que se extendieron sobre la implementación previa de AVISPA con el propósito de dar soporte al manejo de la variabilidad en el análisis visual.

### **3.7.1. Diseño e implementación de vAVISPA**

La propuesta vAVISPA es una extensión al enfoque visual de análisis planteado por Julio Ariel H.[49] y desarrollado en la herramienta AVISPA proponiendo la depuración del proceso de validación tomando en cuenta la resolución de los mecanismos de variabilidad definidos por SPEM 2.0. Debido a estos factores vAVISPA reutiliza la arquitectura y técnicas de visualización establecidas por AVISPA.

La propuesta de este trabajo de grado propone la incorporación de un módulo sobre la arquitectura establecida de AVISPA que permita resolver y representar el impacto que tiene la variabilidad en un modelo de proceso especificado en SPEM 2.0 y descrito en un formato XML con la ayuda de EPF composer. La resolución de la variabilidad es realizada a partir de las reglas definidas por SPEM 2.0 presentadas en la sección 3.3 para cada uno de los mecanismos de variabilidad que ofrece el meta-modelo. Resueltos los mecanismos de variabilidad que sean detectados en la definición del modelo de proceso son calculadas las métricas definidas por vAVISPA para la generación de Blueprints que permitan analizar el proceso. vAVISPA está desarrollada sobre de la plataforma Moose, la cual permite la importación, análisis de datos, modelado, medición, consulta del modelo de proceso utilizando el enfoque visual de los Blueprints que son implementados utilizando Mondrian un paquete que puede ser incorporado a Moose y que permite el diseño y conexión de componentes que a su vez permitirán describir el proceso de forma visual.

Unas de los objetivos que ha alcanzar con esta propuesta es el desarrollo de un prototipo de vAVISPA confiable para realizar el proceso de importe, procesamiento y análisis de

modelos de proceso que contengan mecanismos de variabilidad en su definición descritos en el estándar SPEM 2.0.

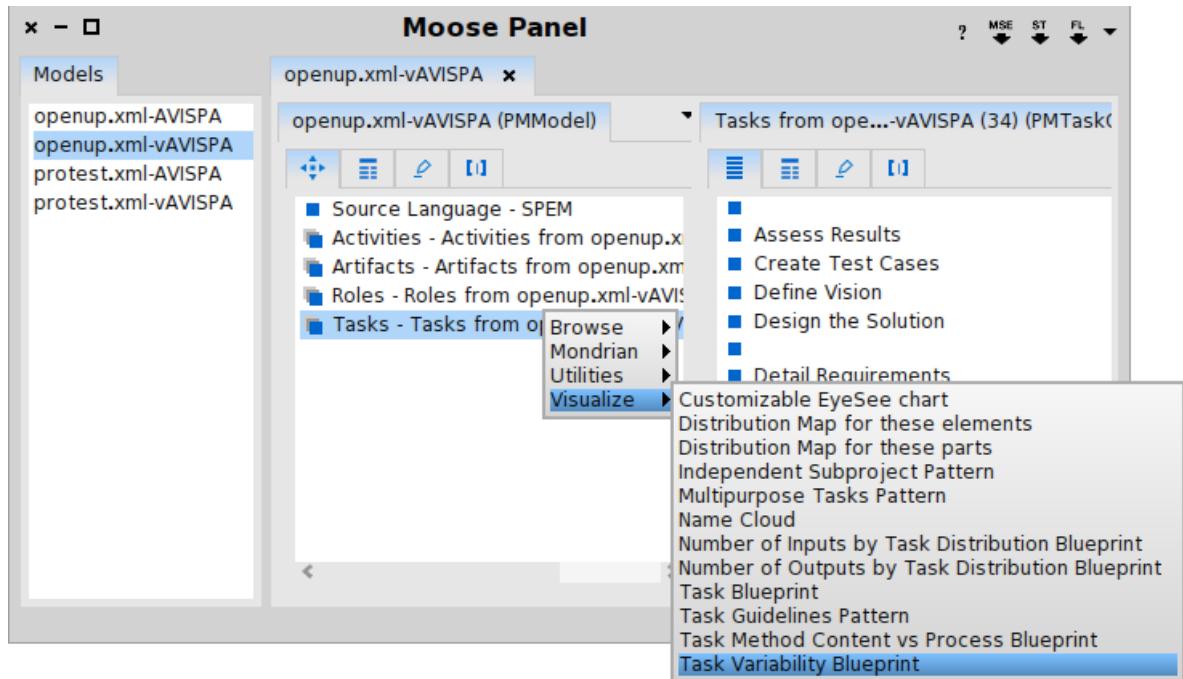
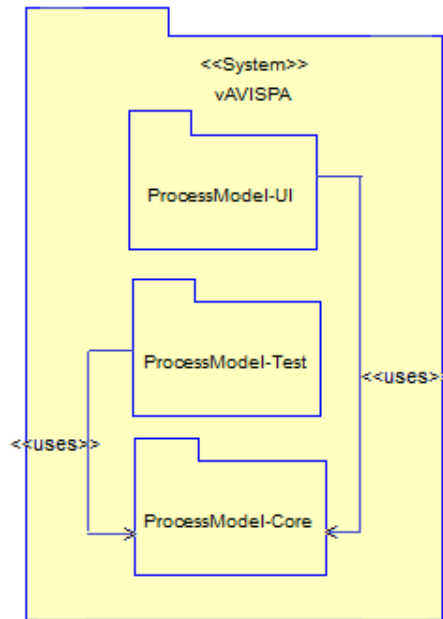


Fig. 26 Interfaz vAVISPA

El panel principal del prototipo vAVISPA (Fig.22), se asemeja al utilizado por AVISPA pero diferenciado al contener el mismo proceso con dos métodos de análisis diferentes el primero de ellos el que presenta AVISPA y el segundo corresponde al análisis generado por vAVISPA, el análisis del proceso está enfocado en tres principales componentes del proceso: tareas (Tasks), roles (Roles) y productos de trabajo (Artifacts) basado en la información presentada en la sección 3.2. Como esta descrito en la sección 3.4 son reutilizadas las métricas propuestas por AVISPA pero calculadas a partir del resultado obtenido al resolver los mecanismos de variabilidad, por lo que los Blueprints obtenidos varían de acuerdo al impacto de la variabilidad en el modelo de proceso. A nivel de tareas son incorporados los dos nuevos Blueprints planteados por la propuesta para el mejor entendimiento del desarrollo de la variabilidad.

A nivel de arquitectura, vAVISPA conserva la distribución de los módulos que tiene la herramienta AVISPA (ver Fig. 27), siguiendo un patrón en capas y de generalización, sobre la cual se desarrolló AVISPA. Cada módulo de vAVISPA es implementado particularizando elementos definidos en módulos abstractos de Moose. Desde una perspectiva de componentes y conectores.

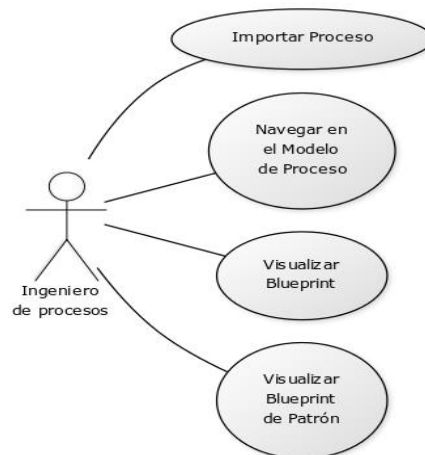


**Fig. 27 Estructura de paquetes vAVISPA**

ProcessModel-UI: Incluye comandos para la inicialización del proceso, es el encargado de iniciar el proceso de estructuración de cada uno de los modelos de procesos que se presentan en vAVISPA

ProcessModel-Core: Contiene la definición del conjunto de elementos encargado de representar el modelo de proceso y los metodos responsables de la resolución de variabilidad, calculo de metricas, la generación de Blueprints y patrones del proceso.

ProcessModel-Test: Contiene una serie de procesos de ejemplo que se utilizaron para la el proceso de validación de AVISPA.



**Fig. 28 Caso de Uso vAVISPA**

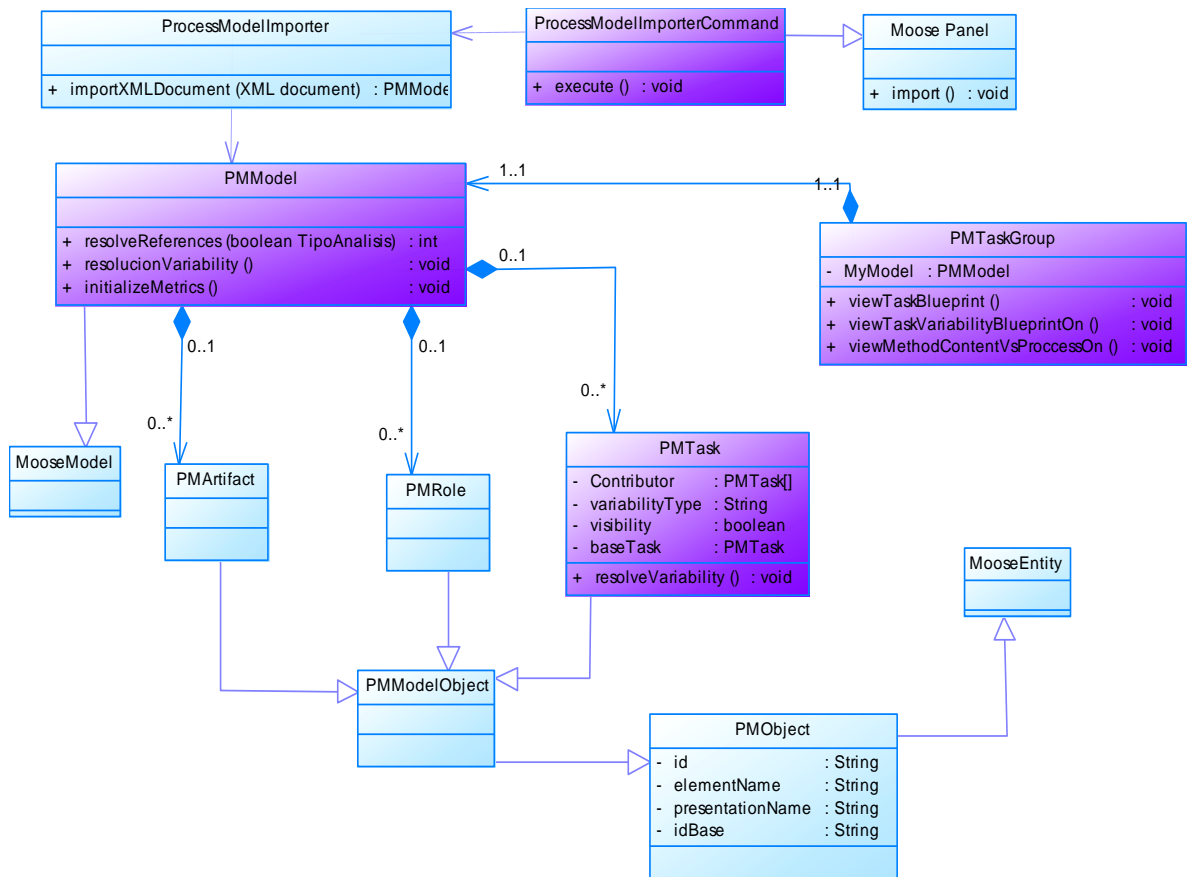
La Fig. 28 representa las funcionalidades que debe cumplir vAVISPA al ingeniero de proceso para que pueda realizar el proceso de análisis del modelo de proceso. Siempre y cuando el ingeniero de procesos proporcione a la herramienta un modelo de proceso en formato XML descrito utilizando SPEM 2.0 para su definición.

Ha continuación describe a alto nivel el proceso de importe del modelo de proceso que implementa vAVISPA. Inicia con la definición previa de un modelo de proceso descrito en SPEM 2.0 e implementado utilizando la herramienta EPF composer. Utilizando la implementación del caso de uso importar proceso, el ingeniero de proceso carga el modelo dos veces. La primera que será para el análisis de AVISPA y el segundo utilizado en el análisis generado por vAVISPA. El primer paso consiste en recuperar los elementos que definen al modelo de proceso y guardarlo por medio de colecciones. El resto del proceso varía de acuerdo al enfoque de análisis (AVISPA-vAVISPA).

- AVISPA: para este caso los elementos recuperados pasan a un proceso de resolución de referencia en el cual se identifican las características necesarias para el cálculo de las métricas así como las asociaciones con otros elementos, recuperada esta información con la cual procedemos al cálculo de las métricas que definirán la representación visual de cada elemento.
- vAVISPA: Los elementos recuperados en el primer paso del importar, pasan a un proceso de resolución de referencia similar al utilizado en AVISPA pero con la diferencia que este realiza una clasificación de las tareas que hacen parte del proceso y las tareas que solo están definidas a nivel de contenido del método, adicionalmente es recuperada la información correspondiente al tipo de variabilidad presentada por cada elemento. Con la descripción y relaciones de cada uno de los elementos se procede a un análisis de **variabilidad** en el cual los elementos que contemplen algún tipo de variabilidad pasan a un proceso de resolución de variabilidad. Finalizado el proceso de resolución de variabilidad los elementos proceden a calcular las métricas utilizadas para su representación visual.

Al final los elementos recuperados y procesados son desplegan en la interfaz de la herramienta para que el usuario pueda navegar en cada una de las categorías dispuestas para representar el modelo y visualizar los diversos Blueprints que faciliten el análisis del modelo de proceso como se puede observar en la Fig.22.

vAVISPA ha actualizado el meta-modelo de AVISPA, como se muestra en la Fig 25, para poder analizar y representar los mecanismos de variabilidad definidos por SPEM 2.0. vAVISPA actualiza algunos de los métodos establecidos por AVISPA y define una serie de variables para control y registro de los elementos involucrados en la variabilidad.



**Fig. 29 Estructura Meta-Modelo vAVISPA**

La clase PMObject contiene todas las operaciones y atributos comunes a todos los elementos de contenido de método de SPEM2.0 (esencialmente un identificador, nombre, descripción, identificador del elemento asociado por variabilidad). Por otro lado las clases PMRole, PMTask, PMArtifact, describen los elementos de contenido de método, en el caso específico de PMTask la incorporación atributos de persistencia para los elementos de variación (Array Contribuidores-baseTask) y variables de control (variabilityType-visibility) para el manejo de la representación de cada tipo de variabilidad. La clase PMTaskGroup contiene las reglas que establecen para cada uno de los Blueprints.

