



Facultad de Informática  
Universidad Politécnica  
de Madrid



Escuela de Posgrado  
Instituto Tecnológico  
de Buenos Aires

# Tesis de Máster en Ingeniería del Software

## “Sistema Experto Asistente de Requerimientos”

***Ing. Francisco Marcelo Rizzi***

***Directores:***

***Dr. Mariano Fernández López (UPM)***

***Dr. Ramón García Martínez (ITBA)***

**Buenos Aires, Argentina  
2001**

# Indice

## **Capítulo 1: Introducción**

1.1 Introducción	2
1.2 Descripción de la composición de la tesis.	3

## **Capítulo 2: Dominio de la aplicación**

2.1 introducción	6
2.2 Problemas de Software	6
2.3 Marcos de Problema	9
2.3.1 Diagramas de marcos	10
2.3.2 Marcos de problema elementales	12
Marco de problema de información	12
Marco de problema de control	14
Marco de problema de transformación	14
Marco de Problema Workpieces	15
Marco de Problema de Conexión	16
2.4 Marcos de Problema Compuestos	17
2.5 Descomposición del problema	19
2.5.1 Descomposición Exterior-Interior	20
2.5.2 Descomposición interior-Exterior	20
2.5.3 Reconocimiento de un marco compuesto	21
2.6 Documentación: Contenido del documento de requerimientos	21

### **Capítulo 3: Definición del problema**

3.1 Introducción	26
3.2 Planteamiento del problema	26
3.3 Objetivos del trabajo	27
3.4 Alcance	28
3.5 Participantes y ámbito del proyecto	29
3.6 Metodología de construcción: IDEAL	29
3.7 Gestión del proyecto	33
3.8 Diagrama de Gantt.	33

### **Capítulo 4: Estudio de la Viabilidad**

4.1 Introducción	36
4.2 Test de viabilidad	37
4.3 Funcionamiento de la técnica	40
4.4 Análisis del test de viabilidad	41
4.4.1 Calculo de los intervalos correspondientes a cada dimensión.	47
4.4.2 Justificación del análisis de viabilidad	49
4.4.2.1 Justificación de la dimensión de Plausibilidad	49
4.4.2.2 Justificación de la dimensión de Justificación	50
4.4.2.3 Justificación de la dimensión de Adecuación	52
4.4.2.4 Justificación de la dimensión de Éxito.	55

## **Capítulo 5: Adquisición de conocimientos**

5.1	Introducción	60
5.2	Ciclo de cada sesión	61
5.2.1	Preparación de las sesiones	
5.2.2	Sesión	
5.2.3	Transcripción	
5.2.4	Análisis de la sesión	
5.3	Sesiones de adquisición	
5.3.1	Sesión 1: Primera reunión con los expertos.	62
5.3.2	Sesión 2: profundización de conocimientos	64
5.3.3	Sesiones 3, 4, 5 y 6: Otras sesiones de profundización de conocimientos	68
5.3.4	Sesión 7: Sesión de emparillado	82
5.3.5	Sesión 8: Estándares	89
5.3.6	Sesión 9: Ajustes de Marcos	91
5.3.7	Sesión 10: Análisis del problema con marcos	94
5.3.8	Sesión 11: Sesión de análisis de protocolos	100
5.3.9	Sesión 12: Profundización de ajuste de marcos al problema	114
5.3.10	Sesión 13: Análisis de conexiones	116
5.4	Sesiones de extracción de conocimientos	118

## **Capítulo 6: Conceptualización**

6.1	introducción	135
6.2	Elaboración del modelo conceptual	136
6.2.1	Identificación, Comparación y Categorización de Conceptos	137

6.2.1.1	Glosario de Términos	138
6.2.1.2	Razones para la elección de los conceptos	141
6.2.1.3	Tabla Concepto/Atributo/Valor	143
6.2.2	Relaciones entre conceptos: El conjunto relacional de base	145
6.2.3	Análisis de los conocimientos estratégicos	146
6.2.3.1	Pasos de Alto nivel	148
6.2.3.2	Subpasos de la tarea	150
6.2.3.3	Comprobación de los conocimientos estratégicos	155
6.2.4	Análisis de los conocimientos tácticos	156
6.2.4.1	Representaciones Intermedias	156
6.2.4.2	Metaconocimientos Tácticos	175
6.2.5	Análisis de conocimientos fácticos	177
6.2.6	Modelo Dinámico o de Proceso	188
6.2.7	El mapa de conocimientos: comprobación del MC	195

## **Capítulo 7: Formalización de conocimientos**

7.1	Introducción	197
7.2	Selección de formalismos	197
7.3	Representación de los conocimientos en marcos	198
7.4	Relaciones entre conceptos	199
7.5	Propiedades de los conceptos	202
7.5.1	Facetas de propiedades	204
7.5.1.1	Facetas que definen propiedades de clase, de instancia y de relación.	204
7.5.1.2	Facetas que definen propiedades de clase y de relaciones.	204

7.5.1.3 Facetas que definen propiedades de instancia	205
7.6 Los marcos, sus propiedades y facetas	206
7.7 El sistema de producción	213
7.7.1 La Base de Hechos (BH)	213
7.7.2 La base de Reglas (BR)	214
7.7.3 Conocimiento de Control	214
7.8 Descripción del funcionamiento del SBM	215
<b>Capítulo 8: Selección de la Herramienta e Implementación del Sistema</b>	
8.1 Introducción	219
8.2 Criterio de selección de la herramienta	219
8.3 Plan de implementación del asistente	222
8.4 La interfaz de usuario	224
<b>Capítulo 9: Evaluación del Sistema Experto</b>	
9.1 Introducción	240
9.2 Verificación del sistema experto	243
9.3 Validación del sistema experto	244
9.4 Usabilidad del sistema experto	253
9.5 Utilidad del sistema experto asistente de requerimientos	253
<b>Capítulo 10: Conclusiones y futuras líneas de investigación</b>	
10.1 Introducción	255
10.2 Conclusiones del trabajo de tesis	255

10.3 Futuras líneas de investigación y desarrollo 256

**Bibliografía 259**

**Apéndices**

Apéndice A: Planificación detallada del proyecto

Apéndice B: Hojas de cálculo del test de viabilidad

Apéndice C: Adquisición de conocimientos

Apéndice D: Formalización de conocimientos

# Capítulo I

## Introducción



# Capítulo I

## 1.1 Introducción

Desarrollar software significa construir una máquina, simplemente mediante su descripción. Esta es una muy buena razón para considerar la actividad de desarrollo de software como una ingeniería. En un nivel más general, la relación existente entre un sistema software y su entorno es clara. El sistema es una máquina introducida en el mundo de modo de provocar ciertos efectos en el mismo. Aquellas partes del mundo que afectarán la máquina y que serán afectadas por ella serán el Dominio de la Aplicación. Es allí donde los usuarios y/o clientes observarán si el desarrollo ha cumplido su propósito. Nadie mide el éxito de un sistema de reserva para un teatro sobre las computadoras que lo soportan. Lo que se observa es el mundo alrededor. ¿Puede la gente reservar asientos fácilmente?, ¿Está lleno el teatro?, ¿Cuánto tiempo insume obtener el boleto?, ¿Se debitan correctamente las tarjetas de crédito?

Esta distinción entre la *máquina* y el *dominio de la aplicación* es la base de la conocida **Qué** versus **Cómo**: **Qué** hace el sistema, se observa en el *dominio de la aplicación*, mientras **Cómo** lo hace se observa en la *máquina* misma. El problema yace en el *dominio de la aplicación*, la *máquina* es su solución.

Una de las mayores deficiencias en la práctica de construcción de software es la poca atención que se presta a la discusión del problema. En general los desarrolladores se centran en la solución dejando el problema inexplorado y por ende no descrito totalmente. El problema a resolver debe ser inferido a partir de su solución. Esta aproximación orientada a la solución puede funcionar en campos donde todos los problemas son bien conocidos y han sido detalladamente descritos, clasificados e investigados, donde la innovación deviene en la detección de nuevas soluciones a viejos problemas. Pero el desarrollo de software no es un campo con tales características. La versatilidad de las computadoras y su rápida evolución hace que exista un repertorio de problemas en constante cambio y cuya solución software sea de enorme importancia.

Un método que resuelva todos los problemas brinda poca ayuda sobre cualquier problema particular. Por lo tanto se debe clasificar los problemas y relacionarlos con metodologías para resolverlo. Para ello surge una idea crucial llamada Marcos de Problema. Un marco de problema define una clase de problema, mediante la provisión de una estructura definida de partes principales en la cuales todos los problemas de dicha clase deben coincidir. Los marcos de problema caracterizan clases de problema que comunmente ocurren como subproblemas de problemas reales mucho mayores. Que un problema particular encaje en un marco particular depende de la estructura y características del dominio de la aplicación y la estructura y características de los requerimientos.

Un buen método resuelve solo aquellos problemas que encajan en un marco de problema particular. Explora las propiedades del marco y sus partes principales de modo de brindar ayuda sistemática y precisa para lograr la solución. Esto significa que el poder del método depende de la calidad y precisión del marco.

Muchos proyectos han fallado porque sus *Requerimientos* fueron inadecuadamente explorados y descriptos. Los *requerimientos* se ubican en el *dominio de la aplicación* donde está el problema. Se debe definir el problema mediante una seria, precisa y explícita descripción.

El objetivo del presente trabajo es el desarrollo de un sistema experto que asista al ingeniero en software en la descripción del problema elaborando el documento de requerimientos de un sistema software.

### **1.2 Descripción de la composición del trabajo de Tesis.**

En el capítulo II se describe brevemente el concepto de marco de problema, su estructura y tipos. Luego se explica su uso para la descripción de los requerimientos y el dominio de la aplicación que se desea construir. Se introducen los elementos fundamentales de dominios, fenómenos compartidos y requerimientos. Finalmente se describe la información que debiera contener todo documento de requerimientos.

El capítulo III esboza la definición del problema, estableciendo claramente los objetivos, alcances, participantes y ámbito en donde se desempeñará el sistema, la metodología escogida para el desarrollo del proyecto y el proceso de gestión realizado.

El capítulo IV se aboca al estudio de la viabilidad del proyecto, procurando identificar la Plausibilidad, Adecuación, Justificación y Éxito que se obtendrá; además al ser abordado el problema desde la perspectiva de la ingeniería del conocimiento, se hace muy recomendable esta actividad dado que es mas complejo y mayores son los riesgos. Se incluye además la justificación de cada una de las características evaluadas.

El capítulo V describe el proceso de Adquisición de Conocimientos seguido, se presentan los conocimientos públicos extraídos de material bibliográfico, papers, posters, etc. y aquellos conocimientos educidos de los expertos que colaboran en el proyecto.

El capítulo VI detalla la fase de conceptualización de conocimientos con sus dos modelos, análisis y síntesis, formando así el modelo conceptual que los expertos poseen del problema.

El capítulo VII pretende describir la transformación del modelo conceptual en el modelo formal, más próximo a la máquina. Se muestra cómo se seleccionan los mecanismos más adecuados para los conocimientos que integran el modelo conceptual del problema.

El capítulo VIII establece las consideraciones que se tuvieron en cuenta para la selección de la herramienta utilizada para la implementación del sistema, con sus ventajas y desventajas.

El capítulo IX se aboca a la evaluación del producto logrado, procurando validar y verificar los diferentes componentes del sistema experto.

En el capítulo X se establecen las conclusiones obtenidas al concluir el desarrollo como así también las futuras investigaciones y tareas sobre el mismo.

En el capítulo XI se detalla la bibliografía utilizada para el desarrollo del trabajo.

El capítulo XII contiene los anexos con toda la documentación adicional obtenida y utilizada en cada capítulo.

# Capítulo II

## Dominio de la Aplicación

## Capítulo II

### 2.1 Introducción

En este capítulo se presenta el dominio de la aplicación del sistema experto que asiste al ingeniero en software en la elaboración del documento de requerimientos para cualquier tipo de software convencional.

Se describe el concepto de "Marcos de Problema" (Problem frames) introducidos por primera vez por Jackson en 1995 [Jackson 1], los tipos de marcos de problema elementales, cómo detectar e identificar las partes de un marco, y las diferentes aproximaciones a la descomposición de un problema real en subproblemas ajustables a marcos elementales.

Posteriormente se trata la descripción del dominio de la aplicación y sus requerimientos según el marco de problema utilizando diferentes metodologías y técnicas, y cómo debe estar constituido el documento de requerimientos, meta final del sistema experto.

### 2.2 Problemas de Software

Sabemos qué clase de problema resuelve un puente: Proveer un camino para que ciertos objetos puedan moverse de un lugar a otro sin caer en el espacio vacío existente entre ambos lugares. Esta definición genérica del problema expone claramente qué clase de preguntas es necesario realizarse para escribir los requerimientos para el puente. Fundamentalmente ellas son: cuál es el tamaño y peso de los objetos que lo cruzarán, cuáles son los puntos que debe conectar?, adicionalmente debe conocerse datos del entorno del puente como por ejemplo: otros tipos de carga tal como vientos, o dónde tiene sus fundaciones tal como tipo de suelo del río.

¿Entonces, qué clase de problemas resuelve el software?

Todos los problemas de software tienen la siguiente forma:

Configurar una máquina  $M$  para que produzca los efectos  $R$  en el  $D$ .

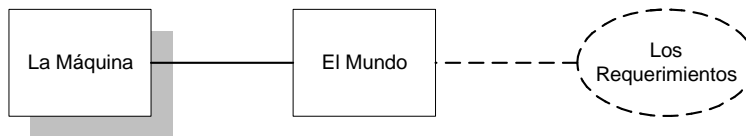


Figura 2.1: Forma general de un problema software

En la figura 2.1 se puede observar que la máquina  $M$  es la computadora que será programada, incluyendo sus dispositivos de entrada/salida. El dominio  $D$  es una parte necesaria en la definición del problema dado que es la parte del mundo en términos de los cuales se definen los requerimientos y porque la máquina raramente pueda producir los efectos deseados por sí misma. El diseño del software explota las propiedades de su entorno haciendo uso de la redundancia para detectar errores, por medio de usuarios u otro software para obtener información, de motores para controlar maquinarias, etc.

Estas son las razones por las que el software es comúnmente malentendido y por las que la ingeniería del software difiere tanto del resto. El software no es un artefacto tangible como un puente, un motor o una computadora. Software es una configuración particular de una computadora.

Estamos en condiciones de realizar algunas definiciones como:

**Dominio del problema:** Es la parte del mundo donde la máquina debe producir los efectos deseados, juntamente con los elementos disponibles para producirlos, directa o indirectamente.

El dominio del problema incluye todo aquello que se considere relevante para describir los efectos deseados: Objetos sobre los que se hacen consultas, personas a ser informadas, objetos a ser controlados, parámetros a ser mantenidos dentro de cierto rango, resultados de consultas, etc.

Los elementos disponibles para los diseñadores del software para producir los efectos deseados también son parte del dominio del problema. Indirectos pueden ser motores que la máquina enciende o apaga, personas que introducen valores, etc. Pero donde hay elementos indirectos también los hay directos. Los únicos efectos que una computadora puede causar directamente son sobre el comportamiento de dispositivos de entrada/salida (teclados, pantallas, etc.)

**Requerimientos:** Son los efectos que la máquina debe ejercer en el dominio del problema, por virtud de su programación.

Hemos hecho una definición precisa limitando los requerimientos a condiciones en el dominio del problema. No estamos interesados en el comportamiento del software sino en los efectos producidos por el comportamiento del software. No debemos confundir la solución con el problema.

El mundo debe separarse en dominios. Cada dominio contiene individuos, esto es, partes distinguibles sobre las cuales se desea decir algo. Son las partes físicas del mundo en términos de las cuales se definen los requerimientos. Los individuos en el dominio de la máquina son las subrutinas, las estructuras de datos que componen la programación así como los dispositivos de entrada/salida. La única regla sobre los individuos es que siempre se puede distinguir uno de otro.

En cada dominio también está incluido todo lo que se desea decir sobre los individuos que lo componen. Todo lo que se esté en condiciones de aseverar o negar sobre un dominio se denomina predicado. Un dominio podría definirse entonces como un conjunto de individuos con sus predicados.

Dado que los requerimientos están expresados en términos del dominio del problema, solo pueden referirse a los individuos en el dominio del problema.

La descripción del dominio del problema ocupa la mayor parte del documento de requerimientos. Inclusive más que la lista de requerimientos. Dependiendo del tipo de problema puede contemplar:

- Qué clase de entidades están o pueden estar en el dominio
- Qué clase de atributos poseen dichas entidades
- Qué relaciones existen entre las entidades
- Qué tipo de eventos pueden ocurrir en el dominio
- Las leyes causales sobre las que el dominio se comporta

Cuando separamos en dominios, existen interrelaciones entre los mismos. Se denomina "Fenómenos compartidos" a todos los estados, eventos y objetos que son compartidos entre dos dominios. Por ejemplo:

- Teclas presionadas por usuarios son teclas recibidas por el software
- Cada pixel mostrado en un monitor es también un pixel visto por el usuario
- La Señal enviada por un sensor de oxígeno a un microprocesador

### **2.3 Marcos de Problema**

La idea del presente trabajo es tomar una aproximación al análisis del problema para poder documentar los requerimientos, basada en la idea de los marcos de problema. Los marcos de problema caracterizan clases de problema que comúnmente ocurren como subproblemas de otros problemas más grandes y reales. La idea es analizar los problemas reales descomponiéndoles en sus subproblemas constitutivos que correspondan a marcos de problemas conocidos.

Cada marco es una elaboración de la forma general del problema software mostrado en la figura 2.1 (pág. 7), y puede ser elemental o compuesto.

Un problema perteneciente a una clase caracterizada por un marco elemental será capturado mediante la construcción de descripciones apropiadas al marco. Un problema de una clase compuesta será primero descompuesto en subproblemas caracterizados por marcos elementales.

Si se restringe el repertorio de marcos de problema a los del tipo elemental, sería necesario descomponer cada problema real en una estructura de subproblemas, cada uno lo suficientemente pequeño y simple de modo que se ajuste a los marcos elementales. Estaríamos desaprovechando la oportunidad de construir un repositorio de experiencia sobre los problemas y su solución. Es por esto que el sistema experto también proveerá los mecanismos para administrar un repositorio de marcos compuestos correspondientes a problemas resueltos.

El uso de marcos de problema otorga dos ventajas importantes. La primera es que si se posee un nutrido repertorio de clases de subproblemas conocidos en los cuales se puedan descomponer los problemas realistas, se obtendrá un proceso de descomposición guiado por una taxonomía de problemas muy sistemática. Esto redundará en excelentes resultados.

La segunda ventaja es que un marco de problema es asociado siempre con uno o más métodos para capturar el problema en detalle y desarrollar su solución. Además se puede utilizar el método apropiado en cada subproblema.

El sistema experto guiará la descomposición, dará consejo y alertará sobre las dificultades que pueden ocurrir y proveerá el contexto en el que la experiencia



previa capturada en el mismo pueda ser explotada efectivamente. Una vez analizado el problema dará guías para describir tanto el dominio de la aplicación como los requerimientos según la descomposición realizada con sus marcos de problema asociados.

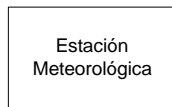
### **2.3.1 Diagramas de marcos**

Se introduce aquí una simple notación gráfica para describir las partes principales de un problema software, como puede observarse en la figura 2.2.

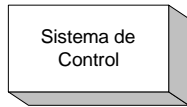
En un marco, cada dominio es representado por un rectángulo, y los fenómenos compartidos por dos dominios se representan por una línea que conecta los dos rectángulos. La máquina a ser programada se representa por un rectángulo sombreado. La palabra utilizada dentro de dicho rectángulo representa el tipo de máquina en la que se convierte la computadora como resultado de la programación. Los requerimientos se simbolizan mediante un óvalo, con una o mas líneas conectándolos a dominios a los cuales ellos pertenecen.

La manera de leer un diagrama de marcos involucra dos pasos. Primero se debe leer el óvalo y detectar con cuáles dominios está relacionado. Estos son los principales dominios de interés. El segundo paso es encontrar el dominio de la máquina y ver cómo, directa o indirectamente, se conecta a los dominios de interés. Es decir, se traza un camino entre la máquina y los dominios de interés.

Debe notarse que el diagrama no intenta mostrar en gran profundidad todos los aspectos del problema. Solo provee una rápida y gráfica manera de mostrar los principales elementos del problema para ayudar a planificar una forma sistemática de documentarlo.



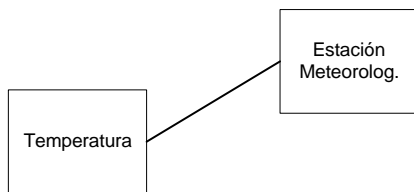
Cada rectángulo es un dominio. Colección de objetos o porción del mundo, descriptas con el propósito de establecer descripciones sobre ellos.



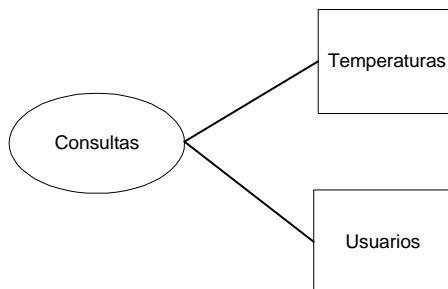
Un rectángulo sombreado es la máquina a construir. La computadora a ser programada.



La elipse representa el conjunto de requerimientos.



La línea que une dos dominios representa fenómenos compartidos: Estados o eventos que corresponden a ambos dominios. Causación o flujo de datos entre ambos siempre involucran fenómenos compartidos.



Una línea que conecta una elipse con uno o mas dominios indica que los requerimientos aplican sobre dichos dominios. Los requerimientos siempre especifican relaciones a ser construidas dentro o entre dominios.

*Figura 2.2 Elementos de Diagramas de marcos.*

### 3.2 Marcos de problema elementales

Se presentan aquí 5 marcos de problema que corresponde a 5 tipos de requerimientos:

Tipo de Requerimiento	Descripción	Marco de Problema
Consultas	Requerimiento de información sobre alguna parte del dominio del problema	Información
Reglas de comportamiento	Reglas que debe seguir el comportamiento del dominio del problema	Control
Mapeos	Mapeos sobre datos de entrada y de salida del software	Transformación
Operaciones sobre dominios creados	Operaciones que realizan los usuarios sobre objetos que existen solo dentro del software	Workpieces
Correspondencias entre dominios	Mantenimiento de dominios que no poseen fenómenos compartidos en sus estados correspondientes.	Conexión

*Tabla 2.1: Tipos de marcos de problema*

Para cada tipo de requerimiento existe un conjunto de información del dominio del problema necesario para describir una especificación que implemente el requerimiento.

Estos marcos no constituyen una lista exhaustiva, solo describen una clase específica de problema. Cuando se detecta un problema que se ajusta a uno de los marcos, se conoce cómo debe documentarse sistemáticamente el problema de una manera que sea útil para el resto del desarrollo.

Los problemas reales involucran distintos tipos de requerimientos a la vez. Es el caso de los marcos compuestos.

El propósito de enmarcar un problema no es forzarlo a que se ajuste a alguna categoría existente, por el contrario, es reconocer un problema familiar y a partir de allí, comenzar el análisis de un problema no familiar.

Se expone a continuación una breve descripción de cada marco.

### *Marco de problema de información*

Todo software que resuelve un problema de información, contesta consultas acerca de cierta parte del mundo real. Documentar un problema de información involucra describir los tipos de requerimientos de información a ser satisfechos, la parte del mundo real sobre la cual aplican tales consultas y cómo el software puede tener acceso a dicha parte del mundo.

La figura 2.3 muestra el diagrama del marco de problema de información.



*Figura 2.3: Marco de problema de información*

El requerimiento es satisfacer consultas iniciadas por los solicitantes de información (usuarios, hardware o software) que necesitan información.

El óvalo de consultas está conectado al mundo real y los solicitantes de información para indicar que el trabajo del sistema es mantener una relación entre ambos de modo de que los solicitantes obtengan la información acerca del mundo real a demanda.

Las consultas están siempre definidas en términos de contenido -preguntas sobre el mundo real. Los usuarios en ocasiones desean la respuesta en formatos especiales como formularios preimpresos, etc. La descripción de dichos preimpresos es parte de la definición del problema.

En la mayoría de los problemas de información la descripción de las consultas es simple. El trabajo mayor es describir el mundo real. Comúnmente los sistemas de información reportan sobre el estado del mundo real en constante cambio. Se denominan sistemas de información dinámico. Por el contrario un sistema de información estático reporta sobre un mundo que cambia muy poco o nada. En el caso dinámico el sistema construye la información disponible durante la operación. Un sistema estático posee la información ya preestablecida antes de comenzar a operar.

Para documentar un sistema dinámico se debe indicar cómo el sistema gana acceso a cada evento que cambia el resultado de las posibles consultas.

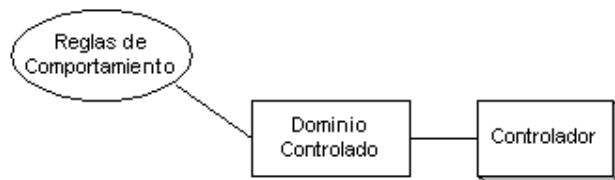
Para documentar un sistema de información estático no se debe indicar cómo el sistema obtiene acceso a la parte relevante del mundo real sino cómo los desarrolladores obtienen dicho acceso.

Hay una subclase de problema de información dinámico llamado *Snapshot*. El sistema reporta sobre el estado actual de alguna parte del mundo real, como la temperatura actual, o por ejemplo una foto de una avenida de Madrid vía internet. Estos problemas se enmarcan mejor como un problema de conexión.

#### *Marco de problema de control*

En un problema de control, el software es responsable de asegurar que alguna parte del mundo se comportará de acuerdo con reglas que se especifican. Documentar un problema de control involucra describir los objetos que habitan en tal parte del mundo y las reglas causales a las que obedecen, las reglas que rigen a dichos objetos por efecto de su propia naturaleza, y los fenómenos compartidos con el software a través de los cuales el mismo puede monitorear el estado del mundo e iniciar cadenas causales que resulten en las reglas que están siendo seguidas.

La figura 2.4 muestra el marco de problema de control



*figura 2.4: Marco de problema de control*

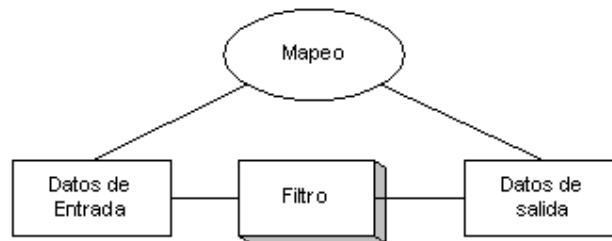
Un problema de control se centra principalmente en causalidad, esto es, en hacer que una parte del mundo se comporte de acuerdo a reglas específicas. Para documentarlo es necesario describir tres cosas: (a) Las propiedades causales de la parte relevante del mundo y las reglas que los objetos en dicho mundo siguen por virtud de su propia naturaleza, independientemente del software, (b) las reglas que se desea que ellos sigan y (c) los fenómenos compartidos entre la computadora y el dominio del problema, a través de los cuales el software monitorea el dominio e inicia acciones que resultan en las reglas (b).

La parte (b) son los requerimientos denominados reglas de comportamiento. El resto es la descripción del dominio del problema.

#### *Marco de problema de transformación*

Para resolver un problema de transformación, el software genera datos de salida que están mapeados contra datos de entrada de acuerdo a reglas específicas. Documentar un problema de transformación involucra la descripción del conjunto completo de todas las posibles entradas y las reglas de mapeo que prescribe, para cada posible entrada, la salida correcta.

La figura 2.5 muestra el marco de problema de transformación.



*Figura 2.5: Marco de problema de Transformación*

Los datos de entrada y de salida son elementos que corresponden a dos conjuntos. Para documentar el problema se debe describir: El conjunto de todas las entradas posibles, el conjunto de todas las salidas posibles y las reglas que relacionan cada posible entrada con su correspondiente salida. Dichas reglas, los mapeos, son los únicos requerimientos.

#### *Marco de Problema Workpieces*

En un problema del tipo Workpieces, el software actúa como una herramienta para carear objetos que existen solo dentro del mismo software. Documentar un problema de este tipo consiste en la descripción del objeto que existirá dentro del software y las operaciones que los usuarios pueden realizar sobre el mismo.

La figura 2.6 muestra el marco de problema workpieces.



*Figura 2.6: Marco de problema Workpieces.*

Las piezas de trabajo son intangibles, son objetos software que existen en dominios creados, no obstante el software puede también generar versiones tangibles de ellos, como documentos impresos. Existen entonces dos requerimientos: Permitir a los usuarios realizar las operaciones en las piezas de trabajo, y crear piezas de trabajo dentro del software.

La mayor parte de la documentación se ocupa de describir las piezas de trabajo.

#### *Marco de problema de conexión*

En un problema de conexión, hay dominios que no comparten fenómenos directamente, sino que lo hacen a través de otro dominio entre ellos, el dominio de conexión. El problema consiste en lograr que dos dominios indirectamente conectados se comporten como si estuvieran directamente conectados.

Hay dos tipos principales de problemas de conexión. En el tipo (a), el sistema necesita interactuar con el dominio de interés, pero debe realizar una conexión con el mismo para obtener información, o ejecutar comandos del sistema sobre el mismo. En el tipo (b), el sistema a construir es el dominio de conexión, responsable de cambiar el sistema B a un estado correspondiente con el estado del sistema A y viceversa.

La figura 2.7 muestra los marcos de problema de conexión de tipo (a) y (b).



Tipo (a)



Tipo (b)

Figura 2.7: Marco de problema de conexión

El requerimiento en ambos casos es nada más que una correspondencia asequible de estados, no una perfecta correspondencia, dado que esta última suele ser imposible de lograr. Normalmente los problemas de conexión ocurren como parte de otro problema mayor.

Documentar un problema de conexión del tipo (a) consiste en la descripción de los mapeos entre los fenómenos compartidos que vinculan el dominio de conexión con el dominio de interés, y los fenómenos compartidos entre el sistema y el dominio de conexión. Este mapeo debe incluir los tipos de distorsión y retraso introducidos por el dominio de conexión: ¿Que tipo de conexión es la menos confiable? ¿cuál es el retraso entre un evento en un extremo del dominio de conexión y el correspondiente evento en el otro?

Es muy valioso documentar las maneras en que el sistema puede detectar que el dominio de conexión no está funcionando apropiadamente.

