



## UNNE - UNLP

Universidad Nacional del Nordeste - Facultad de  
Ciencias Exactas y Naturales y Agrimensura

Universidad Nacional de La Plata - Facultad de  
Informática

### “Fundamentos de Visualización de Requerimientos de Software”

Trabajo Final presentado para obtener el grado de Especialista en  
Ingeniería de Software

**Alumno:** Lic. Bruno Sebastián Ghione

Lic. en Sistemas - Estudiante de Maestría en Ingeniería de Software -  
UNNE-UNLP

[brunoghione@yahoo.com.ar](mailto:brunoghione@yahoo.com.ar)

**Director:** Mg. Pablo Thomas

Prof. Adjunto Dedicación Exclusiva. - Facultad de Informática - UNLP

[pthomas@lidi.info.unlp.edu.ar](mailto:pthomas@lidi.info.unlp.edu.ar)

# Índice General

<b>ÍNDICE .....</b>	<b>I</b>
<b>OBJETIVO .....</b>	<b>IV</b>
<b>PREFACIO .....</b>	<b>V</b>
<b>CAPITULO 1 - VISUALIZACIÓN .....</b>	<b>7</b>
1.1 VISUALIZACIÓN .....	8
1.2 APLICACIONES DE LA VISUALIZACIÓN .....	9
1.2.1 GEO-Visualización .....	9
1.2.2 Gráficas Informativas .....	10
1.2.2 Gráficas Informativas .....	10
1.2.3 Educación Mediante Visualización .....	11
1.2.4 Visualización de Productos .....	11
1.2.5 La Visualización Creativa .....	12
1.2.6 La Visualización Científica .....	13
1.2.7 Visualización de Software .....	14
1.3 HISTORIA DE LA VISUALIZACIÓN .....	15
<b>CAPITULO 2 - INGENIERÍA DE REQUERIMIENTOS Y VISUALIZACIÓN ..</b>	<b>20</b>
2.1 INGENIERÍA DE REQUERIMIENTOS DE SOFTWARE .....	20
2.1.1 Procesos de Ingeniería de Requerimientos según Loucopoulos .....	22
2.1.1.1 Elicitación de Requerimientos .....	24
2.1.1.2 Especificación de Requerimientos .....	24
2.1.1.3 Validación de Requerimientos .....	26

2.2 VISUALIZACIÓN DE REQUERIMIENTOS DE SOFTWARE.....	26
--	----

### **CAPITULO 3 - ENFOQUES DE VISUALIZACIÓN DE REQUERIMIENTOS. 32**

3.1 SYSTEMS MODELING LANGUAGE (SYSML) .....	32
3.2 ENFOQUE SIMBÓLICO.....	34
3.2.1 Enfoque Simbólico – Propuesta de Mejora.....	38
3.3 ENFOQUE ICÓNICO .....	39
3.3.1 Enfoque Icónico – Propuesta de Mejora.....	41
3.4 APROXIMACIÓN METAFÓRICA .....	43
3.4.1 Aproximación Metafórica – Propuesta de Mejora.....	45
3.5 FLOW .....	48
3.6 RED SOCIAL CENTRADA EN REQUERIMIENTOS (RCSN).....	51
3.6.1 Red Social Centrada en Requerimientos - Propuesta de Mejora.....	53
3.7 DISPLAY-ACTION-RESPONSE (DAR) .....	54
3.8 VISMATRIX .....	57
3.9 V-VISUALISE .....	60
3.10 CHAINGRAPH .....	62

### **CAPITULO 4 - CLASIFICACIÓN DE TÉCNICAS DE VISUALIZACIÓN DE REQUERIMIENTOS..... 66**

4.1 UNA CLASIFICACIÓN DE VISUALIZACIONES .....	66
4.1.1 Visualización Tabular.....	67
4.1.2 Visualización Relacional.....	67
4.1.3 Visualización Secuencial.....	67
4.1.4 Visualización Jerárquica .....	67
4.1.5 Visualización Cuantitativa o Metafórica .....	68
4.2 ALTERNATIVAS DE CLASIFICACIÓN DE ENFOQUES DE VISUALIZACIÓN .....	68
4.2.1 Clasificación Según a Quien esta Dirigida.....	68

4.2.1 Clasificación Según el Momento de Utilización .....	69
4.2.2 Clasificación Según a Quien esta Dirigida.....	69
<b>CAPITULO 5 - ANÁLISIS COMPARATIVO DE TÉCNICAS DE VISUALIZACIÓN DE REQUERIMIENTOS .....</b>	<b>71</b>
5.1 CUADRO COMPARATIVO .....	73
5.2 CUADRO COMPARATIVO. CONCLUSIONES.....	75
<b>CAPITULO 6 - CONCLUSIONES Y TRABAJOS FUTUROS.....</b>	<b>76</b>
CONCLUSIONES .....	76
TRABAJOS FUTUROS .....	79
<b>ÍNDICE DE FIGURAS .....</b>	<b>80</b>
<b>REFERENCIAS.....</b>	<b>81</b>
<b>ANEXO I: PUBLICACIONES RELACIONADAS .....</b>	<b>86</b>

# Objetivo

La Visualización de Requerimientos es una metodología que ha surgido recientemente. Diferentes técnicas han sido desarrolladas para transformar Requerimientos expresados en forma verbal o escrita en representaciones visuales. Muchas de estas técnicas utilizan gráficos para “visualizar” esos Requerimientos, los cuales logran capturar la atención de los participantes del proceso inicial de desarrollo de software.

Los objetivos de este trabajo son:

- Estudiar los fundamentos de la Visualización de Requerimientos.
- Estudiar los enfoques existentes sobre Visualización de Requerimientos.
- Realizar un análisis comparativo de estos enfoques.
- Analizar posibles mejoras.

# ***Prefacio***

La Ingeniería de Requerimientos es una disciplina relativamente nueva dentro de la Ingeniería de Software, que establece la necesidad de utilizar un proceso que guíe al analista durante la recolección y especificación de requerimientos. Dicho proceso está asistido por diversas técnicas, aunque muchas veces éstas no resultan suficientes.

Por otra parte, la Visualización se ha utilizado para facilitar la comprensión del producto a construir o transmitir en varias ramas de la Ingeniería y en otras ciencias.

La Visualización contribuye a la Ingeniería de Requerimientos ya que puede mejorar la comprensión de los Requerimientos, así como también sirve de apoyo a la negociación y a la definición de prioridades entre los Requerimientos.

Según [1] en los últimos tiempos, la Visualización se ha utilizado principalmente para el desarrollo de tres aspectos:

- transmitir la estructura y la evolución de las relaciones entre un conjunto de Requerimientos de software y otros artefactos,
- apoyar la organización de las necesidades y la gestión del cambio,
- asistir en las sesiones elicitación de Requerimientos y las actividades de análisis.

Sin embargo, a pesar de la popularidad de los lenguajes visuales de modelado, los Requerimientos aún tienden a ser escritos en un formato textual y narrativo [1], [2] y [3].

El presente trabajo muestra el uso de la Visualización en la Ingeniería de Requerimientos, junto a las técnicas más recientemente difundidas en [4], [5], [6] y [55].

Además, es una profundización del trabajo presentado en [7] y se encuentra organizado de la siguiente manera: en el capítulo 1 se detallan algunos conceptos esenciales de Visualización. En el capítulo 2 se detallan conceptos referentes a Requerimientos e Ingeniería de Requerimientos y el nexo existente con la Visualización. A continuación, en el capítulo 3, se presentan los enfoques más representativos de Visualización de Requerimientos. En el capítulo 4 se presenta una clasificación recientemente difundida que agrupa a las distintas técnicas/enfoques existentes. Posteriormente se define un cuadro comparativo de estos enfoques. Finalmente se establecen conclusiones y se definen los posibles trabajos futuros.

# Capítulo 1

## Visualización

---

En muchas ciencias se utilizan técnicas visuales para comprender conceptos difíciles de entender con la mera utilización de las palabras.

Existe una frase popular: "Una imagen vale más que mil palabras", que simplifica la razón por la cual se utiliza en lo cotidiano la Visualización. Además una de las razones principales por las cuales resulta efectiva la utilización de técnicas visuales, es debido a que el sentido más desarrollado por el ser humano es la vista, y por ende, por él se perciben con mayor facilidad los conceptos.

El uso de gráficos y técnicas de animación se ha aplicado con éxito en prácticamente todas las áreas de la informática (por ejemplo en procesamiento de textos, hojas de cálculo, sistemas operativos, interfaces hombre-máquina, entre otros). Esto puede atribuirse a un rápido aumento en las capacidades gráficas de computadoras en general, a la disminución del costo de la tecnología, y también al aumento de la demanda, en todos los aspectos, de estas características por parte de las comunidades de usuarios.

El uso de la tecnología audiovisual permite que los conceptos y la información compleja estén representados en formas más fáciles de entender [8].

En general, las imágenes proporcionan una mejor representación de las estructuras más complejas, ya que se corresponden más estrechamente con los modelos mentales de estas estructuras [9].

A través de la visualización científica, los investigadores en una amplia gama de disciplinas pueden tomar ventaja del hardware y del

software para representar enormes cantidades de datos, que de otro modo sería casi incomprensible [10].

En la Ingeniería del Software se observa a menudo la implementación de métodos de visualización para comprender diseños de software, y recientemente se hace más hincapié en la aplicación de estas técnicas en la Ingeniería de Requerimientos.

En este capítulo se presentan algunos conceptos básicos referidos a este tema, a fin de proponer un marco teórico.

## **1.1 Visualización**

Según [11], se entiende por Visualización como la acción de representar mediante imágenes ópticas fenómenos de otro carácter, o al acto de formar en la mente una imagen visual de un concepto abstracto. También se define como la acción de imaginar con rasgos visibles algo que no se tiene a la vista.

En [1] y [12] se define a la Visualización como el acto de formación de una visión mental (la imagen).

El ser humano utiliza constantemente la visualización para entender situaciones de la vida cotidiana, muchas veces mediante alguna herramienta visual que puede servir de soporte, y otras simplemente con el pensamiento.

La visualización solo necesita de un elemento, la imaginación. Solo con utilizar el pensamiento para imaginar algo, ya se puede “visualizar”.

En [13] se asegura que la historia de la visualización es la de la búsqueda de nuevos artefactos para amplificar la capacidad de conocer, es la historia de la escritura y de los mapas, la historia del conocimiento.

Desde el comienzo de la historia de la humanidad, las imágenes han sido una herramienta para comunicar lo que el hombre hace y piensa en un determinado momento. Se han encontrado incluso pinturas que datan de hace más de 30.000 años, donde la gente de esa época plasmó

interesantes bocetos de la forma en que se organizaban para cazar [14] [15].

Actualmente, la visualización es utilizada a diario prácticamente para todo. Basta con caminar por la calle y se encuentra todo tipo de imágenes visuales, que muchas veces sin mediar palabra escrita, transmiten claramente información. Un ejemplo de esto son las señales de tránsito y los carteles publicitarios.

Hay avisos publicitarios televisivos que solo están compuestos por tan solo imágenes, y muchas veces logran captar la atención de los posibles compradores con mayor facilidad, que si reprodujeran palabras.

## **1.2 Aplicaciones de la Visualización**

La visualización es una herramienta utilizada para comprender y afrontar diferentes tipos de problemas, ya sean los más simples y cotidianos como también los más complejos.

Algunas de las aplicaciones de esta herramienta, que se mencionan en [16] y [17], son por ejemplo la GEOVisualización, Infografías, Educación mediante Visualización, Visualización Creativa, Visualización Científica, Visualización de Software, entre otras. A continuación se describen brevemente las más relevantes.

### **1.2.1 GEO-Visualización (o Visualización Geográfica)**

Se refiere a un conjunto de herramientas y técnicas de apoyo, para el análisis de datos GEO-espaciales a través del uso de la visualización interactiva.

La Visualización Geográfica considerada como una interfase entre la informática, el conocimiento geográfico y el diseño gráfico, se define como el proceso de representación de información sinóptica, con el propósito de reconocer, comunicar e interpretar patrones y estructuras espaciales [18].

La Visualización Geográfica y los Ambientes Virtuales se perfilan como los nuevos paradigmas en el proceso de visualización y comunicación

de información espacial y por ende del proceso de producción cartográfico. Esto ocurre debido a que redimensionan las formas tradicionales de ver y comunicar la información espacial, al incorporar nuevos elementos que permiten una mayor comprensión del espacio, dando una visión integral y sistemática, lo cual significa una ayuda importante a los usuarios en el proceso de toma de decisiones [18].

Los tradicionales mapas estáticos tienen una capacidad exploratoria limitada, pero la GEO-visualización permite mapas más interactivos, incluyendo la capacidad de explorar diferentes capas del mapa, para acercar o alejar, y para cambiar la apariencia visual del mapa, por lo general en una pantalla de computadora.

Los mapas son medios preponderantes para el almacenamiento y comunicación de información sobre la localización y caracterización del mundo natural, de la sociedad y la cultura. A través de los mapas se reconoce la distribución espacial y las relaciones espaciales, ya que hacen posible la visualización y conceptualización de los modelos y procesos que operan en el espacio [18].

### **1.2.2 Gráficas Informativas o Infografías**

Son representaciones visuales de información, datos o conocimientos. Estos gráficos se utilizan cuando la información a explicar es compleja y se desea hacerlo en forma rápida y clara. Es utilizada a menudo en croquis de mapas, periódicos y en la educación vial. También se utilizan ampliamente como instrumentos por profesionales informáticos, matemáticos y estadísticos para facilitar el proceso de desarrollar y comunicar la información conceptual.

Actualmente la información de gráficos rodea los medios de comunicación, en las obras publicadas, señales de tránsito, entre otros. Ilustran la información que sería difícil de transmitir en forma de texto.

En los periódicos, la infografía se utiliza habitualmente para mostrar el clima, así como mapas y planos de instalaciones para eventos de interés periodístico, y gráficos de datos estadísticos.

Los mapas modernos, especialmente los mapas de rutas para los sistemas de tránsito, utilizan técnicas de infografía para integrar una variedad de información, tales como el diseño conceptual de la red de tránsito, puntos de transferencia y puntos de interés locales.

Las señales de tráfico y otras señales públicas dependen en gran medida de la gráfica de la información, los iconos y emblemas para representar conceptos tales como rendimiento, precaución y sentido de la circulación. Lugares públicos como terminales de tránsito suelen tener algún tipo de sistema integrado de señalización.

### ***1.2.3 Educación Mediante Visualización***

En la Educación suelen utilizarse imágenes visuales para enseñar. Esto es muy útil cuando se enseña acerca de un tema que es difícil de ver de otra manera; un ejemplo de este caso es la estructura atómica, porque los átomos son demasiado pequeños para ser estudiados fácilmente. También se puede utilizar para ver los eventos del pasado, como los dinosaurios, o ver las cosas que son difíciles o frágiles de ver en la realidad, como por ejemplo el esqueleto humano.

### ***1.2.4 Visualización de Productos***

Consiste en técnicas de visualización para modelar distintos tipos de productos, antes de ser fabricados. Es muy común en la industria automotriz. Se utiliza software especial para la visualización y manipulación de modelos en 3D, dibujo técnico y otros documentos relacionados con los componentes fabricados y ensamblados de los productos. Proporciona altos niveles de realismo fotográfico para que un producto pueda ser visto antes de que sea realmente manufacturado. Esto es compatible con funciones que van desde el diseño y el estilo de ventas y marketing.

Originalmente estos dibujos técnicos se hacían a mano, pero con el auge de la tecnología informática, la mesa de dibujo ha sido sustituida por el diseño asistido por computadoras (CAD). Los dibujos y modelos de tipo CAD tienen varias ventajas sobre los dibujos hechos a mano, tales como la posibilidad de modelar en 3-D, realizar prototipos y simulación, todo en menor tiempo y con mayor calidad.

### **1.2.5 La Visualización Creativa**

Es una técnica psicológica para alcanzar una condición emocional deseada a través de imaginar algo concreto.

La visualización creativa es la técnica de utilizar la propia imaginación para crear lo que se desea en la vida.

Esta técnica, básicamente consiste en imaginar los pasos y procedimientos que se deben realizar para alcanzar un objetivo concreto. Por ejemplo un deportista, se imagina los movimientos precisos que debe realizar para ganar una competencia.

No hay nada absolutamente nuevo, extraño o desusado en la visualización creativa. Es utilizada todos los días, en cada momento. Es de naturaleza humana la capacidad de imaginación, la energía creativa básica del universo que utiliza el hombre constantemente, aunque no sea consciente de ello [19].

Existe la realidad objetiva, que es la que sucede en el ámbito externo: las condiciones y estímulos que llegan a través de los cinco sentidos; y la realidad subjetiva, que es la que se da únicamente dentro del individuo.

Es la realidad subjetiva la que rige la conducta humana, es decir, la realidad que sucede dentro del ser humano. La explicación de este fenómeno reside en que el cerebro no distingue entre un acontecimiento real y un acontecimiento imaginado.

En [20] se explica cómo funciona el cerebro: científicamente, el cerebro es el "ordenador central". Controla todas las funciones del cuerpo, tanto las conscientes (por ejemplo caminar, correr o leer) como las inconscientes (por ejemplo la respiración, los latidos del corazón, la digestión, etc.).

Cuando sucede algo, el cerebro da las órdenes pertinentes al cuerpo para responder adecuadamente a lo que esta sucediendo.

Esto pasa tanto cuando el suceso es objetivo como subjetivo: cuando el individuo imagina que algo va mal, el cerebro ordena al cuerpo

una respuesta negativa. Cuando, por el contrario, el individuo elabora en su imaginación algo bueno, el cerebro ordena al cuerpo una respuesta positiva.

Pero más allá de una respuesta física, el cerebro programa una respuesta psicológica. De acuerdo con la información que tiene, el cerebro programa una pauta de conducta: la persona se comporta de una manera o de otra, según sea el caso y, según el comportamiento, se obtienen los resultados deseados.

Esa es la importancia y el "secreto" de la visualización: al crear una realidad subjetiva, el cerebro programa la pauta de conducta adecuada, y esta pauta lleva a los resultados esperados o deseados.

De la persona depende que esta realidad que crea sea la correcta o la que más lo beneficie.

Si el individuo visualiza salud, prosperidad, energía o felicidad, es lo que obtendrá.

Toda persona tiene básicamente en su interior lo que precisa para poder cambiar sus estructuras negativas. Todo lo que un individuo realmente quiera cambiar o corregir lo puede hacer.

### **1.2.6 La Visualización Científica**

Esta disciplina se dedica a la transformación de datos científicos (pero abstractos) en imágenes. Un ejemplo concreto podría ser un diagrama de barras para representar el resultado de una encuesta.

La Visualización Científica se define como el uso de tecnología informática compleja para crear imágenes visuales, a fin de facilitar la comprensión y resolución de problemas [18].

La ciencia siempre ha tratado de entender los fenómenos de la naturaleza, pero sin embargo, estos fenómenos son a veces muy grandes o muy pequeños, muy rápidos o muy lentos para ser estudiados con los métodos tradicionales de laboratorio. La Visualización Científica es una herramienta que permite analizar, entender y comunicar los datos numéricos generados durante una investigación [17].

La Visualización Científica se basa en el uso de imágenes. En palabras simples la visualización científica consiste en la transformación de datos o información en imágenes para explicar y comunicar ideas [14]. Su propósito es promover un nivel más profundo de los datos que se están investigando, y otorgar mayor profundidad de los procesos [17].

Según [17], la visualización científica tiene dos áreas específicas: La Visualización de Volúmenes y la Visualización de Flujos.

La Visualización de Volúmenes se refiere básicamente a los campos escalares. Va desde los exámenes de datos científicos a la reconstrucción de datos dispersos y la representación de objetos geométricos sin una descripción matemática de su superficie.

La Visualización de Flujos se utiliza en general para la visualización de sistemas dinámicos, es decir, en los sistemas donde hay involucrado datos que evolucionan con el pasar del tiempo.

### **1.2.7 Visualización de Software**

La Visualización de Software comprende la visualización de algoritmos y programas [17].

Muchos sistemas tienen millones de líneas de código escritas por decenas de programadores. Tales sistemas son muy difíciles de aprender en su totalidad, situación los hace candidatos excelentes para la visualización, ya que en dichos sistemas interesa encontrar patrones, como por ejemplo de complejidad o de ocurrencia de errores, que permitan tomar decisiones sobre qué partes modificar o cuáles presentan mayor problemática.

Según [13], existen programas de software especiales para afrontar este tipo de problemáticas. Un ejemplo de ello es la herramienta Seesoft desarrollada por los Bell Labs para visualizar software, el cual se apoya en cuatro ideas básicas:

- Representación reducida: cada archivo de programa se representa mediante un rectángulo de base fija y altura proporcional al número de líneas que contiene. Las líneas

de código se representan mediante líneas horizontales dentro del rectángulo, proporcionales a su longitud y tabulación como lo están en el programa.