### 3.7.2. Realización de los casos de uso a nivel de diseño e implementación

Esta sección contiene algunas de las implementaciones de los principales casos de uso que han sido desarrollados para la construcción de vAVISPA.

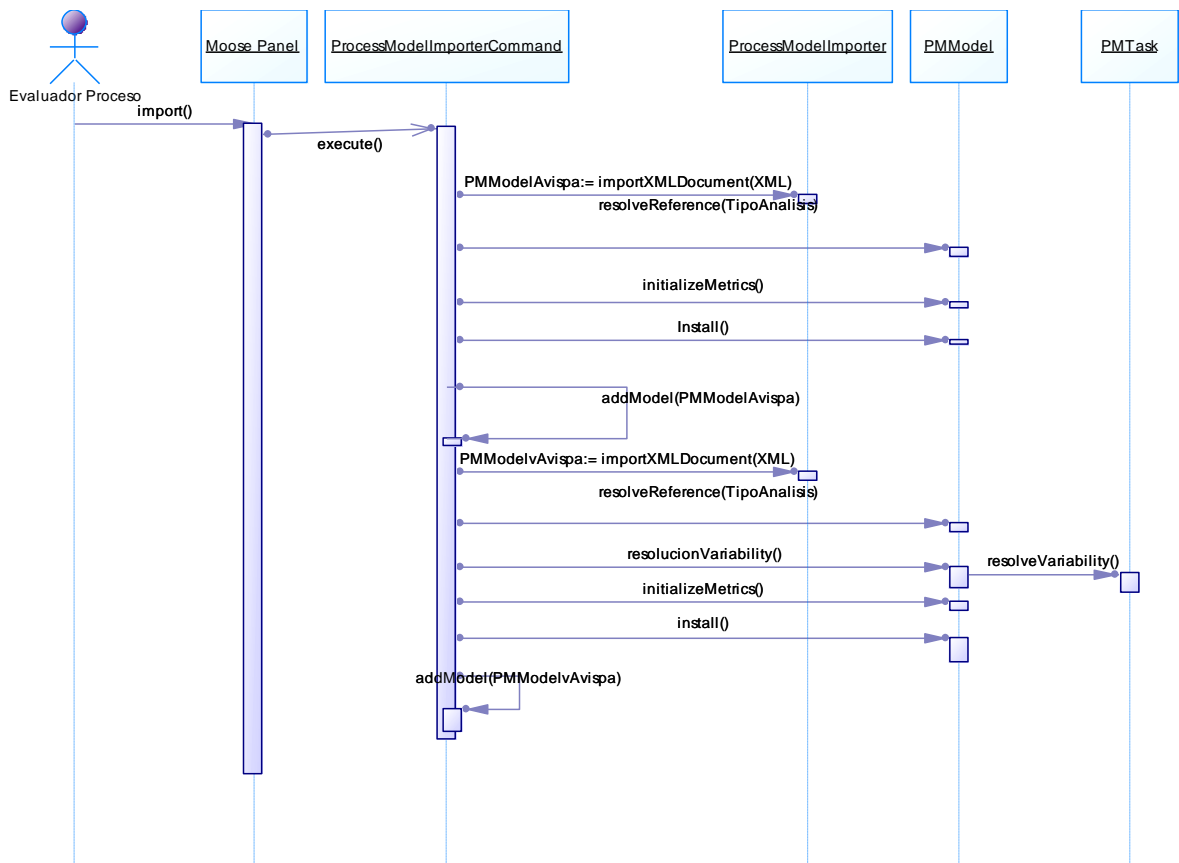


Fig. 30 Diagrama de Secuencia importar vAVISPA

En la Fig. 26 podemos ver la estructura del proceso de importar llevado a cabo por vAVISPA. Inicia cuando un actor (evaluador de proceso) realiza un llamado a la interfaz del moose panel para que dé inicio con el proceso de cargar del archivo que contiene la definición del modelo en formato XML. Con el modelo de proceso cargado, el primer paso consta en recuperar las estructuras para la validación sin tener en cuenta la variabilidad, para ello es invocado el método **importXMLDocument** de la clase ProcessModelImporter el cual es el encargado de descomponer al proceso en unidades elementales y las organiza como colecciones en una instancia de PMModel. Con esta colección es llamado al método **resolveReference** de la clase PMModel donde serán establecidas las métricas y asociaciones que describen al proceso, para poder realizar el siguiente paso que consiste en el cálculo de las métricas para cada uno los elementos que conforman al proceso con el método **initializeMetrics** de la clase PMModel. Por último es agregado el modelo completamente estructurado a la interfaz para que pueda ser navegado por el usuario.

Recuperado el modelo sin resolución de variabilidad al realizar el proceso de importe, vAVISPA al igual que en el primer caso inicia con la obtención de la definición del modelo en formato XML, se descompone al proceso en unidades básicas utilizando el método

**importXMLDocument** de la clase **ProcesosModelImporter**. El resultado obtenido es procesado utilizando el método **resolveReference** de la clase **PMModel** pero adicional a calcular las métricas y asociaciones como es realizado por AVISPA, realiza una clasificación de que tareas son utilizados por el proceso y cuales solo se encuentran definidas en el contenido del método. Las métricas obtenidas son utilizadas por el método **resolucionVariability** de la clase **PMModel** donde es recorrida cada una de las tareas para invocar al método **resolveVariability** de la clase **PMTasks** en donde es resuelto cada uno de los tipos de variabilidad que presenta el estándar SPEM 2.0, re-identificados cada uno de los elementos que componen el modelo de proceso, teniendo en cuenta, la variabilidad se procede a calcular las métricas para describirlo utilizando el método **initializeMetrics** de la clase **PMModel**. Por último es agregado el modelo a la interfaz para que poder ser navegar por el evaluador.

```

tasks := contentElements select: [:el | (el attributeAt: #xsi:type) = 'uma:Task' ].
pmTasks := tasks collect: [:xmlElement | PMTask fromXMLDescription: xmlElement].

roles := contentElements select: [:el | (el attributeAt: #xsi:type) = 'uma:Role' ].
pmRoles := roles collect: [:xmlElement | PMRole fromXMLDescription: xmlElement].

artifacts := contentElements select: [:el | #('uma:Artifact' 'uma:Outcome')
    includes: (el attributeAt: #xsi:type)].

deliverables := contentElements select: [:el | #('uma:Deliverable')
    includes: (el attributeAt: #xsi:type)].

pmArtifacts := artifacts collect: [:xmlElement | PMArtifact fromXMLDescription: xmlElement].

pmDeliverables := deliverables collect: [:xmlElement | PMArtifact fromXMLDescription: xmlElement].

pmDeliverables do[:element| element isDeliverable: true].

pmArtifacts do: [:element | element isDeliverable:false].

pmDeliverables do[:element| pmArtifacts add:element].

taskDescriptors := breakdownElements select: [:el | (el attributeAt: #xsi:type) = 'uma:TaskDescriptor' ].
pmTaskDescriptors := taskDescriptors collect: [:xmlElement | PMTaskDescriptor fromXMLDescription: xmlElement ].

activities := breakdownElements select: [:el | (el attributeAt: #xsi:type) = 'uma:Activity' ].
pmActivities := activities collect: [:xmlElement | PMActivity fromXMLDescription: xmlElement ].

```

**Fig. 31** Descomposición por componentes del método **importXMLDocument**

Como se puede observar en la Fig. 27, el método **importXMLDocument** extrae cada uno de los componentes que conforman el proceso y los organiza en colecciones de elementos. Las estructuras recuperadas por AVISPA correspondían a tareas, roles y productos de trabajo, en el caso de vAVISPA incorpora al método original la recuperación



de los elementos identificados por la etiqueta `'uma:TaskDescriptor'`, los cuales describen que tareas son usadas dentro de la definición del modelo de proceso.

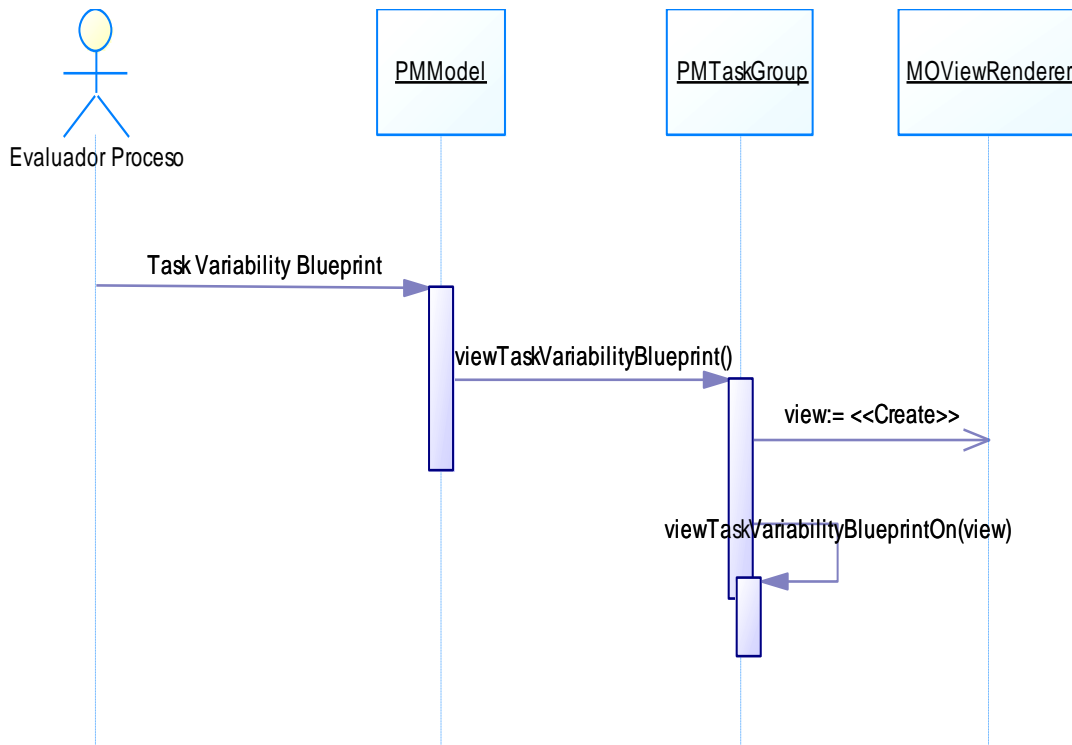
Una de las más importantes actualizaciones obtenidas frente a la versión inicial brindada por AVISPA, es la incorporación de un método encargado de dar soporte a la resolución de los diversos tipos de variabilidad (na - contributes - replaces - extends - extendsReplaces) de acuerdo a las reglas que establecidas en el estándar SPEM 2.0 y adaptadas para su aplicación en el enfoque visual. El método denominado **resolveVariability** de la clase PMModel encargado de redefinir las relaciones que componen a un elemento, modificando así su comportamiento con el modelo de proceso en general de acuerdo a lo establecido para cada tipo de caso de variabilidad que contiene especificado el modelo de proceso con SPEM 2.0. En la Fig. 32 se puede observar un fragmento de este método que corresponde a los casos de variabilidad na y contribución.

```
(self resolvedVariability = false
  ifTrue: [
    (self variabilityType = 'na'
      ifTrue: [
        Transcript show: 'fin de resolución de la variabilidad ', self mooseName; cr.
        self resolvedVariability: true.
      ]
    ).
    (self variabilityType = 'contributes'
      ifTrue: [
        self contributor do:[:contribute |
          contribute resolveVariability .
          contribute artifactOutputs do:[:element|
            (self artifactOutputs includes: element ) = false
            ifTrue: [ self artifactOutputs add: element ] .
          ].
          self artifactMandatoryInputs addAll: contribute artifactMandatoryInputs.
          self roles addAll: contribute roles .
          contribute visibility: false.
        ].
        Transcript show: 'resolucion contribute ', self mooseName; cr.
      ]
    ).
  ]
)
```

Fig. 32 Segmento resolveVariability - Variabilidad tipo Contribute

El procesamiento de la información para el análisis realizado por medio de los Blueprints. Cada uno de ellos puede representar una parte de información sobre el modelo de proceso como por ejemplo: describir las relaciones entre los componentes del modelo de proceso, identificar posibles errores debido a perdidas conceptuales o de especificación a

la hora de la definición del modelo e proceso, representación de anomalías en el modelo de proceso por medio de patrones, entre otras.



**Fig. 33 Diagrama Secuencia generación Task Variability Blueprint**

Como se puede observar en la Fig. 29, la estructura requerida para la generación de los Blueprint, no es tan compleja, esta basada en una solicitud al PMModel que contiene todos los elementos que hacen parte del proceso, el cual delega la tarea a una clase especializada que contiene la información para la construcción de cada uno de los Blueprints desde esta clase es instanciado un objeto de visualización utilizando la clase MOViewRender donde será visualizado y por último hace un llamado al método viewTaskVariabilityBlueprintOn donde establece la forma y estructura de los componentes que harán parte de la vista.