Documentar un problema del tipo (b) involucra la descripción del mismo tipo de mapeos entre estados y/o eventos, excepto que ahora es un mapeo *deseado*, con



características de distorsión y retardo deseado, es decir requerimientos más que descripción del dominio del problema.

#### **2.4 Marcos de problema compuestos**

Idealmente toda vez que se enfrenta un problema software complejo, se puede descomponerlo en distintos subproblemas que se ajustan a marcos de problema elementales, quienes interactúan a través de un estrecho canal lógico, en el cual la descripción de una parte del problema que se ajusta a un marco hace muy poca o ninguna referencia a la parte del problema que se ajusta a otro marco.

El enmarcado del problema total como un conjunto de problemas más pequeños que se superponen sutilmente es el arma mas importante que se dispone para sobrellevar la complejidad del documento de requerimientos. De esta forma se logran descripciones individuales de cada componente pero cubriendo sistemáticamente todo.

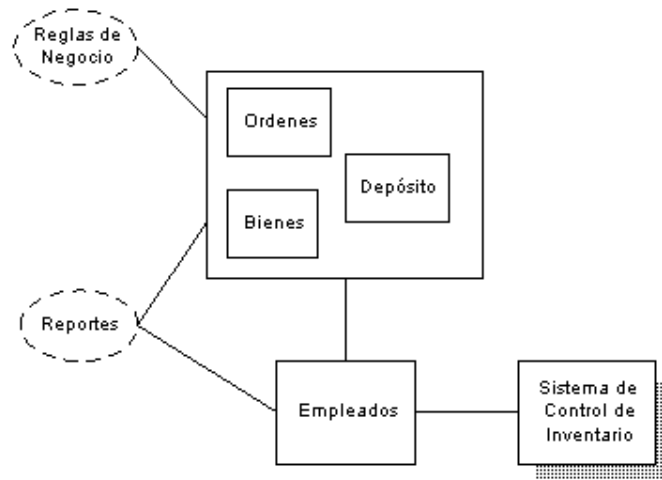
No es necesario utilizar la misma metodología de documentación para todo el proyecto. Se puede utilizar una diferente según la pieza de software sobre la que se trabaja.

Es justamente donde cobra importancia el tener un catálogo de marcos compuestos disponible, dado que la experticia obtenida en otros software es lo que permite enmarcar nuevos problemas de la manera correcta. Tales marcos compuestos son la combinación de marcos elementales donde se comparten dominios, formando un único marco representativo del problema.

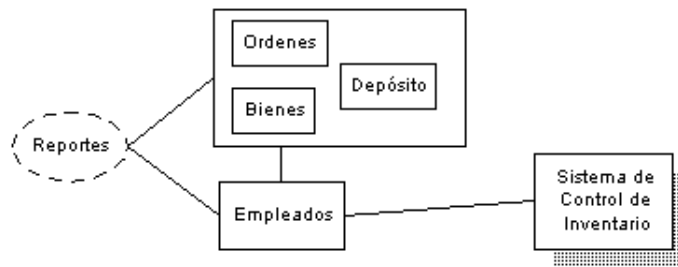
El sistema es capaz de almacenar dichos marcos compuestos y los marcos elementales constitutivos, de modo cuando el analista detecta un problema similar o idéntico obtenga un guiado en la documentación completa del problema.

Usualmente, los problemas complejos poseen grupos de requerimientos de la misma clase, esto es, requerimientos que corresponden a un tipo de marco de problema y que hacen referencia al mismo conjunto de dominios. Haciendo la partición en subproblemas se tienen requerimientos que no se superponen o interfieren entre sí. Además se describen los dominios comunes una única vez sin que interfieran con los requerimientos.

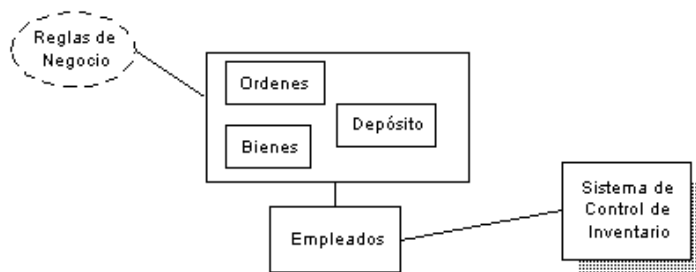
La figura 2.8 muestra un marco compuesto para un sistema de inventario y posteriormente los dos marcos elementales que lo constituyen.



Marco de problema compuesto para el ejemplo del sistema de inventario



Marco elemental de información



Marco elemental de control

Figura 2.8: Marcos elementales de información y control para el sistema de inventario.

## 2.5 Descomposición del problema

Existen variadas aproximaciones a la descomposición de problemas en subproblemas. Se mencionan aquí tres de ellas.

### **2.5.1 Descomposición Exterior-interior**

A veces el problema que no se ajusta a ningún marco conocido aún aproximadamente. Parece entonces adecuado descomponer el problema desde afuera hacia adentro.

En esta aproximación se trata de encontrar partes o aspectos del problema reconocibles que correspondan a marcos conocidos, y analizarlos en el contexto de dichos marcos.

Luego, las partes y aspectos del problema original que permanecen irresueltos pueden ser considerados sin la complicación adicional de los subproblemas ya resueltos.

Es decir, se analiza el problema desde una perspectiva exterior o global tratando de encontrar una parte reconocible y luego proceder a interiorizarse en el resto del problema.

Esta aproximación es una aplicación iterativa de la conocida heurística:

*"Encuentre una pieza del problema que pueda resolver"*

### **2.5.2 Descomposición interior-Exterior**

A menudo el problema parece ajustarse aproximadamente a un marco conocido, pero exhibe dificultades que frustran la aplicación pura del marco. Tales dificultades generan por sí mismas subproblemas que pueden reconocerse como ajustados a otros marcos. Existen diferentes formas de dificultades como por ejemplo:

Dificultad de conexión:

Alguna información que la máquina necesita no está directamente disponible cuando es requerida. Entonces es posible sortear la dificultad como un subproblema de información en la que la máquina original es la parte que realiza las consultas.

Dificultad de Identidades:

La máquina debe poder distinguir múltiples instancias de las entidades del dominio cuando comparte con ellas eventos o estados. Por ejemplo en un sistema de control de semáforos con dos conjuntos de luces. Si las secuencias de los dos conjuntos

son diferentes (es el caso del tráfico más denso en una dirección que en la otra) la máquina debe poder distinguirlos. Generalmente la solución a la dificultad de identidad consiste en la introducción de uno o varios subproblemas de mapeo.

Esta aproximación consiste en trabajar desde el interior hacia afuera, donde el interior es el marco que parece ajustar aproximadamente y el exterior es el conjunto de dificultades circundantes. El problema central puede ser analizado asumiendo que las dificultades serán resueltas en la solución a los subproblemas que las capturan.

Esta aproximación es la aplicación de la heurística:

*"Resolver un problema más simple"*

### **2.5.3 Reconocimiento de un marco compuesto**

Si bien los marcos de problema elementales forman la base de la técnica, es conveniente contar con un rico conjunto de marcos compuestos. De este modo es factible que una parte sustancial, u ocasionalmente el problema completo, se ajuste a un marco de problema compuesto.

Esta aproximación es el resultado de aplicar la heurística:

*"El mejor método es haber resuelto el mismo problema anteriormente"*

### **2.6 Documentación**

El documento de requerimientos debe poseer toda la información necesaria para la concreción exitosa del proyecto. En la tabla 2.2 se muestra una lista exhaustiva del contenido que debe poseer. Nótese que no se define una estructura rígida, como lo hacen los estándares, dado que el software es muy vasto como para definir tal tipo de organización.

<b>El documento de requerimientos</b>	
<b>Requerimientos</b>	Consultas
	Reglas de comportamiento
	Mapeos
	Operaciones en dominios creados
	Correspondencias asequibles
<b>Descripción del dominio del problema</b>	Entidades, atributos y relaciones (modelo de datos)
	Secuencias de eventos
	Reglas causales
	Formatos de archivo
	Fuentes de información
	Hardware y/o software con el que conectarse
	Mapeos entre puertos I/O y hardware
<b>Expectativas</b>	
<b>Invariantes</b>	
<b>Plataforma: hardware y sistema operativo</b>	
<b>Características globales</b>	
<b>Limitaciones de diseño</b>	
<b>Cambios deseables o probables</b>	
<b>Glosario</b>	
<b>Introducción</b>	
<b>Información del documento</b>	

*Tabla 2.2: Contenido del documento de requerimientos.*

**Requerimientos:** Además de la información que se debe incluir para cada tipo de requerimientos, se puede indicar la importancia relativa o prioridad para cada uno de ellos. Es útil para decidir cuál dejar para la siguiente versión, o bien eliminar si el proyecto se retrasa. Dicha priorización puede ser mediante escalas numéricas, no obstante una aproximación simple y precisa es la indicación para cada requerimiento del número de versión donde será implementado.

**Descripciones del dominio del problema:** Normalmente ocupa la mayor parte del documento y se basa en técnicas como entidad-relación, descripción de objetos

y atributos en el dominio, diagramas de transición de estados, eventos, etc. según el tipo de dominio y marco de problema.

Es importante también incluir una estructura preliminar de datos identificando las clases principales y sus relaciones, aún cuando es parte de la especificación de requisitos. Dicha estructura preliminar de datos describe solo los estados del software que pueden ser distinguidos desde el exterior. No especifica una base relacional, un arreglo de bytes en memoria o cualquier otro aspecto sobre cómo los datos son representados en el software.

Los formatos de archivo se deben incluir en la descripción del dominio para aquellos programas que necesitan leer archivos generados por otro software. Si ya existe documentación, solo debe citarse.

**Expectativas:** Son los resultados del software por el cual los clientes pagan, es decir los efectos esperados por haber cumplimentado los requerimientos. Es importante que el grupo de desarrollo conozca las razones por las que se construye el nuevo software.

**Invariantes:** Son condiciones que nunca cambiarán. Hay dos tipos de invariantes:

- a. Requerimientos que establecen condiciones que el sistema se supone debe mantener.
- b. Redundancia que se agrega a los requerimientos para ayudar a asegurar su correctitud.

**Plataforma:** Es la máquina a ser configurada. Se debe incluir hardware y software.

**Características globales:** Son propiedades que el sistema poseerá. Por ejemplo, disponibilidad, confiabilidad, seguridad, performance.

Otra característica típica es la escala. Escala es el número de instancias requeridas de un objeto descrito ya sea en el dominio del problema o en los requerimientos. El software se diseñará diferente según dicha escala.

**Limitaciones de diseño:** Por ejemplo si el cliente impone que se escriba en Lenguaje C dado que posee un staff de programadores en C para mantenimiento y requiere que se utilice una determinada convención de escritura.

**Cambios deseables:** Son cambios que se esperan ocurran en versiones futuras. No se deben describir en demasiado detalle, pero es importante que se mencionen para ayudar a los diseñadores o programadores a realizar su trabajo de modo que haga simples las futuras modificaciones.

**Glosario:** Se debe incluir los términos más importantes del dominio del problema pero también aquellos términos que ciertos lectores pudieran no entender como por ejemplo abreviaturas, etc.

**Introducción:** Siempre es necesario mostrar cómo cada parte del documento (requerimientos y descripciones del dominio del problema) conforman el documento. Si bien la introducción resume lo mismo que se leerá posteriormente, siempre resulta una redundancia saludable y de mucha ayuda.

**Información del documento:** Tabla de contenidos, convenciones tipográficas, datos de modificación o revisión, histórico de cambios, persona que lo preparó, índice, etc.

# Capítulo III

## Definición del Problema



## Capítulo III

### Definición del problema

#### 3.1 Introducción

En el presente capítulo se presenta el problema de la identificación y documentación de los requerimientos de un producto software desde el punto de vista del problema que se desea resolver. Se detalla también los objetivos y el alcance perseguidos en el presente trabajo.

Además se establecen los expertos que colaborarán en la construcción del sistema y demás consideraciones a saber: Ciclo de vida utilizado, las fases y sus etapas y una breve descripción del plan de gestión del proyecto.

#### 3.2 Planteamiento del problema

Para comenzar el planteamiento del problema, es conveniente citar a Frederick Brooks:

*La parte más difícil en la construcción de sistemas software es decidir precisamente qué construir. Ninguna otra parte del trabajo conceptual es tan dificultosa como establecer los requerimientos técnicos detallados, incluyendo todas las interfaces con humanos, máquinas y otros sistemas software. Ninguna otra parte del trabajo puede perjudicar tanto el resultado final si es realizada en forma errónea. Ninguna otra parte es tan dificultosa de rectificar posteriormente.*

Frederick P. Brooks, Jr., *The Mythical Man-Month*, Addison-Wesley, 1995.

La adquisición de requerimientos es una de las más importantes partes del proceso de desarrollo de software, pero es a la vez una de las que cuenta con menor soporte en la actualidad. A su vez es una etapa donde inevitablemente existe ambigüedad, incompletud, contradicciones que atentan contra el correcto comienzo de la vida del producto. (Brooks, 1995).

Las causas primarias de realizar una incorrecta conceptualización del problema pueden clasificarse de la siguiente manera (Sommerville, 1996):

- ◆ Carencia de conocimientos sobre el dominio
- ◆ Dominio de aplicación complejo

- ◆ Carencia de experiencia en detección de requerimientos, conceptualización de los mismos, analogía con sistemas anteriores, etc.

Tales problemas sugieren que existe una creciente necesidad de herramientas que asistan en el proceso de elaborar los requerimientos de un sistema software, que trabaje con dominios complejos de modo que colabore con los analistas en dicha tarea.

La más poderosa contribución de los sistemas expertos es poner al servicio de los analistas noveles la experiencia adquirida por aquellas personas consideradas verdaderos especialistas en el área. (Davis, 1993)

Hay una gran distancia entre la correcta práctica de la ingeniería de software y la práctica media, quizá mayor a la de cualquier otra ingeniería (Brooks, 1995).

Una herramienta que contribuya en el mejoramiento de la práctica de la ingeniería del software y que a la vez incentive su aplicación sería de vital importancia.

### **3.3 Objetivos del trabajo**

El sistema experto que se pretende desarrollar tiene por objetivo asistir al analista y/o Ingeniero en Software en la tarea de elaboración del documento de Requerimientos.

Ante todo conviene aclarar cuál es el significado de la terminología que se utilizará en el trabajo. Se entiende por *Requerimiento* la especificación de la información necesaria para realizar determinadas tareas o funciones, en tanto *Requisito* es una condición y/o especificación técnica u operativa respecto de los requerimientos. Dado que ambos términos suelen ser utilizados como sinónimos, hecha la aclaración, el sistema experto a desarrollar se centrará sobre los *Requerimientos* del sistema.

El proceso de ingeniería de requerimientos comienza con la etapa de elicitación donde se interactúa fluidamente con los usuarios y/o clientes, se obtienen reportes de sistemas anteriores, entrevistas, documentación aportada por los futuros usuarios, observaciones, etc.

Como resultado de dicha fase el ingeniero debe generar el *documento de requerimientos*, donde se debe asentar los requerimientos del producto software a construir atendiendo a un enfoque del problema a resolver, junto con la descripción completa del dominio del problema donde se implantará el sistema (Kovitz, 1999).

Se enumeran a continuación los objetivos perseguidos en el presente trabajo:

- Establecer una forma efectiva de asistencia en el análisis del problema software, su descomposición y la descripción del dominio del problema y sus requerimientos.
- Representar por medio de técnicas de Ingeniería del Conocimiento los criterios y metodología utilizada por los expertos para resolver el problema, entendiéndose por técnicas de Ingeniería del Conocimiento: Técnicas de Adquisición de Conocimiento, Técnicas de Conceptualización, Técnicas de Formalización y Técnicas de Evaluación.
- Construir el sistema mediante el uso de una herramienta para la implementación de sistemas expertos, a través de sucesivos prototipos que iterativamente evolucionen hacia un producto con mejores prestaciones y con más conocimiento incorporado.
- Evaluar el comportamiento y adaptabilidad del sistema mediante casos reales.

### **3.4 Alcance**

El sistema abordará el análisis del problema mediante el uso de marcos de problema para luego guiar al usuario en la información que necesita obtener para poder documentar los requerimientos y la descripción del dominio de aplicación.

El objetivo primario es obtener un marco de problema adecuado al problema que el software intenta dar solución y, a partir del mismo, servir de guía para que toda la información necesaria haya sido obtenida en orden de generar un documento preliminar de requerimientos.

### **3.5 Participantes y ámbito del proyecto**

Para la concreción del sistema experto se tiene:

*Expertos:* Se integran al proyecto la Licenciada Bibiana Rossi y el Licenciado Edgardo Claverie, ambos del Centro de Actualización Permanente en Ingeniería del Software (Instituto Tecnológico de Buenos Aires).

Además se integra en la etapa de evaluación del sistema experto el Ingeniero en Sistemas Diego Linares, quien se desempeña actualmente en la compañía ASECOM de Córdoba

Se entrevistó también a los expertos internacionales M. Jackson (inventor del Jackson Development System JSD) y a Benjamin Kovitz (Autor del Libro Practical Software Requirements), quienes mostraron un gran interés y disposición toda vez que se requirió su consejo a través de internet.

*Usuarios:* Los usuarios serán ingenieros en sistemas de la Universidad Católica de Córdoba interesados en el uso del sistema para los proyectos desarrollados en conjunto con industrias del medio, e ingenieros en software del Instituto Tecnológico de Buenos Aires, interesados en utilizar el sistema como medio didáctico académico para el desarrollo de sistemas experimentales.

*Ámbito:* El ámbito de aplicación será los laboratorios de investigación y desarrollo de ambas Universidades.

*Evaluación:* La evaluación y el desarrollo del prototipo será llevado a cabo con personas de amplia experiencia vinculadas a los laboratorios mencionados. Existen casos de prueba de sistemas ya terminados que el sistema deberá ser capaz de resolver.

*Herramienta:* No existen restricciones en cuanto al hardware donde debe ejecutarse la herramienta. Ambos laboratorios poseen equipamiento de última generación. Resulta apropiado que el prototipo sea desarrollado para funcionar en computadoras personales.

### **3.6 Metodología de Construcción**

Dadas las características del sistema que se desea construir, resulta adecuado adoptar la Metodología IDEAL para el desarrollo del Sistema Basados en Conocimiento (SBC).

La Metodología IDEAL (Juan Pazos y colegas, 1995) propone un ciclo de vida en espiral en tres dimensiones, y se ajusta a la tendencia del software actual, esto es:

- Ser Reutilizable
- Ser Integrable
- Poseer Requisitos Abiertos
- Diversidad de Modelos Computacionales

Los requisitos están sometidos a constantes cambios y por ende el sistema también, por lo que como resultado se obtiene un sistema en constante evolución por lo que puede considerarse como un prototipo en constante perfeccionamiento, mediante el agregado de nuevos marcos compuestos, mediante nuevas técnicas de descomposición del problema, mediante nuevas formas de documentación o estándares a los que debe ajustarse.

Se expone a continuación las fases y etapas que componen la metodología I.D.E.A.L. y que guiarán el desarrollo del sistema experto:

<b>FASES Y ETAPAS DE LA METODOLOGIA I.D.E.A.L.</b>
<b>FASE I: Requerimientos, viabilidad, especificación técnica.</b>
Etapa I.1. Plan de requisitos y adquisición de conocimientos.
Etapa I.2. Evaluación y selección de la tarea.
Etapa I.3 Definición de las características del sistema.
<b>FASE II: Desarrollo de los prototipos de demostración, investigación, campo y operacional</b>
Etapa II.1. Concepción de la solución.
Etapa II.2. Adquisición y Conceptualización de los conocimientos.
Etapa II.3. Formalización de los conocimientos y definición de la arquitectura
Etapa II.4. Selección de la herramienta e Implementación.
Etapa II.5. Validación y evaluación del prototipo.
Etapa II.6. Definición de nuevos requisitos, especificaciones y diseño.
<i>Nota: Las etapas II.1 a II.6 se repiten por cada iteración del prototipo.</i>
<b>FASE III: Ejecución de la construcción del sistema integrado</b>
Etapa III.1 Requisitos y diseño de la integración
Etapa III.2 Implementación y evaluación del sistema integrado
Etapa III.3 Aceptación del sistema por el cliente
<b>FASE IV: Actuación para conseguir el mantenimiento perfecto</b>
Etapa IV.1. Definir el mantenimiento del sistema global.
Etapa IV.2. Definir el mantenimiento de las bases de conocimientos.
Etapa IV.3. Adquisición de nuevos conocimientos.
<b>FASE V: Lograr una adecuada transferencia tecnológica</b>
Etapa V.1. Organizar la transferencia tecnológica.
Etapa V.2. Completar la documentación del SBC construido.

A continuación se describe en forma breve las diferentes etapas del Ciclo de Vida, desarrolladas en el presente trabajo.

- *Estudio de Viabilidad:* cuando se intenta resolver un problema con la tecnología de Sistemas Expertos, previamente debe evaluarse si la tarea es abordable en el campo de la Ingeniería del Conocimiento. Es decir, debe dirimirse si el desarrollo es Plausible, Justificable, Adecuada y procura garantizar su Éxito.
- *Adquisición de Conocimientos:* los problemas abordados con la tecnología de la Ingeniería del Conocimiento intentan imitar a través de un software, el quehacer de un experto humano al desempeñar una determinada tarea. Una de las actividades que requiere mayor esfuerzo, por su complejidad es la extracción y educación de conocimientos, por medio de la cual se intenta descubrir el dominio de la aplicación, el problema y el proceso de solución al problema.
- *Conceptualización:* En esta fase se describe el proceso de organización de los conocimientos adquiridos. Esta actividad está constituida por dos tareas fundamentales: una de Análisis, basada en la detección de conocimientos estratégicos, tácticos y fácticos, y la actividad de Síntesis donde quedan expresados dichos conocimientos en forma estructurada.
- *Formalización:* pretende encontrar una adecuada representación de los conocimientos, garantizando su correcta manipulación. Es el primer acercamiento a la máquina en lo que respecta a su implementación.
- *Implementación:* desarrolla la transformación de los conocimientos representados en el modelo formal en un modelo computable.
- *Evaluación:* establece el grado de experiencia alcanzado por el sistema. De manera tal que expertos en el área que han, o no han participado del desarrollo del proyecto se comprometen a evaluar el desempeño del sistema, tratando de vislumbrar la calidad de asistencia que brinda el sistema experto ante diferentes casos de problema a resolver por software. También se evalúa la amplitud y generalidad de marcos compuestos que posee el repositorio y cómo el sistema guía su uso.

Para el presente trabajo se desarrollarán las fases I y II de la metodología.

En la Fase I, Identificación de la Tarea, se confecciona el plan de requisitos: Documento en el cual quedan plasmadas las necesidades del usuario utilizando todo el espectro de técnicas que brinda la adquisición de conocimientos.

Mediante el Estudio de Viabilidad, se estudia las cuatro dimensiones a saber: Plausibilidad, Adecuación, Justificación y Éxito del problema que se intenta resolver, determinando que el mismo puede ser resuelto utilizando la Ingeniería del Conocimiento.

Una vez considerado viable el desarrollo del proyecto se procede a la definición de las características de la tarea.

En la fase II se desarrolla un primer prototipo que luego se refina gradualmente hasta conseguir las especificaciones precisas del problema. En la primera etapa de esta fase, Concepción de la Solución y Descomposición en subproblemas, se elabora un diseño general del prototipo en conjunto con los expertos considerando las especificaciones del sistema y el plan de requisitos elaborado en la primera fase de la metodología.

Se continúa con la Adquisición de Conocimientos para obtener los conocimientos públicos y educir los conocimientos privados de los expertos, que son modelados en la etapa de Conceptualización.

La Formalización de los Conocimientos determina los modelos de representación adecuados, que permiten definir el diseño detallado del Sistema Experto (motor de inferencia, bases de conocimiento, interfaces, etc.).

La herramienta de implementación del Sistema Experto es Kappa P.C. de la compañía Intellicorp.

Una vez construido el SE, se procede a la Evaluación, utilizando casos de prueba suministrados por los expertos.

En base a los resultados obtenidos se definen nuevos Requisitos y se continúa con el siguiente prototipo.

### **3.7 Gestión del Proyecto**

La gestión del proyecto presupone involucra actividades de planificación, estimación de recursos, seguimiento y control, y evaluación del proyecto.

La determinación del tamaño del producto a desarrollar es una de las primeras tareas en la gestión del proyecto. De acuerdo a las características del problema que se intenta resolver la planificación y estimación será particionada en dos fases. Planificación de Alto Nivel, donde se plantean globalmente las actividades a ser desarrolladas a lo largo del proyecto y Planificación detallada, que se confeccionará luego de las primeras sesiones de Adquisición de Conocimiento y Conceptualización. Dicha planificación constará de mayor detalle, principalmente efectuando estimaciones próximas a lo que será el desarrollo definitivo.

### **3.8 Diagrama de Gantt**

La Tabla 3.1 muestra la correspondiente planificación global de las tareas a ser desarrolladas en el proyecto. En el Anexo A se presenta una Planificación Detallada del proyecto obtenida luego de las primeras sesiones de Adquisición y Conceptualización.

En las sesiones de adquisición iniciales se pudo ajustar las estimaciones realizadas, dado que las sesiones en algunos casos eran con experto locales, otras veces vía internet chat y otras viajando a Buenos Aires.

La conceptualización fue subestimada en cuanto a plazos. En los primeros ciclos de desarrollo se comprobó una extensión mayor a la inicialmente planificada.

Durante todo el proyecto se registra la horas empleadas, recursos utilizados, costos ocasionados, e hitos alcanzados o a alcanzar.



# Capítulo IV

## Estudio de la Viabilidad

## Capítulo IV

### Estudio de la Viabilidad

#### 4.1. Introducción.

El estudio de viabilidad permite determinar si el problema planteado puede ser resuelto mediante un Sistema Experto. Se evalúan los aspectos de Plausibilidad, Adecuación, Justificación y Éxito que caracterizan al problema utilizando el Test de Viabilidad propuesto por la Metodología IDEAL.

Dicho test está conformado por un conjunto de características, a las cuales el Ingeniero de Conocimiento debe asignar valores, de acuerdo al grado de comprensión que este posee del problema, de los expertos, usuarios finales, colaboradores, etc. del proyecto.

El test utiliza las siguientes cuatro dimensiones:

A) *Plausibilidad*: Uno de los requisitos más importantes, por ser condición necesaria, es que existan verdaderos expertos en el área del problema. Estos expertos deberían estar totalmente disponibles para trabajar en el proyecto. Además es imprescindible que el experto sea cooperativo y capaz de articular sus conocimientos y modos de razonamiento. Aquí es crítico disponer de un conjunto de casos de prueba que permitan observar in situ cómo los expertos resuelven el problema, de manera que sea más sencillo entender el proceso real tal como es, así como los conocimientos reales que utilizan.

B) *Justificación*: El esfuerzo de desarrollo de un Sistema Experto se justifica por ejemplo cuando la tarea del experto debe realizarse en entornos hostiles o peligrosos, por lo que no se desea mantener un experto humano en el lugar, o bien, cuando los expertos humanos escasean y una empresa necesita expertos en distintas ubicaciones a la vez. Otra justificación para el desarrollo de un Sistema Experto es la rotación de personal, por ejemplo por jubilación y la experiencia adquirida esta a punto de perderse.

C) *Adecuación*: Para que el desarrollo de un Sistema Experto resulte adecuado, el problema a resolver debe poseer ciertas características intrínsecas, como por ejemplo cuando se necesitan unos conocimientos que son subjetivos, cambiantes, simbólicos, dependientes de los juicios particulares de las distintas personas, o son de naturaleza heurística, etc. Si se cumple alguna de las condiciones mencionadas entonces el problema se ajusta para ser tratado con la INCO.

D) *Éxito*: Existen otras cuestiones no técnicas a tener en cuenta para decidir aplicar la Ingeniería del Conocimiento en la resolución de un problema, como por ejemplo la mentalización de los responsables de modo que los recursos humanos y materiales estén comprometidos en lograr la solución, que las personas implicadas estén lo suficientemente entrenadas, que el Sistema Experto sea finalmente ubicado en el lugar correcto para cumplir su función, que los usuarios lo acepten como una herramienta que mejora su calidad laboral, y que los expertos coincidan en la escuela de pensamiento acerca del problema a resolver.

#### **4.2. Test de Viabilidad.**

El método es de tipo métrico, usa ponderaciones, utiliza la media armónica e incorpora la manipulación de valores lingüísticos mediante intervalos difusos, con los que, además, se pueden definir operaciones básicas de cálculos.

El método integra tres tipos de valores para las características: booleanos, que podrán tener los valores Sí o No, numéricos en el intervalo  $[x,y]$ ; y lingüísticos. Se trata de conservar la naturaleza de cada tipo de valor por lo que cada uno es traducido a un intervalo difuso, desarrollándose todos los cálculos con dichos intervalos. Esto es porque el cerebro humano piensa, en general, con valores lingüísticos en vez de valores numéricos.

Los valores lingüísticos se podrán tomar de entre un conjunto de los cinco valores siguientes: "nada", "poco", "regular", "mucho", "todo". Cuanto más verdadera parece la característica, mayor valor se le asigna, es decir, "mucho" o "todo", "poco" o "nada" se dan a características que parecen falsas. Finalmente, el valor "regular" es para los casos en los que no se sabe muy bien. Estos valores se pueden ver como cuantificadores de las características.

Todos los valores lingüísticos se han traducido en valores difusos. El intervalo dentro del cual se expresarán todos los valores difusos es  $[0,10]$ .

La Tabla 4.1 muestra las funciones de pertenencia para los respectivos valores.

Un valor lingüístico se define por su función de pertenencia del intervalo [0,10] en el intervalo [0,1]; que indica en que grado se ajusta a dicho valor lingüístico, sabiendo que cuanto más se acerca la función a 1, más cierto es el valor lingüístico. Así mismo se muestra en la tabla 4.2 la función de pertenencia para valores booleanos a los efectos de realizar el cálculo en base a intervalos difusos. Por otra parte, a continuación se muestra la función de pertenencia del conjunto difuso cuyo único elemento es un número a.

$$\mu(x) = \begin{cases} 1 & \text{si } x = a \\ 0 & \text{en otro caso} \end{cases}$$

Esta función sirve para manipular características que toman valores nítidos. Como se puede visualizar en la Figura 4.1, las gráficas de las funciones de pertenencia pueden ser definidas gracias a sus puntos de ruptura o puntos angulares. A cada valor lingüístico le será asociado un intervalo difuso, determinado por los siguientes puntos angulares.