- Coloreado por estadística: cada línea de código exhibe un color determinado por una estadística asociada a dicha línea. Por ejemplo si es una línea que ha cambiado recientemente, que ha producido errores en el pasado o que se ha ejecutado n veces. Basta seleccionar una estadística, es decir un color, para que rápidamente se resalte en todos los archivos cuyas líneas tienen esa estadística asociada.
- Manipulación directa: moviendo el cursor sobre porciones de la pantalla se produce la consulta sobre la base de datos que contiene el código, y se iluminan con el color apropiado las líneas que la resuelven.
- Capacidad de leer el código real: para ver el código basta abrir una ventana de lectura y colocar un cuadrado de aumento sobre la representación del mismo.

Otro ejemplo de esto es la compañía aiSee que ha desarrollado un sistema, aiCall, que permite básicamente visualizar código escrito en lenguaje C, lo cual ayuda a los programadores a entender, mantener, optimizar, documentar, revisar y hacer reingeniería de sistemas de software complejos [17]

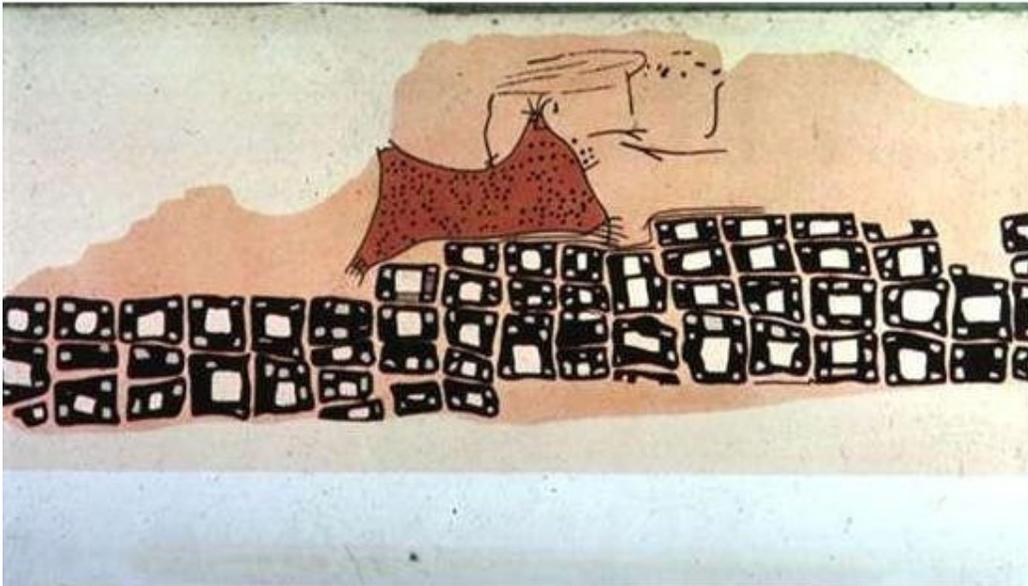
### **1.3 Historia de la Visualización**

Existen indicios del uso de la visualización desde 38.000 años AC. Antes de la aparición de cualquier lenguaje escrito, el hombre primitivo hacía marcas en huesos de animales. En uno de ellos hallado en la Dordoña (Francia) hay grabado lo que parece ser el registro de las fases de la luna durante dos meses y medio [13].

Hace 8.000 años los Sumerios ya utilizaban tablas de marcas para contar sus transacciones comerciales. 2800 años después ya utilizaban un lenguaje pictográfico con casi 2.000 signos, mientras los egipcios

desarrollaban su escritura jeroglífica, que perduraría sin cambios esenciales durante casi 3.000 años más [13].

El plano más antiguo, de una posible ciudad, descubierto hasta el momento es el de “Çatal Höyük” en Anatolia (Turquía) una de las ciudades más antiguas del neolítico. El mapa de la Figura 1, pintado en una pared hace 6.200 años, representa la ciudad mencionada [13].



**Figura 1: “Çatal Höyük”, en Anatolia (Turquía) – El Plano más antiguo de una ciudad encontrado hasta el momento**

En [21] existe una cronología completa sobre los avances de la visualización a lo largo de la historia de la humanidad. A continuación se

mencionan algunos de los hitos más destacados en la historia de la visualización según [21], [22] y [13]:

✓Año 105: invención del papel. T'sai Lun ( 50 A.C. – 121 D.C.) inventa el proceso para el desarrollo del papel, que será utilizado como medio escrito en china ampliamente, recién por el siglo 3

✓Año 950: primer intento conocido para mostrar gráficamente un cambio de valores (posiciones del sol, la luna y los planetas durante todo el año)

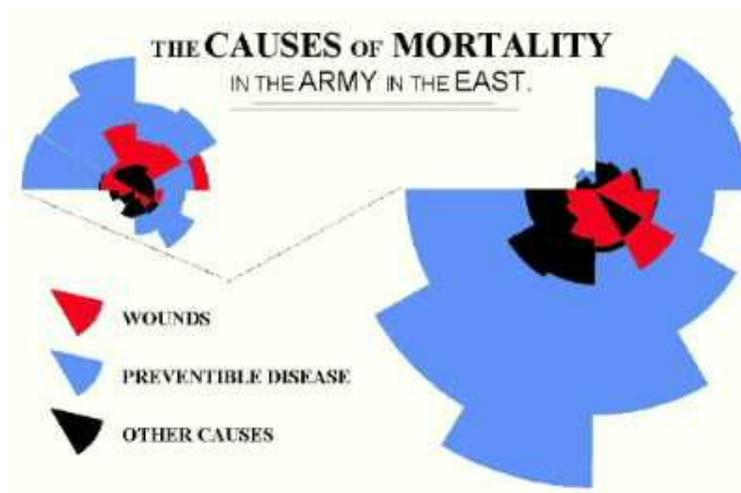
✓Año 1375: aparición del atlas catalán realizado por Abraham y Jafuda Cresques, que cubría todo el mundo conocido en la época (desde Portugal a China, cubriendo Escandinavia y el norte de África): Parte del Atlas se presenta en la Figura 2.



Figura 2: Atlas del año 1375, con parte de Europa, África y oriente [21]

✓Año 1610: Galileo Galilei realiza las primeras imágenes astronómicas jamás impresas, a partir de observaciones a través de un telescopio, para ilustrar los descubrimientos de cráteres en la luna, de 4 satélites de Júpiter y un gran número de estrellas nunca visto por los ojos sin ayuda.

✓ Año 1857, Florence Nightingale realiza el diagrama de área polar para mostrar las causas de la mortalidad en los hospitales británicos durante la guerra de Crimea, que era muy superior a la de los hospitales en Inglaterra. Nightingale hizo amplio uso de gráficos y diagramas para analizar el cuidado médico en la Inglaterra del siglo XIX, protagonizando una auténtica revolución en ese campo y salvando muchas vidas. Dicho diagrama se muestra en la Figura 3.



**Figura 3: Diagrama de Área polar realizado por Florence Nightingale en 1857 [21]**

✓ Año 1869, C.J. Minard realiza el extraordinario mapa sobre el avance y posterior retirada del ejército de Napoleón en la campaña rusa de 1812-1813. La brillante combinación de diferentes tipos de representaciones gráficas para mostrar datos multidimensionales le ha valido la consideración como autor de algunos de los “mejores gráficos jamás realizados” (según E.R Tufte en “The Visual Display of Quantitative Information”).

✓ Año 1911 Henry L. Gantt, desarrolla la planificación industrial sistemática e inventa el diagrama de Gantt ampliamente utilizado hoy en día.

El avance de la ciencia y de la tecnología ha posibilitado que la experiencia del hombre hacia la visualización se haya incrementado en forma exponencial con el correr de los años.

# Capítulo 2

## *Ingeniería de Requerimientos y Visualización*

---

### **2.1 Ingeniería de Requerimientos de Software**

Un Requerimiento es una condición o capacidad que necesita un usuario para resolver un problema o alcanzar un objetivo, o una condición o capacidad que se debe cumplir o poseer un sistema para satisfacer un contrato, norma, especificación, u otros documentos formalmente impuestos [23].

Por otra parte, se define a la Ingeniería de Requerimientos como el proceso sistemático de desarrollar requerimientos a través de un proceso cooperativo e interactivo de analizar el problema, documentar las observaciones resultantes en una variedad de formatos de representación, y chequear la precisión de la comprensión obtenida [24].

Existe consenso respecto a que la Ingeniería de Requerimientos es un proceso iterativo y tiene fases que pueden ejecutarse en forma concurrente, sin embargo hay distintos enfoques de los subprocesos a realizar.

Por ejemplo, Davis [25] propone que la fase de requerimientos consta de dos actividades principales: Análisis del Problema y Descripción del Problema.

El Análisis del Problema incluye las siguientes actividades:

- Delinear las restricción
- Refinar las restricciones
- Negociar las restricciones conflictivas

- Comprender el problema
- Expandir la información

Por otra parte, la Descripción del Problema incluye dos actividades:

- Chequear la consistencia de la información obtenida,
- Consolidar la información .

Otro enfoque es el de Jitnah [26], quien considera. que el proceso de Ingeniería de Requerimientos consta de las actividades de elicitación, análisis y especificación de requerimientos, siendo estas tareas no necesariamente ejecutadas respetando algún orden o periodo de tiempo.

En [27] Kotonya y Sommerville sugieren un proceso iterativo que se puede llevar a cabo durante todo el proceso de desarrollo del software. Este proceso establece cuatro actividades principales que son: elicitación (donde se captura, descubre y adquiere el conocimiento), análisis, especificación y validación de requerimientos. Además establece una actividad de Gestión de Requerimientos para administrar los cambios, el mantenimiento y la trazabilidad de los requerimientos. El proceso se presenta en la Figura 4



Figura 4: Modelo de proceso de Ingeniería de Requerimientos propuesto por Kotonya y Sommerville [27]

Por otro lado Van Lamsweerde [28] asegura que las actividades realizadas en el proceso de la Ingeniería de Requerimientos son análisis del dominio, elicitación, negociación y acuerdo, especificación, análisis de la especificación, documentación, y por último la evolución.

Una de las propuestas más destacadas y de mayor consenso es la presentada por Loucopoulos [24]. A continuación este enfoque es analizado con mayor profundidad.

### **2.1.1 Proceso de Ingeniería de Requerimientos según Loucopoulos**

Básicamente, Loucopoulos propone tres aspectos fundamentales que deben considerarse:

- 1) Comprender el problema,

2) Describir el problema

3) Alcanzar un acuerdo sobre la naturaleza del problema

Para lograr alcanzar estos aspectos establece la ejecución de tres etapas, las cuales se describen en la Figura 5:

- ✓ Elicitación de Requerimientos
- ✓ Especificación de los Requerimientos
- ✓ Validación de los requerimientos

Tal como se menciona en [29] cada etapa necesita de otra etapa y no necesariamente empieza y termina. Se debe iterar en las tres etapas tantas veces como sea necesario. El papel del usuario es crucial en todo este proceso, tanto para transmitir conocimiento, como para certificar que el analista comprende el problema, en el marco de un dominio de problema específico.

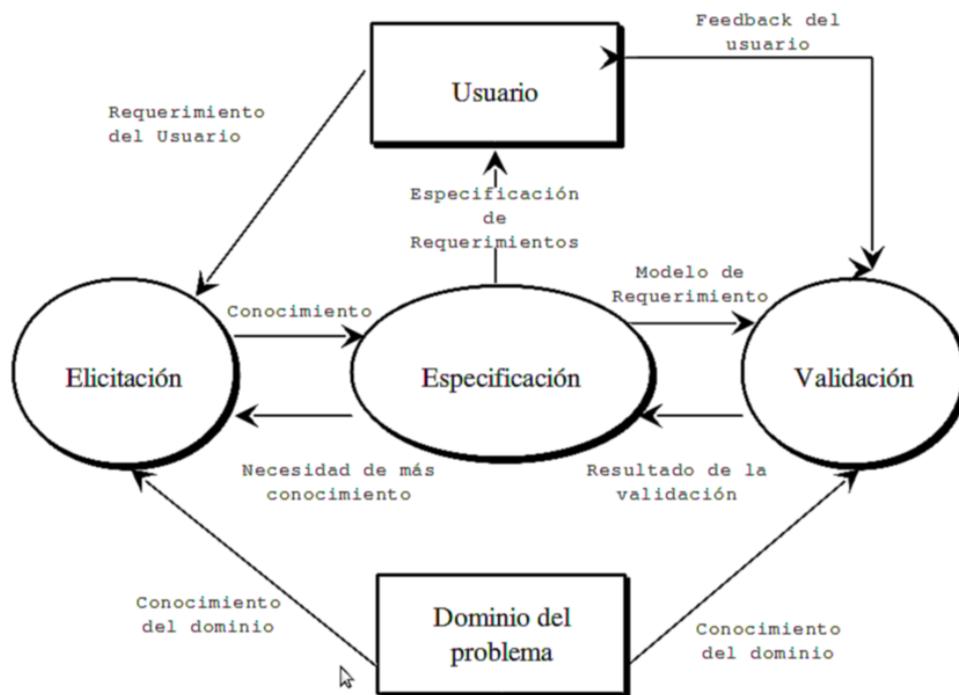


Figura 5: Proceso de Ingeniería de Requerimientos según Loucopoulos [24]

### **2.1.1.1 Elicitación de Requerimientos**

La Elicitación de Requerimientos es el proceso de obtener el conocimiento necesario sobre un dominio para elaborar un modelo de requerimientos de Software.

Durante esta etapa, es necesaria la habilidad de trabajar en colaboración con los clientes para descubrir las necesidades actuales acerca del producto a construir, y acordar la visión y las metas del proyecto propuesto. Debido a que las personas que van a interactuar con el sistema no demandan exactamente las mismas funciones, parte del proceso de elicitación es la identificación temprana de las diferentes clases de usuarios y sus características [30].

Además de comprender el dominio, el analista debe entender las necesidades de los diferentes usuarios que intervienen en él.

Una de las metas más importantes de la elicitación es descubrir cuál es el problema que se debe resolver y, por consiguiente, identificar los límites del sistema a construir. Estos límites definen, a un alto nivel, dónde se adecuará el sistema final entregado en el ambiente operacional actual. La identificación y el acuerdo de los límites de un sistema afectan todas las tareas posteriores a la elicitación [29].

El analista debe realizar la especificación de requerimientos y la validación con el usuario, solamente después de comprender la naturaleza, características y límites de un problema [29].

De lo antes expuesto, se podría mencionar entonces, que en el proceso de elicitación se debe alcanzar la identificación del problema, sus límites y de los usuarios que intervienen en él.

### **2.1.1.2 Especificación de Requerimientos**

El proceso de Especificación de Requerimientos tiene como propósito principal obtener una descripción detallada de los requerimientos del software a desarrollar, luego de que el analista tiene una primera comprensión del problema (primera iteración del proceso de Elicitación).

También se puede definir como el proceso de grabado o el registro de los requerimientos en una o más formas, incluyendo el lenguaje natural y formal, representaciones simbólicas o gráficas [31].

Una especificación puede ser vista como un contrato entre usuarios y desarrolladores de software, el cual define el comportamiento funcional del artefacto de software (y otras propiedades tales como performance, confiabilidad, seguridad, etc.), sin mostrar cómo será obtenido ese comportamiento [29].

La Especificación de Requerimientos puede ser descrita en términos de dos actividades principales [29]:

- ✓ Análisis y asimilación de conocimiento de requerimientos
- ✓ Síntesis y organización de conocimiento en un modelo de requerimientos lógico y coherente.

Es esencial que una Especificación de Requerimientos sea completa, correcta y sin ambigüedades, tanto como sea posible [8].

Esto puede lograrse si se cumplen ciertos criterios durante las primeras etapas de desarrollo de software, por ejemplo según [32]:

1. Que los Requerimientos no varíen sustancialmente durante el desarrollo de sistemas.
2. Que los Requerimientos de los clientes sean claros y sin ambigüedades.
3. Que el desarrollador haya comprendido y representado los requerimientos correctamente.
4. Que las representaciones realizadas por los clientes del sistema requerido, coincidan con las representaciones de realizadas por los desarrolladores.

Hay muchos retos para satisfacer estos criterios mencionados. Uno de los principales problemas que existe es la brecha de comunicación entre el cliente y los desarrolladores [8].

En muchas oportunidades no se cumple el cuarto criterio dado que los clientes no comprenden con exactitud las anotaciones incorporadas por los desarrolladores en las especificaciones de los requerimientos [8].

### **2.1.1.3 Validación de Requerimientos**

Por último, el proceso de confirmación con el usuario del software que los requerimientos son válidos, correctos y completos, constituye la etapa de Validación de Requerimientos. La validación es crítica para resaltar las disparidades entre las perspectivas de los stakeholders y para descubrir suposiciones que pueden quedar enmascaradas en la comunicación oral [30].

Locoupoulos indica que la validación establece y justifica la convicción del analista y del usuario, de que el modelo de requerimientos especifica una solución de software la cual es correcta para las necesidades del usuario [24].

La validación tiene como meta identificar y corregir errores en la fase de requerimientos y no más tarde cuando el software esté desarrollado. Por lo tanto, es una actividad siempre presente en el proceso de requerimientos. Es una actividad no estructurada, por lo que no tiene una solución algorítmica [29].

## **2.2 Visualización de Requerimientos de Software**

### **2.2.1 Visualización de Requerimientos - Concepto**

Se puede definir a la Visualización de Requerimientos como al acto de representar las condiciones o capacidades que necesita un usuario para resolver un problema o para alcanzar un objetivo, mediante la utilización de imágenes visuales [23] y [11]. Es decir, representar las necesidades de los usuarios, respecto a los objetivos que debe alcanzar un sistema de Software, mediante la utilización de técnicas de representación Visuales, transformando los conceptos abstractos en explicaciones visibles y entendibles.

En [33] se concibe a la Visualización de Requerimientos como una técnica gráfica que incluye dibujo de prototipo de pantallas, gráficos y modelos, que pueden ayudar a facilitar una sesión de Requerimientos. Este concepto limita en cierta medida el significado, dado que considera requerimientos meramente de tipo interfaz de usuario, lo cual no es del todo correcto, ya que la idea de Visualización de Requerimientos es poder “ver”, cualquier tipo de requerimiento, inclusive aquellos más abstractos que nada tienen que ver con la interfaz del usuario.

La Visualización es una herramienta capaz de transformar los datos existentes, en un resultado visual comprensible por los individuos interesados [1].

Así como los mapas permiten entender las ubicaciones geográficas, los gráficos de barras interpretar resultados de forma rápida, el propósito de poder visualizar Requerimientos de Software consiste en contar con una idea clara y precisa de las necesidades de los usuarios en relación al software a desarrollar. Esto permite aclarar las ambigüedades que puedan presentarse en la descripción narrativa de los requerimientos y posibilita arribar a un acuerdo mediante la utilización de herramientas gráficas.

En ocasiones se logran aclarar problemas complejos con la utilización de metáforas [1]. Cuanto más aún si estas metáforas pueden ser expresadas en forma visual.

La aplicación de la Visualización en la Ingeniería de Requerimientos es muy variada como se muestran en los distintos enfoques estudiados en este trabajo. Pero sin lugar a dudas es de mucha utilidad cuando el analista de Requerimientos lleva a cabo la difícil tarea de conocer el problema y entenderlo (Elicitación).

La notación visual desempeña un rol particularmente crítico en la comunicación con los usuarios y clientes, ya que los diagramas, se cree que son más eficientes que el texto para transmitir información a los usuarios no-técnicos [34].

También resulta de utilidad a la hora de transmitir productos intermedios que resuman los Requerimientos de software elaborados (Especificación)

Y por supuesto, la Visualización, juega un papel importante a la hora de validar con el usuario los Requerimientos.

También es un soporte muy eficaz para los integrantes del proyecto, ya que resulta de utilidad Visualizar el estatus y trazabilidad de los Requerimientos relacionados con otros artefactos del proyecto. Por lo tanto es una herramienta que bien podría utilizarse en la Gestión de Requerimientos.

Los requerimientos son, además de una descripción textual de necesidades, un conjunto de atributos [1]. Los requerimientos tienen atributos tales como usuarios, tiempo esperado, prioridad, relaciones con otros requerimientos, entre otros. Es decir, como se menciona en [1], un requerimiento no es una mera descripción en lenguaje natural, es además un conjunto multidimensional de datos.

En la representación de estos atributos, en forma correcta y oportuna, es donde radica el verdadero reto de la Visualización de Requerimientos. Por ejemplo, si un cliente desea conocer el costo que tiene modificar un requerimiento en particular, se deberá estudiar el impacto que tiene este requerimiento haciendo hincapié en los atributos que intervienen. Es necesario entonces “ver” los artefactos que se relacionan con cada requerimiento, los stakeholders que intervienen y cualquier otro punto que deba considerarse en el momento de modificar un requerimiento. Seguramente será mucho más fácil interpretar el alcance de las modificaciones, visualizando los atributos intervinientes mediante algún gráfico que los abarque a todos.