En el método viewTaskVariabilityBlueprintOn que se observa en la Fig. 34, grafica cada una de las tareas que contiene el proceso, definiendo su estructura de acuerdo a las métricas planteadas en la sesión 3.4. También son graficadas las asociaciones entre cada una de las tareas de acuerdo al tipo de relación de variabilidad que está definida entre cada uno de los elementos que conforman al proceso.

```

viewTaskVariabilityBlueprintOn: view
| ds components orderedComponents normalizer cycleColor coll auxE auxC|
  coll:=self.
  view shape rectangle
    borderWidth: 1;
    linearFillColor: #numberOfRoles within: coll;
    width: [:each | each numberOfArtifactOutputs * 10];
    height: [:each | each numberOfArtifactMandatoryInputs * 10].
  view nodes: coll.
  auxE := coll select: [:each | each variabilityType = 'na' ].
  view shape arrowedLine width: 1.
  view edges: auxE from: #yourself toAll: #followingTasks.
  auxC := coll select: [:each | each variabilityType = 'contributes'].
  view shape arrowedDashedLine width: 1; color: Color blue.
  view edges: auxC from: #yourself to: #baseTask.
  auxC := coll select: [:each | each variabilityType = 'replaces'].
  view shape arrowedDashedLine width: 1; color: Color red.
  view edges: auxC from: #yourself to: #baseTask.
  auxC := coll select: [:each | each variabilityType = 'extends'].
  view shape arrowedDashedLine width: 1; color: Color green.
  view edges: auxC from: #yourself to: #baseTask.
  auxC := coll select: [:each | each variabilityType = 'extendsReplaces'].
  view shape arrowedDashedLine width: 1; color: Color orange.
  view edges: auxC from: #yourself to: #baseTask.
  view treeLayout.
  view root interaction item: 'inspect group' action: [:v | coll inspect]

```

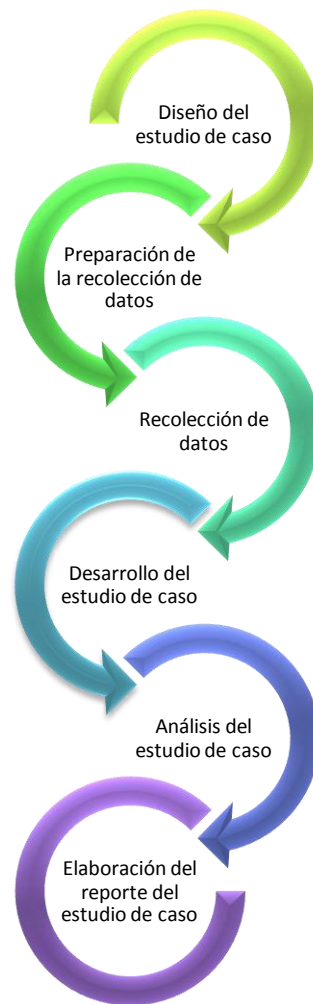
Fig. 34 Definición viewTaskVariabilityBlueprintOn

#### **4. Estudio de caso: Análisis Visual de Variabilidad en Modelos de Procesos SPEM 2.0: vAVISPA**

Desde los inicios de esta investigación el planteamiento de la importancia de la especificación y adaptación de un modelo de proceso, debido a las múltiples ventajas que obtenidas como lo son: la formalización de procesos, mejores estimaciones de tiempos y recursos, optimización de recursos dependiendo del tipo de proyecto (desarrollo nuevo, adaptación, mantenimiento, extensión, etc.), Comprensión por los participantes del proyecto, entre otras. Pero teniendo en cuenta que requiere un gran esfuerzo obtenerlas y a su vez que el modelo depende de factores muy particulares como: la especificación del proceso, adaptación del conocimiento colectivo de una organización para un tipo de proceso en particular, la complejidad de los mecanismos para la especificación de variabilidades, la experticia del diseñador del proceso entre otras. Por lo que desarrollar un estudio de caso para la validación de la propuesta estará ligado estrechamente al contexto sobre el cual sea aplicada. Con este antecedente es definido el estudio de caso que valide la efectividad de la propuesta vAVISPA planteada en este trabajo de grado, para la detección de posibles errores en modelos de proceso especificados con SPEM 2.0 y descritos con EPF composer.

##### **4.1. Metodología**

La metodología para este estudio es desarrollada utilizando como marco de referencia la guía de Runeson [52] y las recomendaciones de Robert K. Yin [53]. Planteando un estudio de caso de tipo “revelatorio” cuyo propósito estará enfocado en determinar la efectividad de la propuesta en el análisis visual de modelos de procesos, verificando ser más efectiva que el enfoque propuesto por AVISPA. Para la validación serán realizadas pruebas cualitativas y cuantitativas sobre un modelo de proceso específico con el fin de determinar que tanto afecta la variabilidad en la detección de anomalías utilizando el enfoque de análisis visual. El desarrollo del estudio de caso está planteado en base al modelo presentado por Runeson:



**Fig. 35 Metodología del estudio de caso**

- Diseño del estudio de caso:** La construcción del estudio de caso debe estar estrechamente ligada con los objetivos del estudio que lo motiva, por lo que requiere que su diseño este enfocado a comprobar o refutar la motivación de la investigación. Para está primera los esfuerzos están centrados en definir el objetivo del estudio, que elementos son necesarios para desarrollarlo y que medios serán utilizados para la obtención de los resultados. El estudio de caso está enfocado en el objetivo de determinar qué tan efectivo resulta vAVISPA para la detención de posibles errores en modelos de procesos que contengan algún tipo de variabilidad en su definición utilizando un enfoque visual. Para validar dicho objetivo está planteada una actividad comparativa de análisis, utilizando los dos mecanismos de análisis visual que se tiene referencia. Por una parte la evaluación utilizando AVISPA, detectando los errores que identificados por medio de los Blueprints de patrones, terminado el proceso de recopilación de datos proceder a una validación del mismo modelo utilizando el enfoque vAVISPA y documentando los errores identificados. Finalizado el proceso de

evaluación realizar un análisis de los datos recopilados en los dos casos con el fin de determinar qué tan efectivo resulta el enfoque de vAVISPA para la depuración de falsos errores.

- **Preparación de la recolección de datos:** Para el estudio de caso no solo basta con generar las herramientas necesarias para el control de la experiencia, sino que deben establecerse de qué forma se van a utilizar. Para este caso contamos con una introducción en que consiste vAVISPA, cuales son los posibles errores que pueden identificarse y patrones de error que pueden detectarse con el enfoque visual de análisis. Para el proceso de evaluación la utilización de una plantilla sobre la cual deben registrarse la información sobre los errores detectados y las observaciones que pueda ofrecer el evaluador.
- **Recolección de Datos:** El proceso de recolección de datos para este estudio de caso tendrá dos enfoques uno cualitativo y uno cuantitativo. Para la evaluación cuantitativa se tendrá como referencias las observaciones, comentario y perspectivas que pueda ofrecer el evaluador. Mientras que el enfoque cuantitativo estará centrado en determinar los errores detectado por cada enfoque y la comparación de que tan efectivo resulta la propuesta vAVISPA en el análisis de modelos de procesos frente al análisis obtenido utilizando un enfoque que no tiene en cuenta la variabilidad en los modelos de proceso.
- **Desarrollo del estudio de caso:** Para el desarrollo de la actividad iniciara con una introducción de conceptos necesarios para su desarrollo, como son los mecanismos de variabilidad dispuestos por SPEM 2.0, en que consiste el enfoque de análisis visual y una capacitación en el manejo de la herramienta vAVISPA en cuanto a importe del modelos de proceso, Blueprint disponibles y patrones de error que pueden ser detectados utilizando el enfoque visual. Finalizado el proceso de capacitación el evaluador cuenta con un modelo de proceso especificado en SPEM 2.0 y generado en formato XML utilizando EPF Composer, para realizar el proceso de análisis así como una plantilla de captura de errores en la que debe registrar la evaluación del modelo. La actividad de análisis del modelo está planteada en dos fases, la primera realizando el análisis del modelo utilizando AVISPA para la detección de posibles errores en el modelo de proceso, mientras que para la segunda fase se realiza la evaluación utilizando vAVISPA para la validación del modelo. Paralelo al desarrollo de la actividad es llevado a cabo el proceso de observación.
- **Análisis del estudio de caso:** Corresponde a la fase de procesamiento de la información obtenida del desarrollo del estudio de caso, los datos recopilados por medio de las planillas de detección de errores serán procesados y cuantificados, así como las observaciones dadas por el evaluador y los datos del proceso de monitoreo de la actividad

- **Elaboración del reporte del estudio de caso:** El reporte del estudio de caso consignado en este documento con el fin de compartir no solo los resultados obtenidos si no las experiencias adquiridas durante el desarrollo de la evaluación

#### 4.2. Pregunta de Investigación

La propuesta vAVISPA está desarrollada como un esfuerzo adicional en el fortalecimiento de la idea de poder realizar una evaluación visual del modelo de proceso de una forma más intuitiva y que facilite la tarea de evaluación para el diseñador del proceso. Extendiendo el enfoque de análisis visual por medio de Blueprints. Incorporando las normas de resolución de cada uno de los métodos de variabilidad se busca que la detección de posibles problemas en el modelo de proceso sea más efectiva, reduciendo inconsistencia o falsos errores que puedan presentarse debido a la no resolución de las variabilidades que pueda contener el modelo de proceso. Con este propósito es planteado el siguiente interrogante para este estudio de caso ¿vAVISPA incorporando la resolución de la variabilidad resulta ser una solución más efectiva que AVISPA en el análisis visual de modelos de procesos especificados en EPFC?

#### 4.3. Objetivo del estudio de caso

El objetivo de este estudio de caso es evaluar la efectividad de vAVISPA depurando posibles falsos errores generados por la incorporación de métodos de variabilidad dentro de la definición de un modelo de proceso especificado en SPEM 2.0 utilizando EPFC además de validar la asimilación del método de análisis visual por parte de los participantes del estudio de caso

#### 4.4. Selección del estudio de caso

Para la selección del estudio de caso son fijados una serie de criterios como:

- **Disponibilidad de procesos en EPFC:** La poca formalización de modelos de procesos existentes y accesibles, la disponibilidad de modelos especificados en SPEM 2.0 y generados con EPFC y la confidencialidad manejada por la organizaciones con respecto a sus modelos de procesos reducen en gran medida el espectro de modelos de procesos disponibles para poder llevar a cabo la evaluación por lo cual el proceso de análisis se llevara a cabo utilizando los pocos procesos disponibles.
- **Disponibilidad de los Ingenieros de procesos:** Las capacidades conceptuales y de abstracción para la definición y estudio del modelo de proceso requiere contar con personas capacitadas que puedan interpretar y verificar de forma más objetiva el modelo de proceso.

- **Revelatorio<sup>4</sup>**: La disponibilidad de modelo de procesos es acotada aún más por la limitante de que para lograr una evaluación más efectiva del enfoque visual presentado por vAVISPA requiere de modelos que contengan algún tipo de variabilidad en su especificación que permita realizar una evaluación más efectiva.
- **Embebido<sup>5</sup>**: El análisis del impacto de la variabilidad en modelos de procesos utilizando un enfoque visual está limitado a la disponibilidad de técnicas o herramientas desarrolladas para dicho tipo de análisis, en nuestro caso se cuenta con la propuesta AVISPA para identificar el impacto que tiene la variabilidad en un modelo de proceso.

#### 4.5. Descripción del estudio de caso

El estudio de caso es de tipo embebido [52] ya que se establecieron dos unidades de análisis correspondientes a dos modelos de procesos que contemplan algún tipo de variabilidad en su definición. Los dos procesos son procesos open source, están descritos utilizando SPEM 2.0 y generados por medio del EPFC lo que los convierten en candidatos ideales para realizar la verificación del enfoque de vAVISPA. El proceso de evaluación estará dividido en dos partes: la primera en donde la evaluación efectuada con la propuesta AVISPA la cual no contempla las variabilidades del modelo de procesos, mientras que para la segunda parte el modelo es evaluado con el enfoque visual pero tomando en cuenta la variabilidad que presenta vAVISPA. De forma que se pueda efectuar una comparación entre cada uno de los enfoques.

Para el proceso de evaluación serán seleccionadas personas con cierto grado de conocimiento en modelos de procesos y SPEM 2.0, pero de igual forma serán unificados conceptos en la fase introducción del estudio de caso, para garantizar que no se presente desacuerdos en cuanto a los mecanismos de representación de variabilidad en modelo de procesos. Al igual que también es necesario la jornada de capacitación de forma que el evaluador pueda interactuar de manera más eficiente con el prototipo vAVISPA.

##### 4.5.1. Los modelos de proceso para el análisis

La búsqueda de modelos de procesos actos para realizar este estudio de caso género en sí misma un gran reto para el desarrollo de la actividad, debido a que para obtener información de la efectividad de la propuesta es necesario contar con modelos de proceso muy específicos que aporten información significativa para detectar la eficiencia en la

---

<sup>4</sup> Estudio del caso tipo revelatorio: en el que se presenta a los ojos del investigador un fenómeno antes no estudiado.

<sup>5</sup> Estudio del caso tipo embebido: donde se estudian múltiples unidades de análisis dentro de un caso.



detección de errores. Un modelo de proceso útil para la prueba debe contar con tres requisitos mínimos: Especificación en SPEM 2.0, Generación en formato XML a partir de EPFC y la más relevante que contenga algún tipo de mecanismo de variabilidad en su definición. Por medio de estos parámetros se realiza una búsqueda entre los posibles modelos de procesos:

| Proceso  | Especificado con SPEM 2.0 | Generado con EPFC | Presenta mecanismos de adaptación | Tipo de fuente |
|----------|---------------------------|-------------------|-----------------------------------|----------------|
| Tutelkan | Si                        | Si                | No                                | Abierta        |
| OpenUP   | Si                        | Si                | Si                                | Abierta        |
| Protest  | Si                        | Si                | No                                | Industria      |
| XP       | Si                        | Si                | No                                | Abierta        |
| Scrum    | Si                        | No                | -                                 | Abierta        |
| Remics   | Si                        | Si                | Si                                | Abierta        |

**Tabla 3 Modelos de procesos identificados para el estudio de caso**

De la búsqueda que realizada solo se encontraron dos modelos actos para el desarrollo del estudio de caso OpenUp y Remics. A partir de la definición de estos modelos del proceso será aplicado el análisis visual en busca de posibles errores en su especificación y analizara el impacto que tiene para el modelo la incorporación de mecanismos de variabilidad.

### **Remics**

Con la creciente demanda de nuevos mecanismo para prestación de servicios, los mecanismos de cloud computing se han vuelto más populares permitiendo esquemas de licencias flexibles y ahorro en costos de infraestructura para los proveedores de servicios. Sin embargo la construcción de estos sistemas implica una gran inversión de tiempo y esfuerzo. Remics propone un progreso en el legado de migración de sistemas a servicio en la nube al proporcionar una metodología ágil de soporte a las organizaciones en la adaptación de sus sistemas heredados al paradigma en la nube [54, 55].