Valor Lingüístico	Intervalo difuso			
Muy Poco o Nada	0,01	0,01	1,2	1,2
Poco	1,2	2,2	3,4	4,4
Regular	3,4	4,4	5,6	6,6
Mucho	5,6	6,6	7,8	8,8
Muchísimo o Todo	7,8	8,8	10	10

Tabla 4.1: Definición de Intervalos Difusos correspondientes a cada valor lingüístico

En la tabla 4.1 se puede apreciar 4 columnas que definen el intervalo difuso. Cada uno de dichos valores se denomina "punto angular" o "punto de ruptura", dado que es en dichos puntos donde el valor de la característica cambia su función de pertenencia. Por ejemplo para el caso del valor lingüístico "regular", el punto de ruptura "3,4" indica que a partir de allí la característica no tiene más el valor "Cero", pero tampoco es "uno". El valor 4,4 indica que a partir de allí la característica tiene valor "Uno", hasta el punto de ruptura "5,6" en el cuál la característica se vuelve nuevamente difusa hasta el valor "6,6" a partir del cuál el valor es "cero". Puede observarse también que se utilizan para "Muy Poco o Nada" los valores 0,01 en lugar de cero. Esto es para evitar, en ciertas ocasiones, la división por cero.

Valor Lingüístico	Intervalo difuso			
No	0,01	0,01	0,01	0,01
Sí	10	10	10	10

Tabla 4.2: Definición de Intervalos binarios correspondientes a cada valor lingüístico

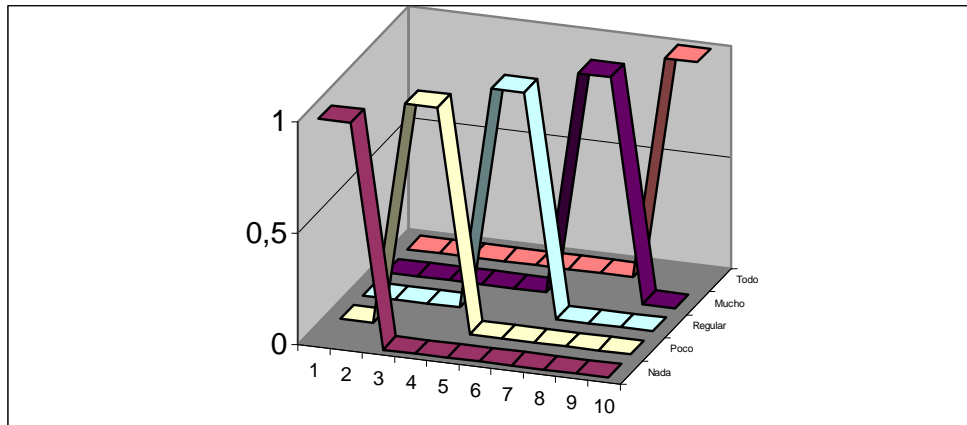


Figura 4.1: Gráfico de las funciones de pertenencia de los valores lingüísticos utilizados en el Test de Viabilidad

Se puede observar en la figura 4.1 los puntos de ruptura o angulares que definen la función de pertenencia.

Además, las características poseen otros componentes indicativos de su naturaleza, que hay que tener en cuenta para su consideración y uso en el Test de Viabilidad. Dichas características son:

**Categoría:** es únicamente de carácter indicativo y muestra a qué o a quién se referirá la característica. Puede ser a la Tarea, a los Directivos/Usuarios o a los Expertos.

**Peso:** Permite dar una importancia relativa a cada característica en la globalidad del test. El peso tiene dos componentes, una de carácter numérico que puede tomar valor entero en el intervalo [1,10]. La otra de carácter binario toma el valor + si la importancia relativa que aporta la característica favorece la construcción del SE, y el valor - si hace disminuir el grado de interés en el desarrollo del SE.

**Naturaleza** del valor asociado a la característica puede ser: booleano, numérico o lingüístico.

*Tipo:* una característica puede ser de dos tipos: deseable o esencial y muestra su importancia. Si es vital para el proyecto, es esencial, una característica de este tipo deberá superar un valor de umbral, de lo contrario el proyecto deberá ser inmediatamente abandonado. En otro caso la característica se considera deseable.

*Umbral:* Es una referencia para características esenciales. El valor del umbral es fijo, pero no necesariamente igual para todas las características y es de la misma naturaleza que el valor de las características.

*Valor:* para cada proyecto concreto hay que asignar un valor a cada característica dentro del conjunto de valores adecuados para cada naturaleza.

### 4.3 Funcionamiento de la Técnica

La representación de los valores en intervalos difusos según los cuatro puntos angulares mencionados anteriormente, permiten trabajar con estos como si fueran valores numéricos.

La media armónica proporciona los valores más aceptables para el problema, con el único inconveniente que si hay un valor "cero" en el conjunto de los valores de los que se hace la media, el resultado obtenido es "cero". Esto se soluciona haciendo la media armónica y la media aritmética del conjunto de intervalos y luego, hacer la media aritmética de los dos intervalos obtenidos. Es decir:

$$VC_i = \frac{1}{2} \frac{\sum_{k=1}^{r_i} P_{ik}}{\sum_{k=1}^{r_i} V_{ik}} + \frac{1}{2} \frac{\sum_{k=1}^{r_i} P_{ik} V_{ik}}{\sum_{k=1}^{r_i} P_{ik}}$$

Donde:

$VC_i$  Valor Global de la aplicación en una dimensión dada.

$V_{ik}$  Valor de la característica k en la dimensión j.

$P_{ik}$  Peso de la característica k en la dimensión j.

$r_i$  número de la característica en la dimensión j.

La suma de intervalos se realiza de la siguiente forma:

$$(V_1, V_2, V_3, V_4) + (W_1, W_2, W_3, W_4) = (V_1 + W_1, V_2 + W_2, V_3 + W_3, V_4 + W_4)$$

Si una sola característica de justificación tiene un valor muy alto, enteramente está justificado el desarrollo del sistema experto.

La viabilidad técnica del proyecto es más dependiente de las Plausibilidad y la Adecuación que de la Justificación o del Éxito.

La Justificación del proyecto es importante únicamente antes de que empiece el desarrollo del sistema. Para determinar la evaluación de viabilidad del proyecto, se calculará el valor final, mediante la media aritmética ponderada de los valores obtenidos para cada dimensión con los pesos:

- 8 Para Plausibilidad y Adecuación.
- 3 Para Justificación.
- 5 Para Éxito.

Con la fórmula siguiente:

$$V_f = \frac{\sum_{i=1}^4 P_i V_i}{\sum_{i=1}^4 P_i}$$

Aquí el producto de un intervalo por un número se define como:

$$a * (V_1, V_2, V_3, V_4) = (a * V_1, a * V_2, a * V_3, a * V_4)$$

La tarea es aceptada si se obtiene un valor mayor o igual a 6.

#### **4.4 Análisis del Test de Viabilidad**

Las siguientes tablas muestran las evaluaciones realizadas para cada dimensión.

Denominación de la característica	Categoría	Dimensión	Peso (P)	Tipo	Naturaleza	Umbral	Valor
Existen expertos, están disponibles y son cooperativos	Experto	P1	+10	Esencial	Booleana	Sí (sí)	Sí
El experto es capaz de estructurar sus métodos y procedimientos de trabajo	Experto	P2	+7	Deseable	Difusa	No	Mucho
La tarea está bien estructurada y se entiende	Tarea	P3	+8	Deseable	Difusa	No	Mucho
Existen suficientes casos de prueba y sus soluciones asociadas	Tarea	P4	+10	Esencial	Numérica	Sí(8)	8
La tarea sólo depende de los conocimientos y no del sentido común	Tarea	P5	+9	Deseable	Numérica	No	8
Resuelve una tarea útil y necesaria	Tarea	J1	+8	Deseable	Difusa	No	Mucho
Se espera una alta tasa de recuperación de la inversión	Directivos/ usuarios	J2	+7	Deseable	Numérica	No	8
Hay escasez de experiencia humana	Experto	J3	+6	Deseable	Difusa	No	Mucho
Hay necesidad de tomar decisiones en situaciones críticas o ambientes hostiles, penosos y, o, poco gratificantes	Tarea	J4	+10	Deseable	Difusa	No	Poco
Hay necesidad de distribuir los conocimientos	Tarea	J5	+10	Deseable	Difusa	No	Todo
Los conocimientos pueden perderse de no realizarse el sistema	Experto	J6	+10	Deseable	Difusa	No	Mucho
No existen soluciones alternativas	Tarea	J7	+8	Esencial	Booleana	Sí (sí)	Sí
La transferencia de experiencia entre humanos es factible	Tarea	A1	+7	Deseable	Difusa	No	Mucho

*Tabla 4.3: Análisis de las características de Plausibilidad, Adecuación, Justificación y Exito.*



Denominación de la característica	Categoría	Dimensión	Peso (P)	Tipo	Naturaleza	Umbral	Valor
la tarea requiere "experiencia"	Tarea	A2	+10	Deseable	Difusa	No	Mucho
Los efectos de la introducción del SE no pueden preverse	Tarea	A3	-2	Deseable	Difusa	No	Poco
la tarea requiere razonamiento simbólico	Tarea	A4	+5	Deseable	Difusa	No	Mucho
La tarea requiere el uso de "heurísticas" para acotar el espacio de búsqueda	Tarea	A5	+7	Deseable	Difusa	No	Mucho
la tarea es de carácter práctico y más táctica que estratégica	Tarea	A6	+8	Deseable	Booleana	No	Sí
Se espera que la tarea continúe sin cambios significativos durante un largo período de tiempo	Tarea	A7	+8	Esencial	Difusa	Si (mucho)	Mucho
Se necesitan varios niveles de abstracción en la resolución de la tarea	Tarea	A8	+8	Deseable	Difusa	No	Poco
El problema es relativamente simple o puede descomponerse en subproblemas	Tarea	A9	+6	Deseable	Difusa	No	Mucho
El experto no sigue un proceso determinista en la resolución del problema	Experto	A10	+3	Deseable	Booleana	No	Sí
La tarea acepta la técnica del prototipado gradual	Tarea	A11	+8	Deseable	Booleana	No	Sí
El experto resuelve el problema a veces con información incompleta o incierta	Experto	A12	+3	Deseable	Difusa	No	Mucho
Es conveniente justificar las soluciones adoptadas	Tarea	A13	+3	Deseable	Difusa	No	Mucho

*Tabla 4.3: Análisis de las características de Plausibilidad, Adecuación, Justificación y Exito.*

Denominación de la característica	Categoría	Dimensión	Peso (P)	Tipo	Naturaleza	Umbral	Valor
La tarea requiere investigación básica	Tarea	A14	-10	Esencial	Booleana	Sí (No)	No
El sistema funcionará en “tiempo real” con otros programas o dispositivos	Tarea	A15	-6	Deseable	Difusa	No	Nada
Existe una ubicación idónea para el SE	Directivos/ usuarios	E1	+7	Deseable	Difusa	No	Regular
Problemas similares se han resuelto mediante INCO	Tarea	E2	+8	Deseable	Booleana	No	Sí
El problema es similar a otros en los que resultó imposible aplicar esta tecnología	Tarea	E3	-5	Deseable	Booleana	No	No
La continuidad del proyecto está influenciada por vaivenes políticos	Directivos/ usuarios	E4	-9	Esencial	Difusa	Si (poco)	Nada
La inserción del sistema se efectúa sin traumas, es decir, apenas se interfiere en la rutina cotidiana	Directivos/ usuarios	E5	+8	Deseable	Difusa	No	Poco
Se dispone de experiencia en INCO	Tarea	E6	+7	Deseable	Difusa	No	Regular
Se dispone de los recursos humanos, hardware y software necesarios para el desarrollo e implantación del sistema	Tarea	E7	+4	Deseable	Difusa	No	Todo
El experto resuelve el problema en la actualidad	Experto	E8	+4	Deseable	Difusa	No	Todo
La solución del problema es prioritaria para la institución	Directivos/ usuarios	E9	+8	Esencial	Difusa	Si (mucho)	Mucho
Las soluciones son explicables	Tarea	E10	+5	Deseable	Difusa	No	Mucho

*Tabla 4.3: Análisis de las características de Plausibilidad, Adecuación, Justificación y Exito.*

Denominación de la característica	Categoría	Dimensión	Peso (P)	Tipo	Naturaleza	Umbral	Valor
Los objetivos del sistema son claros y evaluables	Tarea	E11	+6	Deseable	Difusa	No	Mucho
Los conocimientos están repartidos entre un conjunto de individuos	Experto	E12	-7	Deseable	Difusa	No	Poco
Los directivos, usuarios, expertos e IC están de acuerdo en las funcionalidades del SE	Directivos/ usuarios	E13	+4	Esencial	Difusa	Sí (mucho)	Mucho
La actitud de los expertos ante el desarrollo del sistema es positiva y no se sienten amenazados por el proyecto	Experto	E14	+8	Deseable	Difusa	No	Mucho
los expertos convergen en sus soluciones y métodos	Experto	E15	+5	Deseable	Difusa	No	Mucho
Se acepta la planificación del proyecto propuesta por el IC	Directivos/ usuarios	E16	+8	Esencial	Booleana	Sí (si)	Sí
Existen limitaciones estrictas de tiempo en la realización del sistema	Tarea	E17	-6	Deseable	Difusa	No	Regular
la dirección y usuarios apoyan los objetivos y directrices del proyecto	Directivos/ usuarios	E18	+7	Esencial	Difusa	Sí (mucho)	Mucho
El nivel de formación requerido por los usuarios del sistema es elevado	Directivos/ usuarios	E19	-2	Deseable	Difusa	No	Mucho
Las relaciones IC - Experto son fluidas	Experto	E20	+4	Deseable	Difusa	No	Regular
El proyecto forma parte de un camino crítico con otros sistemas	Tarea	E21	-6	Deseable	Booleana	No	No
Se efectuará una adecuada transferencia tecnológica	Directivos/ usuarios	E22	+8	Esencial	Difusa	Si (mucho)	Mucho
lo que cuenta en la solución es la calidad de la respuesta	Experto	E23	+5	Deseable	Booleana	No	Sí

*Tabla 4.3: Análisis de las características de Plausibilidad, Adecuación, Justificación y Exito.*



#### 4.4.1 Cálculo de los intervalos correspondientes a cada dimensión

Se exponen a continuación los resultados obtenidos para cada una de las dimensiones analizadas.

En el apéndice B se detalla los cálculos realizados para cada una.

Dimensión de Plausibilidad

$$V_r = (7.452, 7.884, 8.346, 8.695)$$

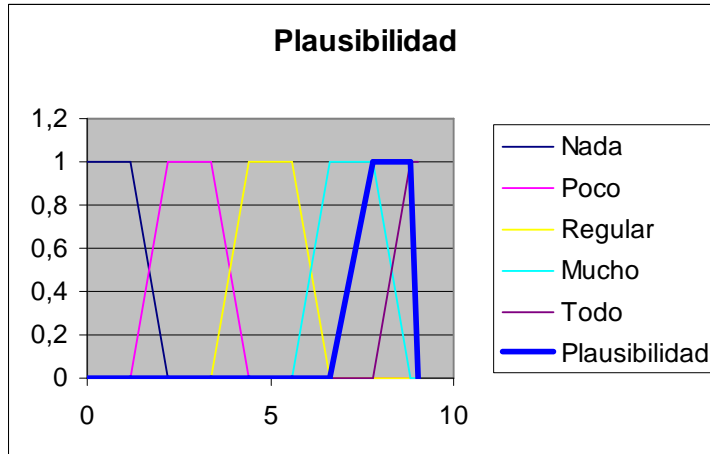


Figura 4.2

Dimensión de Justificación

$$V_r = (7.8, 8.8, 10, 10)$$

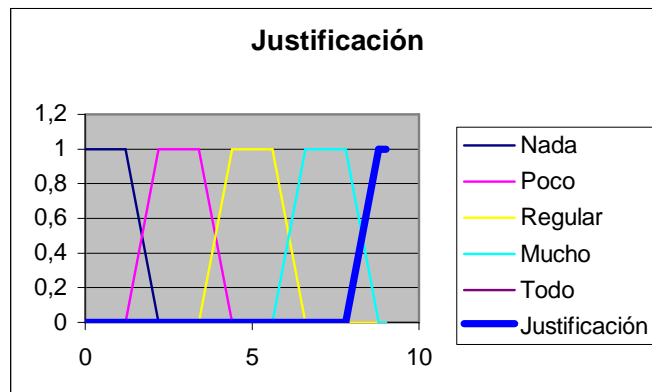


Figura 4.3

### Dimensión de Adecuación

$$V_r = (4.248, 4.729, 5.221, 5.702)$$

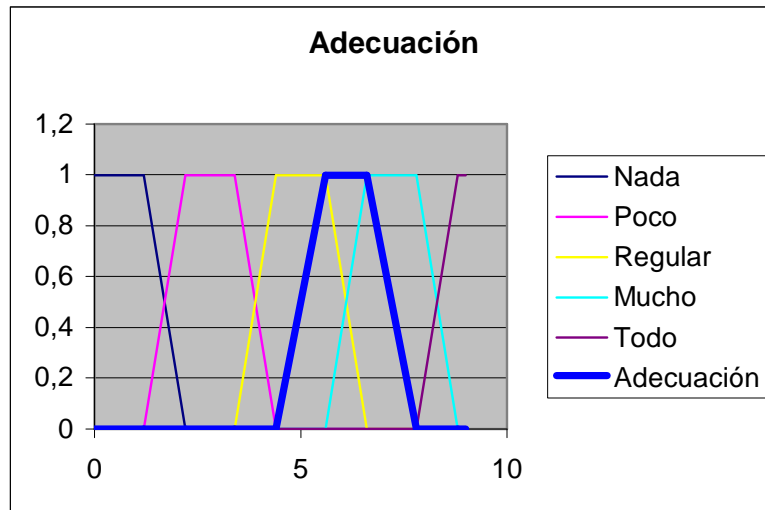


Figura 4.4

### Dimensión de Éxito

$$V_r = (4.178, 4.671, 5.172, 5.607)$$

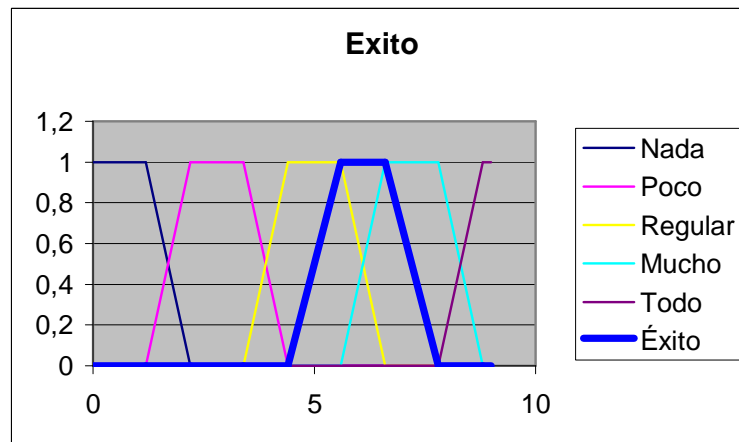


Figura 4.5

Finalmente una vez obtenidos los intervalos resultantes de las cuatro dimensiones, se efectúa el cálculo final para determinar la viabilidad general del proyecto.

El intervalo final dá como resultado:

$$V_f = (5.75 , 6.28 , 6.85 , 7.22)$$

Entonces:

$$\frac{5.75 + 6.28 + 6.85 + 7.22}{4} > 6$$

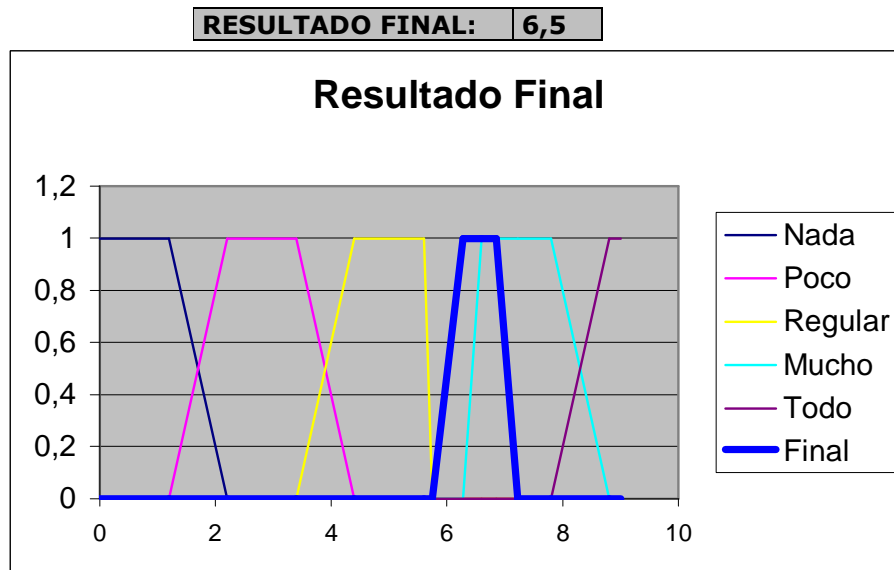


Figura 4.6

**Dado que es superior a 6, es viable la construcción del sistema experto.**

#### 4.4.2 Justificación del análisis de viabilidad

##### 4.4.2.1 justificación de la Dimensión de Plausibilidad

Característica P1: *Existen expertos, están disponibles y son cooperativos.*

Análisis: Se dispone de personas en el ámbito universitario que tienen experiencia en el área de requisitos, tanto en el área de posgrado como externa a ella. En el ámbito de la CAPIS existen personas con experiencia interesadas en la construcción del sistema. Se dispone del recurso de internet para consultar con expertos externos.

Valor: Sí.

↳ Característica P2: *El experto es capaz de estructurar sus métodos y Procedimientos de trabajo.*

Análisis: El experto local ha sido compañero de tareas del autor por años, lo que permite aseverar que es capaz de estructurar sus métodos y procedimientos de trabajo.

Valor: Mucho

↳ Característica P3: *La tarea está bien estructurada y se entiende*

Análisis: La tarea está bien estructurada, dado que se han identificado la mayoría de las funciones que se deben realizar:

- Captura de información acerca de los requisitos elicitados previamente.
- Análisis de dicha información para el posterior guiado al usuario según ciertos criterios y estrategias de acuerdo a patrones de especificaciones.
- Captura de información faltante según avanza el proceso.
- Crítica a los resultados obtenidos
- Recomendaciones en función de un estándar de especificación de requisitos y refinamiento de requisitos elicitados.

Valor: Mucho

↳ Característica P4: *Existen suficientes casos de prueba y sus soluciones asociadas*

Análisis: Tanto en la bibliografía, como en el material recopilado se dispone de documentación que sirve como ejemplo de documento de requerimientos y además se cuenta con varios ejemplos de análisis del problema y descomposición del dominio de la aplicación.

Valor: 8

↳ Característica P5: *La tarea sólo depende de los conocimientos y no del sentido común*

Análisis: Se debe disponer de material como por ejemplo la norma estándar IEEE/ANSI 830-1998, las técnicas de modelado de datos, estados, transiciones, taxonomías, etc. por lo que podemos decir que la tarea depende en gran parte de los conocimientos que se posean acerca de cómo documentar requerimientos y de cómo analizar el dominio de la aplicación y el problema que el software debe resolver.

Valor: 8

#### **4.4.2.2 Justificación de la Dimensión de Justificación**

↳ Característica J1: *Resuelve una tarea útil y necesaria*

Análisis: Dado que el desarrollo de software es una actividad intensamente cognoscitiva, la disponibilidad de una herramienta de asistencia al desarrollo basada en conocimiento sería de gran utilidad. Además permitiría la disponibilidad de conocimientos y experiencia actualizada y en permanente crecimiento.

Valor: Mucho



↳ Característica J2: Se espera una alta tasa de recuperación de la inversión

Análisis: La recuperación de la inversión se espera en términos de una mejor productividad, dado que consiguiendo una especificación de requisitos acertada y de gran calidad, que asegure que los requisitos se adecúen de la mejor manera posible a las necesidades del usuario, se logrará que el resto del proceso software se ajuste a los plazos y esfuerzos previstos.

Valor: 8

↳ Característica J3: *Hay escasez de experiencia humana*

Análisis: No es frecuente encontrar expertos en el análisis de requisitos. Se requieren personas que hayan trabajado en una gran cantidad de proyectos software.

Valor: Mucho

↳ Característica J4: *Hay necesidad de tomar decisiones en situaciones críticas o ambientes hostiles, penosos y, o, poco gratificantes*

Análisis: El trabajo se realiza en ambientes preparados para el trabajo de desarrollo, por lo que las situaciones no son críticas.

Valor: Poco

↳ Característica J5: *Hay necesidad de distribuir los conocimientos*

Análisis: Se espera que el sistema ayude y pueda ser utilizado por toda la comunidad informática.

Valor: Todo

↳ Característica J6: *Los conocimientos pueden perderse de no realizarse el sistema*

Análisis: El trabajo se realiza basándose parte en bibliografía, revistas, etc. Y parte en expertos, dado que para el análisis de requisitos se utilizan heurísticas. No obstante en cualquier organización las personas que se dedican al análisis de requisitos son escasas y existe una gran dependencia para con ellas. Si tales personas abandonan la organización, puede perderse los conocimientos.

Valor: Mucho

↳ Característica J7: *No existen soluciones alternativas*

Análisis: Se han desarrollado muchas técnicas y herramientas para asistir al proceso de desarrollo de software. Tales herramientas y técnicas deben ser empleadas por personas expertas para poder realizar la tarea. Por ejemplo la interacción con el cliente para extraer los requisitos iniciales, controlar las subsiguientes adquisiciones, reconocer requisitos erróneos, explicar la especificación al cliente y validarla, como manejar los conflictos entre requisitos, etc.

Valor: Sí

#### **4.4.4.3 Justificación de la Dimensión de Adecuación**

↳ Característica A1: *La transferencia de experiencia entre humanos es factible*

Análisis: Existen reglas empíricas que permiten transmitir la valoración de ciertos aspectos del dominio de la aplicación para subdividir el problema en subproblemas, para luego poder ser documentados de acuerdo a reglas también claras según el tipo de dominio del cual se trate. Esto hace factible que dicha experiencia pueda transmitirse a otras personas mediante el guiado del proceso.

Valor: Mucho

↳ Característica A2: *la tarea requiere "experiencia"*

Análisis: La actividad de requisitos requiere que la persona que la realiza posea una gran experiencia y haber trabajado en muchos proyectos.

Valor: Mucho

↳ Característica A3: *los efectos de la introducción del SE no pueden preverse*

Análisis: Se espera que la introducción del SE no depare efectos adversos en cuanto a que la tarea que realizará será asistir al analista de requisitos acelerando su trabajo y agregándole confiabilidad.

Valor: Poco

↳ Característica A4: *la tarea requiere razonamiento simbólico*

Análisis: El objetivo es formalizar las necesidades del usuario o cliente en una especificación que sirva de información de entrada al proceso de diseño.

Valor: Mucho

↳ Característica A5: *La tarea requiere el uso de "heurísticas" para acotar el espacio de búsqueda*

Análisis: Cuando se presenta un problema que debe analizarse para luego poder documentar los requerimientos es necesario el uso de heurísticas; las mismas permiten que el espacio de búsqueda quede acotado en el momento de aconsejar al ingeniero de software sobre los pasos a seguir.

Valor: Mucho

↳ Característica A6: *la tarea es de carácter práctico y más táctica que estratégica*

Análisis: La tarea está orientada fuertemente a la meta de especificar los requisitos según el estándar IEEE/ANSI 830-1998. El alcance está dado para el proyecto en particular por lo que afecta en menor medida a la organización. Esto hace que sea eminentemente táctica.

Valor: Sí

↳ Característica A7: *Se espera que la tarea continúe sin cambios significativos durante un largo período de tiempo*

Análisis: El análisis de requisitos utiliza técnicas y metodologías desarrolladas en la década del 70 y los 80. El estándar de requisitos tiene una versión anterior de 1984 por lo que se espera que el objetivo a lograr permanezca estable por un considerable período de tiempo.

Valor: Mucho

↳ Característica A8: *Se necesitan varios niveles de abstracción en la resolución de la tarea.*

Análisis: Los conocimientos para resolver la tarea no son complejos. Se necesitan conocer algunas técnicas de especificación fácilmente comprensibles por usuarios no técnicos.

Valor: Poco

↳ Característica A9: *El problema es relativamente simple o puede descomponerse en subproblemas.*

Análisis: Se puede descomponer el problema en subproblemas, como por ejemplo el guiado según el estándar de requisitos, el proceso de buscar modelos patrones

que coincidan con el modelo que el usuario ha ingresado, el proceso de recomendación y crítica al modelo obtenido, etc.

Valor: Mucho

↳ Característica A10: *El experto no sigue un proceso determinista en la resolución del problema*

Análisis: En absoluto. Cada proyecto a analizar es un problema diferente y su resolución demandará un nuevo análisis, el uso de diferentes técnicas según el dominio de la aplicación, el contexto, el tipo de usuarios, etc.

Valor: Sí

↳ Característica A11: *La tarea acepta la técnica del prototipado gradual*

Análisis: Se han determinado distintos subproblemas para los cuales es posible iniciar el proceso de prototipado del sistema experto. Además los requisitos iniciales no están del todo claros por lo que el sistema puede ser desarrollado mediante el prototipado incremental.

Valor: Sí

↳ Característica A12: *El experto resuelve el problema a veces con información incompleta o incierta*

Análisis: A menudo, con el documento de elicitación, no se tiene una descripción completa de los requisitos, sólo son declaraciones tentativas de las necesidades de los usuarios, o de lo que se espera que el sistema software realice. En algunos casos porque los usuarios tienen una idea vaga de lo que esperan del sistema, o porque existe un desarrollo de hardware en paralelo al sistema software o porque no es suficientemente completa la tarea de elicitación. Por lo tanto el experto suele resolver el problema contando con información incompleta o incierta.

Valor: Mucho

↳ Característica A13: *Es conveniente justificar las soluciones adoptadas*

Análisis: Es absolutamente necesario justificar las soluciones adoptadas explicando al usuario el porqué de la sugerencia o guía para poder determinar los requisitos, dado que es el objetivo del SE.

Valor: Mucho

↳ Característica A14: *La tarea requiere investigación básica*

Análisis: La tarea no requiere investigación básica.