### ***2.2.2 La Visualización y el Proceso de Ingeniería de Requerimientos***

Los avances en la tecnología gráfica han permitido el desarrollo de la visualización, una disciplina que se trata generalmente de la representación de la información basada en texto con gráficos y animación [35], [36], [37] y [38]. El principal argumento para la utilización de la visualización es que puede representar la información en formas que son

más fáciles de comprender, por lo tanto, puede mejorar la comunicación [39] y [40].

Cuando se aplica a la Ingeniería de Requerimientos, se espera que la Especificación de Requerimientos pueda ser transformada en una forma que el cliente y el desarrollador alcancen a comprender más fácilmente [8].

Por lo tanto el uso de técnicas de visualización contribuye claramente a reducir el problema de falta de comunicación entre los clientes y los desarrolladores [8], mejorando notablemente el proceso de validación de requerimientos.

La representación de los requerimientos juega un papel muy importante en la validación de los requerimientos debido a que es vital para que el cliente pueda confirmar que el sistema previsto se ajuste a sus necesidades, y esto significa que el sistema debe ser descripto de una manera que el cliente puedan entender [41].

Así como es importante la visualización para la validación de requerimientos, es de gran utilidad en la elicitación. Es claro que si la idea del analista es comprender el dominio de un problema y el problema en sí, utilizar herramientas visuales para captar el conocimiento permitiría la mejor comprensión del problema.

Si bien no hay un proceso formalmente definido, hay quienes proponen pautas esenciales para utilizarlas como soporte.

En [33] se proponen los siguientes pasos simples:

1. El analista de Requerimientos captura los requerimientos en formato de texto a partir de los futuros usuarios del sistema a construir.
2. El analista crea prototipos de diseño de pantalla en papel (esbozo a mano alzada de las pantallas), y los muestra a los usuarios.
3. Los usuarios informan y proponen Requerimientos a partir de lo que han visto
4. El analista crea en detalle los requerimientos formales.

De este modo, la Visualización contribuye a obtener los Requerimientos de los usuarios, particularmente los relacionados con la interfase del futuro sistema.

La visualización es una herramienta que es de utilidad en todo el proceso iterativo de elicitación, especificación y validación de requerimientos.

Hay muchas áreas de la Ingeniería de Requerimientos donde se puede aprovechar las ventajas de la visualización. En [8] se asegura que es de gran ayuda en la representación de las Especificaciones de los Requerimientos, ya que la representación de los Requerimientos proporciona una apariencia externa de éstos.

La aplicación de la visualización en la Ingeniería de Requerimientos implica el desarrollo de un proceso que permita identificar y entender los conceptos abstractos asociados a los requerimientos, que efectivamente capture el comportamiento de ejecución del sistema de software. La eficacia de este proceso depende en gran medida la representación elegida y el grado de complejidad que presente el proceso [8].

En Figura 6 se sintetiza el proceso de Visualización en la Ingeniería de Requerimientos según [8]. Este proceso consiste en que el Desarrollador o Analista, a partir de la recreación abstracta del problema (recreación que el analista elabora en su mente luego del trabajo de Elicitación), elige y elabora una representación de las especificaciones de los requerimientos (que podrían ser en formato textual o mediante un diagrama o mediante un prototipo), que pueda ser entendida más fácilmente por los usuarios finales del sistema.

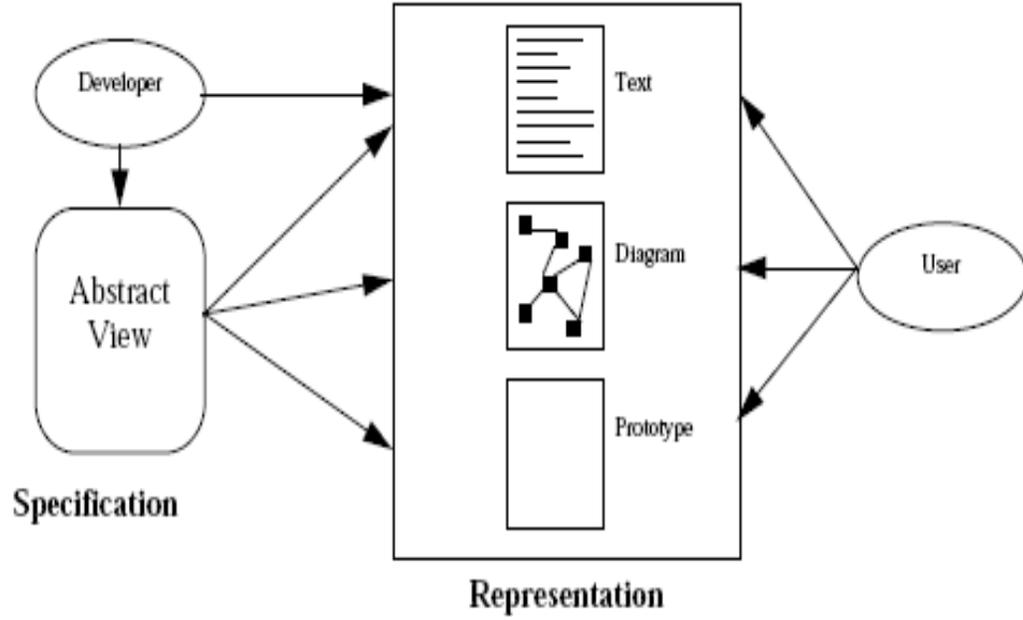


Figura 6: Proceso de Visualización en la Ingeniería de Requerimientos, según [8]

# Capítulo 3

## Enfoques de Visualización de Requerimientos

En este capítulo se presentan los enfoques de Visualización de Requerimientos más destacados de la comunidad vinculada a este tema, algunos de ellos han sido divulgados a través de [4], [5], [6] y [42]

### **3.1 Systems Modeling Language (SysML)**

Este enfoque es un estándar de extensión de UML para representación de requerimientos [43]. El lenguaje SysML es un subconjunto ampliado de UML 2.0, utilizado para especificación de sistemas. Desde Septiembre de 2007 es un estándar del Grupo de Gestión de Objetos (Object Management Group – OMG).

El estándar SysML está desarrollado a partir de UML 2.0 desde la Meta-Object Facility (MOF). Se compone de cuatro tipos de diagramas principales, uno de los cuales es SysML (diagrama de Requerimientos) [43].

SysML provee las construcciones de modelado para representar los requerimientos basados en texto. Un requerimiento se representa mediante un estereotipo de una Clase UML que cuenta con un identificador único y el texto del requerimiento. El estándar permite la representación de relaciones entre requerimientos y otros componentes (como clases o casos de uso), por el cual se especifica si es una relación de dependencia o inclusión (un requerimiento depende de otro requerimiento para su realización, o un requerimiento está compuesto por otros requerimientos).

Existe software de modelado de sistemas que incluye la notación SysML para realizar los diagramas de requerimientos. Un ejemplo es el sistema **Enterprise Architect** de **Sparx Systems** [45], software que actualmente se encuentra disponible en su versión 8 y es totalmente

compatible con UML 2.0 y SysML. Desde su página WEB permite la descarga de un trial para probar el sistema durante 30 días.

Otro software es el **Visual Paradigm** [46], que es una suite de aplicaciones para diseño y mantenimiento de software. Incluye también el módulo para representación de Requerimientos de Software.

A diferencia de **Enterprise Architect** [45], **Visual Paradigm** [46] cuenta con una versión libre para uso no comercial denominado **Community Edition**.

En la Figura 7 se puede contemplar un diagrama de requerimientos realizado con la herramienta **Visual Paradigm** [46] y utilizando el lenguaje SysML.

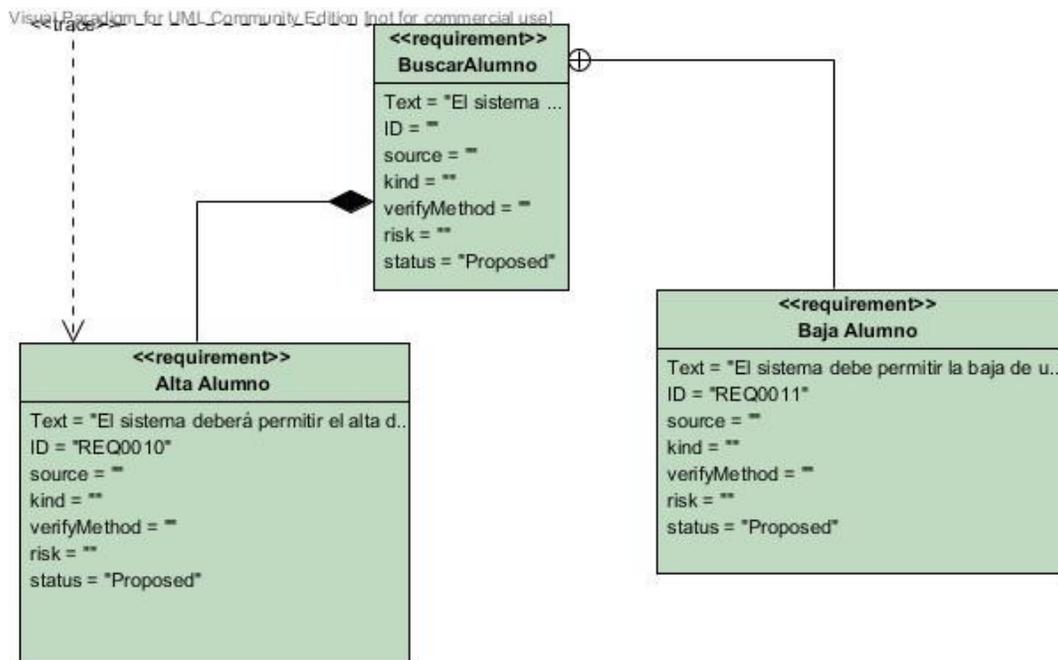


Figura 7: Requerimientos dibujados con Visual Paradigm [46]

En esta Figura se grafica un ejemplo sencillo donde se representan tres requerimientos: “BuscarAlumno”, “AltaAlumno” y “BajaAlumno”. El diagrama permite establecer relaciones de dependencia o pertenencia. Por ejemplo si se desea realizar una baja de un alumno es necesario realizar la

búsqueda del alumno. También es apreciable, que todos los requerimientos tienen al menos, un nombre y una identificación.

Además es posible representar y codificar otros atributos específicos de cada requerimiento, como el estado (por ejemplo, si el requerimiento se encuentra en planeamiento, en proceso o cancelado) y determinar la fuente de origen y por supuesto la descripción del requerimiento en formato texto natural.

### ***Ventajas***

Es una metodología asociada a UML, con lo cual se presume de fácil aprendizaje para aquellos que conozcan esta metodología.

Ha sido creado para documentar los requerimientos y sus relaciones en un formato gráfico. Si bien es una extensión de UML, SysML no se limita meramente sólo a representar requerimientos de software que será desarrollado mediante la metodología de Orientación a Objetos.

Existen sistemas específicos en el mercado que están adaptados a esta metodología, como los mencionados anteriormente **Enterprise Architect** [45], **Visual Paradigm** [46], entre otros.

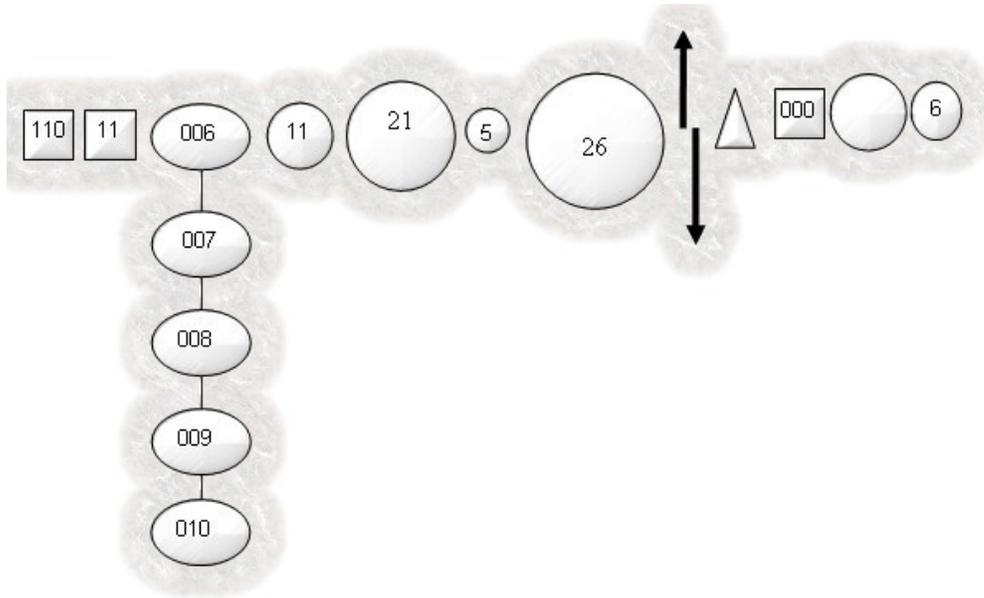
### ***Desventajas***

Si bien este enfoque es una adaptación de UML y es fácil de ser interpretado por un ingeniero de software, no está claro si podría ser comprendido fácilmente por un usuario final (gerente, usuario de la aplicación), por lo que no sería muy eficiente utilizar este método para la captura de conocimiento y su posterior validación.

## ***3.2 Enfoque Simbólico***

El enfoque Simbólico [1] permite representar atributos relacionados a un requerimiento en particular, mediante gráficos compuestos por cuadros, círculos y flechas, lo que facilita la detección de requerimientos más complejos.

Los requerimientos son representados de acuerdo al valor de los atributos que lo componen, por lo que cada representación tomará una forma única. La Figura 8 muestra un ejemplo de este enfoque, en la cual se representan los valores asociados a cada atributo de un requerimiento en una fila de símbolos dispuestos en forma sencilla (“red Rectilínea” [1])



**Figura 8: Enfoque simbólico. Representación adaptada de [1]**

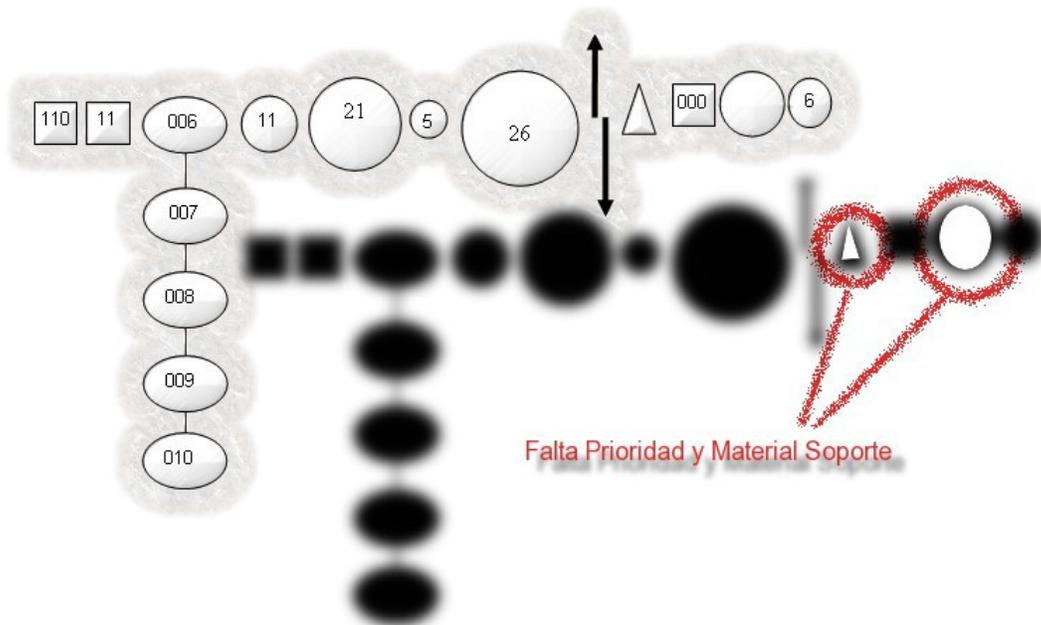
Esta propuesta [1] esta basada en las especificaciones de Volere [47] para describir los atributos de un requerimiento en particular

En el ejemplo de la Figura 8 se incluye en primer lugar, de izquierda a derecha, el número de requerimiento (110), el tipo de requerimiento (11), eventos o casos de uso relacionados con el requerimiento (lista de óvalos, 006, 007, 008, etc.) y así sucesivamente. La lista completa de los atributos representados es la siguiente:

1. Número de Requerimiento (110)
2. Tipo de Requerimiento (11)
3. Evento / Lista de Casos de Uso (006)-(007)-(008)-(009)-(010)

4. Descripción (11 Texto)
5. Justificación (21 Texto)
6. Fuente (5 Texto)
7. Criterio (26 Texto)
8. Satisfacción del cliente (3)
9. Insatisfacción del Cliente (5)
10. Prioridad (Esta ausente)
11. Lista de Conflictos (000)
- 12 Materiales de apoyo (Esta vacío)
- 13 Historia (6 Texto)

Esta representación, se podría transformar perfectamente en una sombra y así detectar los vacíos o faltantes del requerimiento. Como se puede apreciar en la Figura 9, no hay material de apoyo, ni esta establecida una prioridad.



**Figura 9: Enfoque simbólico, con sombra. Representación adaptada de [1]**

Mediante el dibujo de la sombra del Requerimiento, es posible detectar los atributos faltantes, apreciando a simple vista los objetos que quedan en color blanco, es posible identificar cuales son los atributos sin datos.

### ***Ventajas.***

Claramente se puede apreciar que este enfoque permite detectar requerimientos complejos y abultados, y la información faltante o inexistente sobre los mismos, asociando en un solo gráfico varios atributos de un solo requerimiento.

### ***Desventajas***

Resulta costoso de elaborar ya que no se cuentan actualmente con herramientas de apoyo que permitan realizar la representación en forma automática.

Además, no está especificado ningún proceso formal que indique los pasos a seguir para la elaboración del gráfico.

No está claro si es posible que el usuario final, Stakeholders interesados en la realización del sistema, pueda interpretar esta visualización. Es decir que los principales beneficiarios de este enfoque son los diseñadores y aquellos que estiman el tiempo que insume cada requerimiento. En definitiva, está destinado a los Stakeholders que participan en el desarrollo del software.

### **3.2.1 Enfoque Simbólico – Propuesta de Mejora.**

Como se explicó en los párrafos anteriores, no existe una herramienta o proceso automático que permita elaborar de forma rápida y eficiente los gráficos del enfoque simbólico. Al menos no existe explicación o aclaración al respecto en [1].

Realizar un software o una propuesta de sistematización del método de Enfoque Simbólico podría ser un anhelo demasiado ambicioso para este trabajo.

Sin embargo, se puede expresar una secuencia de pasos a seguir que ayuden y faciliten la elaboración del Enfoque Simbólico.

A continuación se presenta como mejora a la propuesta de Enfoque Simbólico, una secuencia de pasos sugeridos que un futuro sistema informático podría considerar para realizar el proceso de transformación de requerimientos de formato Texto a formato Gráfico, con diseño de enfoque simbólico.

1) En primer lugar deberían identificarse todos los atributos que pueden contener los Requerimientos y asignar a cada atributo una simbología:

Por ejemplo, el atributo “Fuente” del requerimiento se grafica con un círculo, el atributo “Número o identificación” del requerimiento se grafica con un cuadrado, y así sucesivamente.

Con este punto, se solucionaría gran parte del problema, debido a que cada atributo que puede tener un requerimiento tendría asignado una simbología correspondiente.

2) Ingresar los requerimientos en formato texto mediante alguna interfaz de usuario, e ingresar los atributos de cada requerimiento tales como fuente, prioridad, entre otros.

3) Por ultimo, el sistema debería posibilitar la generación de un grafico al estilo enfoque simbólico, de acuerdo a la asociación de atributos realizada en el punto 1 y de acuerdo a los atributos efectivamente cargados en el punto 2.

Por ejemplo se podría agregar la funcionalidad de graficar según enfoque simbólico, a los sistemas que se mencionaron anteriormente, y que permiten graficar requerimientos mediante técnica de SysML (**Enterprise Architect** [45] y **Visual Paradigm** [46]). Estos sistemas ya cuentan con la interfaz de usuario para ingresar los requerimientos y sus correspondientes atributos, por lo tanto, solo quedaría implementar la asociación de los atributos con un determinado símbolo y la funcionalidad de elaboración de la vista en formato enfoque simbólico.

### **3.3 Enfoque Icónico**

También mencionado en [1], este enfoque plantea una representación gráfica de los requerimientos utilizando caras o iconos gestuales.