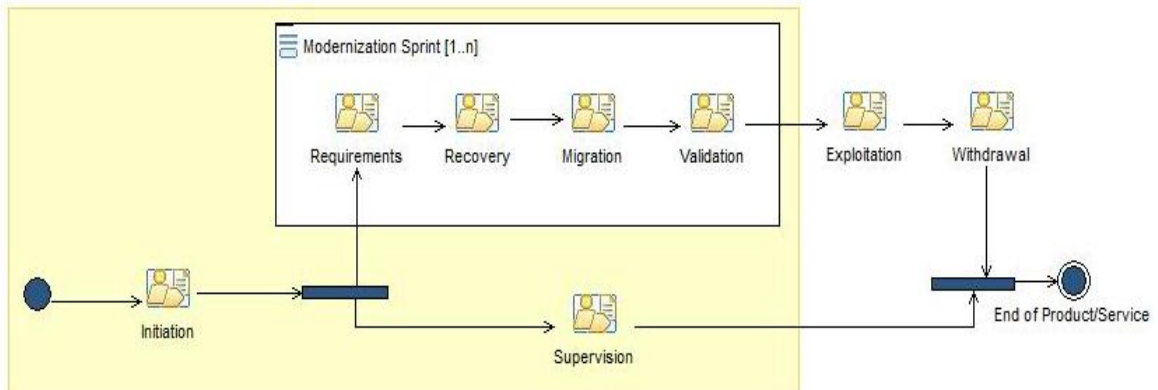


Fig. 36 Flujo de trabajo REMICS[54]

### OpenUP (Open Unified Process)

Consiste en un proceso ágil y unificado que contiene una mínima serie de prácticas para ayudar a los equipos a generar desarrollos software más efectivo. Está diseñado implementando enfoques iterativos e incrementales dentro de un ciclo de vida estructurado. La esencia de OpenUp radica en contener los elementos básicos necesarios para el desarrollo de un proceso software y la adaptabilidad por medio de extensión para adecuarse a diversos proyectos según sean las necesidades del contexto[56, 57].

#### 4.5.2. Indicadores y métricas

Para obtener una evaluación coherente del estudio de caso. Requiere establecer métricas o indicadores que faciliten analizar de manera objetiva la efectividad de la propuesta desarrollada. Para lo que se diseña un esquema de los datos a obtener, resumido en la siguiente tabla:

| Pregunta de Investigación   | Indicadores  | Métricas  | Instrumentos  |
|---|--|---|---|
| ¿vAVISPA incorporando la resolución de la variabilidad resulta ser una solución más efectiva que AVISPA en el análisis visual de modelos de procesos especificados en EPFC? | Efectividad de AVISPA                                  | Porcentaje de errores reales respecto los errores detectados, con AVISPA          | -Modelo del proceso<br>-EPF Composer<br>-Prototipo AVISPA<br>-Planilla de evaluación  |
|   | Efectividad de vAVISPA                                 | Porcentaje de errores reales respecto los errores detectados, vAVISPA             | -Modelo del proceso<br>-EPF Composer<br>-Prototipo vAVISPA<br>-Planilla de evaluación |
|   | Índice de variación porcentual en detección de errores | Variación entre la detección de errores por el evaluador y los errores detectados | Efectividad de AVISPA<br>Efectividad de vAVISPA                                       |

|  |  |  |   |
|--|--|--|---|
|  | Facilidad del enfoque visual y del prototipo | Experiencia del evaluador utilizando el enfoque visual para analizar un modelo de proceso. | Observaciones de los evaluadores y observaciones del desarrollo del estudio de caso |
|--|--|--|---|

**Tabla 4** Tabla de Métricas e Indicadores

A continuación se explica de forma detallada las métricas e indicadores del estudio de caso.

- Efectividad de AVISPA: Al momento de realizar la evaluación el participante determinara la presencia de errores teniendo en cuenta los patrones de error que pueden identificarse por la herramienta AVISPA, cada error detectado es consignado en la plantilla de errores para su posterior análisis. Esta métrica pondera el porcentaje de errores reales, respecto al porcentaje real de errores detectados por AVISPA.

$$EA p_i = \frac{EEp_i * 100}{ERp_i}$$

Donde

**EEp<sub>i</sub>**: Representa el número total de errores detectados al realizar el análisis por cada uno de los patrones

**ERp<sub>i</sub>**: Representa el número total de errores detectados por el evaluador por cada uno de los patrones

- Efectividad de vAVISPA: Al momento de realizar la evaluación el participante determinara la presencia de errores teniendo en cuenta los patrones de error que se pueden identificar por la herramienta vAVISPA, cada error detectado es consignado en la plantilla de errores para su posterior análisis. Esta métrica pondera el porcentaje de errores reales, respecto al porcentaje real de errores detectados por vAVISPA.

$$EvA p_i = \frac{EEp_i * 100}{ERp_i}$$

- Índice de variación porcentual en la detección de errores: utilizado para identificar que tanto varía la detección utilizando uno de los métodos con respecto a la detección lograda con el análisis.

$IVAp_i$ : Índice de variación de AVISPA para el patrón sub i

$$IVAp_i = 1 - \frac{EAp_i}{100}$$

$IVvAp_i$ : Índice de variación de vAVISPA para el patrón sub i

$$IVvAp_i = 1 - \frac{EvAp_i}{100}$$

Cada una de las métricas es aplicada con base a un patrón de los establecidos para el modelo de análisis visual desde la propuesta AVISPA, Los Patrones analizados se resumen en la siguiente tabla:

| Patrones                                |    |  |                     |
|---|----|--|---------------------|
| Nombre                                  | Id | descripción  | Blueprint           |
| No Asociación de Guía                   | P1 | Si un rol, tarea o producto de trabajo no cuenta con una guía que indique como ser ejecutado, genera una alta posibilidad de ser realizado de forma inapropiada. El Blueprint presenta resaltando los elementos que no poseen una guía en color azul.              | Task<br>WorkProduct |
| Patrón de error Rol Sobrecargado        | P2 | Si un rol se ve involucrado en gran cantidad de tareas genera un riesgo, Si llega a fallar, todas las tareas asociadas pueden verse afectadas. Este patrón se identificar resaltando los roles que presentan sobrecarga de color rojo.                             | Roles               |
| Patrón de error multipropósito de tarea | P3 | Si el modelo de proceso presenta una tarea que genera muchos productos de trabajo (salidas) puede significar que no presenta una correcta granularidad y sería conveniente redefinirla. Este patrón se visualiza resaltando las tareas sobrecargadas de color rojo | Task                |
| Demanda de Producto de trabajo          | P4 | Si un producto de trabajo es entrada de un alto número de tareas este elemento se puede transformar en un cuello de botella para el proceso lo cual no es viable en ninguna circunstancia. Estos elementos se  | WorkProduct         |

|                                |    |   |                     |
|--------------------------------|----|---|---------------------|
|                                |    | identifican con los elementos que resaltados en color rojo.   |                     |
| Independencia de sub proyectos | P5 | Las tareas y productos de trabajos están relacionados por medio de arcos indicando una relación de precedencia. Si consideramos que el modelo indica el camino de trabajo un único proyecto. Es incorrecta la presencia de sub-grafos. Cada sub-grafo se identificada con un color diferente cada sub proceso   | Task<br>WorkProduct |
| Desechos o Residuos            | P6 | En los productos de trabajo un arco representa dependencia. Si hay un $WP_a$ que precede $WP_b$ en el grafo, eso significa que hay una tarea donde $WP_a$ es una entrada y $WP_b$ es una salida. A nivel de las hojas del grafo es decir los nodos que no refinan un sucesor se definen como un entregable. Este patrón resalta en azul aquellas que no se definen como entregables y no son entrada para otra tarea dentro del proceso, entonces se considera como residuo | WorkProduct         |

Tabla 5 Patrones de Error

También se realiza una verificación registrando el porcentaje del modelo utilizado por el proceso y variabilidades dispuestas en la definición, información solo de registro para la verificación del estado del modelo de proceso.

#### 4.6. Ejecución del Estudio de caso

Para el desarrollo del estudio de caso, se trabajó con el evaluador en un proceso de revisión dividido en dos partes la primera enfocada al análisis del modelo sin tener en cuenta el factor variabilidad (AVISPA) y la segunda una evaluación sobre el mismo modelo pero donde es tenido en cuenta el factor variable (vAVISPA). Está planteada el desarrollo de la evaluación en 4 pasos:



**Fig. 37 Evaluación del Estudio de caso**

Las fases de evaluación está centrada solo en identificar los patrones en cada uno de las herramientas y reportarlas en la plantilla proporcionada: Mientras que la fase de análisis de errores está diseñada para la revisión de los errores detectados junto con el encargado de la prueba para validar la efectividad del error y en la última fase del proceso determinaran si los errores detectados por la herramienta corresponden o no a un error.

El desarrollo del estudio de caso está estructurado de la siguiente manera:

- **Capacitación:** Se socializan conceptos básicos entre el grupo de evaluadores sobre: modelos de procesos, técnicas de variabilidad que pueden encontrar. Los patrones que serán utilizados para detectar anomalías y el prototipo vAVISPA para el desarrollo de la evaluación. También será socializado el procedimiento para realizar el proceso de evaluación.
- **Preparación de la evaluación:** Adecuación del espacio para poder realizar la evaluación brindando los mecanismos necesarios para su desarrollo. Le es entrega al evaluador el modelo en XML sobre el cual va realizar el proceso de evaluación junto con una plantilla donde consignaran los errores identificados, el formato de la plantilla se puede observar en la Fig. 38.

|   |   |
|---|---|
| <p>Patrones de error</p> <p>P1 Patrón error no asociación de guía</p> <p>P2 Patrón de error Rol Sobrecargado</p> <p>P3 Patrón de error multipropósito de tareas</p> <p>P4 Demanda de Producto de trabajo</p> <p>P5 Independencia de sub proyectos</p> <p>P6 Patrón error desechos</p> | <p><b>Estado</b></p> <p>(C) Comprobado #</p> <p>(FE) Falso Patrón de error #</p><br><p><b>Blueprint</b></p> <p>BA = Blueprint Artefact</p> <p>BR = Blueprint Roles</p> <p>BT = Blueprint Tareas</p> |
|---|---|

| Modelo   | OpenUp  | Método evaluación  | Avispa  |        |
|----------|---|--------------------|---|--------|
| Id_error | Elemento o elementos involucrados (Blueprint) | Id patrón de error | Analisis del error (Participante - Evaluador) | Estado |
|          |   |                    |   |        |

**Fig. 38 Plantilla para reportar errores detectados**

- **Desarrollo de la Evaluación:** El evaluador realiza la evaluación del modelo de proceso utilizando el enfoque de AVISPA el cuál está embebido dentro del prototipo vAVISPA y va reportando cada uno de los errores que puede identificar en los patrones disponibles. Finalizada la evaluación con AVISPA, se inicia el mismo proceso de evaluación pero utilizando el prototipo vAVISPA que maneja variabilidad en el proceso. En la Fig. 39 podemos observar la interfaz con la cual contara el evaluador donde están dispuestos los modelos y los mecanismos de validación de cada uno de los casos.

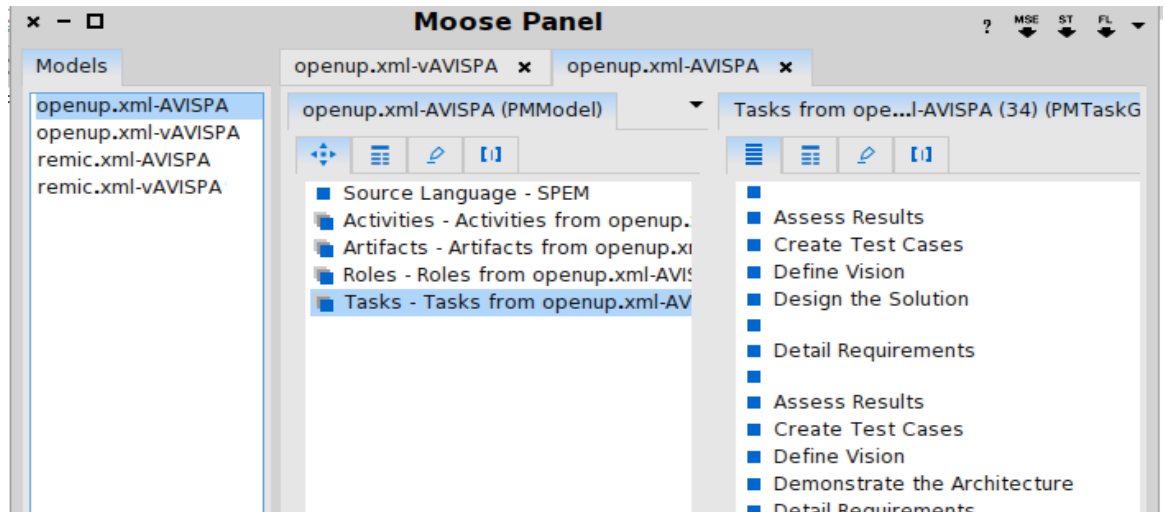


Fig. 39 Prototipo vAVISPA

- **Revisión de la Evaluación:** Corresponde al proceso de análisis de los resultados realizado al terminar la evaluación por los dos métodos (AVISPA y vAVISPA), para no influenciar los datos obtenidos por uno u otro método. Para el desarrollo de este análisis contamos con la participación del evaluador y el encargado de la prueba, serán utilizados como activos: EPF Composer, el modelo en XML y se utilizaron como soporte los Blueprints de variabilidad y MethodContent vs Process, para facilitar la identificación de elementos variables y en uso del proceso.
- **Entrega de Resultados:** Los ingenieros evaluadores entregan las planillas con los resultados incluyendo las conclusiones obtenidas en el proceso de análisis de los errores detectados.
- **Análisis de uso:** Realización de una encuesta al evaluador identificando aspectos positivos y negativos en la utilización del prototipo vAVISPA. También analizando la utilidad de los dos Blueprint que se han generado (variabilidad y methodcontent vs Process).

## 4.7. Resultados

El análisis de los resultados está focalizado solo en la identificación de errores por cada uno de los patrones que fueron evaluados, los resultados cuantitativos y cualitativos obtenidos. De la evaluación de modelo de proceso Remics se presentara información parcial debido a problemas presentados para la estimación de errores por medio de análisis debido a la cantidad de errores identificados e imposibilidad de verificar cada uno contrastándolo con el modelo debido a cuestiones de tiempo y disponibilidad del evaluador.