Valor: No

↳ Característica A15: *El sistema funcionará en “tiempo real” con otros programas o dispositivos*

Análisis: No trabajará en tiempo real con otros dispositivos o programas.

Valor: Nada

#### **4.4.4.4 Justificación de la Dimensión de éxito**

↳ Característica E1: *Existe una ubicación idónea para el SE*

Análisis: El SE será utilizado por los integrantes del equipo encargados de la ingeniería de requerimientos. No obstante el director del proyecto debe poder tener acceso a su utilización para justificar o rectificar decisiones, para coordinar tareas a realizar por el resto del equipo de desarrollo. No parece prioritario una ubicación única y eficaz del SE.

Valor: Regular

↳ Característica E2: *Problemas similares se han resuelto mediante INCO*

Análisis: En la bibliografía encontrada aparecen trabajos de sistemas de análisis de requisitos.

Valor: Sí

↳ Característica E3: *El problema es similar a otros en los que resultó imposible aplicar esta tecnología*

Análisis: No se conoce la existencia de tales casos.

Valor: No

↳ Característica E4: *La continuidad del proyecto está influenciada por vaivenes políticos*

Análisis: No es de esperar este tipo de influencias.

Valor: Nada

↳ Característica E5: *La inserción del sistema se efectúa sin traumas, es decir, apenas se interfiere en la rutina cotidiana*

Análisis: El trabajo será el mismo pero con la ayuda y la asistencia del sistema, por lo que este tipo de herramientas son bienvenidas. No obstante todo cambio genera algo de resistencia.

Valor: Poco

↳ Característica E6: *Se dispone de experiencia en INCO*

Análisis: No existe experiencia previa. Pero se dispone de material realizado por expertos con experiencia previa.

Valor: Regular

↳ Característica E7: *Se dispone de los recursos humanos, hardware y software necesarios para el desarrollo e implantación del sistema*

Análisis: Se cuenta con computadoras para el desarrollo, el software adecuado, material bibliográfico, la disponibilidad de personal universitario y de expertos externos.

Valor: Todo

↳ Característica E8: *El experto resuelve el problema en la actualidad*

Análisis: Se encuentran en el ámbito de la ingeniería del software, donde se resuelve el problema del análisis de requisitos cotidianamente.

Valor: Todo

↳ Característica E9: *La solución del problema es prioritaria para la institución*

Análisis: Es prioritaria dado que es una rama de investigación que se desea comenzar y proseguir con futuros trabajos.

Valor: Mucho

↳ Característica E10: *Las soluciones son explicables*

Análisis: El sistema debe explicar cada solución adoptada de modo de brindar trazabilidad a los requisitos y ayudar al usuario a continuar con el proceso.

Valor: Mucho

↳ Característica E11: *Los objetivos del sistema son claros y evaluables*

Análisis: Los objetivos está claramente establecidos por el estándar a seguir, y del mismo modo puede ser evaluado al finalizar el trabajo o mientras es realizado. Además se puede evaluar mediante la comparación del documento de elicitación antes y después verificando si se produjo una depuración del mismo.

Valor: Mucho

↳ Característica E12: *Los conocimientos están repartidos entre un conjunto de individuos*

Análisis: Si bien se cuenta con varias personas expertas en la materia, todas poseen un nivel similar de conocimientos por lo que no es necesaria su presencia simultánea.

Valor: Poco

↳ Característica E13: *Los directivos, usuarios, expertos e IC están de acuerdo en las funcionalidades del SE*

Análisis: Se ha llevado una tarea de conciliación de objetivos.

Valor: Mucho

↳ Característica E14: *La actitud de los expertos ante el desarrollo del sistema es positiva y no se sienten amenazados por el proyecto*

Análisis: Se espera con entusiasmo el desarrollo para luego continuar con el proyecto global de investigación.

Valor: Mucho

↳ Característica E15: *los expertos convergen en sus soluciones y métodos*

Análisis: Todos los involucrados poseen una formación similar y adoptan soluciones convergentes.

Valor: Mucho

↳ Característica E16: *Se acepta la planificación del proyecto propuesta por el IC*

Análisis: Fue aceptada.

Valor: Sí

↳ Característica E17: *Existen limitaciones estrictas de tiempo en la realización del sistema*

Análisis: Si bien no existen plazos rígidos es deseable finalizarlo en un tiempo razonable en función de los futuros proyectos.

Valor: Regular

↳ Característica E18: *la dirección y usuarios apoyan los objetivos y directrices del proyecto*

Análisis: El sistema es parte de un proyecto de investigación.

Valor: Mucho

↳ Característica E19: *El nivel de formación requerido por los usuarios del sistema es elevado*

Análisis: Está dirigido a Ingenieros en Software o con experiencia similar.

Valor: Mucho

↳ Característica E20: *Las relaciones IC - Experto son fluidas*

Análisis: Son fluidas existiendo múltiples medios adecuados para la comunicación. No obstante la distancia física hace que dicha característica se vea levemente disminuída.

Valor: Regular

↳ Característica E21: *El proyecto forma parte de un camino crítico con otros sistemas*

Análisis: El resto de proyectos no tienen una definición precisa todavía por lo que no forma parte de un camino crítico.

Valor: No

↳ Característica E22: *Se efectuará una adecuada transferencia tecnológica*

Análisis: Se elaborarán los manuales necesarios y clases de capacitación para el uso por parte de los expertos.

Valor: Mucho

↳ Característica E23: *lo que cuenta en la solución es la calidad de la respuesta*

Análisis: Es muy importante la calidad de las respuestas, dado que una mala especificación de requisitos lleva a fallas en el sistema, retrasos y sobre costos.

Valor: Sí



# Capítulo V

## Adquisición de Conocimientos

## Capítulo V

### Adquisición de conocimientos

#### 5.1 Introducción

El proceso de adquisición seguido ha sido subdividido en diferentes perspectivas de profundización gradual, con las siguientes etapas:

1. Primeras reuniones y evaluación de la viabilidad
2. Extracción de conocimientos (A partir de la documentación disponible, como por ejemplo, libros, conferencias, proceedings, papers, etc)
3. Educación de conocimientos (A partir de los expertos)
  - a. Interrogatorios iniciales
  - b. Investigación profunda.

La Adquisición de conocimientos se comenzó con una serie de reuniones con los expertos, y posibles usuarios del Sistema Experto. Estas reuniones han servido para:

- Determinar los requisitos funcionales del Sistema Experto, las necesidades de los usuarios del futuro sistema, y lo que los usuarios esperan del mismo. La sesión 1 fue realizada para alcanzar estos objetivos.
- Introducir al IC en el dominio a un nivel tal que sea capaz de desarrollar un estudio de Viabilidad del Sistema Experto donde se determine si la tarea de los expertos es tratable mediante la INCO. La Sesión 2, que se describe en el presente capítulo refleja la entrevista desarrollada con el objetivo de vislumbrar características esenciales para evaluar la viabilidad del proyecto.

En las primeras reuniones se ha buscado describir conocimientos generales, así como afianzarse con la terminología del dominio.

El siguiente paso en el proceso de adquisición ha sido el estudio de la documentación existente. En este estudio se ha conseguido aprender sobre el dominio de la ingeniería de requerimientos utilizando Marcos de Problema, técnicas de documentación de requerimientos y el proceso de elicitación, lo cual ha

permitido estudiar y asimilar conocimientos, que ha favorecido la interrelación con los expertos.

En el tercer paso, en el ciclo de educación de conocimientos se ha obtenido los conocimientos genuinamente privados de los expertos. La educación es, especialmente, el proceso de interacción con un experto humano con el propósito de construir un SE. El proceso de educación puede dividirse en dos etapas fundamentales: el interrogatorio inicial, y la investigación profunda. En el interrogatorio inicial se ha tratado de obtener una visión general de alto nivel del dominio, donde se ha comprendido el alcance del dominio; la tarea de los expertos y el entorno de la tarea.

En el proceso de investigación profunda se ha tratado de obtener una visión de muy bajo nivel del dominio, donde se llega a comprender el verdadero proceso de la tarea que desempeñan los expertos.

## **5.2 Ciclo de cada sesión**

Para desarrollar la tarea se ha seguido el siguiente ciclo para cada sesión:

### **1. Preparación de la sesión.**

- Información a tratar.
- Amplitud, profundidad, etc.
- Técnica adecuada.
- Preparación de preguntas.

### **2. Sesión.**

- Repaso del análisis.
- Explicación al experto de los objetivos.
- Evaluación de la sesión con el experto.
- Resumen y comentarios del experto.

### **3. Transcripción.**

### **4. Análisis de la Sesión.**

- Lectura para obtener una visión general.
- Extracción de conocimientos concretos.
- Lectura para recuperar detalles olvidados.
- Críticas para mejorar por parte del IC.

### 5.3. Sesiones de Adquisición

#### 5.3.1 Sesión 1: Primera Reunión con Expertos

##### Preparación de la Sesión

- Información a tratar: primer acercamiento al problema.
- Amplitud y profundidad: Se trata de esclarecer la tarea que lleva adelante el experto, sin detallar casos específicos.
- Técnica adecuada: Entrevista no estructurada
- Preparación de Preguntas:
  - ¿En qué consiste la actividad que desarrolla el experto?
  - ¿Dónde se lleva a cabo la actividad?
  - ¿Cómo se lleva a cabo la tarea?
  - ¿A qué estará abocada la tarea del sistema experto?

##### Transcripción de la Sesión 1

**Sesión 1****Fecha:** 1/04/00**Experto:** Lic. Bibiana Rossi**Lugar:** ITBA - CAPIS**IC:** Ing. Marcelo Rizzi**Técnica empleada:** Cuestionario**Objetivos:** Obtener conceptos básicos sobre el dominio y la tarea que debe realizar el SE.

#### 1. Define los términos requerimientos y requisitos

*En general se suelen usar como sinónimos requisitos y requerimientos si te parece bien te proponemos usarlos con el siguiente significado:*

*Requerimientos: es la especificación de la información que se necesita para realizar determinadas tareas o funciones.*

*Requisitos: es una condición y/o especificación técnica u operativa respecto de los requerimientos.*

*Si usamos esta convención, el trabajo lo estamos haciendo sobre requerimientos y no requisitos.*

#### 2. Define la especificación

*Especificación de requerimientos: identificación detallada de la información que se necesita para realizar determinadas tareas o funciones.*

*Analisis de requerimientos: el estudio de los requerimientos especificados con el fin de sacar conclusiones, pueden ser: completitud, consistencia, etc.*

*Desde este punto de vista, el sistema que propones debería estar orientado a especificacion y no a analisis.*

### **3.Cuál es la tarea que debería desempeñar el SE**

*Sería un asistente para definir requisitos sin considerar requisitos anteriores y sin considerar un dominio específico. Es más una ayuda para asistir en la definición de todos los datos que se necesitan para especificar el requerimiento.*

### **4. Dentro de Ingeniería de Requisitos tenemos las áreas de elicitación y de especificación. Sobre qué área trabajará?**

*Obviamente son áreas absolutamente relacionadas. Si estas de acuerdo me parece más conveniente trabajar sobre la especificación. Por ejemplo si los requisitos fueron elicitados y comienzo a usar el SE para especificarlos, el SE me ira pidiendo la informacion necesaria, en ese proceso es posible que me requiera informacion o precision que no se haya elicitado, desde ese punto de vista el SE estaría colaborando en el proceso de depuración y especificación. Por otra parte al solicitar informacion desde un cierto orden también colabora en el ordenamiento de la información elicitada.*

### **5. A qué tipo de usuario estará dirigido: Usuario común, ingeniero de software,...**

*Creo que podemos considerar un ingeniero en software con poca experiencia en el tema y que el SE lo asista en el proceso de especificación.*

## **Análisis de la sesión**

### **Conceptos educidos**

REQUERIMIENTOS

REQUISITOS

ESPECIFICACION DE REQUERIMIENTOS

ANALISIS DE REQUERIMIENTOS

### **Conocimientos obtenidos**

REQUERIMIENTOS

Es la especificación de la información que se necesita para realizar determinadas tareas o funciones.

REQUISITOS

Es una condición y/o especificación técnica u operativa respecto de los requerimientos.

## ESPECIFICACION DE REQUERIMIENTOS

Identificación detallada de la información que se necesita para realizar determinadas tareas o funciones

## ANALISIS DE REQUERIMIENTOS

El estudio de los requerimientos especificados con el fin de sacar conclusiones, Pueden ser: completitud, consistencia, etc.

### **Evaluación de la sesión**

Se obtuvieron los resultados esperados. Se concluye que el SE debe asistir al Ingeniero de Software en la especificación de los requerimientos mediante el pedido de la información necesaria para tal fin, completando llegado el caso la elicitación realizada, sin considerar requerimientos anteriores o dominios específicos.

### **5.3.2 Sesión 2: Sesiones de profundización de conocimientos**

#### **Preparación de la sesión**

- Información a tratar: Identificación de las características que participan en el estudio de viabilidad del proyecto.
- Amplitud y profundidad: Lograr esclarecer la viabilidad que presenta el proyecto al ser abordado con la Ingeniería del Conocimiento, tratando de evaluar diferentes características que son objeto de estudio, en el test de viabilidad propuesto por la metodología IDEAL.
- Técnica adecuada: Entrevista estructurada.
- Preparación de preguntas:
  1. De los casos en los que has trabajado, ¿Se mantiene un registro que permita obtener estudios históricos?. En caso afirmativo, ¿Existe la posibilidad de utilizar dichos casos para evaluar el sistema?
  2. Cuándo se lleva a cabo el análisis del problema, al tomar decisiones, ¿Se utiliza sentido común?
  3. ¿Cómo se calificaría la posibilidad de utilizar un sistema experto que imitara tu forma de razonar ante la tarea de realizar el documento de requerimientos?
  4. Para realizar el análisis y documentación de requerimientos, ¿Se requiere de un volumen de conocimientos elevado?
  5. Los conocimientos necesarios están basados en conocimientos científicos públicos, o además, se requiere de cierta experiencia

propia?.

6. ¿El análisis del problema y enmarcado es sumamente sistemático (algorítmico) o se utiliza algún tipo de atajo para resolver el problema (heurísticas) propio de la experiencia?.
7. Para analizar el dominio y documentar requerimientos ¿Se puede descomponer la tarea en varias subtareas?
8. ¿Se suele llevar adelante el trabajo con información incompleta, es decir, sin contar con toda la información que se desearía?.
9. ¿Es conveniente justificar las soluciones adoptadas?
10. ¿El tiempo esperado para dar una respuesta es limitado?
11. ¿El sistema debe buscar la solución óptima?
12. Una vez que esté construido el sistema, los usuarios tendrán que tener cierta experiencia en el campo de la ingeniería de requerimientos?

## Sesión 2

### Sesión 2

**Fecha: 3/04/00**

**Expertos: Lic. Bibiana Rossi, Lic. Edgardo Claverie**

**Lugar: ITBA - CAPIS**

**IC: Ing. Marcelo Rizzi**

**Técnica empleada: Cuestionario**

**Objetivos: Identificación de la Viabilidad**

1. De los casos en los que has trabajado, ¿Se mantiene un registro que permita obtener estudios históricos?. En caso afirmativo, ¿Existe la posibilidad de utilizar dichos casos para evaluar el sistema?

*Existe la documentación de cada proyecto en los que hemos participado, que bien puede servir para casos de prueba tanto de aquellos proyectos exitosos como de los que fracasaron ya sea por exceso de costo o tiempo, o bien por software defectuoso.*

2. Cuándo se lleva a cabo el análisis del problema, al tomar decisiones, ¿Se utiliza sentido común?

*El proceso de ingeniería de requerimientos, está plenamente basado en*

*conocimientos de técnicas de entrevistas, elicitación, diagramas, etc, en la experiencia que se posee sobre los casos resueltos, innumerables publicaciones que hacen que su proceso carezca de sentido común, basándose pura y exclusivamente, en la experiencia propia.*

3. ¿Cómo se calificaría la posibilidad de utilizar un sistema experto que imitara tu forma de razonar ante la tarea de realizar el documento de requerimientos?

*Es muy importante para aquellos profesionales recién iniciados o con poca experiencia en el campo, dado que les permite sistematizar esta ardua tarea, lo que mejora las chances de que el proyecto llegue a feliz término en costo y tiempo, además de la consiguiente mejora de la calidad. Además el hecho de que pueda convertirse en un repositorio de experiencia ayuda a los mismos expertos a organizar y mejorar su tarea.*

4. Para realizar el análisis y documentación de requerimientos, ¿Se requiere de un volumen de conocimientos elevado?

*Se necesita de una gran versatilidad para encarar el estudio de dominios en los cuales se posee mínima o nula experiencia. Además se necesita manejar unas cuantas técnicas de elicitación.*

5. Los conocimientos necesarios están basados en conocimientos científicos públicos, o además, se requiere de cierta experiencia propia?.

*La experiencia propia es fundamental a la hora de analizar los dominios de aplicación dado que cada problema es un mundo nuevo y tiene sus particularidades que lo diferencian de los demás. Sin embargo se puede trabajar mucho con analogías a problemas ya resueltos anteriormente.*

6. ¿El análisis del problema y enmarcado es sumamente sistemático (algorítmico) o se utiliza algún tipo de atajo para resolver el problema (heurísticas) propio de la experiencia?.

*La descomposición de problemas en subproblemas requiere de heurísticas para llevarla a cabo, cuando se observa un problema no existe un proceso algorítmico que permita detectar subproblemas o realizar el enmarcado en distintos marcos de problema, o para identificar subdominios.*

7. Para analizar el dominio y documentar requerimientos ¿Se puede descomponer la tarea en varias subtareas?

*Se debe realizar primero el análisis del problema para tratar de identificar el marco*



*de problema correspondiente o los marcos correspondientes si es un problema multimarco, luego identificar correctamente las partes del marco para proceder con la descripción de cada una de ellas. Finalmente se debe completar el documento con el listado de requerimientos describiendo correctamente cada uno de ellos.*

8. ¿Se suele llevar adelante el trabajo con información incompleta, es decir, sin contar con toda la información que se desearía?.

*Puede suceder que la elicitación no esté completamente desarrollada, o bien el tipo de ciclo de vida elegido sea prototipado en el que los requerimientos son cambiantes o van evolucionando, por lo tanto en el momento de tomar decisiones puede que no esté toda la información disponible.*

9. ¿Es conveniente justificar las soluciones adoptadas?

*Claro que sí, dado que son documentos que se utilizarán para el todo el proceso de desarrollo, además Marketing y comercial basarán sus tareas y estrategias en función de estos documentos y fundamentalmente sirve como base contractual con el cliente.*

10. ¿El tiempo esperado para dar una respuesta es limitado?

*Si bien no se necesita un tiempo de respuesta muy ajustado, es importante que el usuario pueda realizar la tarea en forma dinámica; el sistema debe cooperar con esa dinámica.*

11. ¿El sistema debe buscar la solución óptima?

*El sistema no debe buscar la solución óptima al problema, sino encontrar la recomendación que el experto hubiese realizado si se le presentase un caso con características similares.*

12:Una vez que esté construido el sistema, los usuarios tendrán que tener cierta experiencia en el campo de la ingeniería de requerimientos?

*Se busca que los usuarios noveles puedan utilizar el sistema y disponer de la experiencia que contenga el mismo.*

## **Análisis de la Sesión 2**

Los conocimientos extraídos de la sesión 2 se encuentran reflejados en la justificación de las diferentes características que se evalúan en test de viabilidad. En la Justificación de las características del Test de Viabilidad, se describen los conocimientos obtenidos de la presente sesión de educación.

### **5.3.3 Sesiones 3,4, 5, 6: Profundización de conocimientos (Continúa)**

Las siguientes sesiones tienen como objetivo comprender en el dominio de la aplicación y obtener conocimientos concretos del mismo.

#### **Preparación de las sesiones 3, 4, 5 y 6**

- Información a tratar: Identificación de los diferentes elementos que conforman el dominio de la aplicación.
- Amplitud y profundidad: En la sesión se tratarán conocimientos amplios descendiendo a un nivel de profundidad adecuado, mediante el cual se procura la búsqueda de mayor conocimiento.
- Técnica adecuada: Entrevista estructurada.

#### **Sesión 3**

**Fecha: 05/04/00**

**Experto: Lic. Bibiana Rossi**

**Lugar: ITBA - CAPIS**

**IC: Ing. Marcelo Rizzi**

**Técnica empleada: Cuestionario**

**Objetivos: La presente sesión se concentrará en el entorno del proceso de especificación de requerimientos, definición de términos generales, metodología global de trabajo, es decir los lineamientos que enmarcan la actividad mencionada.**

**Se pretende ubicar en rasgos generales la actividad que el experto realiza, familiarizarse con los términos y actividades más comunes.**

**Se utiliza la técnica de cuestionario dado la distancia física entre el experto y el IC.**

#### **Transcripción de la sesión**

**1.** Podrías definir qué son los requerimientos?

*Son las necesidades de información, para desarrollar determinadas tareas (operativa, de decisión) o funciones.*

*En varias ocasiones se utiliza el término requisitos como sinónimo de requerimientos, me gustaría aclarar que considero conveniente efectuar una distinción. Requisitos se refiere a un condición de tipo técnica u operativa respecto de los requerimientos.*

*Hay que considerar que los requerimientos pueden no ser respecto de información a obtener, sino respecto de la funcionalidad o de la performance con que se desea esa información. En el hipotético caso que la información que se requiera sea la misma que ya se obtiene pero que se desee con mayor rapidez y por algún medio tecnológico distinto a los actuales, en ese caso desde mi punto de vista los requerimientos deberían igualmente especificarse a partir de cual es la actual información. No creo conveniente obviar este paso (si bien es evidente que puede resultar mas sencillo y rápido) y definir requisitos porque es factible que se requiera algún ajuste en la información. Por otra parte es obviamente la forma de dejar documentado de que información a obtener se hace referencia. Y si se hace abstracción generalizada de esta situación puede pensarse que en muchos casos la motivación inicial es un cambio tecnológico que suele llevar tambien implícitas mejoras respecto del contenido y calidad de la información actualmente usada.*

**2.** De igual modo podrías definir que es una especificación de requerimientos?

*Identificación detallada y documentada de la información necesaria para desarrollar determinadas tareas o funciones.*

**3.** De qué tipo de información, material, etc. necesitas disponer al comenzar el proceso de especificación de requerimientos?

*Obviamente el objetivo del sistema, su alcance y limites, lo mas claro posible porque justamente es la especificación de requerimientos la que agrega precisión respecto del objetivo, los limites y alcances. Aun así debe existir previamente un objetivo definido con el mayor nivel de precisión posible.*

*La documentación de las entrevistas realizadas a los usuarios, si las hubiera.*

*La documentación de las encuestas o cuestionarios realizados por los usuarios.*

*Un bosquejo de los listados, formularios o documentos que los usuarios desean recibir y con que periodicidad. Quienes son los destinatarios, y quienes originan la información.*

*Una copia de los actuales listados, formularios o documentos que se usan actualmente y con que periodicidad, en lo posible completos con la información correspondiente. Con que periodicidad, destinatarios y orígenes de la información.*

**4.** Una vez que has identificado el dominio de la aplicación y el problema a resolver, utilizas ciertos modelos, patrones o marcos, fruto de tu experiencia, en el proceso de identificar y especificar los requerimientos para la aplicación? Por ejemplo, aplicas analogías de sistemas similares resueltos anteriormente, pero atendiendo a las particularidades del actual?

*Si, si dispusiera de requerimientos definidos para otros sistemas que pudiera consultar seria de gran ayuda para completar o al menos consultar a los usuarios sobre mayor grado de precisión en los requerimientos ya definidos o sobre requerimientos a sugerir. Si no dispongo de estos requerimientos recorro a mi experiencia, memoria.*

*En ultimo termino trato de ponerme en el lugar del usuario (depende de cuanto entiendo y /o conozco el dominio) y analizar que información requeriría yo para hacer esas tareas. Pero esto como ultimo recurso ya que si tuviera que usarlo formalmente requeriría de un relevamiento detallado respecto de como se llevan a cabo las tareas o funciones para poder deducir que información.*

*Es conveniente considerar que en todos los casos el que conoce profundamente el dominio es el usuario, y no siempre es posible o conveniente considerar la propia experiencia.*

- 5.** Que características consideras como de vital importancia que debe tener todo requerimiento, como por ejemplo que sea necesario, o verificable o factible, etc.? en otras palabras, describe algunas características que debe tener un requerimiento para que lo consideres como correcto o muy bueno.

*Precisión respecto de los datos necesarios y de como deben estar combinados o relacionados.*

- 6.** En los proyectos que has realizado, cuales son los problemas mas comunes que has enfrentado en la especificación de requerimientos? (Por ejemplo asumir algo incorrectamente, sobre especificar, describir operaciones en vez de escribir requerimientos, etc.)

*En este sentido tengo una anécdota que suele ser altamente representativa de la situación habitual que me encuentro. Por lo que vengo escribiendo hasta ahora se deduce que desde mi punto de vista el analista lo que hace es ordenar, clarificar, agrupar, precisar los requerimientos que deben haber definido los usuarios. Y ahí esta justamente el problema los usuarios no suelen definir con claridad y precisión (en algunos casos en porcentaje muy bajos) la información que necesitan y mucho menos aun de que forma la quieren relacionada. Paso a contar la anécdota: trabaje durante varios años en la Universidad Tecnológica Nacional, en la Regional Buenos Aires, en el Departamento de Sistemas. El director del Departamento es obviamente un especialista en sistemas, a cargo de la cátedra de análisis de sistemas, y ha orientado la carrera hacia un fuerte perfil de análisis y diseño. Es decir, es un usuario altamente entendido y calificado respecto de la tarea de especificar requerimientos ya que obviamente es parte de lo que enseña como*

*punto de partida para cualquier actividad de análisis y diseño. Tenía además una habilidad que ponía en práctica frecuentemente en sus habituales tareas como director. En cuanto comenzábamos a trabajar juntos tomaba una hoja en blanco, un lápiz negro y bosquejaba un diseño de formulario sobre el cual iba ordenando la información que necesitaba para atender el tema de ese momento. Era realmente interesante como ese bosquejo ordenaba el trabajo, lo clarificaba y quien tenía que trabajar con él podía operativizar rápidamente las ideas.*

*Podría suponerse entonces que si estuviera en lugar de usuario facilitaría la tarea del entrevistador solicitándole los datos necesarios y que relación debían tener. En una ocasión un grupo estaba desarrollando un sistema fundamental para las tareas del departamento, pues fue casi imposible poder obtener la información necesaria. Para saber que información necesitaba (listados básicos) fue necesario recurrir a los bosquejos que guardábamos quienes trabajamos con él, y fue necesario organizar sesiones para observar como hacía la tarea, para poder rescatar cual era la información que usaba y como.*

*A mi modo de ver este es realmente el peor de los problemas, pero por ahora no le veo alternativa de mejora, salvo que se considere alguna forma de ayuda para que un usuario defina los requerimientos desde su punto de vista y luego el analista los defina profesionalmente.*

*En segundo lugar el problema que ocurre con más frecuencia es la incompletitud de los requerimientos. Es decir requerimientos definidos parcialmente, o no definidos. Es decir en Ingeniería del Software se proponen dos modelos iniciales (que se depuran a lo largo del análisis y del relevamiento) como son el DER y DFD (al menos el diagrama de contexto) que si bien se amplían, tal como mencioné, también es necesario poder hacer un primer bosquejo para tener la posibilidad de estimar y presupuestar el proyecto. Pues bien todo esto debe poder construirse a partir de los requerimientos. El problema es que usualmente falta información para poder hacerlo, porque los requerimientos son incompletos.*

**7.** Utilizas alguna forma de agrupación de los requerimientos, según una determinada característica u otro parámetro?

*En la mayoría de los casos me resulta fácil y claro agruparlos por subsistemas. Otra alternativa sería por tema, otra puede ser por tipo de soporte en el que se pide. La última me parece la menos adecuada en los términos de requerimientos de los que hablamos. Otra opción puede ser por usuario, esta variante si no se usa de todas formas sería conveniente identificar el o los usuarios que hayan solicitado el requerimiento..*

**8.** Asignas prioridades a los requerimientos como por ejemplo esencial, deseable, etc.?

*Siempre tengo en cuenta el tipo de prioridad que haya definido de alguna forma el usuario.*

*Si puede ser esencial, deseable, posible, simple, complejo, condicionado a., opcionales, si es fijado internamente (por la propia organización) o externamente (casa matriz, otra organización como algún ente gubernamental) si requiere o no hardware, software o algún otro recurso específico, según la prioridad que se establezca para el desarrollo e implementación de los subsistemas.*

*En realidad trato de analizar si en el discurso del usuario existe alguna clasificación que se pueda utilizar al asignar las prioridades.*

**9.** Cuales son los criterios para evaluar dicha prioridad?

*El criterio del usuario en primer lugar, o sea lo que el haya asignado.*

*En el resto analizo el tiempo que implican para su desarrollo, eventualmente en un primer momento en forma estimativa genérica el hardware o software disponible, si existe alguna limitación inicial de tiempo, o recursos de algún tipo.*

**10.** Al momento de escribir la especificación utilizas algún template para describir y caracterizar cada uno de los requerimientos? (ID, categoría, ID de otro requerimiento dependiente o relacionado, riesgos, etc.)

*Podrías aclararme ésta pregunta?*

## **Análisis de la sesión**

Conceptos obtenidos

*Tipos de requerimientos:* Los requerimientos pueden ser de información a obtener como también de funcionalidades y performance. Para un sistema nuevo o para actualizaciones de sistemas existentes.

*Información de entrada:* Los materiales que se tienen al comienzo pueden ser listados, reportes, documentación de entrevistas a usuarios, encuestas, cuestionarios a usuarios, documentos de elicitación.

En situaciones no es suficiente la documentación y es necesario la observación de tareas.

*Clasificación de requerimientos:* Hay varias maneras de agrupar requerimientos: Por subsistemas, por tema, por usuario.

Los requerimientos se pueden priorizar según diferentes criterios asignados por el usuario.

#### **Sesión 4**

**Fecha: 10/04/00**

**Experto: Lic. Bibiana Rossi**

**Lugar: ITBA - CAPIS**

**IC: Ing. Marcelo Rizzi**

**Técnica empleada: Cuestionario**

**Objetivos:** Avanzar en la obtención de conocimientos en la perspectiva del entorno de la tarea del analista de requerimientos y en el "cómo" de la tarea.