Su diseño permite una evaluación “cruda” (es decir, al desnudo, sin velos que desformen la información) sobre la certeza de los datos presentes en los requerimientos.

Se dibuja una cara por cada atributo de requerimiento, y cada requerimiento ocupa una línea de rostros. Se utilizan dos formas para reflejar el tipo de requerimiento. Un círculo se usa para requerimientos funcionales, y un pentágono se utiliza para requerimientos No funcionales. Cabe aclarar que los gestos utilizados se limitan a la presentación de la boca del rostro en tres posiciones.

En la Figura 10 se puede apreciar un ejemplo de esta representación, donde el atributo “valor” está dado por una satisfacción del cliente.

Si el cliente está de acuerdo en el diseño del requerimiento y su posible implementación, el rostro mostrará rasgos de felicidad. Si parte del requerimiento no satisface plenamente al cliente, entonces el rostro presentará aspecto de seriedad. Por último, si el rostro es triste, seguramente habría que replantearse en conjunto el requerimiento ya que el cliente no se siente satisfecho con el requerimiento.

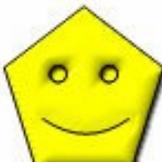
Requerimiento	Valor	Fuente
# 12		
# 13		
# 14		

Figura 10: Enfoque Icónico. Representación adaptada de [1]

Respecto al atributo “fuente”, se evalúa si los datos para cada requerimiento son proporcionados por personas que se encuentran dentro del grupo interesado en su implementación. Un rostro con rasgos de “felicidad” denota que la información está dentro del grupo de Stakeholders, si los rasgos son de “seriedad” parte de la información necesaria esta fuera de este grupo; y si tiene rasgos de “tristeza” toda la información necesaria se encuentra fuera del grupo interesado en el cumplimiento del requerimiento.

Además, se utilizan colores en la representación para indicar la “felicidad” del atributo (Amarillo para Feliz, gris para “Serio”, azul para “Triste”), produciéndose una codificación redundante (expresión de la cara y color).

### ***Ventajas***

El enfoque icónico es visualmente amigable ya que presenta la particularidad de visualizar rostros y es intuitivamente entendible.

Permite evaluar rápidamente el estado y aceptación de los requerimientos.

Si bien el análisis de cada requerimiento es individual, se puede visualizar el estado de varios requerimientos a la vez, por lo que facilitaría una rápida evaluación de varios requerimientos.

### ***Desventajas***

Graficar las caras implica un trabajo tedioso ya que no hay un proceso automático. A mayor cantidad de atributos que se deseen evaluar, mayor será el trabajo a realizar.

#### ***3.3.1 Enfoque Icónico – Propuesta de Mejora.***

La propuesta de Enfoque Icónico tal como se presenta en [1], se limita a la evaluación de los atributos según la valoración que le da un solo grupo de Stakeholders (en los ejemplos se menciona al usuario final). Sin embargo se podría mejorar la representación utilizando el color de los rostros (que actualmente tiene un significado redundante), para representar la perspectiva del mismo atributo desde el punto de vista de otro grupo de Stakeholders, y así poder comparar las distintas posturas.

Por ejemplo, si se utilizase el color verde para indicar un grupo de usuarios, el color amarillo para otro grupo, y el azul para otro y así sucesivamente; se podría analizar en un solo grafico las apreciaciones de varios Stakeholders, respecto de varios atributos de cada requerimiento.

En la Figura 11 se presenta un ejemplo de esta propuesta, donde:

- El color verde indica al grupo de desarrolladores
- El color azul al usuario final del sistema
- Por ultimo, el color amarillo representa al Gerente del usuario Final (Jefe del empleado, que no utiliza directamente el sistema, ya que no lo opera, pero si necesita la obtención de resultados, informes, gráficos de análisis, entre otros.)

Requerimiento	Valor	Fuente
# 12		
# 13		
# 14		

Figura 11: Propuesta de Mejora del Enfoque Icónico.

Con esta mejora se podría mostrar que dos grupos de Stakeholders tienen una apreciación diferente respecto de algún atributo del requerimiento, como se da en el requerimiento número 13 para el atributo “Fuente” de la Figura 11.

En este caso el usuario final interpreta que solo parte de la información necesaria para realizar el requerimiento se encuentra fuera del grupo de Stakeholders. Sin embargo, el grupo de desarrolladores interpreta

que toda la información necesaria para el requerimiento esta fuera del grupo de Stakeholders.

Con esta mejora, es posible detectar interpretaciones divergentes entre distintos grupos de Stakeholders, y sería posible detectar inconsistencias en los atributos de los requerimientos de forma temprana.

### **3.4 Aproximación Metafórica**

La técnica de Aproximación Metafórica tiene como particularidad especial, la utilización de metáforas simples para transmitir la información relativa a la estabilidad de un conjunto de requerimientos.

Este enfoque está centrado en la representación gráfica de un subconjunto de atributos de los requerimientos, para ayudar a responder preguntas tales como: ¿qué requerimientos son susceptibles de cambios? ¿Dónde están los problemas inminentes?

Cada requerimiento se representa como una isla con un pequeño volcán. Si existen dependencias entre los requerimientos, las islas están relacionadas con calzadas, o agrupadas en función de la naturaleza de la dependencia. El tipo de requerimiento no se manifiesta en la visualización, pero esto podría lograrse por la forma o el color del volcán. El tamaño del volcán es proporcional a la cantidad de material de apoyo, es decir si el requerimiento cuenta con mucho material adicional que facilite su interpretación y su elicitación, y las nubes sobre el volcán significan el nombre de una fuente [1].

En la Figura 12 se representan tres requerimientos, de cada requerimiento se aprecia la fuente que lo origina, como así también las dependencias que existen entre los requerimientos (calzadas bajo los volcanes). Además es factible observar, que el requerimiento #56 se caracteriza por tener un abundante material de apoyo.

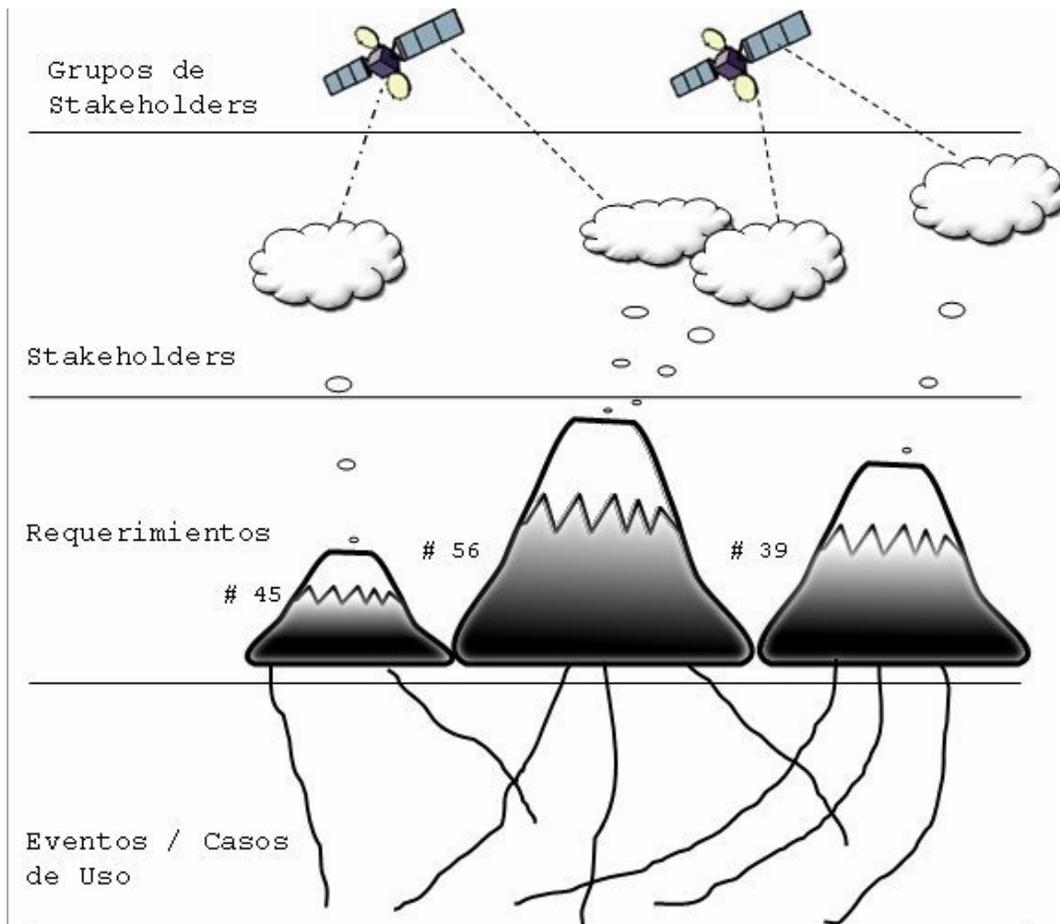


Figura 12: Aproximación metafórica. Representación adaptada de [1]

### **Ventajas**

Esta técnica es visualmente muy expresiva, debido a que utiliza metáforas para facilitar su comprensión.

Permite detectar requerimientos complejos a simple vista, por el tamaño del volcán y el cruce de sus calzadas (requerimientos que se relacionan con otros requerimientos).

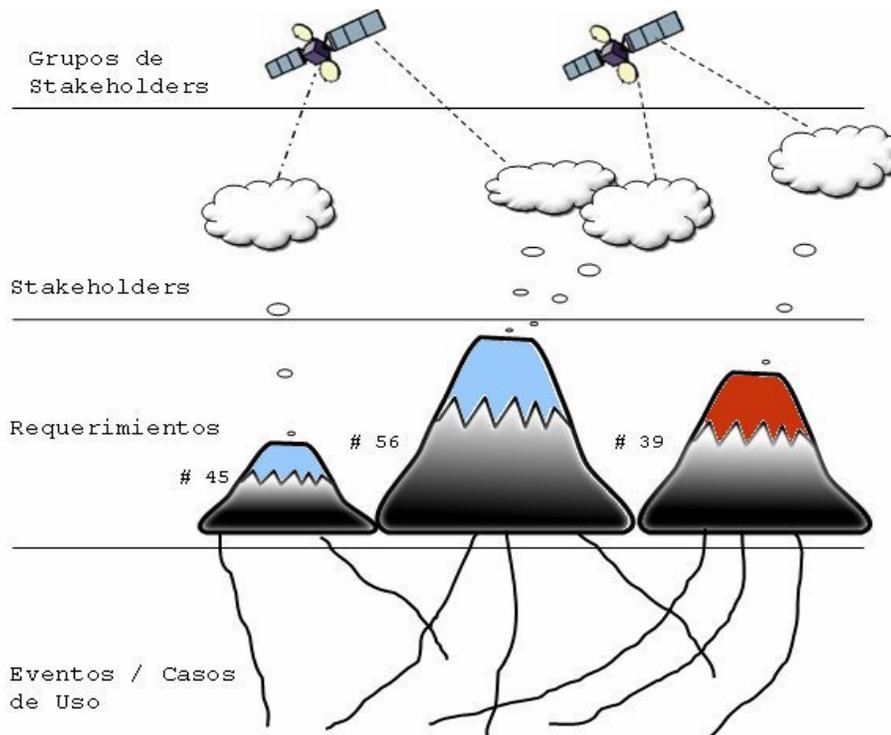
### **Desventajas**

No se cuenta con un proceso sistemático que guíe la elaboración de los gráficos. Tampoco hay evidencia de alguna herramienta que ayude a elaborar estos diagramas.

#### **3.4.1 Aproximación Metafórica – Propuesta de Mejora.**

Una mejora que podría realizarse en la técnica de Aproximación Metafórica mencionada en [1] es la utilización de colores para identificar el tipo de requerimientos.

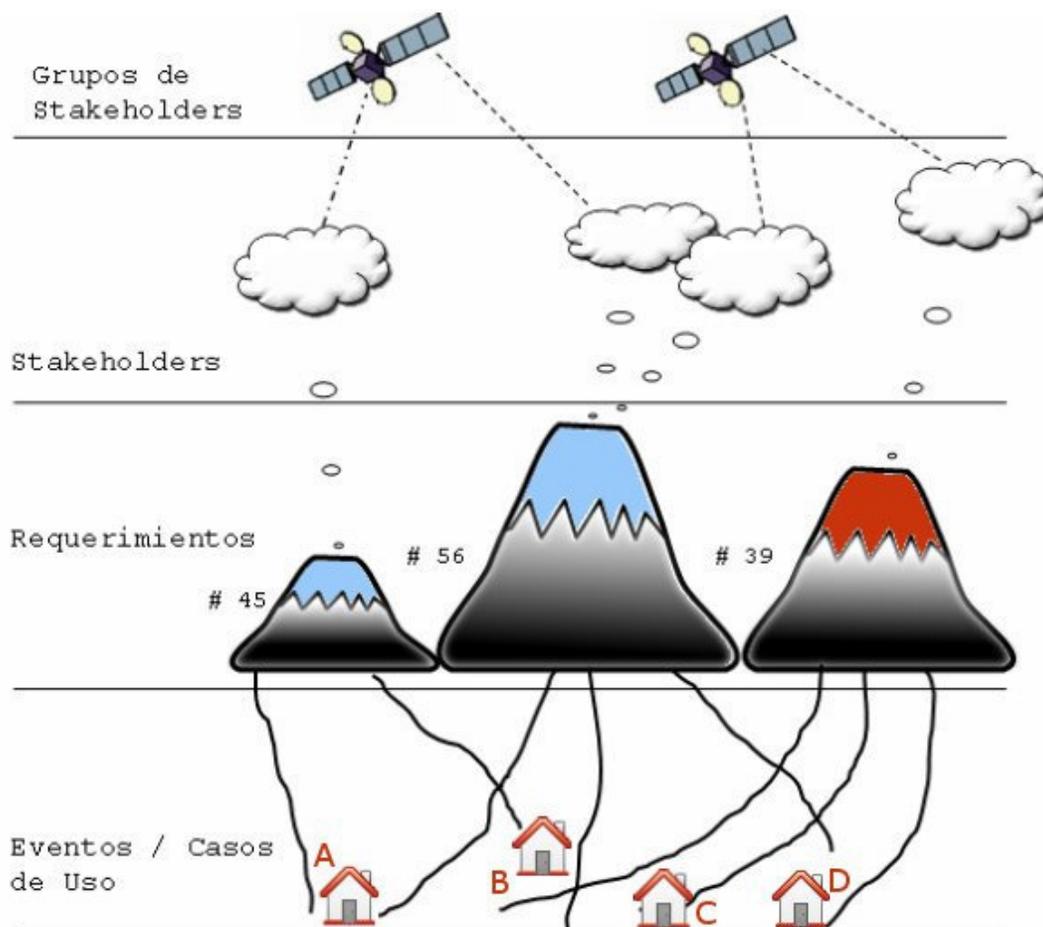
Esta mejora se podría implementar, por ejemplo, coloreando los volcanes de color rojo, aquellos que son requerimientos no funcionales y de color celeste aquellos que son requerimientos funcionales. En la Figura 13 se agregan los colores identificando a los requerimientos #39 como no funcional y a los requerimientos #45 y #56 como requerimientos funcionales.



**Figura 13: Aproximación metafórica. Propuesta de Mejora, identificando requerimientos con colores**

Ya que no está claro en el gráfico presentado en [1], donde finalizan las calzadas, se podría agregar un nuevo símbolo para identificar los casos de usos con los que los requerimientos están relacionados.

Por ejemplo, se podrían agregar dibujos de casas, que identificarían a los casos de uso. En la Figura 14, se incluye esta modificación. Aquí se puede apreciar que los requerimientos 45 y 56 se relacionan (además del cruce de calzadas), porque una de sus calzadas desemboca en el mismo caso de uso (para el ejemplo el caso de uso denominado "A").

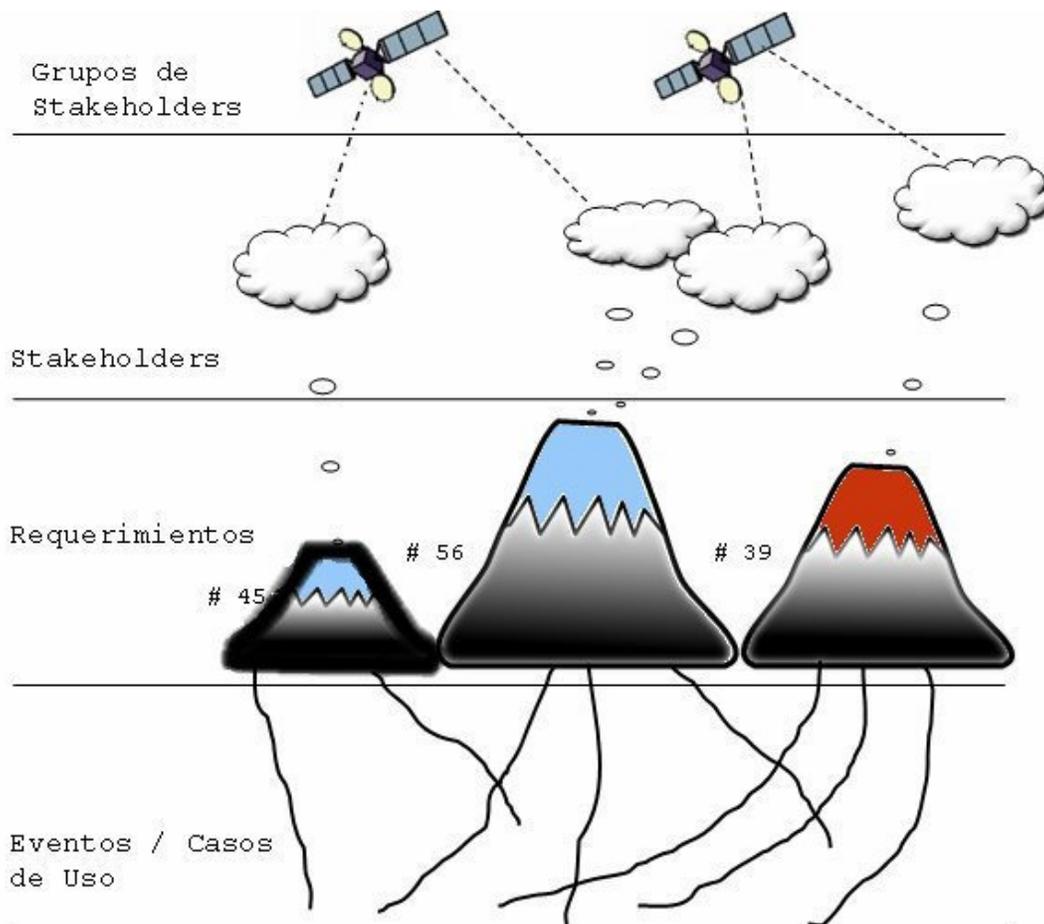


**Figura 14: Aproximación metafórica. Propuesta de Mejora, identificando de casos de usos relacionados con los requerimientos.**

Además se puede interpretar que el requerimiento 39 está relacionado con el Caso de Uso denominado “C” y con el Caso de Uso denominado “D”.

Un atributo que es de mucha importancia es la prioridad del requerimiento. Este atributo está ausente o no es posible identificarse las graficas de aproximación metafórica.

Una alternativa podría ser agregar a cada volcán un borde fino o grueso dependiendo de la prioridad que tenga el requerimiento. Es decir, cuanto más grueso es el borde del volcán, mayor es la prioridad del requerimiento.



**Figura 15: Aproximación metafórica. Propuesta de Mejora, identificando la prioridad de los requerimientos.**

En la Figura 15 se puede observar esta propuesta. En este caso, el requerimiento 45 presenta mayor prioridad que el requerimiento 56 y el 39, a pesar de tener menos material de apoyo.

### 3.5 FLOW

Esta propuesta incorpora un lenguaje de Visualización para representar flujos de los requerimientos. En [48] se presenta la notación sugerida para utilizar en la visualización de los flujos de información que se presentan en los requerimientos.

El objetivo de esta técnica es dejar rastros de la información que subyace a los requerimientos y que muchas veces no es formal. Es decir, intenta mantener un vínculo entre los requerimientos y la comunicación informal entre los stakeholders que los nutre. En la Figura 16 se presenta la notación FLOW propuesta en [48].

Según se menciona en [48], se pensó realizar una notación sencilla y que pueda ser realizada con cualquier herramienta tipo de oficina, sin embargo no se explica con profundidad su utilización.