### 4.7.1. Resultados Cuantitativos

#### Mediciones Directas

Esta sección describe la información obtenida en el desarrollo del estudio de caso. Desde el desarrollo de la capacitación hasta la encuesta final, las medidas de tiempo están especificadas en minutos. En la Tabla 6, se especifica un resumen del estado de los procesos y del modelo de proceso. Describe la información de cada tipo de variabilidad y una comparativa de que porcentaje del modelo es usado realmente por el proceso.

| Modelo de Proceso | Variabilidad (Task) |    | porcentaje de Uso modelo Proceso (Task) |
|-------------------|---------------------|----|---|
| Open UP           | Contribución        | 0  | 70.5%                                   |
|                   | Reemplazo           | 0  |   |
|                   | Extensión           | 9  |   |
|                   | Extensión-Reemplazo | 1  |   |
| Remics            | Contribución        | 0  | 59.8%                                   |
|                   | Reemplazo           | 0  |   |
|                   | Extensión           | 32 |   |
|                   | Extensión-Reemplazo | 0  |   |

Tabla 6 Información variabilidad y uso del modelo de proceso

La Tabla 7 nos registra el tiempo empleado por el evaluador en cada una de las actividades del estudio de caso. El tiempo de evaluación corresponde al tiempo que le tarda al evaluador identificar y registrar los elementos detectados como error por cada uno de los patrones evaluados, mientras que el tiempo de análisis corresponde al tiempo en el que el evaluador y quien está a cargo del estudio de caso realizan un análisis de cada uno de los elementos identificados del proceso evaluación para precisar porque son establecidos como error y si corresponde a un falso error o si en realidad es un problema en la especificación del modelo de proceso.



| Modelo | Actividad                | Tiempo |
|--------|--------------------------|--------|
| OpenUP | Evaluación AVISPA        | 30     |
|        | Evaluación vAVISPA       | 20     |
|        | Análisis errores AVISPA  | 30     |
|        | Análisis errores vAVISPA | 25     |
|        | Conclusiones             | 5      |
| Remics | Evaluación AVISPA        | 60     |
|        | Evaluación vAVISPA       | 70     |
|        | Análisis errores AVISPA  | -      |
|        | Análisis errores vAVISPA | -      |
|        | Conclusiones             | -      |

**Tabla 7 Costo en tiempo de las actividades de evaluación**

La Tabla 8 nos da un resumen de la relación de errores encontrados por cada patrón en el caso de OpenUP y por cada uno de los métodos de evaluación que se disponían.

| OPENUP  | No asociación de guía | Rol Sobrecargado | Tarea multipropósito | Demanda producto de trabajo | Independencia Sub proyecto |              | Residuo |
|---------|-----------------------|------------------|----------------------|-----------------------------|----------------------------|--------------|---------|
|         |                       |                  |                      |                             | Task                       | Work Product |         |
| AVISPA  | 4                     | 8                | 8                    | 0                           | 16                         | 12           | 10      |
| vAVISPA | 4                     | 8                | 4                    | 0                           | 1                          | 10           | 10      |

**Tabla 8 Relación de errores detectados en OpenUP**

La Tabla 9 corresponde al resumen de los errores identificados en el modelo de proceso Remics por cada uno de los patrones analizados, tanto por AVISPA como por vAVISPA.

| Remics  | No asociación de guía | Rol Sobrecargado | Tarea multipropósito | Demanda producto de trabajo | Independencia Sub proyecto |              | Residuo |
|---------|-----------------------|------------------|----------------------|-----------------------------|----------------------------|--------------|---------|
|         |                       |                  |                      |                             | Task                       | Work Product |         |
| Avispa  | 221                   | 22               | 24                   | 22                          | 221                        | 145          | 103     |
| vAVISPA | 221                   | 22               | 45                   | 22                          | 78                         | 139          | 103     |

**Tabla 9 Relación de errores detectados en Remics**

Como se observa en la Tabla 7, para el modelo de proceso Remics no pudo llevarse a cabo el proceso de análisis de errores para ninguno de los dos casos de evaluación debido a tiempos del evaluador y a la gran cantidad de errores identificados en varios de los patrones, lo que convirtió la tarea de análisis en una actividad que involucra un enorme esfuerzo en tiempo, para poder verificar cada uno de los errores que fueron detectados y así determinar su veracidad. Solo se efectúa una revisión evaluando

métricas de número de errores que reflejan un patrón de comportamiento diferente al visto en OpenUP.

La medición de la efectividad solo pudo efectuarse sobre el modelo OpenUp ya que no es posible contrastar los elementos identificados por el evaluador y los elementos detectados en el proceso de análisis para el proceso Remics. Los datos que se obtuvieron son consignados en la siguiente Tabla.

| Patrón  | EE<br>(Avispa) | EE<br>(vAvispa) | ER | Ea     | EvA    | IvA  | IvVA |
|---------|----------------|-----------------|----|--------|--------|------|------|
| P1 (BA) | 4              | 4               | 4  | 100,00 | 100,00 | 0,00 | 0,00 |
| P1 (BT) | 14             | 14              | 14 | 100,00 | 100,00 | 0,00 | 0,00 |
| P2      | 8              | 8               | 8  | 100,00 | 100,00 | 0,00 | 0,00 |
| P3      | 8              | 4               | 4  | 50     | 100,00 | 0,5  | 0,00 |
| P5 (BT) | 16             | 1               | 1  | 6,25   | 100,00 | 0,93 | 0,00 |
| P5 (BA) | 12             | 10              | 1  | 8,33   | 10,00  | 0,92 | 0,90 |
| P6      | 10             | 10              | 10 | 100,00 | 100,00 | 0,00 | 0,00 |

Tabla 10 Valor Índices Obtenido en OpenUP

#### 4.7.2. Resumen Cualitativo

##### Apreciaciones de los evaluadores

Durante el trascurso de la prueba el evaluador realiza sugerencias tanto del proceso de evaluación como del prototipo vAVISPA algunas de las que se realizaron son plasmadas en esta sección.

Una de las dificultades planteadas por el evaluador, está ligada al modelo de proceso, debido al desconocimiento previo del modelo no permite validar de forma más precisa la veracidad de un error. La dificultad presentada durante la evaluación del proceso Remics dada la cantidad de errores que pudieron identificarse con los patrones. La sugerencia que dada para el desarrollo del proceso de validación, contando con una definición previa de modelos debido a la necesidad de mayor tiempo, conocimiento y esfuerzo para plantear una evaluación. En la que se pueda establecer un contexto para desarrollar la variabilidad por el propio evaluador.

Otra de las dificultades percibidas por el evaluador corresponde con el método de registro de errores, vAVISPA al ser una propuesta netamente visual para la identificación de errores no cuenta con un método de conteo que permitan registrar los elementos que son identificados como error, lo que genero dificultades debido al método de conteo para registro de errores en la prueba, más aún si tenemos en cuenta que se utilizó un proceso relativamente grande como lo es Remics.

Entre los aspectos positivos resaltan los 2 nuevos Blueprints incorporados en vAVISPA ya que el Blueprint de variabilidad en el modelo de proceso permite realizar seguimiento en la detección de los mecanismos de variabilidad, permitiendo identificar su impacto y

desarrollo en el modelo de proceso. El Blueprint methodContent vs Process permite identificar fácilmente que elementos del method content han sido descartados por el proceso, lo que facilita un análisis que determine validar si estos elemento deben o no ser descartados del proceso. También se resalta por el evaluador la practicidad de los patrones definidos en el enfoque visual ya que le permiten al evaluador identificar de forma sencilla que elementos presentan un riesgo para el modelo de proceso.

#### **Apreciaciones del investigador a cargo del estudio de caso**

El proceso de validación estuvo apoyado en las observaciones percibidas por la persona encargada de la prueba, las cuales son plasmadas en la siguiente sección.

Durante el transcurso de la prueba pudieron identificarse dificultades por el evaluador para poder interactuar con los elementos del Blueprint. Este problema es presentado cuando se evalúa un proceso demasiado grande debido a que la gran cantidad de arcos generados por la precedencia de los elementos, obstruyendo el poder interactuar con ellos y no permitiendo identificarlos fácilmente.

Entre los aspectos positivos que pudieron identificarse está en la asimilación del concepto de patrón de error por parte del evaluador a pesar de no contar con conocimiento previo del enfoque visual, lo que facilita el funcionamiento de vAVISPA, y favoreció a la detección de errores en el modelo. También se percibe que el evaluador a pesar de no tener una experiencia previa con la herramienta interactúa fácilmente con los Blueprints que permiten el análisis del modelo de proceso, solo con el antecedente de la capacitación previa recibida.

#### **4.8. Análisis de los resultados**

##### ***Análisis Resultados por Medición Cuantitativa***

La investigación previa para la detección de modelos factibles para el estudio dejo percibir lo limitado de modelos que cumplieran con las necesidades que requería el estudio de caso. De los dos modelos obtenidos que cumplían con los requerimientos para la evaluación, se realizó un análisis identificando que solo presentaban dos tipos de variabilidad (extends y Extends-replace), lo que percibe de cierta manera que los mecanismos de variabilidad no están siendo completamente aprovechados en la generación de modelos adaptables al contexto de la organización. Otro de los elementos conseguidos al realizar el análisis inicial, está ligado al potencial de uso del conocimiento colectivo registrado en el method content de los modelos, el porcentaje de componentes utilizados en el proceso no supera el 75% dejando un porcentaje considerable de elementos que permitirían al modelo adaptarse a nuevas condiciones organizacionales.

Los tiempos de análisis de errores observados en cada uno de los métodos de evaluación resultan relativamente cortos con respecto al proceso normal que es requerido para la identificación al utilizar otro enfoque. Tanto AVISPA como vAVISPA obtienen tiempos de

evaluación relativamente cortos, fortaleciendo el objetivo inicial del enfoque de evaluación visual. Al mismo tiempo pudo identificarse un incremento en el tiempo de evaluación para procesos de tamaño considerable, por el proceso de observación y comentarios del evaluador se puede relacionar este incremento a la dificultad de interactuar con los elementos por su despliegue en el Blueprint y el método de registro para los datos de la prueba.

Los patrones de error favorecen enormemente el detectar comportamientos o malas prácticas que pueden llevar a que el proceso no culmine de forma exitosa, pero su uso se limita solo a identificar problemas específicos del proceso. La evaluación realizada con AVISPA y vAVISPA detecto comportamientos similares en la evaluación de los patrones enfocados a detectar problemas de comportamiento individual, mientras que un comportamiento colectivo del modelo, es mayor el impacto por el proceso de variabilidad, permitiendo que el análisis efectuado por vAVISPA resulte más efectivo para la detección de problema como sub-proyectos o elementos desconectados del proyecto.

La cuantificación de efectividad es realizada bajo un sistema de comparación entre errores detectados por los patrones, contra errores detectados por el análisis del proceso. Para que esta comparativa pueda realizarse optamos por que el evaluador solo basara su criterio de identificación del error, a lo resaltado por cada uno de los patrones tanto en AVISPA como para vAVISPA. A su vez teniendo en cuenta que el modelo de proceso era el mismo para ambos casos, se desarrolla el proceso de análisis determinando el porqué de los errores detectados al finalizar las dos evaluaciones, descartando de esta forma sugerir el criterio de evaluación por los datos del análisis.

Algunos de los resultados generados a partir del análisis de los errores detectados y el contraste entre los Blueprints generados para cada uno de los mecanismos de evaluación se describen a continuación:

**Patrón sub proyecto:** observamos una efectividad más marcada del proceso, en la detección de errores en el patrón de sub proyectos. Al analizar el modelo AVISPA por detección de colores del patrón de sub proyecto a nivel de tareas que se observa en la Fig. 40, identificamos 16 proyecto (2 proyectos y 14 elementos), en el proceso de análisis es detectado que los elementos desconectados corresponden a los elementos que son definidos en el method content pero que no son usados para el proceso, mientras que el segundo sub proyecto que puede identificars corresponde a un caso de variabilidad no resuelto, la cual desconecta dos tareas del proceso principal.

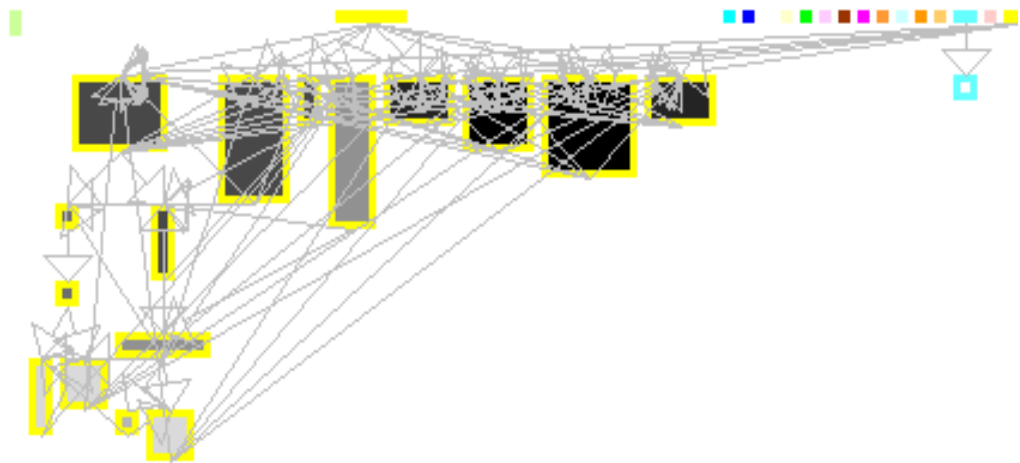


Fig. 40 Patrón sub proyecto (Blueprint Tareas) AVISPA

Mientras que en el caso de vAVISPA estos casos son identificados por el proceso de variabilidad y por tanto se depuran del patrón para la identificación de errores. Generando un Blueprint que se observa en la Fig. 41, que es más acorde a la definición del proceso que el obtenido por AVISPA.