#### **Transcripción de la sesión**

1. Qué tipo de información o documentación generas como resultado de tu trabajo como analista de requerimientos? Cómo la denominas?

*Informe de Requerimientos. Como parte del Informe puede ir, una breve descripción del negocio, un organigrama, breve descripción de las funciones de las áreas del organigrama, objetivo del trabajo, diagrama de contexto, diagrama de nivel 1 de DFD (si corresponde). Enumeración de los requerimientos. Muchos de estos items pueden ser opcionales, y pueden ser acotados según el escenario de trabajo. Por ejemplo si estoy haciendo un sistema de gestión para el Ministerio probablemente no ira todo el organigrama del ministerio sino de aquellas áreas o sectores vinculados al sistema en cuestión y su entorno.*

2. Como información de entrada, Tomas datos suministrados o elicitados de los usuarios (Documentación de entrevistas, listados, formularios, etc.).

*Asi es exactamente, de la documentacion de las entrevistas, de los formularios existentes, de la observacion realizada, y aunque no debiera ser tan asi de mis experiencias.*

3. ¿Consideras las restricciones y/o limitaciones que pudiere imponer el propio dominio de la aplicación por sus propias características?

*Es tan genérico lo que preguntas que no se cual es el sentido. ¿Podes aclararme mejor esta pregunta?*

4. En la sesión 1, pregunta 5, " Que características consideras como de vital importancia que debe tener todo requerimiento, como por ejemplo que sea necesario, o verificable o factible, etc.? en otras palabras, describe algunas características que debe tener un requerimiento para que lo consideres como correcto o muy bueno".

*Tu respuesta fue: "Precisión respecto de los datos necesarios y de como deben estar combinados o relacionados. "*

Podrías aclararme qué significa que haya precisión en la combinación o relación de datos? ¿Cómo se dan dichas combinaciones y/o relaciones?

*Por ejemplo si el usuario solicita "Informacion sobre la cantidad de alumnos inscriptos en el Magister", que sea preciso significa que aclare si necesita o no considerar como inscriptos los que estan de licencia. Si desea una estadistica del tiempo promedio que se demoran los alumnos entre entrega de controles que especifique como se contabiliza en esa estadistica los tiempos de licencia, como se contabiliza el tiempo inicial hasta la entrega del primer control, como se contabiliza los tiempos para los finales, si desea algun tipo de totales por profesor o por epoca del año, en definitiva lo que especifica es como se combinan o relacionan los datos. Este ejemplo es trivial pero mi experiencia es que justamente este es el tipo de información que no suele precisarse y como profesional de sistemas mi tarea no debiera ser "gerenciar" sino organizar los datos.*

5. En la sesión 1, pregunta 6, identificas como problema la incompletitud de los requerimientos, que dificulta la realización de un primer bosquejo. ¿Cómo procedes para tratar de minimizar la imcompletitud de requerimientos? ¿Utilizas alguna sistematología para lograrlo?

*En general trato de ponerme en el lugar del usuario y analizar que requeriria yo en ese rol, obviamente debería luego verificarlo con el usuario. Esto considero que no es conveniente ya que demora desde el inicio el trabajo de análisis del sistema.*

6. Para lograr precisión, utilizas (o deseas que el SE te brinde) algún tipo de formalismo, como por ejemplo algún conjunto finito de símbolos que ayude a disminuir la ambigüedad en la especificación hecha en lenguaje natural?

*Si es una idea excelente, considerar abreviaturas y símbolos y si el sistema experto sobre esos simbolos tuviera un glosario que se pueda imprimir para incorporarlo al informe de especificación de requerimientos sería de ayuda.*

7. Una vez que has reunido toda la información de entrada, considerada suficiente como para comenzar a realizar la especificación, ¿podrías indicarme, en líneas



generales, cómo es el procedimiento que llevas a cabo para realizar la especificación?

*Comienzo a armar el informe completando todo lo que puedo y haciendo sobre el informe sucesivos refinamientos. Me sirve para ordenar las ideas y para detectar lo que pueda estar faltando, me ayuda a comprender el dominio. En general la estructura del informe, dado que tiene una secuencialidad de lectura ayuda a ordenar la información y al tener que explicar a un lector, en general se detecta la información faltante propia. En este sentido si el SE ofrece una estructura base (puede ser mas de una) puede ser tambien de interesante ayuda.*

8. ¿Dicho procedimiento es absolutamente secuencial, realizas un refinamiento iterativo, o bien qué naturaleza te parece que tiene?

*Por lo que contesté tiene ambas, hay momentos de secuencialidad hasta el punto en el que se puede llegar a completar, luego sigo con otro punto y finalmente existe un refinamiento, es posible también que al avanzar sobre algun aspecto del informe provoque alguna modificación o incompletitud en lo que ya estaba definido.*

9. La pregunta 10 de la sesión anterior, apuntaba a determinar si utilizas algún estándar como guía para escribir el documento de especificación, como por ejemplo el ANSI/IEEE 830 de recomendación para especificar requisitos de software.

*No tengo un estandar salvo que el usuario o la organización lo requieran, estuve leyendo el estandar del IEEE y me parecio bastante bueno, si bien es cierto que se especifican requisitos y requerimientos. Quizas una versión de SE que me permita definir un estandar a partir de sugerencias o de propuestas propias tambien seria de ayuda.*

### **Análisis de la sesión**

Conceptos obtenidos

*Documentación de salida:* descripción del negocio, organigrama, funciones del área, diagramas de contexto y nivel 1, enumeración de requerimientos.

*Características de requerimientos:* Precisión, especificación de combinación y relación entre los datos.

*Procedimiento:* La especificación se realiza en forma iterativa, alrededor de una estructura base de organización de la documentación de salida que le dá una secuencialidad.

## Sesión 5

Información a Identificar: La presente sesión se concentrará en el entorno del proceso de especificación de requerimientos, definición de términos generales, metodología global de trabajo, es decir los lineamientos que enmarcan la actividad mencionada.

Se pretende ubicar en rasgos generales la actividad que el experto realiza, familiarizarse con los términos y actividades más comunes.

### Sesión 5

**Fecha:** 12/04/00

**Experto:** Lic. Edgardo Raúl Claverie

**Lugar:** Capis

**IC:**

**Técnica empleada:** Cuestionario

### Transcripción de la sesión 5

**11.** Podrías definir qué son los requerimientos?

*Los requerimientos del software (también denominados requisitos) sintetizan el comportamiento externo del sistema, es decir QUE debe hacer sin precisar COMO conseguirlo.*

*Los requerimientos pueden ser vistos desde dos puntos de vista. Del usuario (son las condiciones necesarias para que el usuario pueda satisfacer su necesidad de información) y del Desarrollador (son las condiciones que debe cumplir el sistema para satisfacer el compromiso formal asumido)*

**12.** De igual modo podrías definir que es una especificación de requerimientos?

*La especificación de Requerimientos es una descripción completa de lo que realizará el software sin describir como.*

*Esta tarea es previa al desarrollo del software y se la sitúa en la fase mas temprana del ciclo de vida. A esta fase se la denomina de "Requisitos software"*

**13.** De qué tipo de información, material, etc. necesitas disponer al comenzar el proceso de especificación de requerimientos?

*La información la dispongo de distintas fuentes:*

*Por un lado necesito contar con todos aquellos manuales de procedimientos o normas referidas al sistema bajo estudio con el debido cuidado de estar atento a su nivel de actualización.*

*Además de ello toda aquella documentación vinculada al sistema que pueda servir de base para comprender acabadamente las necesidades del usuario (documentos y formularios utilizados hasta ese momento).*

*Por otra parte es necesario realizar entrevistas y cuestionarios a fin de recabar la información mencionada anteriormente.*

- 14.**Una vez que has identificado el dominio de la aplicación y el problema a resolver, utilizas ciertos modelos, patrones o marcos, fruto de tu experiencia, en el proceso de identificar y especificar los requerimientos para la aplicación? Por ejemplo, aplicas analogías de sistemas similares resueltos anteriormente, pero atendiendo a las particularidades del actual?

*La respuesta es afirmativa pero teniendo muchísimo cuidado ya que las experiencias anteriores nos facilitan la comprensión del problema pero por otra parte nos quitan objetividad.*

- 15.**Que características consideras como de vital importancia que debe tener todo requerimiento, como por ejemplo que sea necesario, o verificable o factible, etc.? en otras palabras, describe algunas características que debe tener un requerimiento para que lo consideres como correcto o muy bueno.

*Las características que debería reunir todo requerimiento son las siguientes:*

*Correcto*

*No ambiguo*

*Completo*

*Verificable*

*Consistente*

*Comprensible para el usuario*

*Modificable*

*Trazable*

*Independiente del diseño*

*Conciso*

- 16.**En los proyectos que has realizado, cuales son los problemas mas comunes que has enfrentado en la especificación de requerimientos? (Por ejemplo asumir

algo incorrectamente, sobreespecificar, describir operaciones en vez de escribir requerimientos, etc.)

*Los problemas mas comunes son:*

*Ambigüedad (diferentes enfoques de los usuarios acerca de un mismo requerimiento).*

*Incorrectos (ya sea por el desconocimiento del usuario sobre su necesidad o la falta de comprensión del analista al relevarlo). También hay que agregar a esto los cambios en el entorno que se producen durante el desarrollo del proyecto.*

*Imposibilidad de verificarlos (hay situaciones que solamente puede ser verificadas una vez concluido).*

*Independencia del diseño (en ocasiones dependen demasiado sobre todo cuando se desarrolla bajo una tecnología ya disponible en la empresa).*

*Falta de organización (muchas veces se arma un sistema sumando requerimientos particulares de cada usuario sin un criterio sistémico).*

**17.**Utilizas alguna forma de agrupación de los requerimientos, según una determinada característica u otro parámetro?

*Sí. Podemos dividirlos en requisitos funcionales, de rendimiento, de interfaz (con el usuario y con otros sistemas) y específicos (por ejemplo seguridad).*

**18.**Asignas prioridades a los requerimientos como por ejemplo esencial, deseable, etc.?

*Sí. Asigno prioridades en razón de Requerimientos obligatorios, opcionales y deseables.*

**19.**Cuales son los criterios para evaluar dicha prioridad?

*El criterio es la posibilidad de prescindir de ellos. En algunos casos será imposible lo cual determinará que sean obligatorios (por ejemplo la emisión de facturas en un sistema de facturación). En otros , se podría prescindir sin llegar a generar demasiados problemas (opcionales). (para el ejemplo anterior la realización de algunas estadísticas específicas). Y finalmente los deseables, que sin ser imprescindibles mejoran las prestaciones del sistema.*

**20.**Al momento de escribir la especificación utilizas algún template para describir y caracterizar cada uno de los requerimientos? (ID, categoría, ID de otro requerimiento dependiente o relacionado, riesgos, etc.)

*Generalmente empleo técnicas tales como Diagramas de Entidad Relación, Diagramas de Flujo de Datos, Diagramas de Transición de Estados, Redes de Petri, Arboles y Tablas de Decisión, etc.*

### **Análisis de la sesión 5**

Se obtuvo en esta sesión una perspectiva para establecer la prioridad de cada uno de los requisitos. El experto brindó una taxonomía de requerimientos como así también un criterio objetivo para clasificarlos según su prioridad.

### **Sesión 6**

**Fecha:** 15/04/00

**Experto:** *Lic. Edgardo Raúl Claverie*

**Lugar:** *C A P I S*

**IC:** *Ing. Marcelo Rizzi*

**Técnica empleada:** *Cuestionario*

**Objetivos:** Avanzar en la obtención de conocimientos en la perspectiva del entorno de la tarea del analista de requerimientos y en el "cómo" de la tarea.

### **Transcripción de la sesión**

1. Qué tipo de información o documentación generas como resultado de tu trabajo como analista de requerimientos? Cómo la denominas?

*Genero un documento denominado ERS (Especificación de Requerimientos del Sistema) que contiene una descripción completa de lo que hará el software sin describir cómo.*

2. Como información de entrada, Tomas datos suministrados o elicitados de los usuarios (Documentación de entrevistas, listados, formularios, etc.).

*Sí, por supuesto. Es muy importante cruzar esta información con la de otros usuarios (cuando ello sea posible) y además confrontarlo con la observación personal.*

3. ¿Consideras las restricciones y/o limitaciones que pudiere imponer el propio dominio de la aplicación por sus propias características?

*Sí, teniendo mucho cuidado de que esto no quite objetividad al análisis. También considero aquellos casos de excepción que inciden sobre las restricciones mencionadas.*

4. Describe algunas características que debe tener un requerimiento para que lo consideres como correcto o muy bueno".

*Las características que debería reunir todo requerimiento son las siguientes:*

*Correcto*

*No ambiguo*

*Completo*

*Verificable*

*Consistente*

*Comprensible para el usuario*

*Modificable*

*Trazable*

*Independiente del diseño*

*Conciso*

5. ¿Cómo procedes para tratar de minimizar la incompletitud de requerimientos?  
¿Utilizas alguna sistemología para lograrlo?

*La completud es uno de los atributos en el que resulta mas dificil asegurar su corrección. Una buena metodología es la de utilizar diferentes herramientas tales como Diagramas de Entidad-Relación, Diagramas de Flujo de Datos o Tablas de Decisión.*

6. Para lograr precisión, utilizas algún tipo de formalismo, como por ejemplo algún conjunto finito de símbolos que ayude a disminuir la ambigüedad en la especificación hecha en lenguaje natural?

*Trato de emplear, en la medida de lo posible, cualquier tipo de formalización lógica o matemática.*

7. Una vez que has reunido toda la información de entrada, considerada suficiente como para comenzar a realizar la especificación, ¿podrías indicarme, en líneas generales, cómo es el procedimiento que llevas a cabo para realizar la especificación?

*Una vez reunida la información requerida la elaboro produciendo un documento que debe contener como mínimo la información que detallo a continuación:*

*Introducción:*

*Propósito*

*Alcance*

*Definiciones, sinónimos, etc.*

*Referencias*

*Descripción General*

*Perspectiva del producto*

*Funciones del producto*

*Características del usuario*

*Restricciones*

*Dependencias*

*Otros requisitos*

*Requisitos específicos*

*Interfaces externas*

*Funciones*

*Requisitos de rendimiento*

*Restricciones de bases de datos, diseño y de cumplimiento con estándares*

*Atributos del sistema software*

8. ¿Dicho procedimiento es absolutamente secuencial, realizas un refinamiento iterativo, o bien qué naturaleza te parece que tiene?

*Lo inicio secuencialmente pero a posteriori realizo un refinamiento iterativo a fin de ir ajustando los detalles.*

9. ¿Utilizas algún estándar como guía para escribir el documento de especificación, como por ejemplo el ANSI/IEEE 830 de recomendación para especificar requisitos de software.

*En general utilizo los estándares propios de las organizaciones en las cuales realizo la tarea. En aquellos casos en que no se dispone de un estándar propio me baso en el mencionado ANSI/IEEE 830.*

## **Análisis de la sesión 6**

Se obtuvo principalmente una plantilla para la documentación formal de los requerimientos. El experto además expuso que la característica de completud es la que más frecuentemente le causa problemas y explicó una manera de lograrla.

### **5.3.4: Sesión 7: Emparrillado**

Esta sesión de emparrillado se realiza luego de las sesiones de extracción de conocimientos a partir de la documentación existente. Las sesiones de extracción de conocimientos se transcriben en el apartado 5.3 del presente trabajo.

A partir de dichas sesiones se obtuvo información sobre los diferentes tipos de dominio que forman parte de los marcos de problema. Se recurre a los expertos mediante un emparrillado para tratar de clasificar los diferentes tipos de dominios con los que el experto trabaja al realizar el análisis del problema.

<b>Sesión número:</b> 7
<b>Fecha:</b> 17/04/00
<b>Experto:</b> Ing. Diego Lianres
<b>IC:</b> Ing. Marcelo Rizzi
<b>Técnica Empleada:</b> Emparrillado
<b>Objetivos:</b> Obtener una visión más profunda de la caracterización de dominios, clave en el posterior desarrollo de la especificación de requerimientos.

#### **Paso 1: Identificación de los elementos**

Se desea profundizar en cómo el experto caracteriza los dominios para poder luego proceder a su análisis para el documento de requerimientos. Se elegirán entonces diferentes dominios, sobre los que una aplicación software podría trabajar.

E1: Red de Telefonía (Como dominio de una aplicación rodando en centrales telefónicas)

E2: Texto en un sistema procesador de Textos (Como dominio de un procesador de textos)

E3: Máquina expendedora de bebidas (Como dominio de una aplicación de control)

E4: Un archivo de registros en una cinta magnética (Como dominio de parte de un sistema operativo)

E5: La Atmósfera (Como dominio de una aplicación de meteorología)

#### **Paso 2: Identificación de características**



Se eligen a continuación diferentes características atribuibles a los dominios mencionados.

C1: unidimensional

C2: Estático

C3: Activo

C4: Tangible

### Paso 3: Diseño de la parrilla

**Matriz Dicotómica**

	C1	C2	C3	C4
E1	-	+	-	+
E2	+	+	-	-
E3	-	-	-	+
E4	+	+	-	-
E5	-	-	+	+

Luego se evalúa la matriz:

**Matriz Evaluada**

		E1	E2	E3	E4	E5	
C1	Unidimensional	5	1	3	5	5	Multidimensional
C2	Estático	5	3	5	4	5	Dinámico
C3	Activo	2	3	5	3	4	Reactivo
C4	Intangible	5	1	5	2	5	Tangible

### Paso 4: Formalización

Clasificación de los elementos mediante distancia mínima

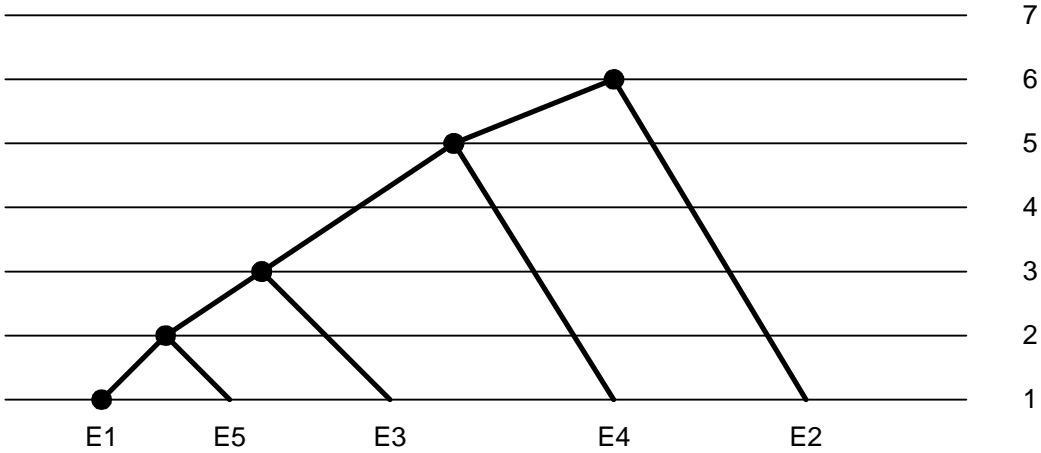
	E1	E2	E3	E4	E5
E1		11	5	5	<b>2</b>
E2			10	6	11
E3				8	3
E4					5
E5					

	[E1,E5]	E2	E3	E4
[E1,E5]		11	<b>3</b>	5
E2			10	6
E3				8
E4				

	[[E1,E5],E3]	E2	E4
[[E1,E5],E3]		10	<b>5</b>
E2			6
E4			

	[[[E1,E5],E3],E4]	E2
[[[E1,E5],E3],E4]		<b>6</b>
E2		

ARBOL ORDENADO DE ELEMENTOS



## Evaluación del Arbol de Elementos

Tanto el elemento E1 (Sistema de Telefonía) como el elemento E5 (La Atmósfera) son dominios con similares características en cuanto a que son tangibles y dinámicos. Además son activos, los fenómenos ocurren espontáneamente y en forma autónoma. Por otra parte ambos son multidimensionales. Esto explica su cercanía en el árbol. El E3 (Máquina expendedora de bebidas) también es tangible y dinámico, pero en vez de ser activo es reactivo. Sus eventos son en respuesta a estímulos provocados por los usuarios. Es razonable que esté separado de E1 y E5 pero en la vecindad.

En cambio E4 (Registros en una cinta) y E3 (Texto en un procesador de textos) son dominios diferentes a los explicados.

E4 es intangible y unidimensional. Es a la vez dinámico ya que sufre alteraciones por parte del sistema que lo mantiene. E3 es también intangible, unidimensional y dinámico, pero no es activo ya que no existen eventos espontáneos y autónomos, y tampoco es reactivo ya que no responde como dominio a eventos provocados por los usuarios. Aquí hay un hueco en la caracterización de dominios que es necesario cubrir.

Clasificación de Características mediante distancia mínima

Matriz Evaluada

		E1	E2	E3	E4	E5	
<b>C1</b>	<b>Unidimensional</b>	5	1	3	5	5	<b>Multidimensional</b>
<b>C2</b>	<b>Estático</b>	5	3	5	4	5	<b>Dinámico</b>
<b>C3</b>	<b>Activo</b>	2	3	5	3	4	<b>Reactivo</b>
<b>C4</b>	<b>Intangible</b>	5	1	5	2	5	<b>Tangible</b>

Matrices de Valores Opuestos

C2	5	3	5	4	5	
C1	5	1	3	5	5	$(C2-C1) = 0+2+2+1+0 = 5$
C1'	1	5	3	1	1	$(C2-C1') = 4+2+2+3+4 = 15$
C3	2	3	5	3	4	
C1	5	1	3	5	5	$(C3-C1) = 3+2+2+2+1 = 10$
C1'	1	5	3	1	1	$(C3-C1') = 1+2+2+2+3 = 10$
C4	5	1	5	2	5	
C1	5	1	3	5	5	$(C4-C1) = 0+0+2+3+0 = 5$
C1'	1	5	3	1	1	$(C4-C1') = 4+4+2+1+4 = 15$

C3	2	3	5	3	4	
C2	5	3	5	4	5	$(C3-C2) = 3+0+0+1+1=5$
C2'	1	3	1	2	1	$(C3-C2') = 1+0+4+1+3 =9$

C4	5	1	5	2	5	
C2	5	3	5	4	5	$(C4-C2) = 0+2+0+2+0 =4$
C2'	1	3	1	2	1	$(C4-C2') = 4+2+4+0+4=14$

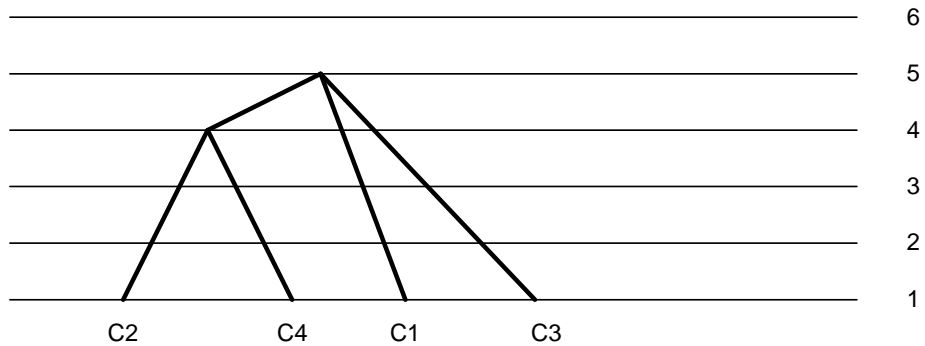
C4	5	1	5	2	5	
C3	2	3	5	3	4	$(C4-C3) = 3+2+0+1+1=7$
C3'	4	3	1	3	2	$(C4-C3') = 1+2+4+1+3=11$

	C1	C2	C3	C4
C1: Unidimensional		5	10	5
C2: Estático	15		5	4
C3: Activo	10	9		7
C4: Intangible	15	14	11	

	C1	C2	C3	C4
C1: Unidimensional		5	10	5
C2: Estático			5	<b>4</b>
C3: Activo				7
C4: Intangible				

	C1	[C2,C4]	C3
C1		<b>5</b>	10
[C2,C4]			<b>5</b>
C3			

ARBOL ORDENADO DE CARACTERÍSTICAS



### Evaluación del árbol de características

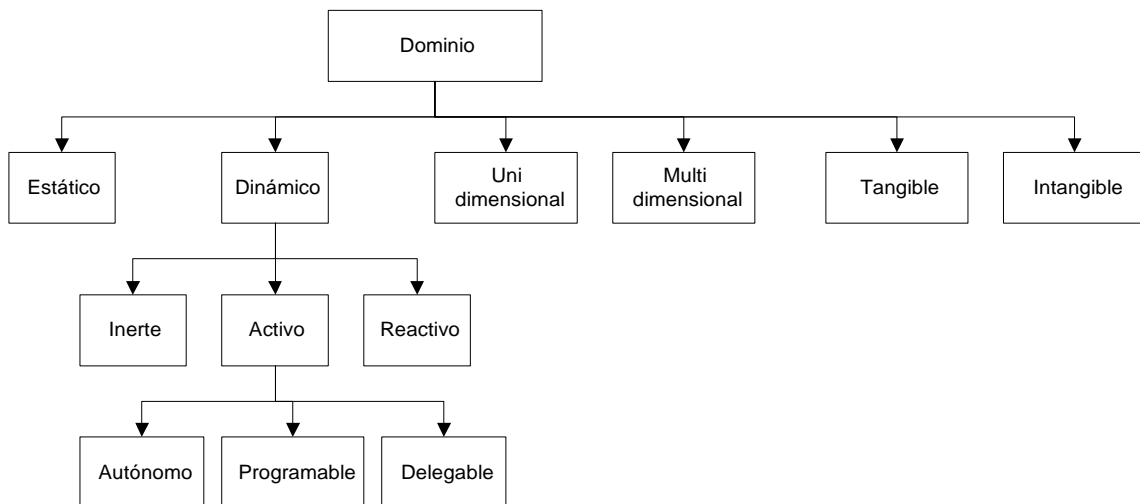
Las características C2 (dinámico) y C4 (Tangible) aparecen cercanas en un valor alto. Es frecuente que un dominio tangible sea también dinámico en el mundo real.

La característica C1 (Multidimensional) está separada de las anteriores siendo aparentemente independiente del carácter dinámico y tangible.

C3 (Reactivo) está al mismo nivel que Multidimensional, sin embargo debería estar cercana a dinámico, dado que el ser reactivo aparenta ser una característica dinámica.

Luego de un análisis sobre las fuentes de conocimientos se obtuvo el siguiente árbol ordenado de las características de los dominios.

### Taxonomía de Dominios según sus características



En el árbol puede observarse que se incluye una nueva característica de dominio dinámico llamada inerte. Un dominio dinámico es inerte cuando no provoca eventos por si mismo. Los cambios son producidos siempre por agentes externos.

Además se agrega una nueva caracterización para dominios dinámicos activos. Autónomo, Programable y delegable.

Autónomo, es el dominio activo clásico tratado en el emparrillado. Los eventos y cambios ocurren espontáneamente. Programable es cuando el dominio contiene una parte que se puede programar. ésta se convierte luego en parte de la máquina que se está construyendo.

Un dominio es Delegable cuando el usuario debe cumplir que se le ha asignado en el funcionamiento normal del sistema.

Los resultados de esta entrevista se ven reflejados en la conceptualización de los conocimientos tácticos para caracterizar los tipos de dominios que componen un marco de problema.

### **5.3.5 Sesión 8: Estándares**

#### **Preparación de la Sesión**

- Información a tratar: Estándares de documentación de requerimientos.
- Amplitud y profundidad: Se trata de esclarecer el estándar que utiliza el experto para documentar requerimientos.
- Técnica adecuada: entrevista semi-estructurada
- Preparación de Preguntas:
  - ¿Utiliza el estándar IEEE STD-830 para describir requerimientos?
  - ¿En caso afirmativo ha enfrentado dificultades al utilizarla?

#### **Transcripción de la Sesión 8**

##### **Sesión 8**

**Fecha: 20/04/00**

**Experto: Benjamin L. Kovitz**

**Lugar: Internet**

**IC: Ing. Marcelo Rizzi**

**Técnica empleada: Entrevista semi-estructurada**

**Objetivos: Esclarecer el problema de la estandarización de documentación**

1. ¿Utiliza el estándar IEEE STD-830 para describir requerimientos?  
Realmente, una de las razones principales por las que escribí el libro "Practical Software Requirements" era una frustración con los estándares específicos de compañías que fueron basados en IEEE STD-830. He reescrito los documentos de requisitos escritos en ese estilo. Solamente no forzando las descripciones en el formato de input/processing/output, he conseguido a veces una reducción de 4:1 en la talla del documento, mientras que agregaba detalles necesarios.

Las tentativas de definir estándares exactos para documentar requisitos generalmente utilizan un índice prefabricado. La idea implícita es que todo al análisis se realiza completando espacios en una estructura preexistente, no inventando una

estructura conceptual apropiada para el proyecto actual. (Lo notable es que nadie propone completar una tabla de contenido como manera de escribir un programa.)

Las secciones de " requisitos funcionales " de la STD-830 tienden a forzar una aproximación tipo análisis estructurado de los requisitos, donde usted describe la estructura de alto nivel del programa más bien que describir el problema que se solucionará. "

El famoso Input, Processing, Output para cada " función " tiende a no corresponder a la estructura verdadera del programa, ni describen el interfaz en bastante detalle para programarlo, ni proporcionan la información del dominio del problema que usted necesita para diseñar un interfaz. Y por supuesto muchos clientes no tienen ninguna idea acerca de qué significan las frases " requisitos funcionales " y " requisitos no funcionales "

2.¿Entonces cómo reemplaza el estándar para describir los requerimientos?

El template que utilizo es una lista de contenido, no una tabla de contenido, dado que tabla implica organización y ello dependerá del software en cuestión. La lista de contenido comprende los requerimientos propiamente dichos ya sea consultas, reglas de comportamiento, mapeos, operaciones en dominios creados, luego una exhaustiva descripción del dominio del problema donde se trata las entidades, atributos y relaciones, secuencias de eventos, reglas causales, formatos de archivo, fuentes de información, hardware y software con los que debe interrelacionarse a través de mapeos entre puertos de entrada /salida y hardware.

Además debe contener las expectativas de los usuarios, sus preferencias, invariantes, etc.

Para ello debe indentificarse primero el marco de problema para cada subproblema y luego describir con diferentes técnicas los requerimientos, el dominio del problema según las partes del marco de problema ajustado, las interfaces con el mundo exterior, etc.