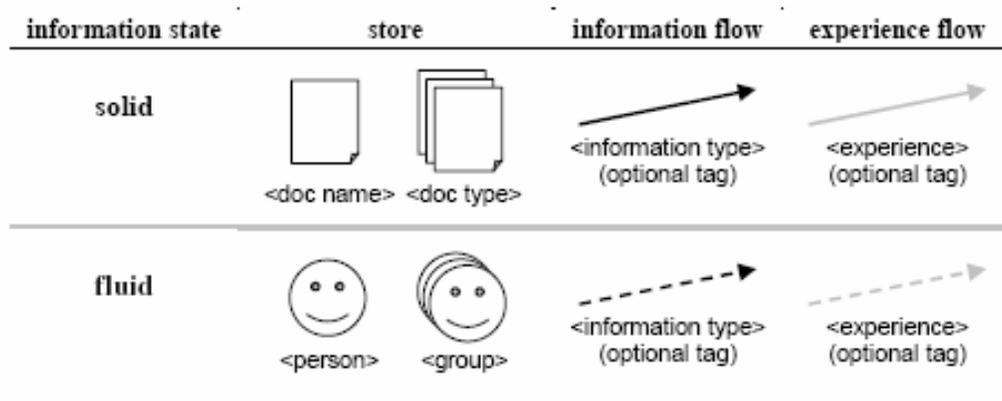


Figura 16: Notación técnica FLOW [48]

La Figura 17 presenta un ejemplo de FLOW donde se describe la representación de una nueva técnica de manipulación de requerimientos de seguridad denominada “SecReq”. Se visualiza las entradas de las partes interesadas y de la utilización de una herramienta llamada “UMLsec”. En este ejemplo se describe mediante grafica los flujos que subyacen al

requerimiento “SecReq”, donde intervienen un Grupo de Stakeholders, proporcionando información al requerimiento. También participa en forma de asesor un ingeniero en seguridad (Flecha gris) y es información necesaria los documentos de seguridad y la documentación UML referente al Sistema “Sec”.

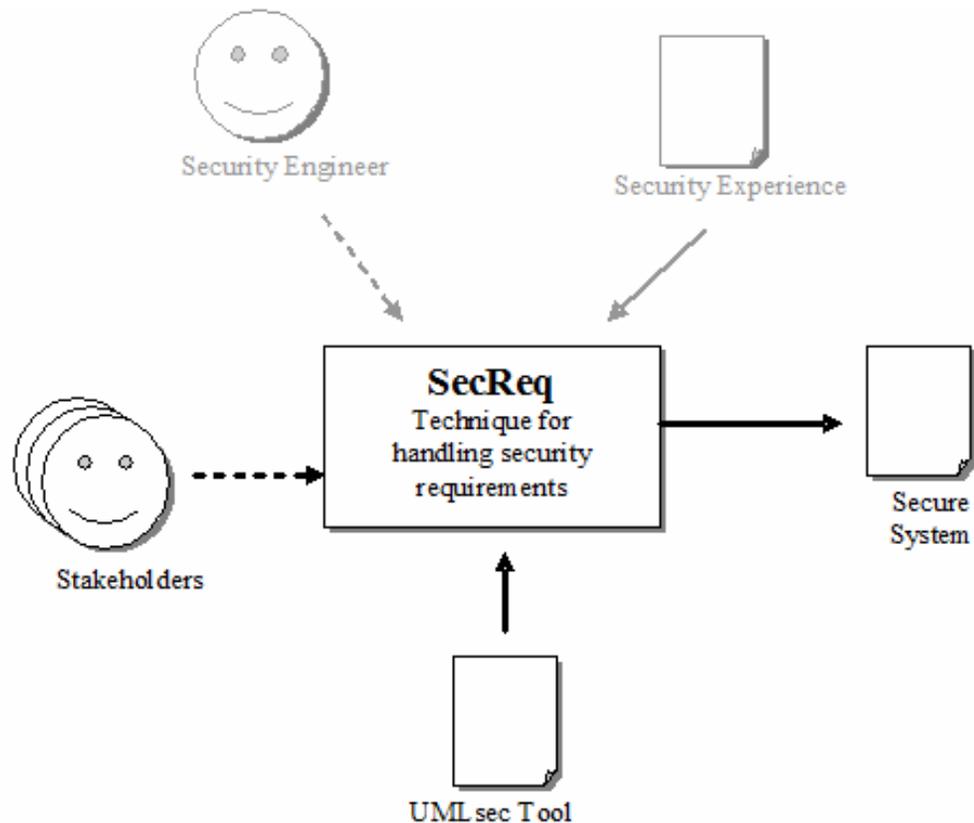


Figura 17: Ejemplo utilización técnica FLOW [48]

Según [48], FLOW fue diseñado para cumplir con los siguientes criterios:

- La notación debe ser fácil de usar en pizarras, en general, en herramientas de dibujo de PowerPoint, y en editores de Texto.
- Los modelos deben ser auto-explicativos a los expertos del dominio sin ningún tipo de formación en el modelado de software. Este criterio es

esencial para utilizar esta técnica como medio de comunicación entre diferentes Stakeholders.

- La notación debe permitir visualizar y distinguir explícitamente el Flujo de la información.
- Deben reflejarse las actividades a fin de vincular los modelos de flujo de los pasos del proceso.
- La notación debe contener el menor número de símbolos y detalles como sea posible, para no distraer de la atención del flujo de información.
- Deben utilizarse las notaciones que resulten más familiares a los stakeholders (para [48] las flechas, círculos y rectángulos son notaciones familiares a los stakeholders), cuando exista algún conflicto con los criterios antes mencionados.

### ***Ventajas***

Permite identificar la información informal que gira entorno a los requerimientos.

Tiene una notación claramente definida.

La notación FLOW utiliza como base otras dos notaciones conocidas DFD (Diagrama Flujo de Datos) y UML (Lenguaje de Modelado Unificado), por lo que esto facilita su utilización y aprendizaje

### ***Desventajas***

Si bien existe definida una notación, no hay evidencias de que exista un proceso que guíe la elaboración de los diagramas.

Tampoco existe una herramienta software que soporte dichos diagramas. La idea de los autores es desarrollar un lenguaje que se adapte a las aplicaciones de oficina existentes, lo cual es una limitación en cierta medida. La realización de la Figura 17 se realizó con Microsoft Word 2003 utilizando las auto-formas de la barra de herramientas de dibujo.

### **3.6 Red Social Centrada en Requerimientos (RCSN)**

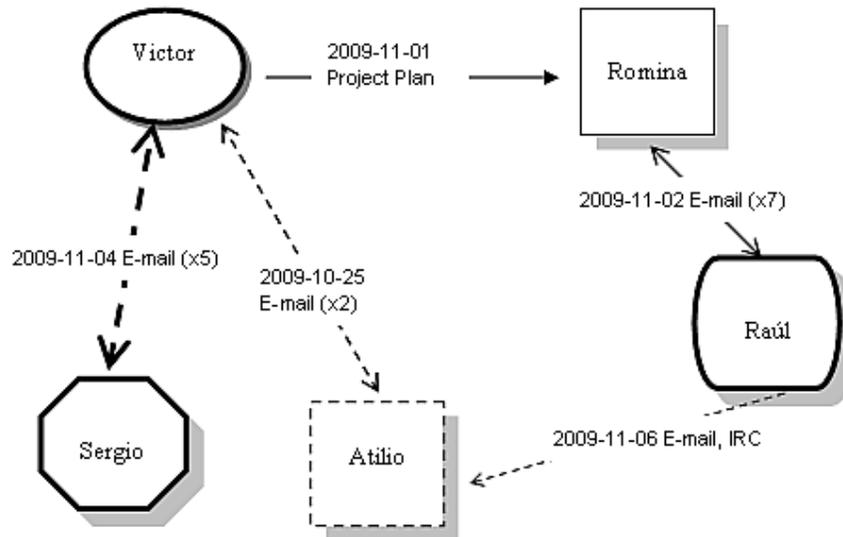
Esta técnica se utiliza fundamentalmente para visualizar los stakeholders que intervienen a lo largo de toda la vida de un requerimiento. Los flujos de información entre los stakeholders y los requerimientos se establecen mediante la utilización de círculos, cuadrados y flechas, lo cual facilita que todas las personas involucradas sean notificadas de las novedades que puedan ocurrir en cada requerimiento, [49].

Los equipos de desarrollo están en continuo cambio durante la vida de un requerimiento, ya que se podrían agregar nuevos interesados, se consultan a otras áreas de la organización o empresa que requiere el desarrollo del software, por lo que resulta necesario mantener toda la información relacionada de los stakeholders involucrados con cada requerimiento.

Una red social es una estructura donde una persona se representa con un nodo, y una línea que conecta dos nodos indica la comunicación procedente de un individuo a otro individuo. Esta estructura puede ser representada visualmente en forma de un gráfico llamado diagrama de red-social.

“Un diagrama de red social centrada en un requerimiento” (RCSN) es una red social que muestra sólo los involucrados con el desarrollo de un requerimiento.

En la Figura 18 se expone un ejemplo de la representación de un requerimiento mediante el diagrama RCSN. Las flechas representan los flujos de información; de acuerdo a su espesor especifican la cantidad de información transmitida, la dirección de la comunicación de información, el medio utilizado (mail, chat, un documento formal, entre otros) y el rol de cada persona que interviene. Se utiliza un símbolo diferente para cada rol.



**Figura 18: Ejemplo de modelo RCSN, adaptado de [49]**

El RCSN permite visualizar la siguiente información respecto a un requerimiento:

- Funciones individuales de los participantes, por ejemplo, diseñador, programador, tester, gerente, espectador, entre otras; que cuenten con acceso a información adicional sobre actividades relacionadas con el requerimiento;
- La dirección del flujo de la comunicación;
- La cantidad de información que fluye de un individuo a otro; el detalle sobre la información intercambiada debe ser fácilmente accesible;
- Fecha de intercambio de información.

### **Ventajas**

Se visualiza claramente el nivel de intervención de cada participante.

La representación gráfica se realiza mediante un conjunto de elementos bien definidos.

### ***Desventajas***

A pesar de que existe una notación que define la representación, no hay evidencia de un proceso que guíe los pasos a seguir.

Además no se dispone de una herramienta que actúe como asistente en la elaboración de los gráficos.

### ***3.6.1 Red Social Centrada en Requerimientos - Propuesta de Mejora***

Se podrían introducir colores en los flujos para distinguir la información que es más reciente respecto de la que es más antigua.

También se podría distinguir con colores a los Stakeholders que más participación tienen en el requerimiento.

En la Figura 19 se puede apreciar esta alternativa, donde se agrega color a los flujos de información y a los Stakeholders, para identificar la información más reciente de la más antigua y los stakeholders que tienen más participación en el requerimiento respectivamente.

Claramente queda identificado que el Stakeholders llamado "Víctor" es el que más participación tiene en el requerimiento y además el que más recientemente ha realizado intercambio de información sobre el requerimiento.

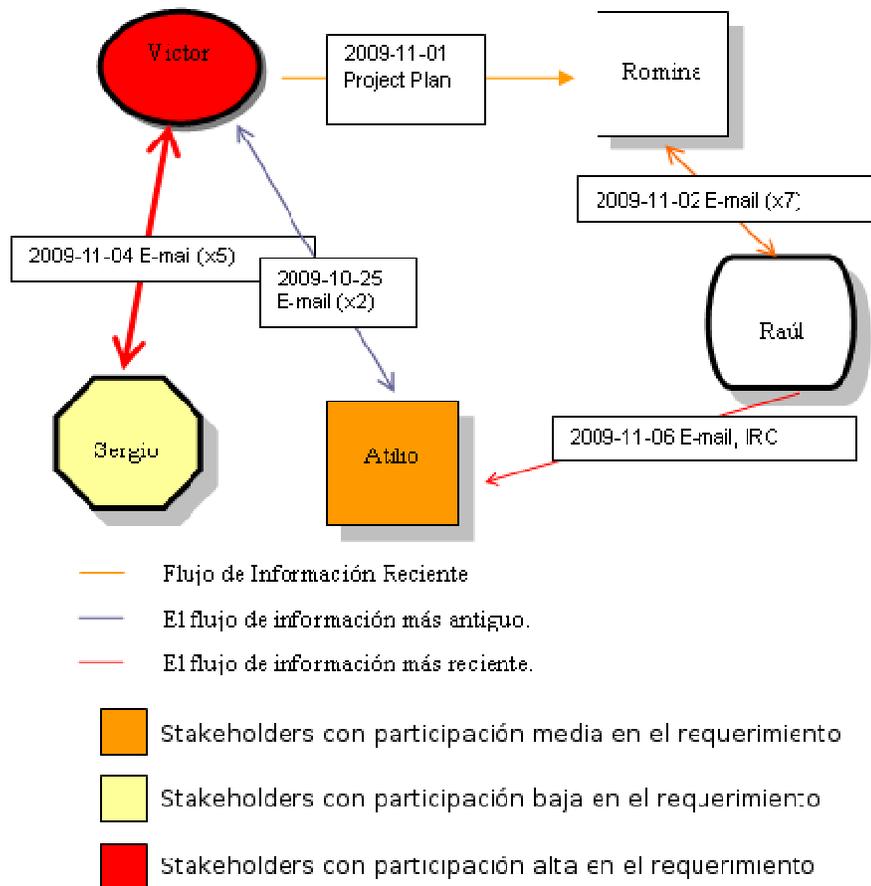


Figura 19: Ejemplo de modelo RCSN, Mejorado agregando colores a los flujos y a los Stakeholders

### 3.7 Modelo de Respuesta de Acción de Pantalla - Display-Action-Response (DAR)

Este modelo se utiliza para capturar requerimientos de interface y reglas de negocio [50]. El objetivo es proporcionar una imagen completa del sistema, vinculando las pantallas de interfaz de usuarios con los requerimientos de negocio.

La información sobre la visualización de interfaz de usuario de software y otros elementos visuales, están basados en diversas condiciones de estado, es por ello que el modelo DAR, enumera el comportamiento de

cada elemento de datos dinámicamente basado en cada condición y acción posible.

El modelo es apropiado para especificaciones de sistemas en los cuales se pone énfasis en la interfaz de usuario. Este modelo está compuesto por una combinación del diseño de las pantallas y tablas de elementos.

En la Tabla 1 se puede visualizar un formato propuesto de la estructura de la tabla, que se realiza a partir de los elementos de la interfaz de usuario. En esta tabla se debe incluir el comportamiento que refleje las posibles precondiciones y los resultados del sistema o eventos de usuarios.

<b>UI Elemento: &lt;nombre del elemento&gt;</b>		
<b>UI Descripción Elemento</b>		
ID	<identificador univoco del elemento en la página >	
Descripción	<descripción del elemento>	
Caso de Uso	<Link al caso de uso que relacionado>	
Size	<tamaño del elemento>	
Validación de datos	<valores a validar en el elemento>	
<b>UI visualización del elemento</b>		
Precondición	Pantalla	
<precondición 1>	<visualización del elemento respecto de la precondición 1>	
<precondición 2>	<visualización del elemento respecto de la precondición 2>	
....	....	
<b>UI Comportamiento del Elemento</b>		
Precondición	Acción	Respuesta
<precondición 1>	Acción de usuario 1	<Respuesta de sistema a la precondición 1>
<precondición 2>	Acción de usuario 2	<Respuesta de sistema a la precondición 2>

**Tabla 1 – Formato propuesto por DAR [50]**

En la Tabla 2 se presenta un ejemplo de la utilización del Modelo DAR, para representar la interfase y el comportamiento deseado del sistema, para el elemento “Número de Teléfono”.

UI Elemento: Número de Teléfono		
UI Descripción Elemento		
ID	ELMT_04006	
Descripción	Número de teléfono para contacto	
Caso de Uso	UC081_ <u>SeleccionarHorario_Contacto.doc</u>	
Size	Ver sección validación de número telefónicos	
Validación de datos	Solo valores numéricos Campo requerido (no se admite valor nulo)	
UI visualización del elemento		
Precondición	Pantalla	
Si esta seleccionado el País EE.UU. en el desplegable	Mostrar en Blanco, cuatro casilleros Número de teléfono ( <input type="text"/> ) <input type="text"/> - <input type="text"/> Ext. <input type="text"/> Codigo Area	
Si esta seleccionado cualquier otro país distinto a EE.UU.	Mostrar en blanco. Solo se visualizan tres casilleros y se debe indicar el código de país Número de teléfono ( <input type="text"/> ) <input type="text"/> Ext. <input type="text"/> Codigo País	
UI Comportamiento del Elemento		
Pre-condición	Acción	Respuesta
El número de teléfono estaba completo en el campo	Se selecciona como País otro distinto a EE.UU.	El sistema añade el campo “código de país” en blanco El sistema no borra el número de teléfono existente en el campo, lo adapta al nuevo formato
El número de teléfono estaba completo en el campo	Se selecciona como País a EE.UU.	El sistema remueve el campo “código de país”. El sistema no borra el número de teléfono existente en el campo, lo adapta al formato propio del país EE.UU.

Tabla 2 – Ejemplo de utilización modelo DAR adaptado de [50]

### Ventajas

Este enfoque vincula la interfaz de usuario con los requerimientos del negocio, identificando y relacionando los requerimientos de interfaz del sistema, con los requerimientos funcionales del sistema. Esto permite, por ejemplo, detectar datos faltantes en la capa de negocio que en las interfaces de usuario son requeridos.

### **Desventajas**

Si bien existe un formato propuesto para elaborar el modelo, no hay un proceso concreto que guíe su construcción.

Tampoco se cuenta con una herramienta de software que actúe como asistente.

### **3.8 VisMatrix**

VisMatrix es una herramienta de visualización de una matriz de trazabilidad de relaciones entre requerimientos y artefactos de diseño del Software [51]. Mediante esta matriz de trazabilidad puede evaluarse el impacto de cambios en los requerimientos. La herramienta no sólo propone visualizar los vínculos candidatos, sino también la fuerza que existe entre estos vínculos para lo cual dispone de dos métricas fundamentales: 'traza la agrupación' (que mide el grado de agrupación y vinculación existente entre los requerimientos) y 'traza el alcance'.

En la Figura 20 se muestra la interfaz principal de VisMatrix, la cual se compone de celdas sombreadas que representan los vínculos entre los requerimientos (eje vertical) y el diseño/aplicación de los artefactos (eje horizontal).

Se puede distinguir que varía la intensidad de los colores en las intersecciones de ambos ejes. Esto es para distinguir el grado de acoplamiento entre los requerimientos y los demás artefactos, es decir más intensidad, corresponde mayor acoplamiento y menos intensidad equivale un acoplamiento menor.

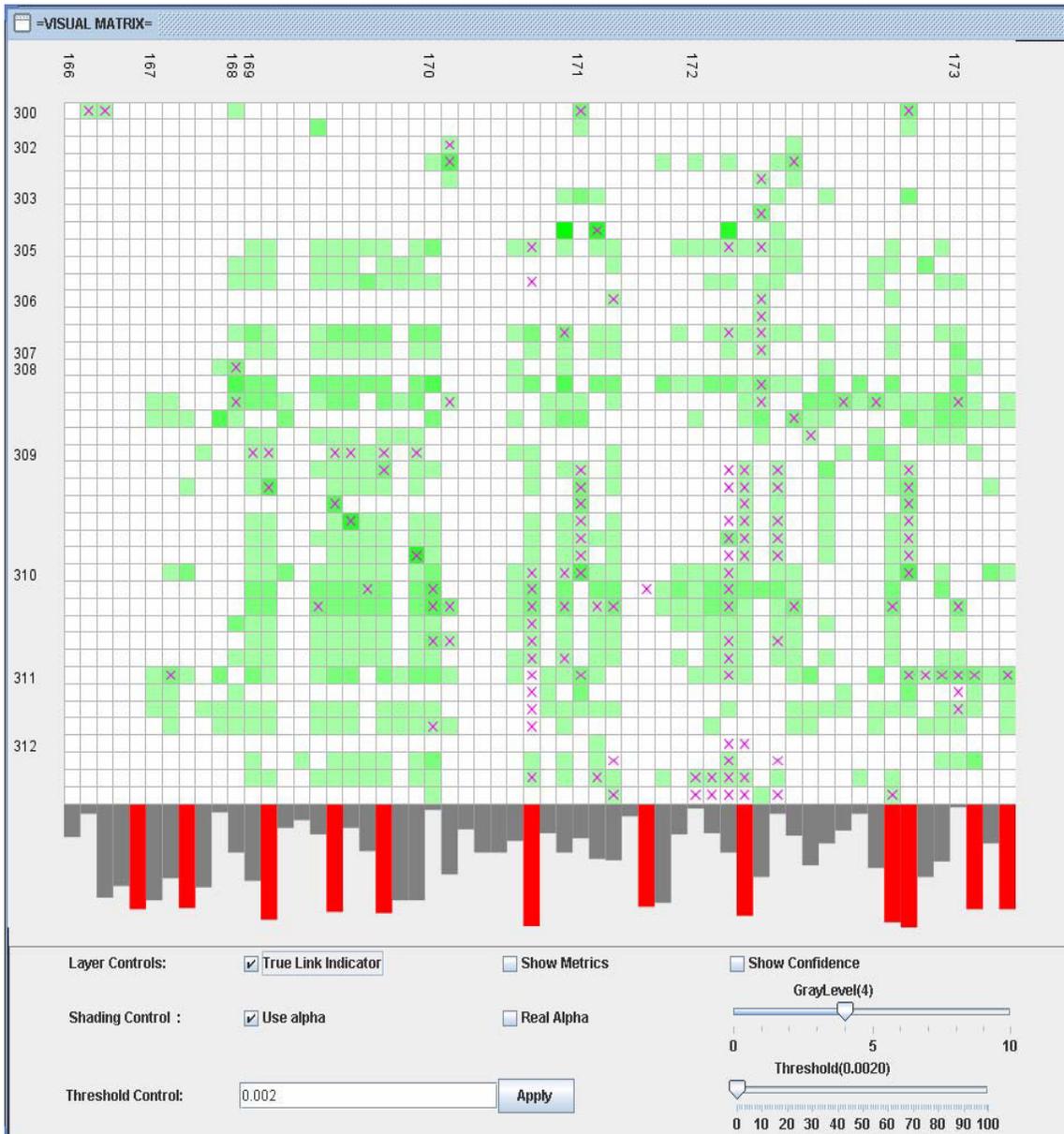


Figura 20: Interfaz principal de VisMatrix [51]

Básicamente se utiliza para los procesos de validación, verificación y gestión de cambios.