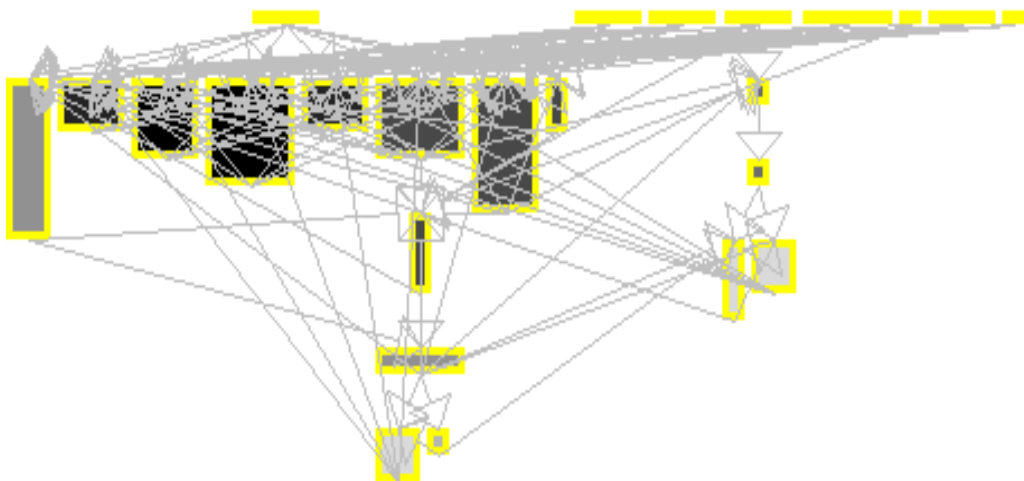


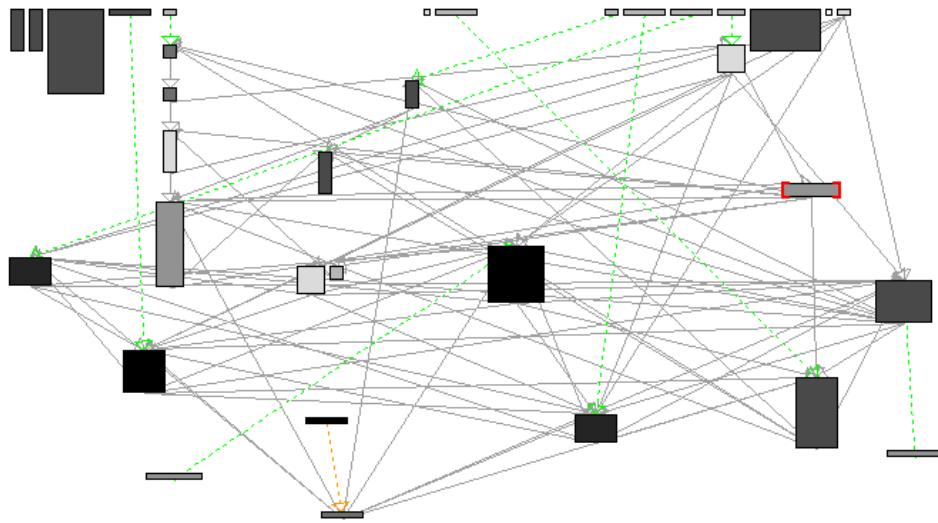
Fig. 41 Patrón sub proyecto (Blueprint Tareas) vAVISPA

**Patrón multipropósito Tareas:** En cuanto al patrón multipropósito son detectados mayor número de casos por parte de AVISPA que por parte de vAVISPA, pero en ningún momento esto estableció un criterio para determinar que herramienta había desarrollado un análisis más efectivo del proceso para ello desarrollamos un análisis caso a caso de los elementos detectados, estableciendo cuales son los que realmente afectan al proceso. Los elementos detectados por cada uno de los casos de evaluación se resumen en la siguiente tabla.

| Tareas multipropósito detectadas AVISPA |                  | Tareas multipropósito detectadas vAVISPA |                  |
|---|------------------|--|------------------|
| Elemento                                | Conclusión       | Elemento                                 | Conclusión       |
| define_vision                           | Falso Positivo   | manage_iteration                         | Error confirmado |
| find_and_outline_requirements           | Error confirmado | find_and_outline_requirements            | Error confirmado |
| implement_solution                      | Error confirmado | implement_solution                       | Error confirmado |
| manage_iteration                        | Error confirmado | dsdm_manage_iteration                    | Error confirmado |
| plan_the project                        | Falso Positivo   |  |                  |
| assess_result                           | Falso Positivo   |  |                  |
| detail_requirements                     | Falso Positivo   |  |                  |
| plan_iteration                          | Falso Positivo   |  |                  |

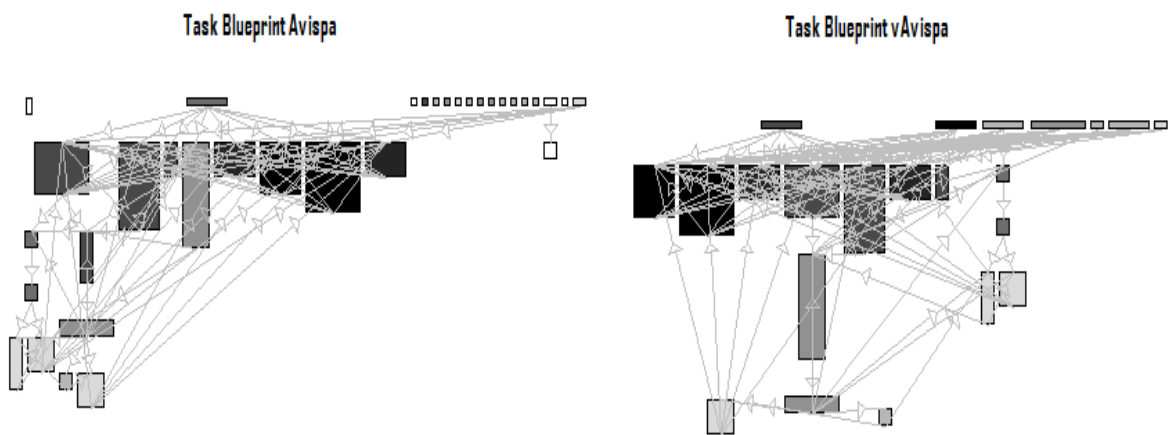
**Tabla 11 Lista de Errores detectados en el proceso de evaluación patrón multipropósito tarea**

El primer análisis es realizado sobre los casos detectados por AVISPA donde por medio del Blueprint de variabilidad que se observa en la Fig. 43, detectamos que ocho de los elementos detectados como error siete de ellos presentan una relación de variabilidad que afecta su comportamiento, lo cual genero una primera idea para analizar por qué son descartados por vAVISPA. El primero en analizarse es la tarea “define\_vision” en la cual se detectó una relación de variabilidad tipo extends-replace, en la cual especifica que el elemento al final de la resolución será reemplazado con una nueva tarea los cual explica por qué resulta ser descartado por vAVISPA.



**Fig. 42 Task Variability Blueprint**

De los seis elementos restantes con variabilidad, se detecta que todos presentan variabilidad de tipo extends, como resultado de este tipo de variabilidad elementos que producían pocos artefactos en el proceso empiezan a generar mayor cantidad de entregables. Teniendo en cuenta que para el desarrollo del patrón esta planteado que la identificación de los elementos es realizada a partir del cálculo de la desviación estándar del número de salidas de todas las tareas, este valor es afectado por los aportes de las tareas resultantes del proceso de resolución de variabilidad, ampliando el margen del criterio para identificar si una tarea es o no multipropósito. Dando como resultado que gran parte de las tareas identificadas por AVISPA sean descartadas por vAVISPA y a su vez que otras tareas que no se consideraban por el patrón como posibles errores, ahora si lo sean debido a los aportes obtenidos por el proceso de variabilidad. En la Fig. 40 podemos observar como aumenta el número de productos de trabajo generado por algunas tareas.



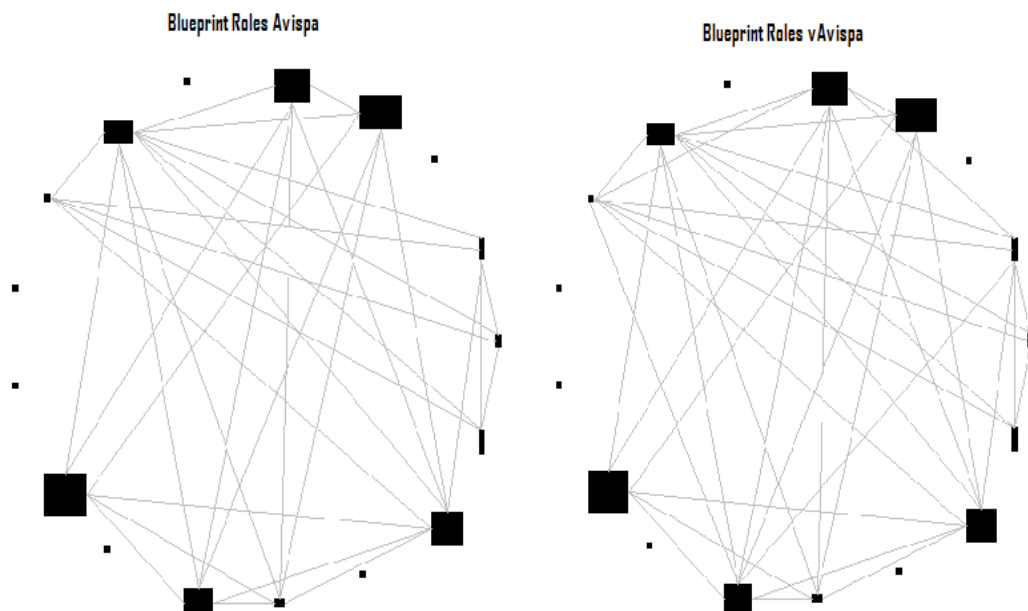
**Fig. 43 Task Blueprints**

## **Análisis visual de los procesos**

La identificación por medio de patrones resulta un proceso que facilita detectar problemas particulares del proceso pero solo esta enfoca en analizar algunas fases específicas del proceso por lo que requiere de una evaluación más afondo para detectar anomalías particulares. Para este proceso se realizó una evaluación del Blueprint entre el evaluador y el responsable del estudio de caso buscando identificar posibles falencias en el modelo o cambios debido a la incorporación de variabilidad.

### **OpenUP**

Al analizar el Blueprints de roles de OpenUP que se observan en la Fig. 45, son identificados varios roles que no interactúan con los demás, una revisión particular detecta que algunos de los roles observados en el Blueprint pertenecen a adaptaciones del proceso OpenUP, en el caso del modelo de proceso utilizado para la prueba este incorpora elemento del DSDM (Dynamic System Development Method) y algunos otros roles que se logran identificar si presentan este tipo de comportamiento, Por lo cual vale la pena analizar en que circunstancias seria apropiada un rol que no trabaje con el resto de participantes del proyecto.



**Fig. 44 Blueprints Roles**

Otro aspecto que vale la pena resaltar fue detectado al realizar una comparativa del Blueprint de AVISPA y vAVISPA, donde identificamos dos nuevas relaciones de colaboración entre roles. Una que corresponde entre el visionary y los stakeholder, y la otra colaboración identificada se da entre exec\_sponsor y los stakeholder. Presumiendo que este tipo de colaboraciones aumenta la efectividad del proceso dada la mejor



comunicación entre lo que se está buscando con el proyecto y los participantes del mismo.

En el caso del Blueprints de artefactos que se observa en la Fig. 46, solo identificamos la incorporación de dos nuevos elementos a la interacción del conjunto total de artefactos, estos elementos corresponde a artefactos del uc\_model que modifican la interacciones de los demás elementos. Tiendo en cuenta que uno de los cambios en precedencia se da en que el uc\_model antecede a los use\_cases estimamos que el proceso estará afectado en su desarrollo desde su inicio debido a la incorporación de la variabilidad y la precedencia de productos de trabajo necesarios para el proceso.

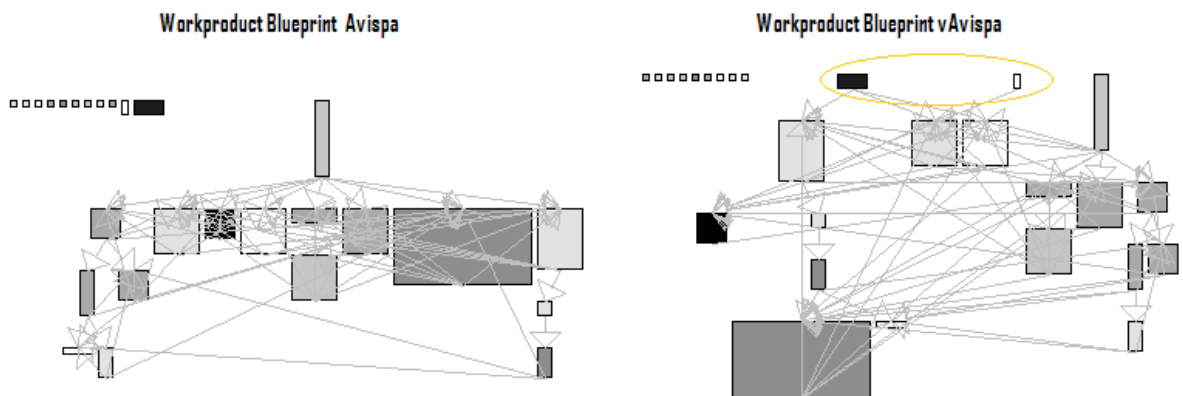
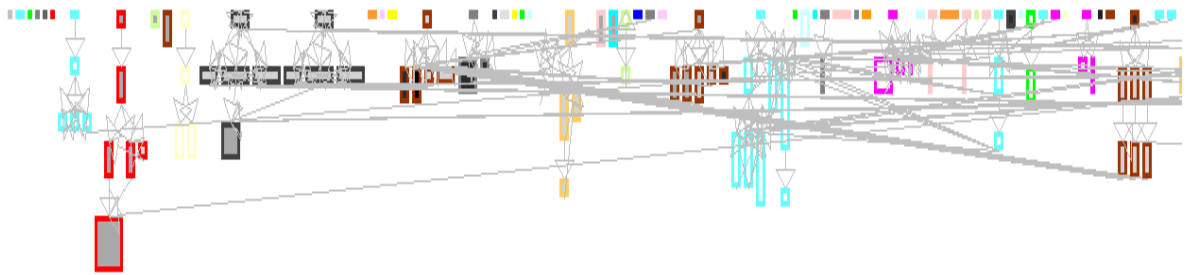