### **Análisis de la sesión**

Se evidencia una clara frustración con el estándar IEEE 830 para describir los requerimientos. En contraposición se obtuvo una lista de contenidos que debe estar presente en la documentación de requerimientos que se profundizará luego junto con las técnicas correspondientes en la etapa de extracción de conocimientos del libro del experto.

### **5.3.6 Sesión 9: Ajustes de Marcos**

## Preparación de la Sesión

- Información a tratar: Análisis de problema y ajuste a un marco.
- Amplitud y profundidad: Se trata de esclarecer qué preguntas se realiza a si mismo el experto cuando analiza un problema para identificar cual marco de problema se ajusta mejor.
- Técnica adecuada: entrevista semi-estructurada

## Transcripción de la Sesión 9

### Sesión 9

**Fecha: 22/04/00**

**Experto: Benjamin L. Kovitz**

**Lugar: Internet**

**IC: Ing. Marcelo Rizzi**

**Técnica empleada: Entrevista semi-estructurada**

**Objetivos: Análisis de un problema para ajustar a un marco.**

1. Estoy trabajando en un proyecto que será utilizado para manejar información financiera; Los usuarios ingresan capital inicial y datos periódicos (es decir, renta por un período de tiempo específico). Ejecutarán una serie de cálculos para determinar los datos finales. ¿Cómo debo analizar este problema?

*Si entiendo el proyecto correctamente, lo clasificaría como problema de transformación. La responsabilidad del sistema es realizar cálculos: para convertir datos de entrada de información a los datos de la salida según reglas especificadas.*

*No pensaría en esto como problema de información, puesto que \* obtener \* la información de un mundo real no suena como un factor importante. En un problema de información el requerimiento es sincronizar los informes o pregunta entregadas por el sistema con una cierta parte del mundo real. Un elemento importante de la definición de problema es identificar cómo el ordenador puede descubrir el estado de ese mundo real: a través de los detectores infrarrojos montados en un edificio, a través de los convertidores analógico a digital asociados a los sensores de temperatura, o, lo más típicamente posible, a través de personal de entrada de datos.*

*Tampoco pensaría en esto como problema de control, puesto que el software no es responsable de causar el suceso de alguna cosa en el mundo real, sólo realizar cálculos. El requisito en un problema del control es hacer una cierta parte del*

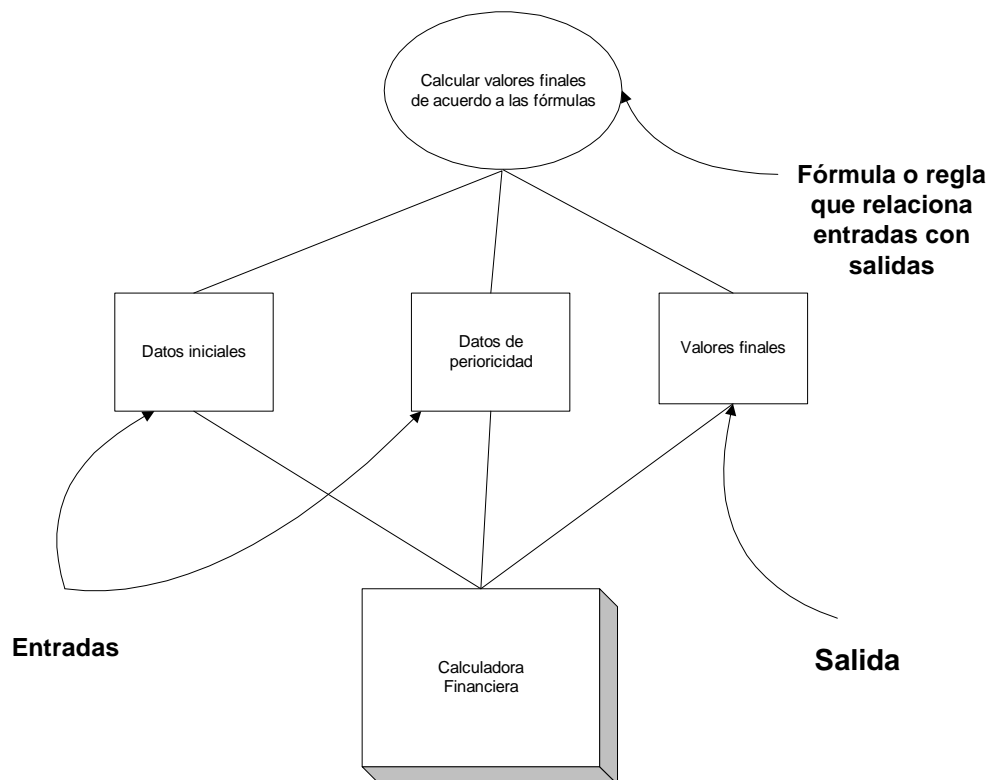


*mundo real \* comportarse \* según reglas especificadas. Una de las partes principales de un problema del control son los medios disponibles para que el ordenador pueda causar el comportamiento deseado: cómo el ordenador puede cambiar los inyectores de combustible de encendido a apagado, que el ordenador puede pedir que muevan items desde el almacén al muelle del envío y a cómo comunicarse con esa persona, etc.*

*Naturalmente, la clasificación no es tan importante como la obtención de las clases de información del dominio del problema que usted necesita para diseñar un interfaz útil. Pero basado en lo que he oído hasta ahora, suena como que usted tiene un problema clásico de " datos de entrada, datos de salida" del problema, donde las interacciones causales con el mundo real son de poca preocupación para el diseñador y los programadores de interfaz.*

*Aquí está un diagrama del marco que muestra las tres partes principales de un problema de transformación: entradas de información, salidas, y una regla o un fórmula que especifica una salida deseada para cada entrada de información posible. Describa exhaustivamente cada uno de éstos, y el problema se define totalmente:*

## Problema de Transformación



*No se muestra el usuario quien es el que provee la información de entrada y lee los resultados.*

*Este problema puede ser complicarse más si el usuario tiene la capacidad de definir la fórmula, si la fórmula misma necesita ser sincronizada con las últimas políticas del gobierno, o si las entradas de información o las salidas necesitan venir indirectamente de una cierta fuente, tal como un plan contable. Todo esto introduciría subproblemas en el problema principal: qué conjunto de elementos puede el usuario combinar para crear una fórmula? (problema Workpieces), de qué fuente puede el sistema aprender las políticas del gobierno? (problema de conexión), cómo puede el sistema comunicarse con el plan contable? (también un problema de conexión) o, si este programa es responsable de realizar cambios en las cuentas llevadas a cabo en un banco, usted tendría un problema genuino de control. En que el caso, usted también tendría que investigar el interfaz a través del cual su programa podría manipular las cuentas, y usted tendría que definir un requisito de comportamiento, indicando la regla según la cual las cuentas se suponen deben cambiar.*

### **Análisis de la sesión 9**

- Para identificar un problema de información debe existir un mundo real de donde obtener información.
- Para identificar un problema de control debe existir un elemento a ser controlado donde deben cumplirse reglas de comportamiento.
- Se debe analizar el problema buscando identificar las partes de marcos de problema conocidos, luego asociar el resto del dominio a otras partes del mismo marco y verificar que se ajuste.
- Una vez que se tiene el marco debe simplemente describirse sus partes y el problema queda documentado.

### **5.3.7 Sesión 10: Análisis del problema con marcos**

#### **Preparación de la Sesión**

- Información a tratar: Análisis de problema y ajuste a un marco.
- Amplitud y profundidad: Se trata de esclarecer qué preguntas se realiza a si mismo el experto cuando analiza un problema para identificar cual marco de problema se ajusta mejor.
- Técnica adecuada: entrevista semi-estructurada

## Transcripción de la Sesión 10

### **Sesión 10**

**Fecha: 24/04/00**

**Experto: Benjamin L. Kovitz**

**Lugar: Internet**

**IC: Ing. Marcelo Rizzi**

**Técnica empleada: Entrevista semi-estructurada**

**Objetivos: Análisis de un problema para ajustar a un marco.**

Supongamos una organización responsable de regular el espectro de radiofrecuencia. Se entregan licencias a las organizaciones para utilizar el espacio del espectro. El espacio del espectro es un área tridimensional, y además el ancho de banda de la frecuencia. Se necesita saber quién tiene qué licencia, que sea capaz de cargar estas licencias en una base anual pero se necesita saber qué espacio está disponible para ventas futuras. El objetivo es aumentar la eficacia en el uso del espacio espectral.

Aparentemente es un problema de información y hay un problema de conexión en términos de que los usuarios incorporan la información sobre frecuencias y áreas. ¿Pero tengo un problema de Workpieces en el sentido de que el espacio del espectro está creado en el sistema y manejamos estos espacios dentro del sistema?

*Con respecto al tema de si la " propiedad del espacio del espectro " es un dominio creado, aquí está lo que preguntaría si hago el análisis: usted desea que el sistema \* decida \* que partes del espectro concede a los nuevos aspirantes, o usted desea que el sistema solamente \* informe \* al personal qué partes del espacio del espectro están disponibles?*

*Para los propósitos de la ingeniería de requerimientos, la diferencia importante es que diversos tipos de problema le conducen a hacer diversas preguntas. La idea es que el marco ayuda a hacerse buenas preguntas.*

*Si la responsabilidad del sistema es tomar las decisiones sobre los segmentos del espacio del espectro concederá a los nuevos aspirantes, usted tiene dos preguntas*

(1) cuáles son las reglas según las cuales el sistema debe tomar estas decisiones?  
(2) qué tipos de actos usted quisiera que la gente pudiera hacer con el sistema? Un probable acto sería " petición de segmento del espacio del espectro. " Una regla simple podría ser, " quienquiera que primero ingrese una petición válida para un segmento dado consigue ese segmento. " (otro acto podría ser, " petición de reemplazar una asignación. " Y la regla correspondiente podría ser, " solamente un miembro de la comisión de comunicaciones puede reemplazar una asignación. ")

Por otra parte, si la responsabilidad de tomar estas decisiones es algo usted desea que la tenga el gobierno, el trabajo del sistema podría ser solamente informar a los usuarios las asignaciones actuales y propuestas del espectro. En este caso, hay diversas preguntas a realizar: (1) cuáles son todos los eventos que cambian la asignación del espacio del espectro? (2) cómo puede el sistema descubrir cuando ocurren estos eventos , junto con toda la información pertinente asociada a ellos?

Una tercera alternativa es que quizás usted no desea dar al sistema la responsabilidad de tomar las decisiones sobre a quién qué espacio del espectro , pero usted quisiera que el sistema sirviera como ayuda a la gente que toma esas decisiones. La gente podría pulsar una petición en el sistema, y el sistema podría proponer una manera posible de satisfacer la petición, o quizás deje a un usuario visión una variedad de escenarios " qué-pasa-si ". En este caso, usted tiene un problema clásico de Workpieces : el dominio del espacio espectral es puramente ficticio, incorporado solamente al ordenador. El software proporciona un conjunto de operaciones que los usuarios pueden realizar en este mundo ficticio para ver los resultados.

Usted podría también llamar esto una clase de problema de transformación, en que el trabajo del sistema es realizar cálculos sobre los datos incorporados por el usuario. De hecho, pequeños problemas de transformación son subproblemas de casi todo problema software. Pero note que las preguntas principales a realizar son, " qué entidades ficticias usted quisiera que creara en el ordenador, y qué operaciones y/o cálculos desea poder realizar en ellas? "

Pero cual es la primera pregunta que debe hacerse en el análisis -- una pregunta que usted puede hacer incluso antes de saber nada sobre el dominio del problema. "¿Qué beneficios desea que el software sea responsable de alcanzar? " Eso hace que el cliente se centre en los efectos más importantes que se alcanzarán en el dominio del problema -- gente que informa algo, haciendo que las cosas sucedan según ciertas reglas, proporcionando los resultados de ciertos cálculos, etc.

una vez que usted tiene el principal problema identificado, entonces puede empezar a investigar los subproblemas implicados, como qué servicio de entrada de datos

*necesita proporcionar, cómo tratar los inevitables errores que ocurren durante la entrada de datos, etc.*

2. ¿Podría describir cuales son las preguntas mas frecuentes que deben hacerse para analizar el dominio e intentar ajustar un marco?

*Los marcos del problema son guías para hacer buenas preguntas, no para clasificar problemas como un fin en sí mismo.*

*Hay que evitar el clasificar por el hecho de clasificar. Los usuarios pueden finalizar concluyendo, " esto es un problema workpieces, eso es un problema de información, y eso es un problema de transformación. Poner cosas en categorías es solamente beneficioso si nosotros tomamos diversas acciones dependiendo de qué categoría reconocemos.*

*Una de las ventajas principales de los marcos de problema reside en ayudar a definir completamente problemas: para incluir no solamente los requerimientos, sino también la información descriptiva asociada del dominio. Es decir, para cada tipo de requerimiento, hay diversas clases de preguntas que usted necesita hacer acerca del dominio del problema (así como diversas clases de soluciones son las apropiadas).*

*Estas son algunas de las preguntas que hay que realizar cuando se enfrentan diferentes tipos de problema:*

*PROBLEMAS de INFORMACIÓN (versión dinámica solamente):*

*Requerimientos:*

*Los usuarios deben poder obtener respuestas a ciertas preguntas. (los requerimientos deben indicar todas las preguntas posibles que los usuarios puedan hacer.)*

*Preguntas del dominio:*

*Cuál es el sujeto de esas preguntas? Qué \* cosas \* (entidades, objetos, o lo que sea ) son el objeto de las preguntas, y qué atributos o relaciones entre esas cosas son necesarios para contestarles?*

*¿Qué eventos cambian las respuestas a las preguntas? Es decir, para cada pregunta, hay una contestación correcta, determinada por, estado del mundo real.*

*¿Cómo puede averiguar el sistema cuándo estos eventos ocurren? ¿Por ejemplo existen base de datos con esta información? ¿Puede subscribir el sistema a alguna*

*fuente automatizada para proporcionarlo? ¿O debe confiar el sistema en usuarios humanos que ingresan la información? ¿En ese caso, cómo consiguen esos usuarios la información?*

*¿Si exactitud o temporización de las respuestas son especialmente importantes, entonces usted también necesita preguntar: De qué maneras puede propagarse la información errónea al sistema o incluir errores? ¿Qué redundancia está allí, en el dominio del problema que el sistema puede aprovechar para descubrir los errores?*

## **PROBLEMAS DE CONTROL**

*Requerimientos:*

*Ciertas cosas se comportan según ciertas reglas. (Los requisitos deben detallar todas las posibles situaciones que se supone que el sistema encontrará, y las acciones deseadas que las cosas controladas deben tomar en cada situación.)*

*Preguntas del dominio:*

*¿Cuáles son las cosas cuya conducta nosotros necesitamos controlar? ¿Cuáles son las reglas de conducta que estas cosas obedecen sin tener en cuenta cómo nosotros diseñamos el sistema? ¿Es decir, las leyes causales inherente en las cosas? Esto incluye describir los eventos espontáneos a los que el sistema debe reaccionar, así como acciones que el sistema puede causar directamente o indirectamente. ¿Cómo puede descubrir el sistema eventos o el estado actual de las cosas en el dominio del problema? (Ésta es la misma pregunta como en un problema de información.) ¿Cuáles son los eventos que el sistema puede causar directamente? ¿Es decir, los dispositivos de salida de la computadora que la computadora puede controlar directamente?*

*¿Cómo se conectan los dispositivos de salida a las cosas cuyo conducta nosotros queremos controlar? ¿Directamente o indirectamente? Si hay entidades intermedias con las que la computadora puede comunicarse a través de los dispositivos de input/output, y eso hará que se cause conducta deseada, nosotros necesitamos también saber controlar éstas entidades, cada manera en la que ellos pueden fallar, y cómo el sistema puede descubrir cuando ellos han fallado.*

## **PROBLEMAS DE TRANSFORMACIÓN**

*Requerimientos:*

*El sistema entrega datos a la salida que corresponden a ciertas reglas aplicadas a los datos de la entrada. (Los requisitos deben detallar la salida deseada para cada posible entrada.)*

*Preguntas del dominio:*

*¿Cuáles son todas las posibles entradas? ¿Cómo consigue el sistema las entradas? ¿(De un archivo, de la entrada del usuario, de algún otro sistema vía TCP/IP? Si es el último caso, entonces usted también necesita saber direcciones del puerto, etc.)*

*¿Cómo entrega el sistema las salidas? ¿(Guárdelos a un archivo, despliegúelos en la pantalla, envíelos vía el Procedimiento Remoto etc.?, etc.)*

*Problemas WORKPIECES*

*Requerimientos:*

*Los usuarios puede crear/editar/manipular/ver/etc. cosas que existen dentro del sistema, como cuentas o documentos.*

*Preguntas del dominio:*

*En un problema workpiece, el dominio del problema no existe aún. En cambio, es creado por el software en respuesta a las órdenes del usuario. No hay realmente ningún dominio del problema a "investigar." ¿Pero hay un dominio del problema a especificar: Juegos posibles de atributos del workpiece? Las relaciones entre ellos (ej. ¿uno-a-muchos, cuando una sección del documento puede contener muchos párrafos, o una sola cuenta puede contener muchas transacciones)?*

*¿Cuáles son las operaciones que los usuarios pueden realizar en los objetos? (Fijando horarios en un cuarto de conferencias, modificando objetos en una herramienta CAD, etc.) Las operaciones constituyen los requerimientos.*

*Otra buena pregunta es si hay cualquier invariante que nosotros queremos ver conservado a lo largo de las operaciones.*

*PROBLEMAS de CONEXIÓN (normalmente parte de otro problema)*

*Requerimientos:*

*Los estados de dos (o más) dominios se corresponden según ciertas reglas. (Los requerimientos deben declarar la regla de la correspondencia precisamente.)*

*Preguntas del dominio:*

*¿Cómo puede averiguar el sistema cuándo los eventos ocurren en los dominios? ¿Es decir, los fenómenos compartidos con la computadora y cómo relacionan ellos a los dominios en cuestión? ¿Qué distorsión o retraso introduce cualquier dominio intermediario?*

*¿Qué redundancia está allí, en los dominios que el sistema puede aprovechar para descubrir o corregir datos erróneos? ¿Si hay fuentes múltiples de datos cuál es la más fiable?*

**Análisis de la Sesión**

Se han obtenido fundamentalmente preguntas concretas que el sistema experto debe incluir cuando se realiza el análisis del problema con el objeto de enmarcar el mismo.

De esta forma se puede comenzar la descripción del dominio y de los requerimientos utilizando también dichas preguntas.

**5.3.8 Sesión 11: Un análisis de protocolo**

<b>Sesión número:</b> 11
<b>Fecha:</b> 25/04/00
<b>IC:</b> Ing. Marcelo Rizzi
<b>Experto:</b> Ing. En Sistemas Diego Linares
<b>Técnica Empleada:</b> Análisis de Protocolo
<b>Material:</b> Cinta de cassette número 1 - Protocolo 1
<b>Objetivos:</b> Profundizar cómo el experto realiza la tarea de búsqueda de un marco de problema, chequea el ajuste del problema a ese marco y procede luego a la especificación de requerimientos a partir de dicho ajuste.



## **Presentación del caso**

El caso presentado al experto para la realización de la especificación de requerimientos, consiste en un sistema de información sobre fotografías almacenadas en un archivo.

La compañía editorial XYZ S.A. dispone de un archivo compuesto de 100.000 fotos de temas y personajes históricos. Dado la complejidad de acceder manualmente al mismo, se pretende su informatización. Para ello se debe construir un sistema completo de archivo fotográfico que contemple subsistemas para digitalización, Alta-Baja-Modificación, control de acceso y consulta al mismo. El caso que nos ocupa corresponde al subsistema de consulta específicamente. Se asume que las fotografías ya han sido digitalizadas y grabadas en diferentes discos compactos, en formatos estándar. Las fotografías se utilizan en diferentes publicaciones de la editorial.

Se han realizado múltiples entrevistas con los usuarios y directivos de la compañía y se dispone de la suficiente documentación como para comenzar la especificación.

## **Paso1: Grabación del protocolo**

(Desgrabado de la cinta número 1)

Bien, ya he estudiado la documentación relevada, la elicitación de requerimientos realizada, las entrevistas, reportes de usuarios. Procedo entonces a realizar la especificación.

La necesidad detectada comprende la realización de consultas por parte de los usuarios de la compañía a un archivo de fotografías, por lo que hago la hipótesis de que se trata de un sistema de información. Veamos entonces si se ajusta con el marco de problema para sistemas de información simple.

El *Sistema* es la maquina que se debe construir.

El *Mundo Real* comprende la colección de las 100.000 fotos disponible y sus datos asociados tanto históricos como técnicos de la fotografía, los usuarios editores que necesitan consultar el archivo para decidir cuales fotografías incluir en sus artículos, los usuarios fotomecánicos que necesitan consultar los datos técnicos de las fotografías para su procesamiento.

La *Información Requerida* comprende las consultas realizadas por los usuarios.

La *Información Entregada* comprende la información y material que brinda el sistema sobre las fotografías.

La *Función de Información son los Requerimientos* que se deben especificar. Los usuarios desean realizar consultas al sistema para que este les entregue la

información en la forma pedida, determinada en parte por los datos ingresados en la consulta y que refleja fielmente la información almacenada en el archivo.

Las consultas de los usuarios ocurren de forma espontánea y autónoma. El dominio de la *Información Requerida* no tiene un estado interno definido o una estructura. Es una fuente de pedidos de información ordenada en el tiempo pero no estructurada. Por lo que resulta ser un *dominio activo dinámico*.

El *Mundo Real* es un dominio estático ya que al ser un conjunto de fotografías históricas, no existirán eventos que provoquen cambios de estado en el mismo.

Una vez que he identificado las cuatro partes del marco de problema de sistemas de información, procedo a realizar los tests informales de ajuste de un marco a un problema. El primero de ellos es el test de separabilidad. Me pregunto: ¿He sido capaz de separar las partes del marco del problema? Si, aparentemente son perfectamente identificables. Segundo, el test de completitud. ¿Cada parte del problema ha podido ser ajustada de una forma natural y razonable en alguna parte del marco del problema? ¿Ninguna parte ha quedado fuera? No, todo parece estar ajustado en formal razonable y natural y nada ha quedado fuera de consideración. Tercero, el test de las características de las partes. ¿Las diferentes partes del problema tienen características demandadas por las partes principales del marco donde fueron asignadas? Si se corresponden. Cuarto y ultimo, el test de proporcionalidad. Cuando se ajustaron las partes del problema a cada parte del marco, ¿la carga de cada parte es similar a las otras? ¿hay partes del marco que están casi vacías de contenido y otras sobrecargadas? No, todo parece estar razonablemente balanceado.

Los cuatro test resultaron exitosos, tenemos entonces que el marco de problema apropiado es un **marco de problema de sistema de información estático**.

Esto quiere decir que debo concentrarme sobre la parte denominada *Función de Información* que conforma los Requerimientos del sistema.

Para llevar a cabo la especificación de la relación entre la información requerida, la entregada y el mundo real, debo identificar las diferentes **entidades** que forman parte del dominio de aplicación y que son relevantes a los efectos de los requerimientos. Ellas son: Fotografías de alta resolución, Minifotos (Thumbnails), Compact Disc, copia impresa de una foto, copia magnética de una foto, usuario editor, usuario fotomecánico.

Bien, ya tengo identificadas las entidades principales, ahora debo identificar sus **atributos**.

*Fotografías de alta resolución:* imagen, Persona o evento, lugar, fecha, descripción, nombre, formato digital, ancho, alto, color/bn, resolución, fecha ultima publicación, numero de publicaciones, numero de CD.

*Minifoto:* imagen, Foto de alta resolución a la que pertenece.

*CD:* capacidad total, numero.

*Copia impresa de la foto:* No tiene atributos

*Copia magnética de la foto:* No tiene atributos

*Usuario Editor:* Nombre

*Usuario Fotomecánico:* Nombre

Finalizada la identificación de atributos, procedo a determinar la **cardinalidad de relaciones** entre las entidades.

- Para cada foto hay una minifoto.
- Cada minifoto pertenece a una única foto
- Un CD puede albergar muchas fotos
- Una copia impresa pertenece a una única foto de alta resolución
- Una copia magnética pertenece a una única foto de alta resolución

Ahora procedo a identificar las informaciones requeridas y entregadas sobre el mundo real en cuestión. Para ello divido los requerimientos funcionales de comportamiento y los de no comportamiento.

Para proceder a especificar los requerimientos de comportamiento divido al sistema en subfunciones, identificando básicamente una función de información para toma de decisión de fotografía a publicar e información técnica de la misma, y otra función de entrega de material.

### **Función de información**

1. El sistema de información debe reportar todas las Minifotos de las fotografías que pertenecen a un intervalo de fechas requerido por el usuario editor.

Entradas: Fecha desde, Fecha Hasta. Formato: DD/MM/AAAA

Procesamiento: buscar en el archivo todas las fotografías cuya fecha esté comprendida entre ambas fechas de entrada.

Salida: Reportar las mismas en pantalla, incluyendo en el reporte el atributo nombre y la entidad minifoto cada una. El reporte debe estar organizado por páginas. Cada página del reporte debe incluir hasta 8 fotografías.

2. El sistema de información debe reportar todas las Minifotos de las fotografías que corresponden a una persona o evento requerido por el usuario editor.

Entradas: Persona o evento. Formato: Cadena de caracteres

Procesamiento: buscar en el archivo todas las fotografías cuyo atributo persona o evento coincida total o parcialmente con el dato de entrada.

Salida: Reportar las mismas en pantalla, incluyendo en el reporte el atributo nombre y la entidad minifoto cada una. El reporte debe estar organizado por páginas. Cada página del reporte debe incluir hasta 8 fotografías.

3. El sistema de información debe reportar todas las Minifotos de las fotografías que corresponden a un lugar requerido por el usuario editor.

Entradas: lugar. Formato: Cadena de caracteres

Procesamiento: buscar en el archivo todas las fotografías cuyo atributo lugar coincida totalmente con el dato de entrada.

Salida: Reportar las mismas en pantalla, incluyendo en el reporte el atributo nombre y la entidad minifoto cada una. El reporte debe estar organizado por páginas. Cada página del reporte debe incluir hasta 8 fotografías.

4. El sistema de información debe reportar todas las Minifotos de las fotografías cuya ultima fecha de publicación sea anterior a una fecha indicada por el usuario editor.

Entradas: Fecha de Publicación. Formato: DD/MM/AAAA

Procesamiento: buscar en el archivo todas las fotografías cuyo atributo fecha de publicación sea anterior a la fecha ingresada como dato de entrada.

Salida: Reportar las mismas en pantalla, incluyendo en el reporte el atributo nombre, la entidad minifoto y el atributo última fecha de publicación para cada una. El reporte debe estar organizado por páginas. Cada página del reporte debe incluir hasta 8 fotografías.

5. El sistema de información debe reportar todas las Minifotos de las fotografías cuyo numero de publicaciones sea inferior a un numero indicado por el usuario editor.

Entradas: Número de publicaciones. Formato: Número Entero

Procesamiento: buscar en el archivo todas las fotografías cuyo atributo número de publicaciones sea menor al número ingresado como dato de entrada.

Salida: Reportar las mismas en pantalla, incluyendo en el reporte el atributo nombre, la entidad minifoto y el atributo número de publicaciones para cada una. El reporte debe estar organizado por páginas. Cada página del reporte debe incluir hasta 8 fotografías.

6. El sistema de información debe reportar la imagen minifoto, el ancho, alto, resolución , formato digital y numero de CD de una fotografía cuyo nombre sea indicado por el usuario Fotomecánico.

Entradas: Nombre de la foto. Formato: Cadena de Caracteres.

Procesamiento: buscar en el archivo la fotografía cuyo atributo nombre coincida plenamente con el nombre ingresado como dato de entrada.

Salida: Reporte en pantalla, incluyendo el atributo minifoto, el ancho, alto, resolución , formato digital y numero de CD.

7. El sistema de información debe poder combinar los criterios de intervalo de fechas, persona o evento, lugar, ultima fecha de publicación y numero de publicaciones indicados por el usuario editor, y reportar todas las Minifotos de las fotografías que cumplen con dicho criterio combinado.

#### **Función de Entrega de material**

8. El sistema de información debe generar una copia impresa color de una fotografía determinada por el usuario editor.

Entrada: Nombre de la fotografía. Formato: Cadena de Caracteres.

Procesamiento: Buscar la fotografía de alta resolución cuyo nombre coincide plenamente con el dato de entrada, en el CD correspondiente y los atributos de la misma en el archivo. Preparar la generación de una copia impresa de la misma.

Salida: Reporte impreso en hoja tamaño A4 de la imagen de alta resolución de la fotografía acompañada por los atributos Nombre, Descripción, persona/evento y Lugar.

9. El sistema de información debe generar una copia impresa color de una fotografía determinada por el usuario Fotomecánico.

Entrada: Nombre de la fotografía. Formato: Cadena de Caracteres

Procesamiento: Buscar la fotografía de alta resolución cuyo nombre coincide plenamente con el dato de entrada, en el CD correspondiente y los atributos de la misma en el archivo. Preparar la generación de una copia impresa de la misma.

Salida: Reporte impreso en hoja tamaño A4 de la imagen de alta resolución de la fotografía acompañada por los atributos: Nombre, ancho, alto, resolución, formato digital y CD donde se ubica la fotografía de alta resolución.

10. El sistema de información debe generar una copia de alta resolución en soporte magnético de una fotografía de determinada por el usuario editor o el usuario Fotomecánico.

Entrada: Nombre de la fotografía, Soporte magnético donde se desea la copia.

Formato: Cadena de Caracteres

Procesamiento: Buscar en el CD Correspondiente la fotografía de alta resolución cuyo nombre coincide con el ingresado como dato de entrada y preparar la copia de dicha fotografía a un soporte magnético ingresado como dato de entrada.

Salida: Copia en soporte magnético de la fotografía de alta resolución y confirmación en pantalla de la tarea realizada.

### **Funcionalidades Varias**

11. Si una consulta cualquiera de las especificadas en los requerimientos funcionales de información 1 al 6, excede la cantidad de 100 fotografías que cumplen con los criterios de búsqueda, el sistema no generará reporte alguno y alertará al usuario de tal situación.

12. Si para una consulta cualquiera de las especificadas en los requerimientos funcionales de información 1 al 6, la cantidad de fotografías que cumplen con los criterios de búsqueda es cero, el sistema no generará reporte alguno y alertará al usuario de tal situación.