La herramienta apoya a tres objetivos específicos:

En primer lugar, proporciona una representación visual de los enlaces candidatos, con el fin de mostrar gráficamente los rastros generados entre los objetos.

En segundo lugar, proporciona información visual sobre la exactitud de los vínculos que genera; esto es útil si una organización desea llevar a cabo un estudio piloto para evaluar el uso potencial de métodos automatizados de rastreo, o para controlar su eficacia en un proyecto en curso.

Por último, se combina la información de acoplamiento y la cohesión con el conocimiento de la trazabilidad de los vínculos generados, con el fin de detectar posibles riesgos en el diseño, e identificar las necesidades que pueden ser de gran importancia arquitectónica.

### ***Ventajas***

La metodología en sí está acompañada por una herramienta para automatizar su aplicación.

Es una buena alternativa como matriz de seguimiento de cambios.

### ***Desventajas***

Los informes proporcionados por la herramienta requieren un estudio minucioso, ya que una interpretación errónea podría acarrear errores graves en la gestión de cambios de requerimientos.

### **3.9 Metodología para visualizar requerimientos en SPLE**

V-Visualise es una herramienta desarrollada especialmente para visualizar los requerimientos de software del tipo “Línea de Producto de Software” [52].

La Ingeniería de Línea de Producto de Software (SPLE) es una solución de software para la reducción de los costos de desarrollo de software propiamente dicho, los plazos de comercialización, la mejora de la calidad del software, y el aumento de la capacidad de respuesta de las empresas de software para hacer frente a las expectativas de clientes y al cambio tecnológico.

En rasgos generales, consiste en generar una línea de producción de software masivo para varias empresas con requerimientos o necesidades similares, teniendo un número manejable de variables para adaptar el software a las necesidades de cada empresa.

Una línea de productos de software se define como un conjunto de productos de software que comparten un conjunto de características comunes para satisfacer las necesidades del mercado, pero con una variabilidad significativa y predecible [52]. Otra definición establece que es un conjunto de sistemas de software que comparten un conjunto común y gestionado de aspectos, satisfacen las necesidades específicas de un segmento de mercado o misión, y son desarrollados a partir de un conjunto común de activos fundamentales (de software) de una manera prescripta [53].

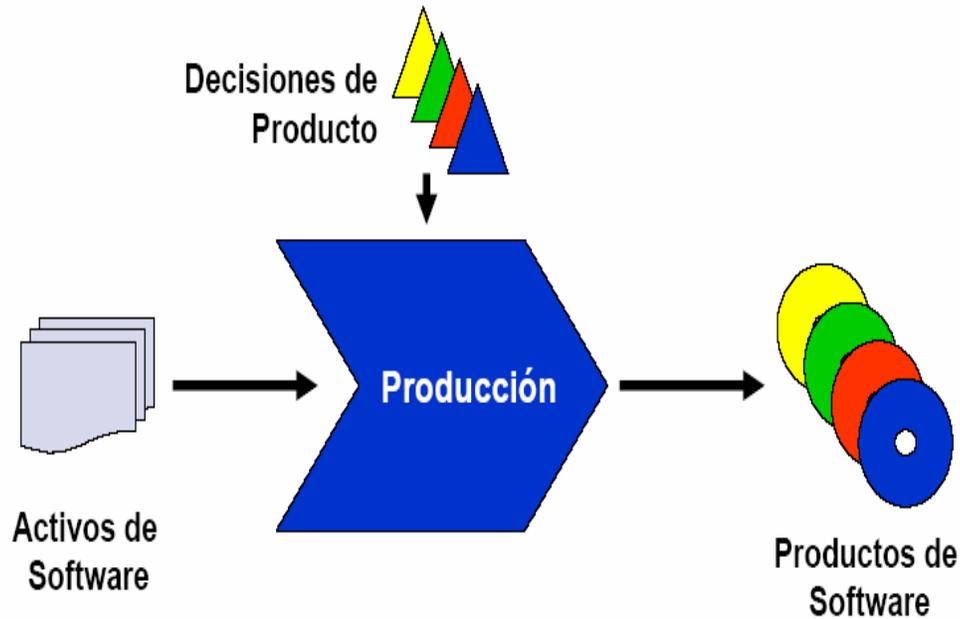


Figura 21: Modelo de Ingeniería de Línea de Producto de Software [54]

El modelo propuesto en [54], presentado en la Figura 21, consta de **Archivos de software**, es decir, una colección de partes (requerimientos, diseño de componentes, casos de prueba, etc.) que se configuran de una manera preestablecida para producir los productos de la línea. Cada producto de la línea es definido por un conjunto de **decisiones de producto** o **modelo de decisiones** que describen los aspectos de las variables y aspectos opcionales. El proceso de Producción establece los pasos para ensamblar y configurar los productos, a partir de los archivos de entradas siguiendo las pautas de las decisiones de producto. Como resultado se obtiene la salida de Productos de Software, es decir, el conjunto de todos los productos que son generados por la línea de productos.

En [52] se propone una herramienta llamada V-Visualise para visualizar los requerimientos en proyectos del tipo de línea de producto de software, para poder gestionar y visualizar los cambios de las variables.

Se establece que son necesarias dos visualizaciones para apoyar la línea de productos de ingeniería de requerimientos:

- Una visualización del modelo de decisión y
- Visualización del modelo de interdependencia

La herramienta entonces permite visualizar el modelo de decisiones que interviene en la línea de producción o lo que es lo mismo, la interdependencia que existe entre estas decisiones.

### **Ventajas**

El modelo de visualización detallada de interdependencia proporciona una ayuda para la comprensión de la interacción entre las decisiones en los requerimientos de software producido con la técnica SPLE[52].

En V-Visualise resulta fácil determinar las decisiones que forman parte de una interdependencia y cuales se ven afectadas por una o más interdependencias. También es fácil determinar cuáles son las decisiones más difíciles (es decir, en términos de impacto sobre las demás decisiones del modelo de decisión) para la reutilización, cambio y evolución [52].

### **Desventajas**

Una desventaja que presenta la herramienta es que no permite la interacción con otras herramientas utilizadas para el análisis y diseño de software, como por ejemplo Enterprise Architect [45] o Visual Paradigm [46].

La herramienta no contempla manejo de conflictos e inconsistencias entre los modelos de decisión e interdependencia.

## **3.10 ChainGraph**

El enfoque ChainGraph presentado en [55] está soportado por una herramienta que lleva el mismo nombre, ChainGraph, que propone una alternativa para visualizar las relaciones existentes entre los requerimientos. Esto permite una mayor flexibilidad y la representación de múltiples dimensiones de las relaciones entre requerimientos, por lo tanto, facilita una mejor comprensión de estas relaciones.

En particular se propone una forma de presentación que permite la exploración de los requerimientos a lo largo de sus relaciones, y facilita la comprensión de las dependencias entre ellos.

En la Figura 22 se muestra una captura de pantalla de un gráfico generado por aplicación ChainGraph, donde se representan las distintas relaciones entre requerimientos.

Dado que los requerimientos se refieren a menudo a metadatos diferentes, están conectados por cadenas diferentes que en última instancia resultará en una red de relaciones entre los requerimientos.

El usuario puede definir fácilmente las relaciones ya que se establecen de a dos requerimientos a la vez.

A fin de facilitar las relaciones de metadatos compartidos, las cadenas pueden ser resaltadas en diferentes colores. Por ejemplo en la Figura 22 es posible ver que el requerimiento “Spam assassin” (sección A de la figura) está relacionado con otros requerimientos por medio de las cadenas compartidas KW:junk y KW:spam en color verde y naranja, respectivamente. En estos casos como está relacionado a más de un requerimiento por la misma cadena, presentará alrededor tantos anillos de colores como relaciones existan.

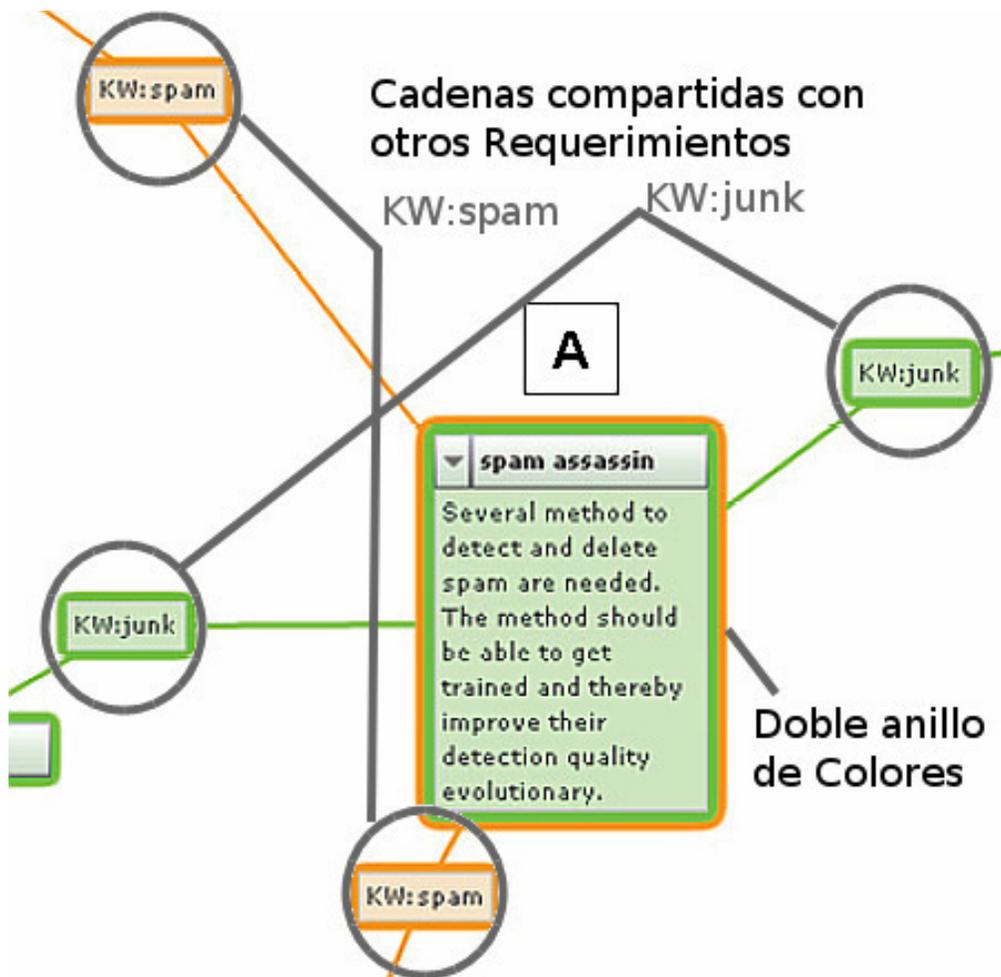


Figura 22: Ampliación zona de doble anillo, Herramienta ChainGraph [55]

### **Ventajas**

En [55], se mencionan las siguientes ventajas que aporta esta herramienta

- *Visualización unificada:* Las relaciones entre requerimientos se muestran en un sólo gráfico.
- *Representación de las relaciones directa e indirectas existentes:* las relaciones entre los requerimientos están explícitamente representadas con las etiquetas y los bordes.

- *Ampliación*: La posibilidad de ampliación flexible de un gráfico de visualización en combinación con las capacidades y el enfoque ChainGraph, proporciona un apoyo sofisticado para el análisis de las relaciones y dependencias multidimensionales de los Requerimientos de Software.

### ***Desventajas***

Solo proporciona relación entre requerimientos y no posibilita la interacción con otros artefactos, aunque según lo mencionado en [55], está previsto en futuras versiones del software, agregar dicha interacción.

# Capítulo 4

## *Clasificación de Técnicas de Visualización de Requerimientos*

---

Para visualizar requerimientos pueden utilizarse distintas técnicas dependiendo que es lo que se quiere “ver” del requerimiento y en qué momento del ciclo de vida del software, como se pudo apreciar en el capítulo anterior donde se describieron algunos enfoques de visualización de requerimientos.

De acuerdo a la actividad que realice un ingeniero de Requerimientos (Elicitar, Especificar, Validar, Analizar, entre otras), podría utilizar una técnica de Visualización de Requerimientos específica, dado que dispone de diversas posibilidades de visualizar requerimientos según la necesidad puntual del proceso.

En los últimos años (de 2006 al 2008) solo se habían presentado métodos y técnicas de visualización de requerimientos utilizadas para distintas finalidades y contextos, pero estas no estaban enmarcadas dentro de ninguna clasificación que las agrupe.

En el congreso de Visualización de Requerimientos de Software realizado el 30 de agosto del 2009 REV [42], se presentó una clasificación que intenta agrupar las distintas técnicas conocidas hasta ese momento.

### **4.1 Clasificación de enfoques de Visualización**

A continuación se detalla la clasificación sugerida en [56], que claramente agrupa a las distintas metodologías de Visualización de Requerimientos existentes según la técnica gráfica utilizada (tablas, gráficos relacionales, secuencias o metáforas)



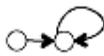
**4.1.1 Visualización Tabular.** Consiste en Visualizaciones que utilizan tablas para representar información, que generalmente se compone de una serie de intersección de las filas y columnas, cuyo contenido generalmente es información textual.

El método DAR (Modelo de Respuesta de Acción de Pantalla), detallado en el capítulo anterior, pertenece a esta clasificación.



**4.1.2 Visualización Relacional.** Este criterio incluye las técnicas de visualización que consisten en una colección de nodos y conectores, para describir o indicar una relación entre los componentes de un sistema (podría ser entre los requerimientos, o entre los requerimientos y otros artefactos del sistema), pero no describen implícitamente el orden que ocupan en las operaciones del sistema.

Un ejemplo de este tipo de técnica es la de Red Social Centrada en Requerimientos (RCSN), la técnica FLOW y SysML.



**4.1.3 Visualización Secuencial.** Abarca a las técnicas visuales que suelen transmitir la orden de la cooperación entre las partes del sistema, o de un usuario y el sistema. Estos son similares a los diagramas de flujo tradicionales o los diagramas de estado de la máquina. Es decir, las conexiones que se presentan entre los nodos denotan un orden o secuencia, a diferencia de la Visualización Relacional que solo indica que existe en una relación.-

Cabe aclarar que la técnica FLOW también podría calificar dentro de esta categoría, como así también SysML.



**4.1.4 Visualización Jerárquica.** Engloba a las Técnicas de Visualización que representan la descomposición de un sistema y sus partes, como se suele utilizar en los enfoques de modelado basado en objetivos.



**4.1.5 Visualización Cuantitativa o Metafórica.** Incluyen a las técnicas que generalmente involucran gráficos de barras, gráficos circulares u otras figuras que transmiten los datos relativos a Requerimientos. También se incluyen técnicas más sofisticadas que hacen uso de metáforas visuales y otras “pistas” visuales, valiéndose de la utilización de colores, formas, grosor de línea y de tamaños específicos para transmitir conceptos relacionados a los Requerimientos.

En esta subclasificación están incluidas las técnicas de Aproximación Metafórica, Enfoque Icónico, Enfoque simbólico y VisMatrix.

De acuerdo a la clasificación presentada, una técnica puede pertenecer a varias subcategorías. Por ejemplo las técnicas FLOW y SysML que clasifican como Visualizaciones Relacional y Secuencial.

## **4.2 Alternativas de Clasificación de enfoques de Visualización**

La clasificación presentada en los párrafos anteriores [56] corresponde a un punto de vista de la forma en que se representan las técnicas de visualización.

Sin embargo, podría proponerse otra clasificación que esté centradas en otros aspectos; como por ejemplo según a quien esta dirigida la visualización (si esta dirigida a un usuario final, a un gerente de proyecto o analista, entre otros), o de acuerdo al momento en que son utilizadas durante el proceso de Ingeriría de Requerimientos.

A continuación se proponen dos alternativas de clasificación.

### **4.2.1 Clasificación Según a Quien esta Dirigida**

Según a quien se dirige una técnica de Visualización de Requerimientos podría clasificarse en: Dirigida al Usuario del Sistema o Dirigida al Desarrollador/Analista.

- **Dirigida al Usuario del Sistema:** engloba a las técnicas visuales que están desarrolladas para poder se interpretadas por el usuario final del sistema. Un ejemplo

de esto es la metodología DAR, que es utilizada para recabar información de requerimientos, y es necesario que el usuario final del sistema la interprete.

- **Dirigida al Desarrollador / Analista:** agrupa a las metodologías que solamente puedan ser interpretadas por los Stakeholders encargados del desarrollo y elaboración del Sistema. Un ejemplo de esto podría ser las metodologías SysML y VisMatrix entre otras.

#### **4.2.2 Clasificación Según el Momento de Utilización**

Esta clasificación se refiere al momento en que se utiliza la técnica visual dentro del proceso de Requerimientos de Software.:

- **Visualización para Elicitar:** engloba a las Visualizaciones que son utilizadas en el proceso de elicitación. Un ejemplo de estas podrían ser las técnicas DAR y Red Social Centrada en Requerimientos (RCSN).
- **Visualización para Especificación:** esta clasificación agrupa a las Visualizaciones que son utilizadas como una alternativa para documentar los requerimientos del sistema. Si bien no es normal encontrar especificaciones de requerimientos descritas solamente con gráficos, puede ocurrir que las especificaciones textuales se acompañen con diagramas que representen las relaciones entre los requerimientos y otros atributos de éstos.
- **Visualización para Validación:** abarca enfoques que pueden contribuir en el proceso de validación de requerimientos con el usuario. Es necesario por lo tanto que el usuario pueda interpretarlas. Un ejemplo de esto podrían ser las técnicas de Aproximación metafórica, Enfoque Icónicos y Enfoque simbólico, entre otras.
- **Visualización para el Mantenimiento:** se incluyen los enfoques utilizados para facilitar el mantenimiento del Software. Generalmente, son a aquellos que permiten representar trazabilidad de los requerimientos y las

relaciones que existen entre los distintos artefactos del software. Un ejemplo de esta alternativa son ChainGraph, VisMatrix y V-Visualise, entre otras.

# Capítulo 5

## *Análisis comparativo de Técnicas de Visualización de Requerimientos*

---

En este capítulo se presenta una mejora al análisis comparativo propuesto en [7] de las técnicas explicadas en el capítulo 2. Se incluye a que categoría de clasificación pertenecerían según [56].

El cuadro que se presenta a continuación se elaboró a partir de la identificación de las siguientes características de cada modelo:

- 1) **Tiene soporte de Software.** Para determinar si la metodología de visualización analizada tiene o se conoce algún software que soporte su proceso de construcción/elaboración.
- 2) **Proceso formalmente definido.** Mencionando si se conoce algún proceso formal que describa los pasos a seguir para la elaboración/construcción de la representación visual en cuestión.
- 3) **Momento de Utilización.** Identifica el momento en que se utiliza el método de visualización dentro del proceso de Requerimientos de Software de acuerdo a la clasificación presentada en el punto 2 del capítulo anterior (“Clasificación Según el Momento de Utilización”).
- 4) **Entendible por el usuario.** Se refiere a si el usuario final (interesado en el desarrollo del software), es capaz de interpretar los gráficos que resultan de la aplicación del método.
- 5) **¿Qué se visualiza?** Presenta en forma breve una descripción acerca de qué permite “ver” la técnica analizada.

6) **Clasificación.** Se refiere a la categoría que pertenece la técnica, dentro de las presentadas en el capítulo anterior.