Fig. 45 Blueprints Work Product

## Remics

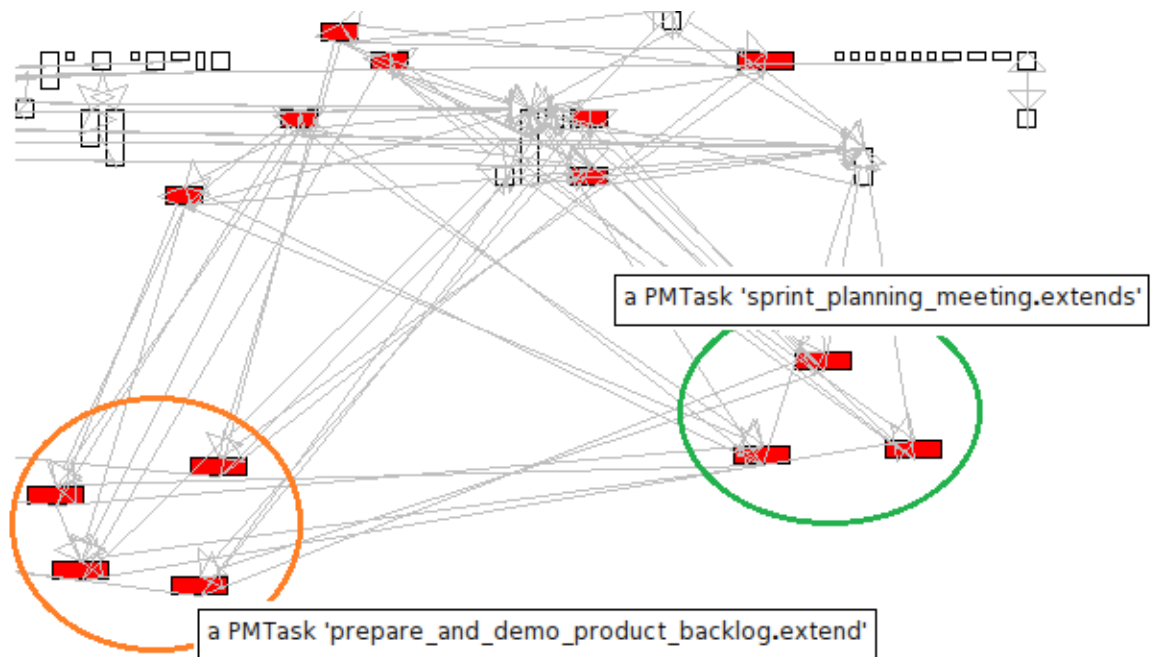
En cuanto a Remics se presentaron dificultades al momento de registrar los datos debido al gran número de errores detectados y a la dificultad de poder identificar algunos elementos debido a la gran cantidad de arcos que dificultaban su visibilidad e interacción con dichos elementos. Pero a pesar de este percance se optó por realizar un proceso de análisis de los Blueprint para identificar qué factores pudieron generar problemas en la definición del proceso, los resultados obtenidos son registrados en esta sección

Una de las grandes dificultades al analizar Remics es debido a la gran cantidad de elementos que contenía y la gran cantidad de errores que fueron detectados en el proceso de análisis, lo cual dificultaba el registro y proceso de verificación caso por caso, con el fin de determinar a que se debían estos problemas. Uno de los ejemplos que más sorprende en el proceso de evaluación corresponde con el análisis del patrón de sub proyectos, dada la cantidad de elementos sueltos que son detectados y la dificultad para poder identificar un proceso central entre la gran cantidad de subprocesos que pudieron identificarse como se observa en un fragmento del Blueprint resultante en la Fig. 47.



**Fig. 46 Fragmento Blueprint Independent Subproject Pattern**

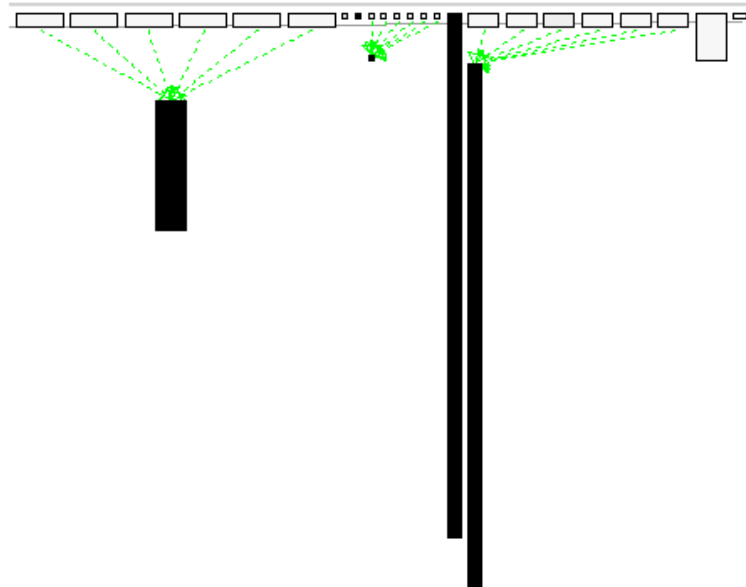
Entre las particularidades detectadas mientras era evaluado el proceso, la identificación de múltiples elementos que presentaban error y que contaban con el mismo nombre e igual estructura en un mismo Blueprint (ver Fig. 44.), llamo la atención tanto del evaluador como el encargado de desarrollar el estudio de caso, dada la nomenclatura encontrada en los nombre de los elementos (.extends) se analiza si este comportamiento es debido a un problema en la solución de variabilidad o que lo provoca.



**Fig. 47 Fragmento Task Blueprint - Presencia de múltiples objetos similares**

Con la ayuda del bluerpint de variabilidad (Ver Fig. 45), se detectó que la presencia de todos estos elementos es debido a implementaciones de variabilidad sobre un mismo elemento, lo cual genera la aparición de esto componentes en un mismo modelo de proceso. Queda como interrogante el propósito de disponer en un mismo modelo de proceso de un elemento (Task) repetido varias ocasiones, ya que estos elementos pueden generar sobrecarga de labor al realizarse el mismo trabajo repetidamente. Otra de

las posibilidades que se plantea es la posibilidad que sea planteada como un plugins diferente dependiendo del tipo de proyecto, pero que por inexperiencia del diseñador del proceso hayan terminado todos incorporados en el mismo proceso, solo se podría determinar con exactitud que genera este problema realizando el análisis con el diseñador. Pero este es un claro ejemplo de lo ligado que resulta el proceso a los criterios del diseñador o el daño que ocasiona una mala especificación de variabilidad a un modelo de proceso.



**Fig. 48 Fragmento Task Variability Blueprint**

Entre los pendientes que quedaron en el análisis del modelo de proceso Remics está el analizar la gran cantidad de elementos denominados residuos que son identificados por medio del patrón Waste (ver Fig. 50), ya que se generó cierta desconfianza del modelo de proceso dada la especificación de gran cantidad de artefactos sin ningún propósito para el modelo, a pesar de catalogarse a sí misma como una metodología ágil para para migración de servicios a la nube.



**Fig. 49 Fragmento Workproduct Waste Pattern**

#### **4.9. Amenazas de Validez**

El resultado de los datos cuantitativos, cualitativos y de análisis obtenidos en el estudio de caso, no representan ninguna generalización para todos los procesos, como ha sido especificado desde los inicios de esta propuesta el modelo de proceso se encuentra estrechamente ligado al contexto organizacional, Experticia del diseñador, tipo de proyecto, entre muchos otros factores que impiden generalizar una evaluación global. También el limitado número de procesos con los que dispuso la realización de la evaluación limitan el alcance de los resultados obtenidos. Pero a pesar de todos estos factores se establece una situación comparativa entre los enfoques visuales de AVISPA y vAVISPA que permiten afirmar que el proceso de evaluación incorporado a los métodos de variabilidad dispuesto por SPEM 2.0 mejoran sustancialmente el análisis de los procesos depurando los Blueprint de ciertas anomalías y permitiendo que el evaluador concentre sus esfuerzos en identificar las posibles causas de problemas en la definición del modelo.

En cuanto al tiempo el enfoque de análisis visual presenta mejores resultados en comparación con otro tipo de enfoques y aunque no es desarrollado un análisis de tiempo para evaluar el proceso con otro tipo de enfoque, se puede identificar esta mejora dada la capacidad de experticia que fue requerida para evaluar el modelo, ya que solo con una pequeña capacitación el enfoque y descripción de la representación de los Blueprints el evaluador pudo obtener conclusiones significativas de deficiencias en el modelo de proceso.

Otro de los factores a discutir de la propuesta de análisis visual de variabilidad radica en su efectividad para ser utilizado en modelos de gran tamaño, dadas las dificultades presentadas para registrar los datos con el modelo Remics, estas no representaron de ninguna manera un problema que evitara realizar el análisis del proceso identificando algunos problemas que contenía la definición del modelo de proceso. Aunque si representa un factor a considerar la usabilidad obtenida debido al despliegue de la información, ya que se detectaron algunas dificultades para interactuar con los Blueprints.

#### **4.10. Síntesis y Discusión**

Tanto para un evaluador como para un diseñador de proceso vAVISPA muestra ser una herramienta de gran utilidad que permite realizar una depuración efectiva para detectar anomalías o inconsistencias en los procesos que incorporen variabilidad en su definición. La adaptación de los patrones que se planteaban en la propuesta AVISPA para su utilización en el enfoque de análisis visual de variabilidad permite de manera ciertamente sencilla concentrar los esfuerzos del evaluador en los problemas base que pueden tener los procesos.

El desarrollo del prototipo de vAVISPA extendido a los modelos de procesos definidos bajo el estándar SPEM 2.0 permiten cubrir un amplio margen de modelos de proceso formalmente definidos gracias a la acogida y soporte que posee este lenguaje. Aunque también se debe tener en cuenta que el evaluador del proceso requiere de una serie de conceptos, experiencias con respecto a variabilidad y modelos de proceso que le permitan hacer un seguimiento más efectivo del desarrollo de la variabilidad así como su impacto en el modelo de proceso en general. Otro de los factores a tener en cuenta para el análisis más efectivo del proceso radica en su contexto desde el inicio de la propuesta se ha recalcado en la estrecha relación del contexto organizacional y factores específicos que pueden afectar el desarrollo de un modelo de proceso por lo que requiere conocer con más exactitud los motivos de diseño para algunos de los casos que se pueden detectar al momento de desarrollar el análisis visual con vAVISPA.

En cuanto al análisis de la correctitud de los modelos de procesos vAVISPA resulta ser una herramienta efectiva para la identificación de problemáticas relacionadas a la definición del proceso así como a la definición de variabilidad dentro del proceso, con tiempos relativamente cortos y un enfoque que permite identificar ciertas anomalías de forma rápida y sencilla mejorando sustancialmente la efectividad de la evaluación permitiendo focalizarse en problemas evidentes dentro del proceso. La adaptación de los Blueprints y patrones planteados en el enfoque visual inicial dada la implementación de variabilidad de los modelos resulta ser bastante efectiva para depurar el proceso de evaluación de elementos que no representan riesgo para el proceso, de esta manera favorece el análisis que se realiza al momento de evaluar un proceso, así como también favorece el proceso de análisis el Blueprint de variabilidad el cual permite al evaluador hacer seguimiento de forma sencilla y efectiva a los diferentes mecanismos de variabilidad que dispone el modelo de proceso, aunque la efectividad de la evaluación también requerirá conocimientos de los mecanismos de variabilidad establecidos en SPEM 2.0 que le permitan interpretar correctamente como el modelo de proceso se ha afectado por la incorporación de variabilidad en la definición del proceso.

Aunque por factores de usabilidad es necesario evaluar la efectividad del prototipo vAVISPA en el despliegue de procesos de gran tamaño, de ninguna manera representa esto un efecto que limite el análisis del modelo de proceso, debe tenerse en cuenta que la sobrecarga de información en algunos Blueprints depende mucho de la cantidad de errores que puedan ser identificados tras el proceso de análisis. Pero para mejora de la experiencia del usuario se debe mejorar el proceso de despliegue de los Blueprints que mejore aún más la experiencia del evaluador con la herramienta vAVISPA.

## **5. CONCLUSIONES, LIMITACIONES Y TRABAJO FUTURO.**

Los modelos de procesos no pueden ser estáticos. El área de la ingeniería es una de las pocas áreas de conocimiento que evoluciona constantemente y de manera vertiginosa, estos cambios se ven reflejados en la forma en que diseñamos nuestros procesos. Las organizaciones cada vez toman mayor conciencia de cómo deben desarrollar sus procesos de manera formal para poder tener éxito y perdurar con un modelo de negocio en un ambiente tan competitivo y cambiante como lo es el de la industria del software. La definición de modelos adaptables y desarrollados a la medida del contexto de la empresa favorecen y establecen una identidad organizacional que le permiten sobresalir frente a sus competidores, el implementar estrategias efectivas de desarrollo y registro de los conocimientos que adquiere una organización en proceso de desarrollo de un producto son invaluable para el crecimiento y fortalecimiento de la organización. Pero la complejidad que conlleva desarrollar estos modelos de procesos adaptables para aprovechar efectivamente los recursos que posee una organización hacen visible una problemática existente en el área de ingeniería software, el requerimiento de estándares, y herramientas de evaluación de proceso de forma más eficiente para las organizaciones genera un ambiente de rechazo por parte de la industria que no cuentan con los recursos para poder establecer sus procesos.

En la actualidad la definición y evaluación de modelos de proceso se encuentra rezagada entre las áreas de investigación de ingeniería software, el desarrollo de modelos de procesos cuenta con pocos mecanismos de validación para la generación de procesos adaptables y algunos de los mecanismos de evaluación de procesos que se tienen disponibles resultan poco accesibles a la pequeña y mediana empresa debido a los requerimientos económicos o históricos que se necesitan para ser implementados. La propuesta vAVISPA responde a cubrir de cierta manera estas necesidades proponiendo un modelo efectivo para la depuración de errores y complementario para otros modelos de validación, siendo un modelo de bajo costo y relativamente sencillo para ser implementado en la evaluación de modelos de proceso que incorporen variabilidad en su definición. También es de resaltar las bases sólidas sobre las cuales se desarrollan los modelos de proceso que puede analizar vAVISPA dado que el estándar SPEM 2.0 es un lenguaje de modelado de proceso altamente acogido por la industria, su capacidad de adaptarse a cualquier tipo de ciclo de vida de desarrollo favorecen su uso por diferentes tipos de organizaciones y adicional cuenta con el soporte de una gran comunidad como lo es la de EPF composer, que aparte del conocimiento que aporta también cuenta con una herramienta de desarrollo de modelos bastante robusta que permite especificar los modelos de proceso SPEM 2.0

### **5.1. Conclusiones**

La capacidad que adquiere el análisis visual en los modelos de proceso ha venido demostrando su efectividad y aporte al momento de validación de los procesos dando sus

primeros pasos con AVISPA, una herramienta desarrollada para el análisis de la correctitud de modelos pero limitada a procesos no variables, AVIMO PS que apporto en el análisis de la modularidad en los procesos, permitiendo entender de manera más efectiva la distribución y cohesión de los elementos del contenido del modelo y ahora en esta propuesta que se enfoca en extender el análisis visual a modelos de procesos adaptables que contienen mecanismos de variabilidad en su especificación. Dando soporte a la generación de modelos más efectivos a partir de especificaciones ya establecidas, con la finalidad de obtener día tras día mejores especificación en el desarrollo software y fortaleciendo el área de ingeniería software. Para ello son adaptados los procesos de evaluación existentes a un enfoque diseñado en validar y dar seguimiento a los impactos que pueden generarse en el modelo de proceso debido a incorporación de variabilidad. Con esta propuesta buscamos seguir dando un nuevo avance en el enfoque de análisis visual sobre el cual aún queda todo un universo de conocimiento que puede ser aprovechado.