### **Requerimientos de No comportamiento**

13. El tiempo de respuesta para todas las consultas especificadas en los requerimientos funcionales de información 1 al 6, no podrá exceder los 5 segundos hasta el comienzo del reporte.

14. Se debe soportar una cantidad de 6 terminales para realizar las consultas.

15. Se debe soportar una cantidad de 6 usuarios simultáneos, realizando consultas.

16. El sistema debe poder administrar las 100.000 fotografías.

### **Paso 2: Transcripción del Protocolo**

(Se realiza parte de la transcripción dado su longitud, hasta el momento en que el experto chequea el ajuste del problema al marco de problema para sistemas de información)

1. Bien, ya he estudiado la documentación relevada, la elicitación de requerimientos realizada, las entrevistas, reportes de usuarios. Procedo entonces a realizar la especificación.
2. La necesidad detectada comprende la realización de consultas por parte de los usuarios de la compañía a un archivo de fotografías
3. por lo que hago la hipótesis de que se trata de un sistema de información
4. Veamos entonces si se ajusta con el marco de problema para sistemas de información simple.
5. El *Sistema* es la maquina que se debe construir.
6. El *Mundo Real* comprende las fotografías ... y sus datos históricos y técnicos ..., los usuarios editores ... los usuarios fotomecánicos
7. La *Información Requerida* comprende las consultas realizadas por los usuarios.
8. La *Información Entregada* comprende la información y material que brinda el sistema sobre las fotografías.
9. La *Función de Información son los Requerimientos* que se deben especificar.
10. Las consultas de los usuarios ocurren de forma espontánea y autónoma.
11. El dominio de la *Información Requerida* ... es una fuente de consultas de información ordenada en el tiempo pero no estructurada. Por lo que resulta ser un *dominio activo dinámico*.
12. El *Mundo Real* es un dominio estático ... no existirán eventos que provoquen cambios de estado en el mismo.
13. Una vez que he identificado las cuatro partes del marco de problema de sistemas de información, procedo a realizar los tests informales de ajuste de un marco a un problema.
14. Test de separabilidad: He sido capaz de separar las partes del marco del problema? Si, aparentemente son perfectamente identificables.
15. test de completitud. ¿Cada parte del problema ha podido ser ajustada de una forma natural y razonable en alguna parte del marco del problema? No, todo parece estar ajustado en formal razonable y natural y nada ha quedado fuera de consideración
16. test de las características de las partes. ¿Las diferentes partes del problema tienen características demandadas por las partes principales del marco donde fueron asignadas? Si
17. test de proporcionalidad. Cuándo se ajustaron las partes del problema a cada parte del marco, ¿la carga de cada parte es similar a las otras? ¿hay partes del marco que están casi vacías de contenido y otras sobrecargadas? No, todo parece estar razonablemente balanceado.

18. Los cuatro test resultaron exitosos, tenemos entonces que el marco de problema apropiado es un marco de problema de sistema de información estático.
19. Esto quiere decir que debo concentrarme sobre la parte denominada *Función de Información* que conforma los Requerimientos del sistema.



### **Paso 3: Codificación del protocolo**

#### **3a. Identificación de conceptos, características, valores y relaciones**

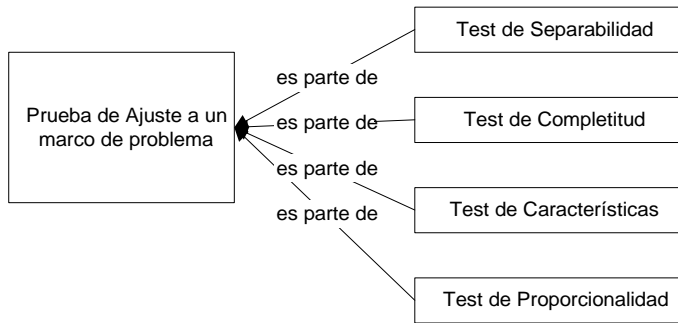
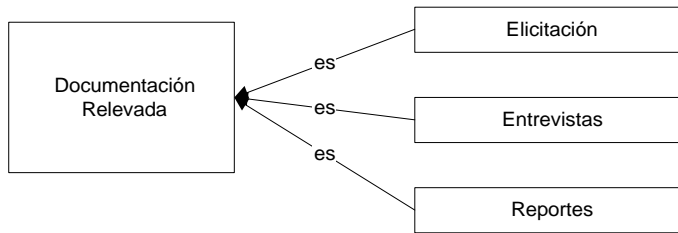
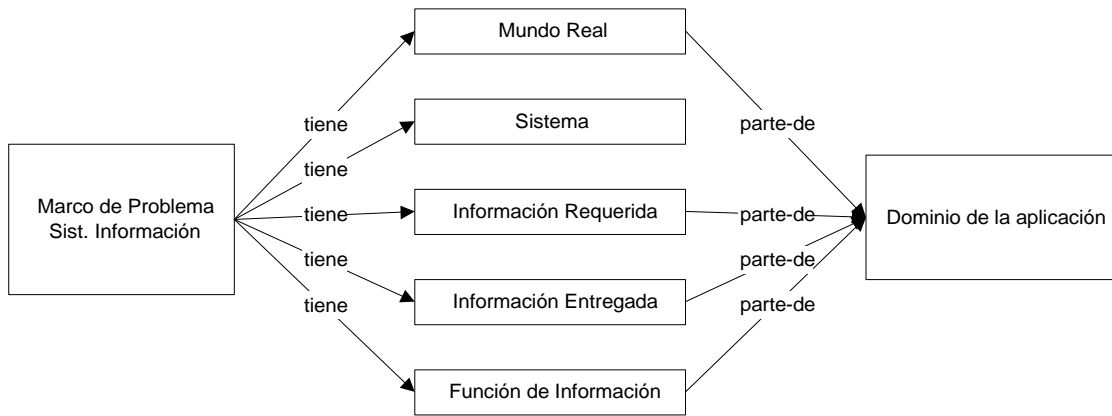
##### **Elementos del protocolo**

Documentos	Concepto
Elicitación	Valor
Procedo entonces	Operador
Comprende	Relación
entrevistas	valor
reportes	valor
Fotografías	Concepto
Datos	Característica
Históricos	Valor
Técnicos	Valor
Usuarios	Concepto
Editores	valor
Fotomecánicos	Valor
Hipótesis	Operador
Sistema de información simple	Valor
Marco de problema	Concepto
Veamos si se ajusta	Operador
Sistema	Valor
Máquina	Característica
Mundo Real	Característica
Información requerida	Característica
Información entregada	Característica
Consultas	Concepto
Información y material	Valor
Función de información	Característica
Requerimientos	Valor
Espontanea y autónoma	Valor
Dominio	Concepto
No estructurada	Valor
Activo	Valor
Dinámico	Valor
Es	Operador
Resulta ser	Operador
Estático	Valor
He identificado	Operador
Partes del marco	Característica
Test	Concepto
separabilidad	Característica
completitud	Característica
características de partes	Característica
proporcionalidad	Característica
Aparentemente son	Operador
Identificables	Valor
Ajustado	Característica
Razonable y natural	Valor
Tienen	Valor
Carga de partes	Característica
Razonablemente balanceado	Valor
Exitoso	Valor
Resultado	Característica

### Conceptos, Características y Valores del Protocolo

Concepto	Características	Valores
Documento	Tipo	Elicitación Entrevistas Reportes
Fotografía	Datos	Históricos Técnicos
Usuarios	Tipo	Editores Fotomecánicos
Consultas	Tipo	Espontáneas y autónomas.
Marco de Problema	Tipo	Sistema de Información Simple
	Sistema	Máquina
	Mundo Real	fotografías, usuarios.
	Información requerida	datos y material fotográfico
	Información Entregada	información y material fotográfico.
	Función de Información	Requerimientos
Dominio	Tipo	Activo Dinámico estático
Test de Separabilidad	resultado	Exitoso
	chequeo	Partes perfectamente identificables
Test de completitud	resultado	Exitoso
	chequeo	Todo ajustado en forma natural y razonable
Test de Características de partes	resultado	Exitoso
	chequeo	Tienen características demandadas
Test de Proporcionalidad	resultado	Exitoso
	chequeo	carga de partes razonablemente balanceadas.

## Relaciones



### **3b. Identificación de la búsqueda**

El modelo de la marcha intelectual del experto nos indica que primero trata de ajustar el problema a un marco de problema, luego validar dicha hipótesis y finalmente comenzar a especificar los requerimientos en función de dicho marco. La parte que analizamos incluye el protocolo de ajuste a un marco y su validación. Para ello comienza haciendo la hipótesis 3, a partir del razonamiento en 1 y 2. Luego relaciona cada parte del marco con el problema analizado, esto es, el Sistema en 5, el mundo real en 6, información requerida en 7, información entregada en 8 y función de información en 9. Luego avanza sobre las características de cada parte identificada en términos de tipo de dominio, líneas 10,11 y 12. Posteriormente realiza los 4 test para verificar la hipótesis, líneas 14, 15, 16, 17, concluyendo con la verificación en la línea 18 acerca de los resultados obtenidos. Finalmente en la línea 19 identifica dónde debe concentrar el problema de la especificación.

### **3c. Identificación de los operadores**

- *Se detectó la necesidad de consultas a un archivo...entonces se establece la hipótesis de un sistema de información...*

Aquí tenemos que el experto establece una hipótesis. Es explícitamente un operador.

#### **Establecer hipótesis**

- *Las consultas de los usuarios ocurren de forma espontánea y autónoma.*

En éste caso se asigna un valor a la característica tipo del concepto consultas.

#### **Asignación de valor a una característica**

- *El dominio de la Información Requerida ... es una fuente de consultas de información ordenada en el tiempo pero no estructurada. Por lo que resulta ser un dominio activo dinámico.*

El experto establece con certeza absoluta una verdad.

#### **Establecer con certeza absoluta**

- *todo parece estar ajustado en formal razonable y natural y nada ha quedado fuera de consideración*

Hay una incertidumbre que el experto utiliza para aumentar la credibilidad de su hipótesis.

#### **Aumentar credibilidad**

Se obtuvieron 4 operadores a saber:

- Establecer hipótesis
- Asignación de valor a una característica
- Establecer con certeza absoluta
- Aumentar credibilidad

### **3d. Identificación de las inferencias**

Se procede a continuación a identificar las inferencias que explícitamente se encuentran en el discurso del experto.

Si (Requerimientos.Tipo = consultas) ENTONCES

Establecer-hipótesis(Sistema de Información)

Si (Dominio.Tipo= Estático) Y (Información-Requerida = Dominio Activo Dinámico)  
ENTONCES

Aumentar-Credibilidad(Marco de Problema Sistema de Información)

Si (Test-Separabilidad) = Si ENTONCES

Aumentar-Credibilidad(Marco de Problema Sistema de Información)

Si (Test-Compleitud) = Si ENTONCES

Aumentar-Credibilidad(Marco de Problema Sistema de Información)

Si (Test-Características-Partes) = Si ENTONCES

Aumentar-Credibilidad(Marco de Problema Sistema de Información)

Si (Test-Proporcionalidad) = Si ENTONCES

Establecer-Certeza-Absoluta(Marco de Problema Sistema de Información)

### **3e. Sinónimos, Metacomentarios e Incertidumbres**

#### **Sinónimos**

Fotos --- Imágenes Alta resolución

Consultas --- Información pedida

Material brindado --- Copias impresas y magnéticas

### Metacomentarios

Ya he estudiado la documentación relevada	Indica que el trabajo con los usuarios debe ser previamente realizado para comenzar con la tarea.
Veamos entonces si se ajusta...	Indica secuencialidad
Una vez que he identificado...procedo a ...	Indica secuencialidad de pasos
Esto quiere decir que debo concentrarme...	Cambio de prioridad. Descartar otros aspectos y abocarse a una tarea o área específica.

### Incertidumbres

Las incertidumbres con que el experto se maneja son las siguientes:

- Veamos entonces si...
- Aparentemente son...
- Todo parece estar..
- Razonablemente...

### Paso 4: Interpretación

El razonamiento comienza con el chequeo de la información existente obtenida en el trabajo de elicitación para la obtención de los requerimientos informales. Luego hace un breve análisis para establecer la hipótesis de que tipo de problema podría tratarse para tratar de ajustarlo a el marco de problema correspondiente.

SI Existe (Documentación-relevada) ENTONCES

Ejecutar(Especificación)

Si (Usuarios.Necesidad = consultas) ENTONCES

Establecer-hipótesis(Sistema de Información)

Luego identifica cada parte del marco de problema para sistemas de información obtenido de la hipótesis planteada.

SI (Marco-De-Problema = MP de Información-simple) ENTONCES

Identificar(Mundo-Real) Y Identificar(Información-Requerida) Y  
identificar(Información-Entregada) Y identificar(Función-de-información) Y  
identificar(Sistema)

SI (Tests-de-ajuste = SI) ENTONCES

Establecer-Certeza-absoluta(Marco de problema sistema de información)

### **Evaluación de la sesión**

La sesión nro 11 de análisis de protocolos ha resultado exitosa para determinar el proceso de selección de un marco y ajuste al mismo para posteriormente realizar las descripciones pertinentes. No obstante el caso corresponde a un problema que se ajusta a un marco único. Los problemas reales de software generalmente se componen de varios marcos simultáneos y problemas de conexión entre el mundo real y el software con las distorsiones y retardos que ellos causan.

#### **5.3.9 Sesión 12: Profundización de ajuste de marcos al problema**

No ha quedado claro aún cómo el experto, al analizar el problema, comienza a discernir los marcos de problema a los que podría ajustarse el problema. Para ello necesitamos una nueva sesión con el experto para profundizar sobre el tema en particular, que heurísticas o reglas particulares sigue para sistematizar el análisis.

- Información a tratar: Profundización de la tarea del experto en el desglose del análisis buscando el ajuste a un marco.
- Amplitud y profundidad: Se trata de esclarecer la tarea que lleva adelante el experto, sin detallar casos específicos.
- Técnica adecuada: Entrevista estructurada

**Esta entrevista fue hecha al experto Michael Jackson, autor del libro Software Requirement and Specification. El experto al analizar la pregunta, envió su respuesta por correo electrónico, que contenía una metodología de análisis general teniendo en cuenta bibliotecas de marco existentes o bien conceptos de dominios y marcos elementales.**

**Sesión nro. 12**

**Fecha: 27/04/00**

**Experto: Michael Jackson**

**Lugar: Inglaterra (Internet)**

**IC: Ing. Marcelo Rizzi**

**Técnica empleada: Entrevista estructurada**

**Objetivos: Profundizar el método y heurísticas de análisis de problema y ajuste a marcos.**

¿Podría describir el razonamiento que sigue en el momento de realizar la descomposición del problema de modo de reconocer subproblemas ajustables a marcos conocidos?

### ***Descomposición del problema***

*Los marcos de problema elementales sólo tratan con problemas idealizados simplificados. Incluso cuando se extienden a través de variantes, el conjunto de marcos elementales no abarca muchos problemas de tamaño y complejidad real. Un problema real generalmente debe descomponerse en subproblemas. Un conjunto adecuado de marcos nos da un gran soporte para encontrar un juego de subproblemas de modo de generar una descomposición apropiada de cualquier problema real. Hay varias aproximaciones a la tarea de descomposición.*

### ***Descomposición de afuera hacia adentro***

*A veces el problema parece incluso no encajar ni siquiera aproximadamente en ningún marco conocido. Puede ser entonces útil descomponer el problema trabajando del exterior hacia el interior. La aproximación aquí es intentar encontrar partes o aspectos del problema reconocibles que correspondan a los marcos conocidos, y analizarlos en el contexto de esos marcos. Entonces ellos pueden considerarse como problemas resueltos, y las partes y aspectos del problema original que deben todavía ser resueltos pueden ser considerados sin la complicación adicional del subproblema ya resuelto.*

*Este acercamiento es esencialmente una aplicación iterativa de la conocida heurística "hallar una porción del problema que pueda ser resuelta." Si la aproximación tiene éxito, el problema original se achica a un núcleo simple que encaja un marco conocido.*



### **Descomposición de adentro hacia afuera**

*A veces el problema parece encajar un marco conocido aproximadamente, pero exhibe dificultades que frustran la pura aplicación del marco. Estas dificultades dan lugar a subproblemas que puede ser reconocibles como que se ajustan a otros marcos en si mismos. Por ejemplo, una forma de dificultad es una dificultad de conexión: puede ser alguna información necesitada por la máquina no está directamente disponible cuando se necesita. Puede ser entonces posible definir la dificultad como un subproblema de de información en el que la máquina original juega el papel de requiriente de información. Otro tipo de dificultad es una dificultad de identidades en la que la máquina comparte un juego de eventos o estados (fenómenos ) con el dominio del problema pero no comparte los papeles asociados que identifican las entidades del dominio participantes.*

*Esta aproximación puede pensarse como que se trabaja del interior hacia el exterior, donde el interior es el marco que parece encajar aproximadamente y el exterior es el juego circundante de dificultades. El problema del centro puede analizarse en la asunción que las dificultades se superarán en las soluciones al subproblema que los captura o parece capturarlos. Este acercamiento, también, es una aplicación de una muy conocida heurística: "resuelva un problema más simple."*

### **Reconociendo un Marco Compuesto Estándar**

*Si bien los marcos elemental forman la base de la técnica de análisis del problema, una técnica totalmente desarrollada deberá tener un juego rico de marcos compuestos. Puede esperarse que una parte sustancial de un problema real, o incluso, el problema completo, encajará en un marco compuesto conocido. Reconocer y explotar este ajuste significa aplicar la heurística "el mejor étodo es haber resuelto el mismo problema antes."*

### **Análisis de la sesión 12**

Con esto, según el problema observado se tiene el mecanismo adecuado para la descomposición del problema.

### **Evaluación de la sesión 12**

Ha sido muy fructífera dado que se ha podido obtener una heurística para descomponer el problema desde diferentes ópticas.

### **5.3.10 Sesión 13: Análisis de conexiones**

Queda por dilucidar cómo el experto realiza el tratamiento de los problemas de conexión, dado que en algunos casos se los trata como un problema en toda su dimensión y en otros forma parte de otro marco de problema.

- Información a tratar: Profundización de la tarea del experto en el tratamiento de los problemas de conexión.
- Amplitud y profundidad: Se trata de esclarecer la tarea que lleva adelante el experto, sin detallar casos específicos.
- Técnica adecuada: Entrevista estructurada

**Esta entrevista fue también hecha al experto Michael Jackson, autor del libro Software Requirement and Specification. El experto al analizar la pregunta, envió su respuesta por correo electrónico.**

**Sesión nro. 13**

**Fecha: 03/05/00**

**Experto: Michael Jackson**

**Lugar: Inglaterra (Internet)**

**IC: Ing. Marcelo Rizzi**

**Técnica empleada: Entrevista estructurada**

**Objetivos: Profundizar el método y heurísticas de análisis de problema de conexión.**

1. ¿Cómo percibimos que puede existir un problema de conexión al analizar el problema?

Los fenómenos compartidos, vistos desde diferentes perspectivas desde diferentes dominios, son la esencia de la interacción y comunicación entre dominios de cualquier clase. Si lo que se observa desde dos dominios se considera como un mismo fenómeno se deben cumplir las siguientes dos precondiciones: Primero la correspondencia entre la observación en los dos dominios debe ser confiable. Segundo si el fenómeno compartido es un evento, debe estar suficientemente sincronizado entre ambos dominios, es decir no debe existir retardo significativo. Cuando ambas precondiciones de velocidad y confiabilidad no se cumplen, se debe reconocer la presencia de otro dominio: El dominio de conexión.

2. ¿Cómo debemos proceder ante la presencia de un dominio de conexión?

Usted tiene 3 opciones:

- a. Ignorar el dominio de conexión y tratar los fenómenos de cada lado como compartidos, en desmedro de conocer que no lo son del todo. La desventaja es que se pierde la dimensión de lo que está sucediendo en el mundo real.

- b. Tratar el dominio de conexión como si fuera el dominio de interés. De ésta forma se ignora la existencia del mundo real. En algunos casos podría tener sentido, pero no recomiendo esta opción.
  - c. Aceptar que el dominio de conexión y el dominio al otro lado del mismo son ambos importantes. Su interrelación es importante de modo de considerarla como un marco de problema. Utilizar entonces el marco de problema de conexión.
3. ¿Básicamente dónde debe indagarse la presencia de un dominio de conexión en los marcos de problemas estudiados?
- En el marco de problema de información la posible presencia de un dominio de conexión ocurre entre el dominio mundo real y la máquina. En el caso de marco de problema de control suele ocurrir entre la máquina y el dominio controlado.

### **Análisis de la sesión 13**

Se obtuvo el modo de analizar los puntos críticos donde pueden aparecer problemas de conexiones y como proceder según el caso.

#### **5.4. Extracción de Conocimientos**

En ésta sección se han agrupado todas las sesiones de adquisición de conocimientos a partir de la documentación existente. Debido a la longitud de las mismas se han incorporado las más significativas. El objetivo es obtener la terminología utilizada, y las metodologías de análisis y documentación más frecuentemente utilizadas.

<b>Sesión número:</b> 1
<b>Fecha:</b> 15/04/00
<b>IC:</b> Ing. Marcelo Rizzi
<b>Técnica Empleada:</b> Análisis estructural de texto.
<b>Texto Analizado:</b> Software Requirements & Specifications, Michael Jackson, Addison-Wesley, 1995. ISBN: 0-201-87712-0
<b>Objetivos:</b> Obtener conceptos fundamentales acerca de requerimientos, terminología utilizada, el entorno de la especificación de requerimientos, identificación de la problemática de la especificación de requerimientos.

Se realizó una lectura general del texto y luego se procedió a la búsqueda sistemática de relaciones existentes en las construcciones gramaticales correspondientes a afirmaciones y definiciones, tales como:

ES-UNA-CARACTERISTICA-DE, SE-COMPONE-DE, SE-DEFINE-COMO, TRATA-SOBRE, ESTA-RELACIONADO-CON, SE-USA-PARA, ES-LA-FINALIDAD-DE, ES-UN, ES-CAUSA-DE, etc.

Luego se realizó un análisis de contexto de dichas construcciones gramaticales para corroborar que la frase en la que se inscribe corresponda a un concepto, definición o afirmación útil.

Como resultado del procedimiento se obtuvieron los siguientes conceptos:

### **Conceptos**

REQUERIMIENTOS

DOMINIO

DOMINIO DE APLICACION

MAQUINA

FENÓMENO

ESPECIFICACION

MARCO DEL PROBLEMA

CONTEXTO DEL PROBLEMA

### **Conocimientos Extraídos**

DOMINIO

Es una porción particular del mundo que puede ser distinguida porque es convenientemente considerada como un todo, y puede ser considerada independientemente de otras partes del mundo.

CONTEXTO DEL PROBLEMA

Es la parte del mundo donde la MÁQUINA será instalada, y en la cual sus efectos serán percibidos y evaluados.

El CONTEXTO DEL PROBLEMA se compone de dos DOMINIOS bien diferenciados: El DOMINIO DE LA APLICACIÓN y la MAQUINA.

DOMINIO DE LA APLICACION

- Es la parte del CONTEXTO DEL PROBLEMA en la cuál los efectos de la MAQUINA creada serán percibidos, evaluados y (si se tiene éxito) aprobados por el cliente o usuario.

REQUERIMIENTOS

- Los REQUERIMIENTOS tratan sobre los FENÓMENOS del *DOMINIO DE LA APLICACIÓN*, no sobre la *MAQUINA*.

- Hay un espacio abierto entre los REQUERIMIENTOS del cliente y lo que la *MAQUINA* puede ejecutar directamente, porque los REQUERIMIENTOS de cliente no están limitados a los FENÓMENOS compartidos entre el *DOMINIO DE LA APLICACIÓN* y la *MAQUINA*.

#### MAQUINA

- La finalidad de la *MAQUINA* consiste en asegurar que dichos REQUERIMIENTOS se satisfagan dado que comparte algunos FENÓMENOS con el *DOMINIO DE LA APLICACIÓN*, como eventos o estados comunes a ambos.
- Cuando se construye software se construye una *MAQUINA*. Se construye el comportamiento y las propiedades de la *MAQUINA* que la hará útil para algún propósito particular.

#### FENOMENO

- Un FENÓMENO es aquello que aparenta existir, o estar presente o ser el caso cuando se observa el mundo o alguna parte del mismo en un DOMINIO.

#### MARCO DE PROBLEMA

- Un problema es caracterizado por la estructura y características del *DOMINIO DE LA APLICACIÓN* y por los REQUERIMIENTOS del cliente en el *DOMINIO DE LA APLICACIÓN*.
- Un MARCO DE PROBLEMA es una estructura consistente de actores principales y tareas para la solución.
- Un MARCO DE PROBLEMA es una generalización de una clase de problema.
- Un MARCO DE PROBLEMA estipula que los actores principales deben estar interconectados en alguna forma y deben tener ciertas características
- Un MARCO DE PROBLEMA caracteriza una clase de problema despojándolo de todo excepto de los elementos esenciales.

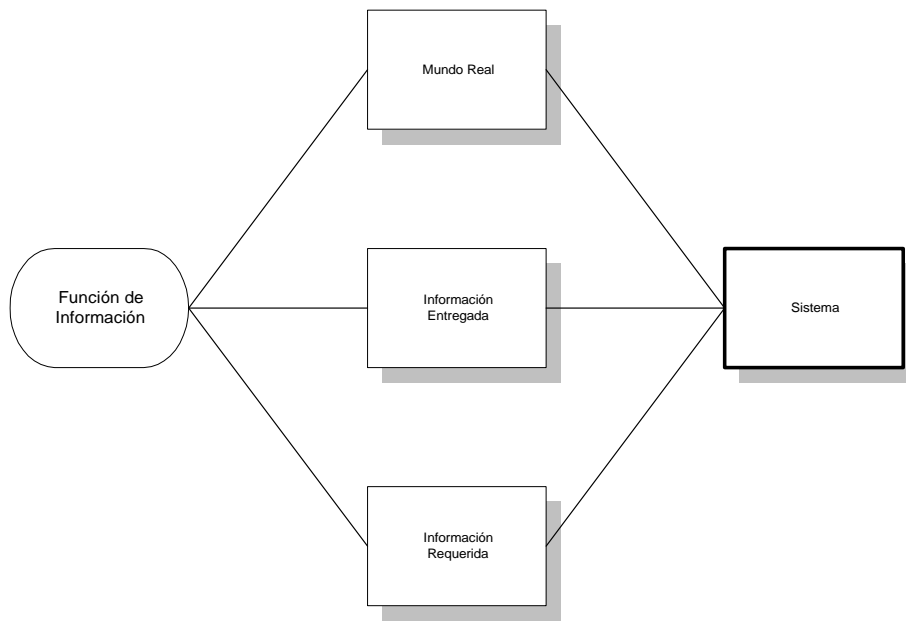
#### DOMINIO (Características)

- Estático, Dinámico
  - Unidimensional, Multidimensional
  - Tangible, Intangible
  - Inerte, Reactivo, Activo
  - Autónomo, Programable
- En un DOMINIO dinámico hay una dimensión del tiempo, hay eventos, estados.

- En un DOMINIO estático no hay una dimensión del tiempo, no hay eventos y nada cambia , nada sucede.
- Un archivo de registros en una cinta magnética es unidimensional pero una base de datos relacional es multidimensional.
- Un DOMINIO dinámico tangible está sujeto a limitaciones físicas que introducen consideraciones de tiempo real.
- Un DOMINIO dinámico es inerte si nunca realiza algo por iniciativa propia. Existe la dimensión del tiempo pero los eventos son producidos por agentes externos al DOMINIO.
- Un DOMINIO es reactivo si realiza alguna acción por propia iniciativa pero como respuesta a estímulos externos.
- Un DOMINIO dinámico es activo cuando realiza acciones sin estímulos externos. El usuario de un procesador de textos es un DOMINIO dinámico activo. Para un programa meteorológico, la atmósfera es un DOMINIO activo. Los cambios son espontáneos.
- UN DOMINIO es autónomo cuando sus acciones no pueden ser controladas a pesar que sus efectos puedan estar limitados por otros dominios.
- Un DOMINIO es programable cuando mediante programación se lo convierte en una *MAQUINA* de propósitos especiales necesaria para resolver el problema.
- Cuando el problema contiene un DOMINIO activo programable posible de ser programado, el mismo se convierte en parte de la *MAQUINA* que se está construyendo.

#### MARCO DE UN SISTEMA DE INFORMACIÓN SIMPLE

- Un Sistema de Información provee información, en respuesta a requerimientos sobre algún dominio de interés del mundo real.



- Sistema es la máquina que se debe construir.
- Mundo Real es el dominio conectado al sistema por medio de fenómenos compartidos.
- Información Entregada e Información Requerida están también conectados al Sistema.
- Función de Información son los Requerimientos en este marco. Es la relación requerida entre el Mundo Real, la información entregada y la información requerida.
- La información de salida debe reflejar precisamente el estado del mundo y también debe estar relacionada con el pedido de información, dado que es producida en respuesta a un requerimiento de información y porque su contenido es en parte determinado por el pedido.
- El mundo real puede ser un dominio estático o dinámico.
- El dominio información requerida es siempre un dominio dinámico activo, los pedidos ocurren en forma autónoma y espontánea

### **Análisis de la sesión**

El objetivo de conocer la terminología y conceptos fue logrado. Además se logró vislumbrar cuál es la problemática del trabajo de especificación de requerimientos, y el marco de problema para un sistema de información simple. Es imperativo entonces continuar con la extracción de conocimientos buscando en otros materiales que profundicen sobre marcos de problema, qué elementos considerar en un dominio, cómo subdividir dominios, etc.

<b>Sesión número:</b> 2
<b>Fecha:</b> 17/04/00
<b>IC:</b> Ing. Marcelo Rizzi
<b>Técnica Empleada:</b> Análisis estructural de texto.
<b>Texto Analizado:</b> Software Requirements, Objects, Functions & States, Alan M. Davis, Prentice Hall, 1993. ISBN: 0-13-805763-X
<b>Objetivos:</b> Profundizar el conocimiento sobre cómo dividir dominios, qué elementos se deben tener en cuenta en un dominio, obtener nuevos conceptos y agudizar el conocimiento sobre marco de problema para sistemas de información simple.

### **Conceptos extraídos**

Objetos  
 Funciones  
 Estados  
 Partición  
 Abstracción  
 Proyección  
 Requerimientos

### **Conocimientos Extraídos**

#### REQUERIMIENTOS

Condición o capacidad necesitada por un usuario para resolver un problema o llevar a cabo un objetivo.