7) **Metodología asociada.** Identifica si existe una metodología conocida que se asocie con la técnica de visualización. Por ejemplo UML, DFD, etc.

### 5.1 Cuadro Comparativo

#	Método	Tiene Software de soporte	Proceso Formal Definido	Momento de Utilización	Entendible por el Usuario Final	¿Qué se visualiza?	Clasificación	Metodología Asociada
1	FLOW	No	Solo una notación	Indefinido	No Indica	Flujo de información hacia los requerimientos	Relacional y Secuencial	DFD, UML
2	SysML	Si	No indica	Indefinido	No Indica	Requerimientos Funcionales - No funcionales	Relacional y Secuencial	UML
3	Enfoque simbólico	No	Solo una notación	Validación	No Indica	Atributos Específicos de cada Requerimiento	Cuantitativa o metafórica	No Indica
4	Enfoque icónico	No	Solo una notación	Validación	No Indica	Caras con gestos, que identifican la satisfacción del usuario respecto de los atributos de los requerimientos	Cuantitativa o metafórica	No Indica
5	Aproximación Metafórica	No	Solo una notación	Validación	No Indica	El material de apoyo de cada requerimiento y el impacto entre requerimientos	Cuantitativa o metafórica	No Indica

#	Método	Tiene Software de soporte	Proceso Formal Definido	Momento de Utilización	Entendible por el Usuario Final	¿Qué se visualiza?	Clasificación	Metodología Asociada
6	Red Social Centrada en Requerimientos (RCSN)	No	Solo una notación	Elicitación	No Indica	Las intervenciones de los colaboradores durante la vida de un requerimiento	Relacional	No Indica
7	DAR	No	No Indica	Elicitación	Debería poder entenderlo	Interfaz de usuarios y la relación con Requerimientos de negocio	Tabular	No Indica
8	VisMatrix	Si	No Indica	Posterior a la validación	No es necesario	Trazabilidad de las relaciones entre los requerimientos	Cuantitativa o metafórica	No Indica
9	V-Visualise	Si	No indica	Posterior a la validación	No es necesario	Visualizar los requerimientos de software del tipo "Línea de Producto de Software" (SPLE)	Relacional	No Indica
10	ChainGraph	Si	No indica	Posterior a la validación	No es necesario	Visualización de Relación entre requerimientos en distintas dimensiones	Cuantitativa o metafórica	No Indica

## **5.2 Cuadro Comparativo. Conclusiones**

A partir de analizar el cuadro presentado en el punto anterior, se destacan las siguientes características:

- Solo dos de las diez técnicas estudiadas tiene una metodología conocida asociada. Por lo tanto, es posible inferir que el 80 % de las técnicas de Visualización de Requerimientos de Software presentadas, no están asociadas a ninguna otra metodología, por lo que carecen del respaldo de una notación estandarizada.
- Solo cuatro de las diez técnicas presentadas disponen de una herramienta o software como soporte. Es decir que más del 50% de las técnicas, no cuentan aún con un apoyo de Software que permita la fácil y correcta elaboración de los gráficos.
- Ninguna de las técnicas presenta un proceso formalmente definido que guíe en la elaboración de las representaciones visuales propuestas por cada una de ellas, dificultando la utilización de otros usuarios que no sean lo propios creadores de las técnicas.
- Ninguna de las técnicas especificadas son factibles de ser interpretada por el usuario final, aún las que son utilizadas en el proceso de elicitación. No esta debidamente documentado (mediante explicación de casos testigos), respecto a que las técnicas puedan ser interpretadas por el usuario final de Software.

Las técnicas de Visualización son muy variadas y utilizadas en diversas circunstancias y momentos del proceso de Requerimientos de Software. Esta variedad puede ocasionar que el Ingeniero en Requerimientos tenga inconvenientes al momento de elegir una técnica para graficar los requerimientos. Es deseable que se generen herramientas de software que faciliten el uso de alguno de estos enfoques.

# Capítulo 6

## Conclusiones y Trabajos Futuros

---

### **Conclusiones**

Este trabajo ha presentado un marco teórico sobre la Visualización de Requerimiento de Software, a través de la descripción de los conceptos más destacados de este tema incipiente.

Además, se describieron las técnicas de Visualización de Requerimientos de software más relevantes y sus diferencias, fundamentalmente en cuanto a su utilización y qué se representa con cada una de ellas, ampliando el trabajo presentado en [7].

En algunos de los enfoques presentados, como por ejemplo, “Enfoque Simbólico”, “Enfoque Icónico”, “Aproximación metafórica” y “Red social centrada en Requerimientos”, se han propuesto mejoras que podrían potenciar la Visualización de Requerimientos en trabajos e investigaciones futuras.

Una de las principales dificultades de la visualización consiste en hallar una adecuada representación gráfica y, por supuesto, la aplicación de visualización a requerimientos no es una excepción. Los desarrolladores de software poseen conocimientos de su disciplina, sin embargo, no son artistas gráficos. Desarrollar y perfeccionar una imagen visualmente efectiva requiere de ciertas habilidades que muchos de los desarrolladores de software, probablemente no las hayan adquirido. Por lo tanto, los desarrolladores deben utilizar los servicios de profesionales del diseño gráfico para maximizar el potencial de la visualización [8].

Es evidente que la Visualización de Requerimientos puede utilizarse con varios fines, pero su uso está asociado esencialmente a la Elicitación de Requerimientos y a la Gestión de Cambios.

Si bien intuitivamente se puede vincular la Visualización de Requerimientos a la técnica de Prototipos, este vínculo no está presente en los enfoques analizados.

Una cuestión que no está debidamente aclarada en los enfoques presentados, es el destinatario de las representaciones visuales construidas. Se puede percibir en algunos casos, que los futuros usuarios del software a construir pueden comprender dichas representaciones, pero hay enfoques que sugieren ser utilizados o estar destinados, a quienes analizan y desarrollan sistemas de software.

Por otra parte, los elementos que se visualizan son distintos en cada caso, aunque siempre bajo el concepto de Visualización de Requerimientos.

También se puede apreciar que en general, los métodos presentados carecen de procesos definidos que agilicen la construcción de las representaciones.

Además, como se menciona en [34] la comunidad de Ingeniería de Software en general no ha prestado mucha atención a las representaciones visuales de los requerimientos. Una de las posibles razones es que en su mayoría las visualizaciones tienden a ser realizadas en forma subjetivas, sin referenciar a ninguna teoría, evidencia empírica o justificación de algún tipo. Además, los expertos tienden a considerar a las visualizaciones como una cuestión estética o complementaria, en vez de considerarlas eficaces y necesarias.

Se ha presentado la única clasificación de Visualizaciones de Requerimientos reconocida, y se han propuesto dos posibles clasificaciones de técnicas de Visualización según el momento de utilización de la técnica y según a quien está dirigido.

Por último, es importante señalar que esta disciplina, Visualización de Requerimientos, es aún incipiente y requiere maduración. No obstante, constituye un avance para la Ingeniería de Requerimientos y la Ingeniería de Software en general.

Es necesario que la comunidad de Ingeniería de Requerimientos comience a dar lugar y utilizar más la Visualización para Elicitar, Especificar y Validar Requerimientos. Es importante comprender que es una herramienta necesaria, y que en los próximos años podría convertirse en imprescindible si se logra estudiar y avanzar adecuadamente en la utilización de todo su potencial.

## **Trabajos Futuros**

Una línea de investigación futura consiste en investigar los nexos posibles entre algún enfoque de Visualización de Requerimientos y los Prototipos.; o bien formalizar la técnica de prototipos como un enfoque de visualización alternativo.

Además, es necesario profundizar el estudio hacia quién están dirigidos los métodos de Visualización de Requerimientos, y definir alguna otra clasificación que pueda orientar aún más la utilización de cada enfoque, independientemente de la clasificación presentada en [56] y otras alternativas propuestas en este trabajo en el capítulo 4.

Finalmente se pretende poner en práctica el método de Aproximación Metafórica con las modificaciones y mejoras sugeridas en el capítulo 3, a modo de validar la utilidad con resultados concretos, y también estudiar la posibilidad de desarrollar una herramienta de software, de modo tal que se automatice la secuencia de pasos del Enfoque Simbólico, descrito en el capítulo 3 sección 2.1.

# Índice de Figuras

Figura 1: “Çatal Höyük”, en Anatolia (Turquía) – El Plano más antiguo de una ciudad encontrado hasta el momento .....	16
Figura 2: Atlas del año 1375, con parte de Europa, África y oriente [21] .....	17
Figura 3: Diagrama de Área polar realizado por Florence Nightingale en 1857 [21] .....	18
Figura 4: Modelo de proceso de Ingeniería de Requerimientos propuesto por Kotonya y Sommerville [27] .....	22
Figura 5: Proceso de Ingeniería de Requerimientos según Loucopoulos [24] .....	23
Figura 6: Proceso de Visualización en la Ingeniería de Requerimientos, según [8] .....	31
Figura 7: Requerimientos dibujados con Visual Paradigm [46] .....	33
Figura 8: Enfoque simbólico. Representación adaptada de [1] .....	35
Figura 9: Enfoque simbólico, con sombra. Representación adaptada de [1] .....	37
Figura 10: Enfoque Icónico. Representación adaptada de [1] .....	40
Figura 11: Propuesta de Mejora del Enfoque Icónico .....	42
Figura 12: Aproximación metafórica. Representación adaptada de [1] .....	44
Figura 13: Aproximación metafórica. Propuesta de Mejora, identificando requerimientos con colores .....	45
Figura 14: Aproximación metafórica. Propuesta de Mejora, identificando de casos de usos relacionados con los requerimientos. ....	46
Figura 15: Aproximación metafórica. Propuesta de Mejora, identificando la prioridad de los requerimientos. ....	47
Figura 16: Notación técnica FLOW [48] .....	48
Figura 17: Ejemplo utilización técnica FLOW [48] .....	49
Figura 18: Ejemplo de modelo RCSN, adaptado de [49] .....	52
Figura 19: Ejemplo de modelo RCSN, Mejorado agregando colores a los flujos y a los Stakeholders .....	54
Figura 20: Interfaz principal de VisMatrix [51] .....	58
Figura 21: Modelo de Ingeniería de Línea de Producto de Software [54] .....	61
Figura 22: Ampliación zona de doble anillo, Herramienta ChainGraph [55] .....	64

# Referencias

- 1- O.C.Z. Gotel, F. T. Marchese And S. J. Morris, "On Requirements Visualization", Second International Workshop on Requirements Engineering Visualization (REV 2007)
- 2- Alexander, I.F. and Stevens, R, "Writing Better Requirements", 2002
- 3- Kovitz, B.L., "Practical Software Requirements: A Manual of Content and Style, Manning", 1998
- 4- REV'06, "First International Workshop on Requirements Engineering Visualization (REV'06)", 2006, <http://swen.uwaterloo.ca/~chpkim/re06/>
- 5- REV'07, "Second International Workshop on Requirements Engineering Visualization (REV'07)", 2007, <http://csis.pace.edu/~ogotel/professional/REV07.html>
- 6- REV'08, "Third International Workshop on Requirements Engineering Visualization (REV'08)", 2008, <http://csis.pace.edu/~ogotel/professional/REV08.html>
- 7- Bruno Ghione y Pablo Thomas, "Visualización de Requerimientos de Software: un estudio comparativo", CACIC 2009 - Jujuy - Argentina
- 8- P.Parry, M. Ozcan and J. Siddiqi, "The Application of Visualisation to Requirements Engineering", Computing Research Centre,Sheffield Hallam University, England.
- 9- P.T. Cox, F.R. Giles, T. Pietrzykowski, " PROGRAPH: A Step Towards Liberating Programming from Textual Conditioning", 1989 IEEE Workshop on Visual Languages, October 4-6, 1989, Rome, Italy, IEEE ComputerSociety Press, ISBN 0 8186 2002 1
- 10- Blaine A. Price, Ian S. Small, Ronald M. Baecker, "A Taxonomy of Software Visualisation", Proceedings of 25th Hawaii International Conference on System Sciences, 7-10 January, 1992,p595-606
- 11- Real Academia Española, "Diccionario de la Real Academia Española ", Consultado diciembre 2009, <http://www.rae.es/RAE/Noticias.nsf/Home?ReadForm>

- 12- Oxford University," Oxford English Dictionary", 1989
- 13- InfoVis.net, "La revista digital de InfoVis.net", Consultado en diciembre del 2009,  
<http://www.infovis.net/printRec.php?rec=revista&lang=1>
- 14- Luis Miguel de la Cruz Salas , "Visualización Científica", Revista Digital Universitaria - dic  
2005 - Volumen 6 Número 12 • ISSN: 1067-6079
- 15- Christopher L. C. E. Witcombe, "Art History: Resources for the study of art history",  
Consultado Abril 2010, <http://witcombe.sbc.edu/ARTHprehistoric.html>
- 16- Wikipedia, "Visualización en Wikipedia", consultado Abril 2010,  
<http://en.wikipedia.org/wiki/Visualization>
- 17- Suárez Roldán, "Animación y Visualización de Fenómenos Naturales", 2003 - Tesis  
Licenciatura. Ingeniería en Sistemas Computacionales. Departamento de Ingeniería en  
Sistemas Computacionales, Escuela de Ingeniería, Universidad de las Américas Puebla
- 18- Bosque Sendra, J. y Zamora Ludovic, H, "Visualización Geográfica y nuevas  
Cartografías", 2002 - GeoFocus(Artículos), nº 2, p. 61-77. ISSN: 1578-5157
- 19- Shakti," Visualización Creativa Título Original: Creative Visualization", 2005
- 20- Dr. Maxwell Maltz," Psico Cibernética", Editorial Open Prject Books
- 21- M. Friendly and D.J. Denis, "Hitos en la Historia de la cartografía temática, gráficos  
estadísticos y visualización de datos", Consultado en diciembre del 2009,  
<http://datavis.ca/milestones/>
- 22- Michael Friendly," Milestones in the history of thematic cartography, statistical graphics,  
and data visualization", 2008
- 23- The Institute of Electrical and Electronics Engineers, " IEEE Standard Glossary of  
Software Engineering Terminology", New York, Std.610.12-1990, 1990
- 24- Loucopoulos P. Karakostas V.," System Requirements Engineering", 1995, London
- 25- Davis Alan M.," Software Requirements, Objetos,Funciones y Estados", Editorial Prentice  
Hall, ISBN 0-13-805763-X, 1993.
- 26- Jitnah, D., Han, J., Steele, P., "Software Requirements Engineering: An Overview",  
Technical Report 95-04, Peninsula School of Computing and Information Technology,  
Monash University, Melbourne, Australia, September (1995).

- 27- Katonya, G., Sommerville, "Requirements Engineering Processes and Techniques", Chichester, John Wiley & Sons (1998)
- 28- Van Lamsweerde, A., "Requirements engineering in the year 00: a research perspective", International Conference on Software Engineering archive, Proceedings of the 22nd international conference on Software engineering, Limerick, Ireland, ACM Press, ISBN:1-58113-206-9, (2000), 5 - 19
- 29- Pablo Javier Thomas, "Definición de un Proceso de Elicitación de Objetivos", Octubre 2005, Facultad de Informática - Universidad Nacional de La Plata
- 30- Borland®, "Mitigating Risk With Effective Requirements Engineering", Abril 2005
- 31- Angela Tuffley, "CIT3190 IT Project Course. Requirements Elicitation & Management", 2005
- 32- M. Stephens, P. Bates, "Controlling Prototyping and Evolutionary Development", Proceedings 4th International Workshop on Rapid System Prototyping: Shortening the Path From Specification To Prototype - Junio 1993
- 33- S. Wadhwa, "Visualizing Requirements, Requirements Quarterly: The Newsletter of the Requirements Engineering Specialist Group of the British Computer Society - RQ42", Diciembre 2006
- 34- Daniel L. Moody, Patrick Heymans, Raimundas Matulevicius; "Improving the Effectiveness of Visual Representations in Requirements Engineering: An Evaluation of i\* Visual Syntax", 2009 17th IEEE International Requirements Engineering Conference.
- 35- Philip T. Cox, "Picture The Future", Object Magazine, Vol 3, No 2, p46-49, 1993
- 36- G.O. Domik, "Scientific Visualisation", Educational Multimedia & Hypermedia Annual Conference - Proceedings of ED-MEDIA '93 -World Conf. on Education Media and Hypermedia, 23-26 June 1993, Orlando, USA, p153-160, ISBN 1 88009 406 1
- 37- P.E. Mantey, "A View of Visualisation - Its Origins, Development and New Directions", SPIE - Proceedings of the Society of Photo-Optical Instrumentation Engineers
- 38- G. Webster, "Scientific Visualisation - Great Hope or Great Hype?", Proceedings of Computer Graphics '90 Conference 6-8 November 1990, London, UK
- 39- Heinz-Dieter Bocker, Gerhard Fischer, Helga Nieper., "The Enhancement of Understanding Through Visual Representations", Proceedings of the CHI'86 Conference -

Human Factors In Computing Systems III 13-17-1986, Boston, USA, Edited by M. Mantei, P. Orbeton ISBN 0-444-70052-8, p44-50

- 40- T.G. West, "Visual Thinkers, Mental Models, and Computer Visualisation", Proceedings of Conference on Interactive Learning Through Visualisation, 1991
- 41- Colin Potts, "Expediency and Appropriate Technology: An Agenda for Requirements Engineering Research in the 1990s", Lecture Notes in Computer Science (550) - ESEC '91 Proceedings, 21-24 October 1991
- 42- REV'09, "Fourth International Workshop on Requirements Engineering Visualization", 2009, <http://csis.pace.edu/~ogotel/professional/REV09.html>
- 43- OMG, OMG Systems Modeling Language (OMG SysML™), 2007
- 44- wikipedia, "SysML en wikipedia", consultado en Mayo 2009, <http://es.wikipedia.org/wiki/SysML>
- 45- Sparx Systems, "Pagina oficial del Software Enterprise Architect", Consultada en Abril 2010, <http://www.sparxsystems.com.au>
- 46- visual-paradigm, "Pagina Oficial de Visual Paradigm", Consultado en Abril 2010, <http://www.visual-paradigm.com>
- 47- James & Suzanne Robertson, "Volere Requirements Specification Template", consultado en diciembre 2009, <http://www.volere.co.uk/template.htm>
- 48- Kurt Schneider, Kai Stapel, Eric Knauss, "Beyond Documents: Visualizing Informal Communication", Requirements Engineering Visualization(REV'08)
- 49- I. Kwan, D. Damian and Margaret-Anne Storey, "Visualizing a Requirements-centred Social Network to Maintain Awareness Within Development Teams", First International Workshop on Requirements Engineering Visualization (REV'06)
- 50- Joy Beatty and Mike Alexander, "Display-Action-Response Model for User Interface Requirements: Case Study", Second International Workshop on Requirements Engineering Visualization (REV'07)
- 51- C. Duan and J. Cleland-Huang, "Visualization and Analysis in Automated Trace Retrieval", First International Workshop on Requirements Engineering Visualization (REV'06)

- 52- David Sellier, Mike Mannion, "Visualising Product Line Requirement Selection Decision Inter-dependencies", Second International Workshop on Requirements Engineering Visualization (REV'07)
- 53- Clements, P. & Northrop, L., "Software Product Lines: Practices and Patterns", Boston, 2002
- 54- SPLE, "Software Product Lines", Consultado Enero 2010, [www.softwareproductlines.com](http://www.softwareproductlines.com)
- 55- Philipp Heim, Steffen Lohmann, Kim Lauenroth, Jürgen Ziegler, "Graph-based Visualization of Requirements Relationships", Requirements Engineering Visualization(REV'08)
- 56- John R. Cooper Jr., Seok-Won Lee, Robin A. Gandhi and Orlena Gotel, "Requirements Engineering Visualization: A Survey of the State-of-the-Art ", Fourth International Workshop on Requirements Engineering Visualization (REV'09)

**Anexo I:**  
**Publicaciones**  
**Relacionadas**

Se adjunta a continuación el trabajo presentado en el CACIC 2009 relacionado al tema de Visualización de Requerimientos: "Visualización de Requerimientos de Software: un estudio comparativo", CACIC 2009 - Jujuy - Argentina

# Visualización de Requerimientos de Software: un estudio comparativo

Bruno Sebastián Ghione<sup>1</sup>, Pablo Thomas<sup>2</sup>

<sup>1</sup>Lic. en Sistemas – Estudiante de Maestría en Ingeniería de Software – UNNE-UNLP  
[brunoghione@yahoo.com.ar](mailto:brunoghione@yahoo.com.ar)

<sup>2</sup> Prof. Adjunto Dedicación Exclusiva. III-LIDI-Facultad de Informática – UNLP  
[pthomas@lidi.info.unlp.edu.ar](mailto:pthomas@lidi.info.unlp.edu.ar)

**Resumen.** La Visualización de Requerimientos es una metodología que ha surgido recientemente. Diferentes técnicas han sido desarrolladas para transformar Requerimientos expresados en forma verbal o escrita en representaciones visuales. Muchas de estas técnicas utilizan gráficos para “visualizar” esos Requerimientos, los cuales logran capturar la atención de los participantes del proceso.