Mediante el estudio de caso analizamos la efectividad de vAVISPA para identificar problemáticas del modelo de proceso en cuanto a variabilidad se refiere, el desarrolló de un estudio de caso enfocado a analizar procesos variables con los patrones adaptados por vAVISPA para la evaluación de modelos de procesos, aunque presento dificultades para la obtención de métricas para uno de los modelos utilizados, de ninguna manera afecto el proceso de evaluación, si no que permitió aprovechar el enfoque visual para detectar las anomalías que no son clasificadas dentro de los patrones de error establecidos pero que pueden ser identificas para su posterior seguimiento con los Blueprints que dispone vAVISPA de esta forma se demuestra la utilidad que tiene el enfoque y la herramienta en el proceso de análisis de correctitud e impacto que trae la incorporación de variabilidad en un modelo de proceso.

El prototipo vAVISPA demostró su efectividad de análisis en modelos de procesos que contienen variabilidad, a pesar de algunas dificultades en cuanto a visualización de procesos de gran tamaño, se pudo establecer la efectividad del prototipo en el análisis e identificación de los mecanismos de variabilidad establecidos en el meta-modelo SPEM 2.0 dando soporte a nuevos procesos de evaluación que permitirán avances en la industria del desarrollo software y dan un aporte en el área de ingeniería de procesos. La eficiencia demostrada en el prototipo vAVISPA para el proceso de evaluación de modelos de proceso da un aporte significativo en cuanto a análisis se refiere dada la poca exploración que se ha desarrollado en el campo investigativo para el análisis de variabilidad en los modelos de proceso.

## **5.2. Limitaciones**

Cabe notar que el enfoque está limitado solo a procesos SPEM 2.0 dado sus mecanismos para implementar variabilidad en modelos de proceso este puede catalogarse como el

primer elemento limitante dentro de la aplicabilidad de la propuesta vAVISPA debido a la alta complejidad que contiene el meta-modelo para su utilización sin contar con la experticia requerida por parte del diseñador para desarrollar un modelo de proceso que implemente efectivamente procesos de adaptación. Adicional a este limitante también es debe contar con que el proceso debe estar en formato XML que se puede generar utilizando la herramienta EPF composer. Otro de los factores que no es limitante si no un poco restrictivo consiste en el tamaño del proceso, el análisis de un proceso considerablemente grande presenta algunas dificultades para el despliegue de información que se puede generar en los Blueprints lo que limitaría el análisis del proceso.

Otro de los limitantes lo encontramos en el análisis del method content del modelo de proceso, que aunque se realizó un análisis superficial de los elementos que componen el method content para ser utilizados en el proceso de análisis, este solo se realizó sobre los elementos de tipo tareas dada la complejidad e investigación necesario para determinar patrones y métricas que permitan establecer un proceso de análisis lo suficientemente sólido para ser presentado, lo cual superaba los objetivos y tiempos planteados para esta propuesta.

### **5.3. Lecciones Aprendidas y problemas enfrentados**

El componente técnico consiste en factor determinante para el desarrollo de la propuesta, desde el inicio del planteamiento del anteproyecto estaba estimado que debíamos contar con un tiempo prudencial para adaptarse al enfoque y herramienta de desarrollo que se iba a trabajar. Esta estimación presento problemas para el cronograma de desarrollo de la propuesta dado que la curva de aprendizaje fue un poco lenta lo que ocasiono retrasos en el desarrollo de este trabajo. Así mismo la experiencia y conocimientos sobre modelos de procesos que disponía al momento de iniciar el proyecto eran relativamente escasos, se desarrolló un proceso de aprendizaje de concepto apoyado por el tutor asignado a la tesis fortaleciendo los conocimientos requeridos para poder abordar esta temática.

Otra de las limitaciones que presentadas al abordar el problema de variabilidad se presenta debido a la disponibilidad de modelo de procesos que cumplieran las condiciones para poder realizar el estudio de caso, uno de los modelos de proceso con que se contaba para realizar el análisis era la definición del modelo protest un modelo de proceso que corresponde a la industria, pero que al realizar el proceso de validación de los requerimientos, identificamos que no posee el elemento variable en su definición lo cual lo descarta para realizar el estudio de caso.



#### 5.4. Trabajos Futuros

Tomando en cuenta el área de investigación del grupo IDIS en ingeniería software y el trabajo que ha venido siendo desarrollado en el enfoque de análisis visual, se seguirá realizando investigaciones en busca de generar modelos de procesos de mejor calidad. El aprovechamiento de los datos que pueden obtenerse para evaluar del modelo de proceso ha sido limitado, aun cuenta con una gran cantidad de información que contribuiría a la meta de mejorar los procesos desarrollados en las empresas. A la vez que identificamos tanto por diferentes autores así como por la experiencia de evaluación que los mecanismos de variabilidad son de cierta forma complejo y poco utilizados por los desarrolladores de procesos, se plantea la necesidad de contar con mecanismos más simples que permitan su generalización en la industria como es el caso de la propuesta vSPeM, o en su defecto contar con metodologías, patrones o diseños que permitan extender los mecanismos de variabilidad ya definidos.

Uno de los trabajos relacionados a desarrollar es un análisis complementario del enfoque de variabilidad en el proceso a nivel de contenido del método la estructuración en paquetes y complementos para la modularidad del proceso que se presentó en AVIMO puede ser analizada para la estructuración de procesos adaptables no desde un enfoque del elemento, sino de plugins para el desarrollo de procesos.

Otra de las propuestas interesantes que surgen de este trabajo consta en analizar la implementación del proceso visual no para el análisis del modelo resultante, sino como un soporte en la estructuración y definición del modelo de proceso. Mientras se realizaba el estudio del estado del arte se encontraron varias propuestas para la estructuración de líneas de producto utilizando un enfoque de nodos y arcos que facilitaba su desarrollo. Utilizando los conocimientos y experiencias aprendidas en el proceso de evaluación de los modelos, extrapolarlo a la estructuración del modelo permitiendo que en tiempo de implementación se puedan detectar errores en la estructuración del proceso.

## 6. Referencias

1. Conradi, R., C. Fernströmy, and A. Fuggettaz, *A Conceptual Framework for Evolving Software Processes*. Software Process Modelling and Technology, 1993.
2. Deming, W.E., *Out of the crisis : quality, productivity and competitive position*1988, New York: Cambridge University Press.
3. Juran, J.M. and A.B. Godfrey, *Juran's Quality Handbook*. 5 ed. McGraw-Hill1999.
4. Simmonds, J., et al., *Modeling Variability in Software Process Models*.
5. Simmonds, J., et al., *Analyzing Methodologies and Tools for Specifying Variability in Software Processes*. 2012.
6. Alegría, J.A.H., et al., *An MDE Approach to Software Process Tailoring*. International Conference on Software and Systems Processes, 2011: p. 43-52.
7. Cockburn, A., *Selecting a Project's Methodology*. IEEE SOFTWARE, 2000: p. 64-71.
8. OMG, O.M.G., *Software & Systems Process Engineering Meta-Model Specification*, 2008.
9. OMG, O.M.G., *Business Process Model and Notation (BPMN)*, 2011. p. 538.
10. Coalition, T.W.M., *XML Process Definition Language V. 2.1*, 2008.
11. Cass, A.G., et al., *Little-JIL/Juliette: a process definition language and interpreter*. International Conference on Software Engineering, 2000. **22**: p. 754 – 757.
12. Ruiz, T.M., et al., *Modelling software proccess variability: an empirical study*. IET Software, 2011. **5**: p. 172-187.
13. Ribó, J.M. and X. Franch, *PROMENADE: A PML Intended to Enhance Standarization, Expressiveness and Modularity in Software Process Modelling*. 2000.
14. Ferreyra, N.E.D., *OVM2PN: Herramienta de análisis de configuraciones en Líneas de Productos de Software*.
15. Alegría, J.A.H., M.C. Bastarrica, and A. Bergel, *Analyzing software process models with AVISPA*. International Conference on Software and Systems Processes, 2011: p. 23-32.
16. Hurtado, J.A.A., et al., *Software Process Model Blueprints*. New Modeling Concepts for Today's Software Processes, 2010. **1**(273-284).
17. Lanza, M. and S. Ducasse, *A categorization of classes based on the visualization of their internal structure: the class blueprint*. Proceeding OOPSLA '01 Proceedings of the 16th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, 2001. **36**(11, 11/01/2001 ): p. 300-311.
18. Mohagheghi, P., V. Dehlen, and T. Neple, *Towards a Tool-Supported Quality Model for Model-Driven Engineering*. Proceedings of the 3rd Workshop on Quality in Modeling, 2009: p. 74-88.
19. Lonchamp, J., *A Structured Conceptual and Terminological Framework for Software Process Engineering*. Second International Conference on the Software Process. Continuous Software Process Improvement, 1993: p. 41- 53
20. Francisco Ruiz, J.V., *Guía de Uso de SPEM 2 con EPF Composer*, U.d.C.-L. Mancha, Editor 2008. p. 93.
21. Tate, J., *Software Process Quality Models: A comparative evaluation*, in *Department of Computer Science*2003, University of Durham.
22. Shaffer, G.R., *Composing And Orchestrating Mission Capability Packages Through Business Process Execution Language*. 2004.

23. Jaccheriy, M.L., G.P. Picco, and P. Lago, *Eliciting software process models with the E3 language*. ACM Transactions on Software Engineering and Methodology (TOSEM), 1998: p. 42.
24. Emmerich, W. and V. Gruhn, *Software process modelling with FUNSOFT nets*. 1995.
25. Bröckers, A., et al., *MVP-L Language Report Version 2* 1995.
26. Marcos, C.L., M. Pantelis, and H.D. Rombach, *A MVP-L Solution for the Software--Process Modeling Problem*. 1991.
27. Bendraou, R., M.-P. Gervais, and X. Blanc, *UML4SPM: A UML2.0-Based Metamodel for Software Process Modelling*. Model Driven Engineering Languages and Systems, 2005: p. 22.
28. OMG, O.M.G. *OMG's MetaObject Facility*. 2012 [cited 2012; Available from: <http://www.omg.org/mof/>].
29. Becker, J., M. Rosemann, and C.v. Uthmann, *Guidelines of Business Process Modeling*, in *Business Process Management2000*. p. 30-49.
30. Min, S.-Y., et al., *An Approach to Software Process Management Based on Formal Process Modeling and Analysis*. Software Engineering Conference, 1997. Asia Pacific. and International Computer Science Conference 1997. APSEC '97 and ICSC '97. Proceedings, 1997: p. 292 - 301.
31. Aalst, W.M.P.v.d., et al., *Preserving correctness during business process model configuration*. Formal Aspects of Computing 2010. **22**(3-4): p. 459-482.
32. Team, C.P., *CMMI for Development, Version 1.3*, 2010, Carnegie Mellon University.
33. Committee, S.S.E.S., *International Standard ISO/IEC 12207 IEEE Std 12207-2008*, 2008.
34. Team, S.U., *Standard CMMI® Appraisal Method for Process Improvement (SCAMPISM ) A, Version 1.2: Method Definition Document*2006.
35. Schwaber, K. and J. Sutherland, *The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game*. 2011.
36. *ISO. /IEC 15504 : Information technology - software process assessment and improvement.*, 1998, Int. Organization for Standardization.
37. Torossi, A.U.S.G., *El Proceso Unificado de Desarrollo de Software*.
38. Pohl, K., G. Böckle, and F.J.v.d. Linden, *Software Product Line Engineering: Foundations, Principles and Techniques*2005.
39. Roos-Frantz, F. and S. Segura, *Automated Analysis of Orthogonal Variability Models. A First Step*.
40. Bosch, J., et al., *Variability Issues in Software Product Lines*. 2002.
41. Crosby, P.B., *Quality is free: the art of making quality certain*1980, New York.
42. Canfora, G., et al., *A family of experiments to validate metrics for software process models*. The Journal of Systems and Software, 2005: p. 113-129.
43. Wangenheim, C.G.v., A. Anacleto, and C.F. Salviano, *MARES - A Methodology for Software Process Assessment in Small Software Companies*. 2004.
44. Gruhn, V., *validation and verification of software process model*. Lecture Notes in Computer Science, 1991. **509**: p. 15.
45. Brockers, A. and V. Gruhn, *Computer-Aided Verification of Software ProcessModel Properties*. Lecture Notes in Computer Science, 1993. **685**: p. 25.
46. Cook, J.E. and A.L. Wolf, *Software process validation: Quantitatively measuring the correspondence of a process to a model*. ACM Transactions on Software Engineering and Methodology (TOSEM) 1999. **8**: p. 147-176.

47. Lee, M.D., R.E. Reilly, and M.A. Butavicius, *An Empirical Evaluation of Chernoff Faces, Star Glyphs, and Spatial Visualizations for Binary Data*. Australian Computer Society, Inc, 2010: p. 10.
48. Batory, D., *Feature Models, Grammars, and Propositional Formulas*. 2005.
49. Alegría, J.A.H., *A META-PROCESS FOR DEFINING ADAPTABLE SOFTWARE PROCESSES*, in *Departamento de ciencias de la computación2012*, Universidad de Chile. p. 172.
50. Black, A.P., et al., *Pharo by Example*. 1 ed2009: Square Bracket Associates, Switzerland.
51. Rathman, C. *Language Notes - Smalltalk*. Available from: <http://www.angelfire.com/tx4/cus/notes/smalltalk.html>.
52. Runeson, P. and M. Höst, *Guidelines for conducting and reporting case study research in software engineering*, 2008.
53. Yin, R.K., *Case study research: Design and methods*. Vol. 5. 2008: SAGE Publications, Inc
54. project, E.P.F. *REMICS Methodology Scope*. Available from: <http://methodology.remics.eu/>.
55. IICT-BAS, SINTEF, and WUT, *REMICS methodology with agile extension, interim release 2012*.
56. Balduino, R., *Introduction to OpenUP (Open Unified Process)*
57. project, E.P.F. *OpenUp*. Available from: <http://epf.eclipse.org/wikis/openup/>.