En el presente trabajo, se realiza una introducción a esta nueva metodología y se analizan diferentes técnicas de Visualización, con un enfoque que pretende establecer un marco de base sobre el tema. Se presenta un análisis comparativo de las técnicas estudiadas.

**Keywords:** Visualización de Requerimientos, Visualización en la Ingeniería de Requerimientos, Ingeniería de Requerimientos.

## 1. Introducción

La Ingeniería de Requerimientos es una disciplina relativamente nueva dentro de la Ingeniería de Software, que establece la necesidad de utilizar un proceso que guíe al analista durante la recolección y especificación de requerimientos. Dicho proceso está asistido por diversas técnicas, aunque muchas veces éstas no resultan suficientes.

Se ha utilizado la Visualización en varias ramas de la Ingeniería y en otras ciencias, para facilitar la comprensión del producto a construir o transmitir.

La Visualización contribuye a la Ingeniería de Requerimientos ya que puede mejorar la comprensión de los Requerimientos, así como también sirve de apoyo a la negociación y a la definición de prioridades de los Requerimientos.

Además, se ha utilizado la Visualización principalmente para desarrollar tres aspectos [1]: transmitir la estructura y la evolución de las relaciones entre un conjunto de Requerimientos de software y otros artefactos, apoyar la organización de las necesidades y la gestión del cambio, y asistir a sesiones de estimulación con los Requerimientos y las actividades de análisis.

Sin embargo, a pesar de la popularidad de los lenguajes visuales de modelado, los Requerimientos aún tienden a ser escritos en un formato textual y narrativo [1], [2] y

[3]. El presente trabajo muestra el uso de la Visualización en la Ingeniería de Requerimientos y algunas de las técnicas más recientemente difundidas en [4], [5] y [6].

En la sección 2 se detallan algunos conceptos esenciales. A continuación se presentan los enfoques más representativos de Visualización de Requerimientos. En la sección 4 se define un cuadro comparativo de estos enfoques. Finalmente se establecen conclusiones y se definen los posibles trabajos futuros.

## **2. Conceptos básicos sobre Visualización de Requerimientos**

Según [7], se entiende por Visualización como la acción de representar mediante imágenes ópticas fenómenos de otro carácter, o al acto de formar en la mente una imagen visual de un concepto abstracto. También se define Visualización a la acción de imaginar con rasgos visibles algo que no se tiene a la vista.

En [1] y [8] se define a la Visualización como el acto de formación de una visión mental (la imagen).

Por otra parte, un Requerimiento es una condición o capacidad que necesita un usuario para resolver un problema o alcanzar un objetivo, o una condición o capacidad que se debe cumplir o poseer un sistema para satisfacer un contrato, norma, especificación, u otros documentos formalmente impuestas [9]

Finalmente, la Visualización de Requerimientos es el acto de representar esas condiciones o capacidades que necesita un usuario para resolver un problema o para alcanzar un objetivo, mediante la utilización de imágenes visuales.

En [10] se concibe a la Visualización de Requerimientos como una técnica gráfica que incluye dibujo de prototipo de pantallas, gráficos y modelos, que pueden ayudar a facilitar una sesión de Requerimientos.

Respecto del uso de la Visualización durante la Elicitación de Requerimientos, no hay un proceso formalmente definido; sin embargo, hay pautas esenciales que suelen respetarse para utilizarla como soporte.

En [10] se proponen los siguientes pasos simples:

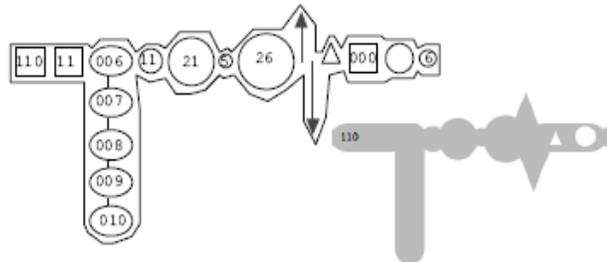
1. El analista de Requerimientos captura los requerimientos en formato de texto a partir de los futuros usuarios del sistema a construir.
2. El analista crea prototipos de diseño de pantalla en papel (esbozo a mano alzada de las pantallas), y los muestra a los usuarios.
3. Los usuarios informan y proponen Requerimientos a partir de lo que han visto.
4. El analista crea en detalle los requerimientos formales.

De este modo, la Visualización contribuye a obtener los Requerimientos de los usuarios, particularmente los relacionados con la interfase del futuro sistema.

## **3. Enfoques de Visualización de Requerimientos**

A continuación, se describen brevemente los enfoques de Visualización de Requerimientos más destacados, presentados en [5], [6] y [4].

- **Systems Modeling Language (SysML):** Standard de extensión de UML para representación de requerimientos [11]. Es un lenguaje, subconjunto ampliado de UML 2.0, utilizado para especificación de sistemas. Desde Septiembre de 2007 es un estándar del Grupo de Gestión de Objetos (Object Management Group - OMG). Es un lenguaje desarrollado como perfil de UML 2.0 desde la Meta-Object Facility (MOF). Se compone de cuatro tipos de diagramas principales, uno de los cuales es SysML diagrama de Requerimientos [12]. SysML provee las construcciones de modelado para representar los requerimientos basados en texto. Un requerimiento es representado mediante un estereotipo de una Clase UML que cuenta con un identificador único y el texto del requerimiento. El estándar permite la representación de relaciones entre requerimientos y otros componentes (como clases o casos de uso), por el cual se especifica si es una relación de dependencia o inclusión (un requerimiento depende de otro requerimiento para su realización o un requerimiento esta compuesto por otros requerimientos.).
- **Enfoque simbólico** [1]: mediante gráficos compuestos por cuadros, círculos y flechas, se permite la visualización de atributos asociados a cada requerimiento, lo cual facilita la detección de requerimientos más complejos. Cada requerimiento se representa gráficamente con un formato distinto, según el valor de los atributos. La figura 1 muestra un ejemplo de este enfoque. Los autores [1] se basan en las especificaciones de Volere [13] para describir los atributos de un requerimiento. En este ejemplo se incluye en primer lugar, de izquierda a derecha, el número de requerimiento (110), el tipo de requerimiento (11), eventos o casos de uso relacionados con el requerimiento (lista de óvalos, 006, 007, 008, 009, 010); de esta manera se provee a cada requerimiento una representación diferente dependiendo del valor de los atributos que lo componen.



**Fig. 1** Ejemplo de representación de un requerimiento utilizando el enfoque simbólico [1].

- **Enfoque icónico:** también mencionado en [1], consiste en una representación gráfica de los requerimientos utilizando caras o iconos gestuales. Su diseño permite una evaluación “cruda” sobre la certeza de los datos presentes en los requerimientos. Se dibuja una cara por cada atributo de requerimiento, y cada requerimiento ocupa una línea de rostros. Se utilizan dos formas para reflejar el tipo de requerimiento. Un círculo se usa para requerimientos funcionales, y un pentágono se utiliza para requerimientos No funcionales. Cabe aclarar que los gestos utilizados se limitan a la presentación de la boca del rostro en tres posiciones. Por ejemplo, para representar el grado de satisfacción del cliente

respecto al requerimiento, si el cliente está muy contento, la boca sonríe, si el cliente muestra desinterés la boca se representa recta, y cuando existe un conflicto o esta disconforme se representará con una cara de disgusto.

- **Aproximación Metafórica:** utiliza una metáfora simple para transmitir la información relativa a la estabilidad de un conjunto de requerimientos. Se centra en la representación gráfica de un subconjunto de atributos de los requerimientos para ayudar a responder a las preguntas: ¿qué requerimientos son susceptibles de cambios? ¿Dónde están los inminentes problemas? Cada requerimiento se representa como una isla con un pequeño volcán. Si existen dependencias entre los requerimientos, las islas están relacionadas con calzadas o agrupadas en función de la naturaleza de la dependencia. El tipo de requerimiento no se manifiesta en la visualización, pero esto podría lograrse por la forma o el color del volcán. El tamaño del volcán es proporcional a la cantidad de material de apoyo, y las nubes sobre el volcán significa el nombre de una fuente [1]. En la figura 2 se representan tres requerimientos, se aprecia la fuente de cada requerimiento, como así también las dependencias que existen entre los requerimientos (calzadas bajo los volcanes). Además es factible observar, que el requerimiento #75 se caracteriza por tener un abundante material de apoyo.

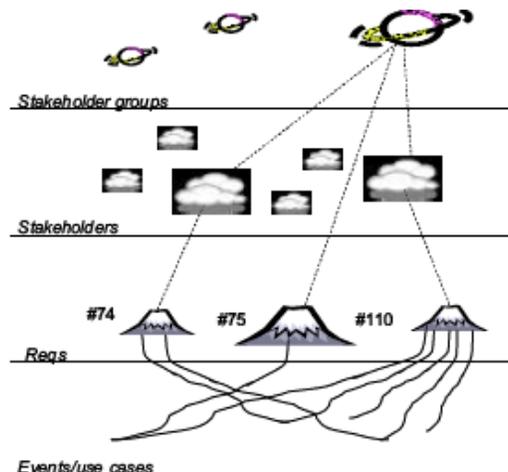


Fig. 2 Ejemplo de Aproximación Metafórica [1].

- **FLOW:** es una propuesta de lenguaje de Visualización para representar flujos de los requerimientos. En [14] se presenta la notación sugerida para utilizar en la visualización de los flujos de información que se presentan en los requerimientos. El objetivo de esta técnica es dejar rastros de la información que subyace a los requerimientos y que muchas veces no es formal. Es decir, intenta mantener un vínculo entre los requerimientos y la comunicación informal que los nutre. La figura 3 presenta un ejemplo de FLOW donde se describe en su lenguaje la representación de una nueva técnica de manipulación de requerimientos de

seguridad denominada “SecReq”. Se visualiza las entradas de las partes interesadas y de la utilización de una herramienta llamada “UMLsec”.

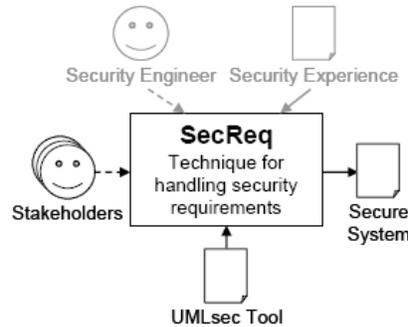


Fig. 3 Ejemplo de Visualización de Requerimientos utilizando la técnica FLOW [14].

- Red Social Centrada en Requerimientos (RCSN);** se utiliza fundamentalmente para visualizar los colaboradores que intervienen a lo largo de toda la vida de un requerimiento. Mediante la utilización de círculos, cuadrados y flechas se establecen los flujos de información entre los colaboradores y los requerimientos, lo cual garantiza que todas las personas involucradas sean notificadas de las novedades que puedan ocurrir sobre cada requerimiento [15]. En la figura 4 se expone un ejemplo de la representación de un único requerimiento mediante el diagrama RCSN. Las flechas representan los flujos de información, especifican según su espesor la cantidad de información transmitida, la dirección de la información, el medio utilizado (mail, chat, un documento formal, entre otros) y hasta el rol de cada persona que interviene, dibujando cada rol con diferentes símbolos.

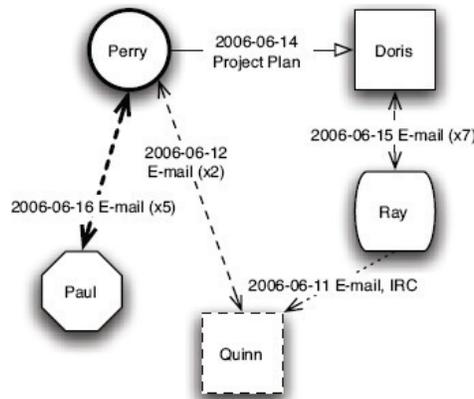


Fig. 4 –Ejemplo de diagrama de Red social centrada en Requerimientos (RCSN) [15]

- **Modelo de Respuesta de Acción de Pantalla - Display-Action-Response (DAR)** este modelo se utiliza para capturar requerimientos de la interfaz de usuario y reglas de negocio [16]. El objetivo es proporcionar una imagen completa del sistema, vinculando las pantallas de interfaz de usuarios con los requerimientos de negocio.
- **VisMatrix:** es una herramienta de visualización de una matriz de trazabilidad de las relaciones entre requerimientos y artefactos de diseño del Software [17]. Mediante dicha matriz, puede evaluarse el impacto de cambios en los requerimientos. La herramienta no sólo se propone visualizar los vínculos candidatos, sino también la fuerza que existe entre estos vínculos para lo cual dispone de dos métricas fundamentales: ‘traza la agrupación’ y ‘traza el alcance’. En la figura 5 se muestra la interfaz principal de VisMatrix, la cual se compone de celdas sombreadas que representan los vínculos entre los requerimientos (eje vertical) y el diseño/aplicación de los artefactos (eje horizontal)

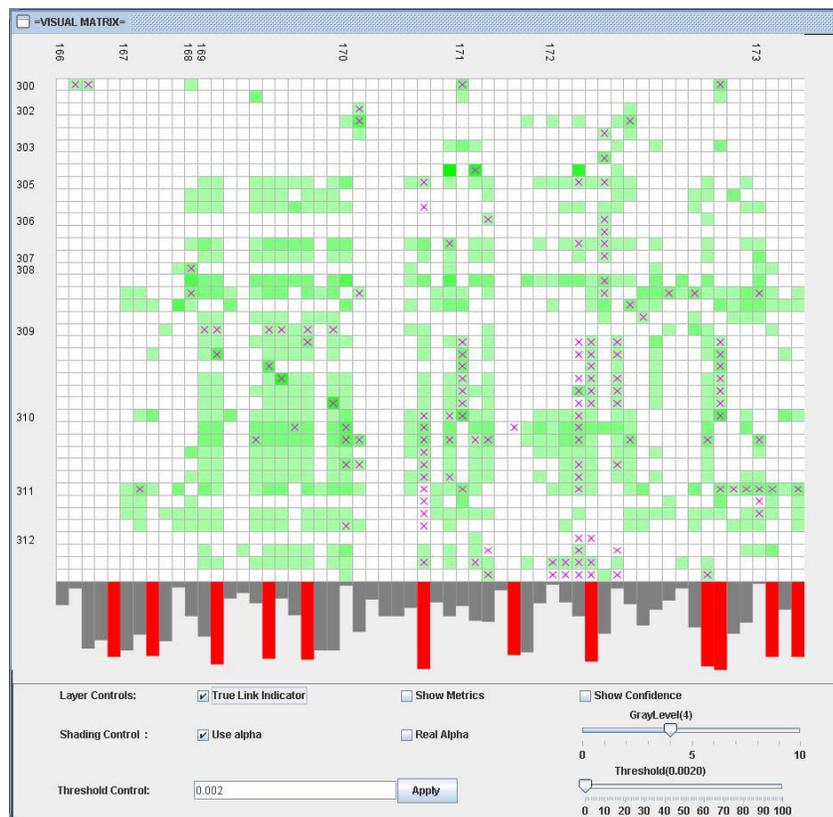


Fig. 5 –Interfaz principal de VisMatrix [17]

#### 4. Cuadro Comparativo de enfoques

A continuación, se presenta un cuadro de análisis comparativo entre los métodos y ejemplos de visualización presentados.

#	Método	Tiene Software de soporte	Proceso Formal Definido	Utilizado en Elicitación	Entendible por el Usuario	¿Qué se visualiza?	Metodología Asociada
1	FLOW	No	Solo una notación	No	No Indica	Flujo de información hacia los requerimientos	DFD, UML
2	SysML	Si	No indica	No	No Indica	Requerimientos Funcionales - No funcionales	UML
3	Enfoque simbólico	No	Solo una notación	No Indica	No Indica	Atributos Específicos de cada Requerimiento	No Indica
4	Enfoque icónico	No	Solo una notación	No Indica	No Indica	Caras con gestos	No Indica
5	Aproximación Metafórica	No	Solo una notación	No Indica	No Indica	El material de apoyo de cada requerimiento y el impacto entre requerimientos	No Indica
6	Red Social Centrada en Requerimientos (RCSN)	No	Solo una notación	Si	No Indica	Las intervenciones de los colaboradores durante la vida de un requerimiento	No Indica
7	DAR	No	No Indica	Si	Debería poder entenderlo	Interfaz de usuarios y la relación con Requerimientos de negocio	No Indica
8	VisMatrix	Si	No Indica	No	No es necesario	Trazabilidad de las relaciones entre los requerimientos	No Indica

## 4. Conclusiones

Se han presentado las técnicas de Visualización de Requerimientos más relevantes y sus diferencias, fundamentalmente en cuanto a su utilización y qué se representa con cada una de ellas.

Es evidente que la Visualización de Requerimientos puede utilizarse con varios fines, pero su uso está asociado esencialmente a la Elicitación de Requerimientos y a la Gestión de Cambios.

Si bien intuitivamente se puede vincular a la Visualización de Requerimientos a la técnica de Prototipos, este vínculo no está presente en los enfoques analizados.

Una cuestión que no está debidamente aclarada en los enfoques presentados, es el destinatario de las representaciones visuales construidas. Se puede percibir en algunos casos, que los futuros usuarios del software a construir pueden comprender dichas representaciones, pero hay enfoques que sugieren ser utilizados o estar destinados a quienes analizan y desarrollan sistemas de software.

Por otra parte, los elementos que se visualizan son distintos en cada caso, aunque siempre bajo el concepto de Visualización de Requerimientos.

También se puede apreciar que en general, los métodos presentados carecen de procesos definidos que agilicen la construcción de las representaciones.

Por último, esta disciplina (Visualización de Requerimientos) es aún incipiente y requiere maduración. No obstante, constituye un avance para la Ingeniería de Requerimientos y la Ingeniería de Software en general.

## 5. Trabajos Futuros

Un línea de investigación futura consiste en investigar los nexos posibles entre la Visualización de Requerimientos y los Prototipos.

Además, es necesario estudiar hacia quién están orientados los métodos de Visualización de Requerimientos, y definir alguna clasificación que pueda orientar aún más la utilización de cada enfoque.

Finalmente, se pretende definir un proceso que guíe la construcción de representaciones visuales para alguno de los enfoques presentados.

## Referencias

1. O.C.Z. Gotel, F. T. Marchese And S. J. Morris - On Requirements Visualization - Second International Workshop on Requirements Engineering Visualization (REV 2007)
2. Alexander, I.F. and Stevens, R. - Writing Better Requirements - Addison-Wesley, 2002.
3. Kovitz, B.L. - Practical Software Requirements: A Manual of Content and Style, Manning - Publications Co., 1998.
4. First International Workshop on Requirements Engineering Visualization (REV'06) - <http://swen.uwaterloo.ca/~chpkim/re06/>

5. Second International Workshop on Requirements Engineering Visualization (REV'07) - <http://csis.pace.edu/~ogotel/professional/REV07.html>
6. Third International Workshop on Requirements Engineering Visualization (REV'08) - <http://csis.pace.edu/~ogotel/professional/REV08.html>
7. Diccionario de la Real Academia Española (<http://www.rae.es/RAE/Noticias.nsf/Home?ReadForm>)
8. Oxford English Dictionary, 1989
9. IEEE Standard Glossary of Software Engineering Terminology, The Institute of Electrical and Electronics Engineers, New York, Std.610.12-1990, 1990
10. S. Wadhwa - Visualizing Requirements, Requirements Quarterly: The Newsletter of the Requirements Engineering Specialist Group of the British Computer Society - RQ42, Diciembre 2006.
11. OMG Systems Modeling Language (OMG SysML™), V1.0 2007
12. SysML en wikipedia - <http://es.wikipedia.org/wiki/SysML> (consultado en Mayo 2009)
13. James & Suzanne Robertson - Volere Requirements Specification Template - <http://www.volere.co.uk/template.htm>
14. Kurt Schneider, Kai Stapel, Eric Knauss - Beyond Documents: Visualizing Informal Communication - 2008 Requirements Engineering Visualization(REV'08)
15. I. Kwan, D. Damian and Margaret-Anne Storey - Visualizing a Requirements-centred Social Network to Maintain Awareness Within Development Teams - First International Workshop on Requirements Engineering Visualization (REV'06)
16. Joy Beatty and Mike Alexander - Display-Action-Response Model for User Interface Requirements: Case Study - Second International Workshop on Requirements Engineering Visualization (REV 2007)
17. C. Duan and J. Cleland-Huang -Visualization and Analysis in Automated Trace Retrieval - First International Workshop on Requirements Engineering Visualization (REV'06)