Condición o capacidad que debe cumplir o poseer un sistema para satisfacer un contrato, estándar, especificación u otro documento formalmente impuesto.

Todo Requerimiento define un Objeto, una función o un estado.

Todo Requerimiento limita o controla las acciones asociadas con un objeto, función o estado.

Todo Requerimiento define relaciones entre objetos, funciones y estados.

#### OBJETO

Desde el punto de vista de los requerimientos, un objeto es cualquier entidad del mundo real, relevante para la discusión de los requerimientos del problema a resolver, con límites bien definidos.

#### FUNCIÓN



Desde el punto de vista de los requerimientos, una función es una tarea, servicio, proceso, función matemática o actividad que es actualmente llevada a cabo en el mundo real y/o a ser llevada a cabo por el sistema bajo especificación para resolver el problema en el mundo real.

#### ESTADO

Desde el punto de vista de los requerimientos, un estado es una condición de alguna cosa que captura algo de su historia y es utilizada por la misma para determinar como se encuentra en circunstancias específicas.

#### PARTICION

Principio que captura la relación estructural "parte-de" entre objetos, entre funciones o entre estados en el dominio del problema.

La finalidad de la partición es descomponer el problema en subproblemas para ayudar al análisis.

#### ABSTRACCION

Principio que captura la relación estructural "General/Específico" o "Ejemplo-de" o "Instancia-de" entre objetos, entre funciones o entre estados en el dominio del problema.

#### PROYECCION

Principio que captura la relación estructural "Visión-de" entre objetos, entre funciones o entre estados. En el dominio del problema. Permite observar el problema desde distintas perspectivas.

#### CLASIFICACIÓN DE PROBLEMAS

Un Problema se puede clasificar según:

- Dificultad
  - Difícil – No Difícil
  - Un problema es difícil si nunca antes fue resuelto, es nuevo o se desconoce o es inaplicable una solución.
- Relación de tiempo entre datos y procesamiento
  - Estático – Dinámico
  - Estático: Todos los datos de entrada están disponibles al programa previo a que comience su ejecución.
  - Dinámico: Los datos de entrada continúan arribando durante el procesamiento causando efectos sobre el programa.

- Número de tareas simultáneas a realizar
  - Secuencial – Paralelo
  - Secuencial: Se espera que la solución software realice solo una tarea a la vez.
  - Paralelo: Múltiples tareas simultáneamente.
  - Este atributo aplica solo a la solución software y no al entorno donde es contenida.
- Dificultad relativa del problema en sus aspectos de datos, control y algoritmos
  - Datos: Dificultad en la descripción, organización, definición y formato de los datos que se mueven a través de la interface entre el sistema y su entorno.
  - Control: Dificultad en la definición y descripción de cómo el entorno controlará al sistema o como el sistema controlará al entorno. Los Requerimientos de tiempo son estrictos.
- Determinista versus No determinista
  - Determinista: Se prevee que el sistema brindará las mismas respuestas todo el tiempo dadas las mismas entradas.
  - No Determinista: Las respuestas del sistema no son bien entendidas. El sistema es menos predecible. Usualmente tales sistemas intentan imitar el comportamiento humano.

### **Evaluación de la sesión**

Se obtuvieron nuevos e importantes conceptos sobre requerimientos, como así también una profundización en el enfoque a realizar sobre la información disponible al momento de comenzar la especificación de requerimientos, como lo son los objetos, funciones y estados, efectuando también abstracciones, proyecciones y particiones.

Se obtuvo una manera de clasificar el tipo de problema de la aplicación en 5 dimensiones ortogonales que permite luego abocarse a la especificación de requerimientos de una manera diferente según el problema. Sin embargo ésta sesión no deja en claro el objetivo de conocer más profundamente el marco de problema para un sistema de información simple.

Se necesita entonces una nueva sesión de extracción aplicada a documentos que traten marcos de problema.

<b>Sesión número:</b> 3
<b>Fecha:</b> 19/04/00
<b>IC:</b> Ing. Marcelo Rizzi
<b>Técnica Empleada:</b> Análisis estructural de texto.
<p><b>Texto Analizado:</b></p> <ol style="list-style-type: none"> <li>1. Problem Decomposition for Reuse. Michael &amp; Daniel Jackson. Paper obtenido de Carnegie Mellon University. January 2, 1995.</li> <li>2. Knowledge-intensive Software Engineering Tools. Mitsubishi Electric Research Laboratories. Rich &amp; Waters. Technical Report 91-03. 1991</li> <li>3. Requirements Engineering, A Survey of methods and tools. H. Hoffmann, Instituto de Informática, Universidad de Zurich.</li> </ol> <p>Requirement Assistant technical description, MIT</p>
<p><b>Objetivos:</b></p> <p>Profundizar el conocimiento sobre cómo dividir dominios, qué elementos se deben tener en cuenta en un dominio, obtener nuevos conceptos y agudizar el conocimiento sobre marco de problema para sistemas de información simple.</p>

### Conocimientos adquiridos

#### MARCO DE PROBLEMA ESTÁTICO DE INFORMACION

El *marco de problema* estático se compone de 5 partes:

- El *sistema*, máquina a ser construida
- El *dominio objeto*, es la parte estática del mundo no conectada al sistema. No posee eventos.
- Los *eventos asociativos* que forman un canal sin estructurar de eventos en el cual se definen asociaciones entre los fenómenos del *dominio objeto* y los fenómenos del *sistema*
- Los *Pedidos de Información* es un conjunto no estructurado de preguntas y respuestas asociadas sobre el *dominio objeto*.
- Las *reglas de información*, son las relaciones requeridas entre el *dominio objeto* y los *pedidos de información*.

La resolución de un problema estático de información requiere que el sistema tenga un modelo interno del dominio objeto. Los eventos asociativos inicializan este modelo y subsecuentemente el sistema lo utiliza pero no lo actualiza.

#### MARCO DE PROBLEMA DINAMICO DE INFORMACION

El *marco de problema* estático se compone de 4 partes:

- El *sistema*, máquina a ser construida

- El dominio objeto el cuales la parte dinámica del mundo directamente conectada al sistema. Es totalmente autónomo. Los eventos son comunes al sistema como al dominio objeto
- Los *Pedidos de Información* es un conjunto no estructurado de preguntas y respuestas asociadas sobre el *dominio objeto*.
- Las *reglas de información*, son las relaciones requeridas entre el comportamiento del *dominio objeto* y los *pedidos de información*.

Un problema de este tipo requiere que el sistema colecte y mantenga información sobre los eventos autonomos en el dominio objeto. Esta informacion es requerida para responder a los pedidos de información. Puede tomar la forma de un modelo elaborado del estado cambiante del dominio objeto o puede ser algo asi como un registro de eventos pasados seleccionados.

#### ASISTENTE INTELIGENTE

Un asistente inteligente, mas alla de simplemente aceptar y ejecutar comandos, debe chequear la razonabilidad de decisiones tomadas, completar detalles no tenidos en cuenta, y requerir asistencia sobre como ejecutar operaciones complejas.

Otra tarea de un asistente inteligente es la capacidad de explicar sus acciones y decisiones en términos entendibles por el ingeniero. Esto permite al ingeniero chequear qué ha hecho la herramienta. Además le permite a la herramienta describir el problema que ha encontrado cuando pide datos o decisiones.

#### BIBLIOTECA DE CLICHÉS

Es un repositorio declarativo de información relevante a requerimientos en general y a dominios de interés en particular.

El proceso de creación de requerimientos se nutre de información específica sobre el problema en particular provista por el ingeniero de software, y de información global sobre el dominio provista por la biblioteca de clichés.

Un cliché de sistema de información está compuesto de información común a sistemas de información sobre dominios tales como sistema de personal, bases bibliográficas, archiveros, control de inventario, etc.

El rol central de un sistema de información es un conjunto de reportes que muestran porciones de datos almacenadas y un conjunto de transacciones que crean, modifican o borran dichos datos.

### Análisis de la sesión

Se encontró que los expertos coinciden en describir las problemas de información como de dos tipos: estáticos y dinámicos, dependiendo del comportamiento del mundo real sobre el que se desea obtener información.

<b>Sesión número:</b> 4
<b>Fecha:</b> 21/04/00
<b>IC:</b> Ing. Marcelo Rizzi
<b>Técnica Empleada:</b> Análisis estructural de texto.
<b>Texto Analizado:</b> Problem Analysys Using Small Problem Frames, M.A. Jackson, Proceedings of WOFACS '98, South African Computer journal 22, Pages 47-60, Marzo 1999.
<b>Objetivos:</b> Obtener una clasificación detallada de cada uno de los marcos de problema elementales.

Conocimientos Adquiridos

Marcos de Problema Elementales:

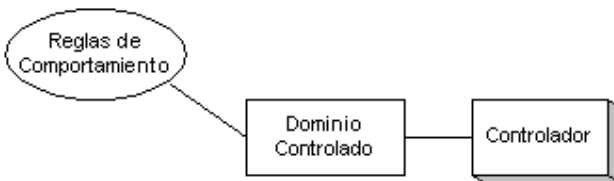
<b>Tipo de Requerimiento</b>	<b>Descripción</b>	<b>Marco de Problema</b>
Consultas	Requerimiento de información sobre alguna parte del dominio del problema	Información
Reglas de comportamiento	Reglas que debe seguir el comportamiento del dominio del problema	Control
Mapeos	Mapeos sobre datos de entrada y de salida del software	Transformación
Operaciones sobre dominios creados	Operaciones que realizan los usuarios sobre objetos que existen solo dentro del software	Workpieces
Correspondencias entre dominios	Mantenimiento de dominios que no poseen fenómenos compartidos en sus estados correspondientes.	Conexión

Estructura de cada marco

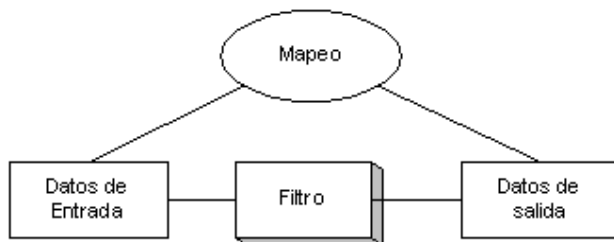
**Marco de problema de información**



**Marco de problema de control**



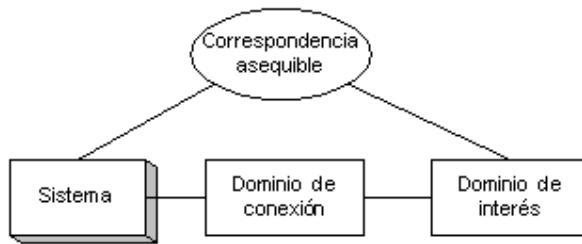
**Marco de problema de transformación**



**Marco de Problema Workpieces**



### Marco de problema de conexión



Tipo (a)



Tipo (b)

### Análisis de la sesión

Se obtuvo la estructura depurada y simplificada de cada uno de los marcos de problema elementales, luego de varias depuraciones.

<b>Sesión número:</b> 5
<b>Fecha:</b> 23/04/00
<b>IC:</b> Ing. Marcelo Rizzi
<b>Técnica Empleada:</b> Análisis estructural de texto.
<b>Texto Analizado:</b> Practical Software Requirements, Benjamin L. Kovitz, Manning Publications, 1999. ISBN: 1-884777-59-7
<b>Objetivos:</b> Profundizar sobre las técnicas de descripción de requerimientos y dominio del problema de acuerdo al marco de problema ajustado.

### Conocimientos Adquiridos

Para generar el Documento de Requerimientos debe obtenerse la siguiente información.

#### a) Problemas de Información:

- Objetos en el Mundo Real, sus atributos y relaciones
- Datos a ser almacenados sobre los objetos
- Todos los eventos del Mundo Real que cambian el resultado de las consultas, y todas las posibles secuencias en las que ocurren dichos eventos.

- Las Consultas
- Cómo puede el sistema acceder los objetos y eventos (O en el caso de un sistema de información estático, cómo pueden los desarrolladores accederlos).
- Formatos de archivo para archivos existentes que el sistema necesite acceder (o referir a documentación existente).
- distorsiones y retardos introducidos por cualquier dominio de conexión.

b) Problemas de control

- Objetos en el dominio controlado. Modelo de datos si correspondiere.
- Leyes causales del dominio controlado, incluídos los eventos de los objetos son capaces de generar.
- Reglas de Comportamiento
- Acciones en el dominio del problema que la computadora es capaz de iniciar.
- Fenómenos compartidos a través de los cuales la computadora puede monitorear el dominio controlado.

c) Problemas de transformación

- Conjuntos de entrada.
- Conjuntos de salida.
- Fuente y destino de los datos.
- Mapeos entre los conjuntos de entrada y salida.

d) Problemas de Piezas de Trabajo (Workpieces)

- La propia pieza de trabajo (mediante clases, atributos, etc.)
- Las operaciones en el dominio controlado describiendo los eventos que recibe y cómo responde a los mismos.

e) Problemas de Conexión

- Estados y eventos en el dominio de interés
- Redundancias en el dominio de interés
- Mapeos, actuales o deseados, entre estados y eventos entre los diferentes dominios
- Distorsiones y retardos introducidos por el dominio de conexión, reales o deseados.
- Reglas para determinar cuál de varios dominios de conexión tiene los datos más confiables.



Para los diferentes elementos a describir según cada marco de problema se debe describir como se indica a continuación:

#### Documentación de Clases:

- Nombre de la clase
- Una definición de que clase de cosas la clase puede contener
- Lista de atributos, su definición, posibles valores, significado de cada posible valor.
- Atributos que identifican unívocamente miembros de la clase
- Cuales clases se relacionan con la clase en cuestión.
- Cada evento, si hay, que afectan a miembros de la clase y cuales atributos y relaciones dichos eventos afectan.

#### Documentación de relaciones

- Nombre, su tipo, binaria, ternaria-aria, cardinalidad

#### Documentación de consultas

- Escribir la sentencia de la consulta. Tener en cuenta que se deben describir dos cosas: La información que el usuario ingresa y la información que le usuario recibe como respuesta.

#### Documentación de secuencias y eventos:

- Describir las secuencias en forma textual acompañadas de un diagrama de Jackson. En el texto debe describirse cada elemento de la secuencia y un ejemplo de la misma.
- Eventos: Describir los conjuntos afectados por los eventos: Clases, miembros, atributos, relaciones.
- Los parámetros de los eventos: Atributos que pueden variar de una instancia a otra del evento.
- Describir como el sistema puede detectar que el evento ha ocurrido y cuales son sus parámetros: Las fuentes de las fuentes de información.

#### Documentación de estados:

- Listado de todos los estados
- Para cada estado, que realiza el objeto durante ese estado, o cualquier diferencia externamente detectable sobre tal estado.
- Para cada estado cuales son los eventos posibles, y para cada evento posible, como responde el objeto mientras esta en dicho

estado: Cuales acciones el objeto realiza y el estado final luego del evento.

- Cualquier información adicional sobre el estado
- Cual es el estado inicial
- Cual estado o estados son los finales

#### Documentación de Acciones

- En un problema de control puede haber tres tipos de acciones a documentar:
  - Espontáneas: Aquellas iniciadas en el dominio del problema
  - Inmediatas: Las que le software puede iniciar directamente
  - Mediatas: Las que son causadas por otras acciones.
  - No son mutuamente exclusivas.
- Describir: El tipo de causa (espontánea, inmediata o mediata)
- Todos los tipos de objetos involucrados en la acción
- Los parámetros que tiene la acción
- En el caso indirecto las condiciones o eventos que disparan la acción
- La duración de la acción
- Los posibles resultados de la acción.
- Si hay mas de un posible resultado como el software detecta cual ocurrió.

#### Documentación de Interrupciones (Suceden en problemas de control)

- La naturaleza de la interrupción:
- Los efectos de la mima
- Los parámetros que pueden variar entre instancia e instancia de la interrupción.
- Como puede detectarse la interrupción
- Que acciones puede interrumpir la interrupción.
- Cómo responder a la interrupción

#### Documentación de Reglas y Mapeos

- Realizar siempre matrices con las reglas
- Mantener excepciones complicadas y definiciones fuera de la matriz en notas que la acompañan.

### **Evaluación de la sesión**

Esta sesión ha sido muy fructífera dado que para cada tipo de dominio [resente en loas marcos y para cada tipo de requerimiento, se tiene una lista de los aspectos a tener en cuenta para indagar en el dominio de la aplicación y en las necesidades de los usuarios y clientes y de éste modo plasmarlos en el documento de requerimientos.

En el apéndice C se transcriben las sesiones adicionales en idioma inglés de las entrevistas realizadas a expertos internacionales.

# Capítulo 6

## Conceptualización

## Capítulo 6

### Conceptualización

#### 6.1 Introducción

La conceptualización constituye la segunda etapa de la segunda fase de la Metodología IDEAL, la cual consiste básicamente en el entendimiento del dominio del problema y de la terminología usada, como así también la modelización de la tarea que lleva a cabo el experto a la hora de resolver el problema

Esta etapa permite al Ingeniero del Conocimiento conformar un marco inicial o mapa mental de los diferentes conocimientos del dominio que el experto utiliza durante la realización de su tarea

Una vez que ha sido identificado el dominio, el siguiente paso consiste en estructurar los conocimientos para modelizar el comportamiento del experto en la solución del problema que son de su competencia.

La conceptualización conlleva un proceso de estructuración de los conocimientos adquiridos y, se desarrolla en una etapa doble: la primera se corresponde con una actividad de **Análisis** y, la segunda de un trabajo de **Síntesis**.

La actividad de *Análisis* está basada en la identificación de conocimientos Estratégicos, Tácticos y Fácticos. El trabajo de **Síntesis** establece en mayor o menor medida cómo dichos conocimientos son transformados en el *Modelo Estático y Dinámico* que conforma el *Modelo Conceptual del Sistema*.

La conceptualización persigue a través del Análisis y el proceso de Síntesis, la obtención de un *Modelo Estático* y un *Modelo Dinámico*, que juntos modelarán el comportamiento del experto en lo que compete a la solución del problema.

En lo referente a la tarea de **Análisis** se persigue la identificación de tres tipos de conocimientos: serán estos *estratégicos, tácticos y fácticos*.

*Estratégicos*: los cuales especifican qué hacer, dónde y por qué hacerlo, es decir, los conocimientos estratégicos fijan la secuencia de pasos que se deberán seguir para ejecutar la tarea.

*Tácticos*: de acción u operativos, que especifican cómo y cuándo el Sistema Experto puede añadir a sus conocimientos genéricos información actual acerca del caso.

*Fácticos o declarativos*, que especifican lo que es, o se cree que es verdad acerca del mundo en general y acerca del caso particular para el cual se está ejecutando la tarea.

Una vez finalizado el análisis de los diferentes conocimientos: estratégicos, tácticos y fácticos, se pasa al trabajo de Síntesis. En este estadio se pretende establecer el *Submodelo Dinámico o de Procesos*. Recordando las tareas estudiadas durante la fase de análisis de conocimientos estratégicos se debe lograr definir la jerarquía entre tareas. Una vez logrado este esquema se intenta plasmar el *Submodelo Estático de Conocimientos* que está constituido por el *Diccionario de Conceptos*, la *Tabla de Concepto-Atributos-Valor* y el *Mapa de Conocimientos*.

El **Diccionario de Conceptos** albergará los diferentes conceptos, o subconjuntos funcionales, del más alto nivel, así como la terminología clave. Para cada uno de los conceptos se especificará su utilidad o función, sinónimos y acrónimos, los atributos que los definen, sus valores, y de dónde pueden ser derivados los datos, es decir, cuáles son las fuentes para obtener información sobre cada uno de los conceptos incluidos, como por ejemplo textos, bibliografía, informes, experiencia de alguna persona, procedimientos, etc. El diccionario de conceptos es una "lista viva" que sintetiza y sumariza todo lo que se obtiene por distintos métodos de adquisición de conocimientos.

Otro documento importante es la **Tabla Concepto-Atributo-Valor**, la cual, permite representar por cada concepto que atributos encierra y, de cada atributo especificar los diferentes valores que puede llegar a poseer a lo largo de la resolución del problema.

Por último uno de los elementos más importantes que resume el Modelo Dinámico y el Modelo Estático es el **Mapa de Conocimientos**, el cual conforma el *Modelo Conceptual Completo* del comportamiento del Experto.

## **6.2. Elaboración del modelo conceptual**

En el presente trabajo, luego de comenzadas las sesiones de adquisición de conocimientos, se ha comenzado a esbozar el modelo conceptual que nos permite entender la tarea que desempeña el experto.

En una primera instancia se realiza una Identificación, Categorización y Clasificación de los diferentes conceptos participes en la resolución del problema. Dichos conceptos son plasmados para su mejor entendimiento y descripción en el **Glosario de Términos, Diccionario de Conceptos y la Tabla de Concepto-Atributo-Valor.**

Luego se describen las relaciones obtenidas entre los diferentes conceptos.

Luego de la obtención de este material para el comienzo de la conceptualización, se ha profundizado el análisis de los conocimientos adquiridos y se ha continuado el ciclo de adquisición de conocimientos para poder redondear la descripción de **Conocimientos Estratégicos, Tácticos y Fácticos.**

Por último, todos los conocimientos representados son sintetizados en el **Modelo Dinámico y Modelo Estático de Conocimientos.**

#### **6.2.1. Identificación, Comparación y Categorización de Conceptos**

Al iniciar la Conceptualización se debe conformar un glosario de todos los términos que se utilizarán a lo largo del proyecto. Esto permite eliminar ambigüedades que pudiesen ocurrir al ser interpretados o manipulados determinados términos.

Para lograr una adecuada interpretación del **Glosario de Términos** se ha dedicado primordial interés en la calidad y exactitud de sus descripciones. En la siguiente tabla se observan los términos que son empleados en la resolución de la tarea por parte de los expertos.

Además el definir un adecuado glosario de términos ha permitido realizar con mayor precisión las sesiones de educación de conocimientos, tendientes a evaluar y refinar el Modelo Conceptual, en su primer etapa de gestación.

### 6.2.1.1 Glosario de Términos

El objetivo del mismo es lograr minimizar o eliminar las ambigüedades al interpretar o utilizar los términos.

A su vez permite profundizar en futuras sesiones de adquisición sobre aquellos huecos que pudieren quedar en la tarea del experto.

<b>Término</b>	<b>Descripción</b>
Acciones	Puede considerarse como dos o mas eventos concatenados, con una estructura de tiempo interna. Se diferencia de un evento dado que sucede en un intervalo de tiempo en vez de en un punto del tiempo.
Característica del dominio	Cuando se identifican las diferentes partes de un marco de problema es importante averiguar la característica de cada dominio que interviene en dicho marco. Ello ayudará a identificar el marco de problema y por consiguiente dónde se debe concentrar el analista a la hora de especificar los requisitos.
Contexto del problema	Es la parte del mundo donde la máquina será instalada, y en la cual sus efectos serán percibidos y evaluados.
Documento de requerimientos	Identificación detallada de la información que se necesita para realizar determinadas tareas o funciones
Documentación de salida	Descripción del negocio, organigrama, funciones del área, diagramas de contexto y nivel 1, enumeración de requerimientos.
Dominio	Es una porción particular del mundo que puede ser distinguida porque es convenientemente considerada como un todo, y puede ser considerada independientemente de otras partes del mundo.
Dominio creado	Es un dominio que no existe tangiblemente fuera del software y es, por lo tanto, creado dentro del mismo con el objeto de ser controlado.
Dominio de la aplicación	Es la parte del CONTEXTO DEL PROBLEMA en la cuál los efectos de la MAQUINA creada serán percibidos, evaluados y (si se tiene éxito) aprobados por el cliente o usuario.
Dominio de conexión	Dominio que comparte fenómenos con otros dos dominios causando que un dominio tenga una conexión indirecta con el otro.
Entidad (objeto)	Desde el punto de vista de los requerimientos, un objeto es cualquier entidad del mundo real, relevante para la discusión de los requerimientos del problema a resolver, con límites bien definidos.
Estado	Desde el punto de vista de los requerimientos, un estado es una condición de alguna cosa que captura algo de su historia y es utilizada por la misma para determinar como se encuentra en circunstancias específicas.



<b>Término</b>	<b>Descripción</b>
Eventos	Es la designación de hechos que involucran un punto en el tiempo. Son atómicos e instantáneos. Un evento representa una transición del mundo real de un estado a otro sin pasar por estados intermedios.
Fenómeno	Un fenómeno es aquello que aparenta existir, o estar presente o ser el caso cuando se observa el mundo o alguna parte del mismo en un dominio.
Fenómenos compartidos	Eventos o estados compartidos por dos o mas dominios. Por ejemplo el evento de vender un producto en el dominio vendedor es también el evento de comprar un producto en el dominio comprador. Es únicamente a través de los fenómenos compartidos que un dominio causa un efecto en otro dominio.
Función	Desde el punto de vista de los requerimientos, una función es una tarea, servicio, proceso, función matemática o actividad que es actualmente llevada a cabo en el mundo real y/o a ser llevada a cabo por el sistema bajo especificación para resolver el problema en el mundo real.
Información de entrada	Los materiales que se tienen al comienzo pueden ser listados, reportes, documentación de entrevistas a usuarios, encuestas, cuestionarios a usuarios, documentos de elicitación. En situaciones no es suficiente la documentación y es necesario la observación de tareas.
Máquina	Cuando se construye software se construye una MAQUINA. Se construye el comportamiento y las propiedades de la MAQUINA que la hará útil para algún propósito particular.
Marco de problema	-un marco de problema es una estructura consistente de actores principales y tareas para la solución.
Marco de problema de conexión	Se define sobre el problema a resolver por un software que simule fenómenos compartidos entre dos o mas dominios que están conectados por un dominio de conexión. Usualmente se encuentran junto a otros problemas.
Marco de problema de control	Se define sobre un problema en el que el software debe causar que un cierto dominio se comporte de una determinada manera.
Marco de problema dinámico de información	Se refiere a un marco de problema definido como una generalización del problema consistente en sistemas de información sobre entidades del mundo real que sufren variaciones continuas o discretas en su estado.
Marco de problema estático de información	Se refiere a un marco de problema definido como una generalización del problema consistente en Sistemas de Información sobre entidades del mundo real que no sufren cambios ni perturbaciones en el tiempo.
Marco de Problema de Transformación	Se define sobre un problema de mapeo de elementos de un conjunto de entrada a un conjunto de salida, tal como la conversión de archivos de un formato a otro, datos en gráficos, conversiones entre CMYK a RGB, etc.
Marco de Problema de workpieces	Se define sobre un problema donde un usuario/s puede crear y editar objetos que existen dentro del propio software que lo resuelve, tal como documentos o gráficos.
Prueba de Ajuste al marco	Son una serie de pruebas que debe realizar el experto par asegurarse que ha elegido correctamente el marco de problema, es decir, que el marco de problema se ajusta al problema que se desea resolver.

<b>Término</b>	<b>Descripción</b>
Relación	Un conjunto de Tuplas que mapean elementos de una entidad o clase a elementos de una o mas entidades.
Requerimiento	Condición o capacidad necesitada por un usuario para resolver un problema o llevar a cabo un objetivo. Condición o capacidad que debe cumplir o poseer un sistema para satisfacer un contrato, estándar, especificación u otro documento formalmente impuesto.

*Tabla 6.1: Glosario de términos*

La identificación de conceptos o subconceptos funcionales, del más alto nivel así como la terminología clave, se describen en un **Diccionario de Conceptos**, como se puede apreciar en la tabla 6.2.

Para cada uno de los conceptos del diccionario se especificará: Su utilidad o función, sinónimos y acrónimos, los atributos que lo definen, y de donde pueden derivarse los datos.

<b>Concepto</b>	<b>Función</b>	<b>Sinónimos/ Acrónimos</b>	<b>Elementos</b>	<b>Relaciones</b>
Dominio	Se utiliza para caracterizar una parte del dominio del problema dentro de un marco de problema y luego según su tipo proceder a su descripción.	Parte	Tipo Nombre Identificado	- Marco de problema (Parte-de)
Descripción del Dominio del problema	Descripción imprescindible en el documento de requerimientos construida a partir de la descripción de cada parte del tipo dominio de cada marco de problema.	Descripción del dominio de la aplicación, descripción del entorno del problema	- Descripción de dominios parte de un MP - Descripción de Fenómenos compartidos entre dominios	-Dominio (Compuesto-por) -Marco de problema (Describe-a) -Documento de requerimientos (parte-de)
Fenómenos Compartidos	Eventos o estados compartidos por dos o mas dominios. Se deben describir como parte del dominio del problema.	FC	Tipo Descripción Dominio1 Dominio2	-Descripción del dominio del problema.(Descrito-en) -Marcos de problema (Parte-de)

Concepto	Función	Sinónimos/ Acrónimos	Elementos	Relaciones
Marco de Problema	Estructura consistente de partes tales como: Dominios, Fenómenos compartidos, Requerimientos y la máquina.	MP	Nombre Partes Tipo	-Dominios (Compuesto-por) -Fenómenos compartidos (Compuesto-por) -Máquina (Compuesto-por) - Requerimientos (Compuesto-por) -Descripción dominio del problema. (Descripción-en)
Proyecto	Es el elemento contenedor de toda la información útil en pos de la solución al problema.	Solución, Descripción.	Nombre, Objetivo, Partes, Marcos, Documento de Requerimientos Aproximación descomposición Dificultad	-Marcos de problema (Compuesto-por) -Documento de Requerimientos (Compuesto-por)
Requerimientos	Efectos que se desea que la máquina produzca en el dominio del problema. Su descripción es parte fundamental del documento de requerimientos.	Requisitos.	Identificación Tipo Descripción	-Marco de problema (Parte-de) -Documento de requerimientos (Descrito-en)
Documento de Requerimientos	Es el producto resultante del sistema experto. Contiene la descripción del dominio del problema y de los requerimientos	Especificación de requerimientos.	Nombre de archivo, Proyecto Partes	-Requerimientos (Compuesto-de) Descripción del dominio del problema. (Compuesto-de)

Tabla 6.2: Diccionario de Conceptos.

### 6.2.1.2 Razones para la elección de los conceptos

Existe documentación al comienzo de la tarea, producto de la fase de elicitación de requerimientos, en donde mediante diferentes técnicas se ha obtenido información compilada en el documento de elicitación. Además se dispone de documentos de entrevistas con usuarios, bocetos de listados o reportes que los usuarios desearían obtener, listados o reportes de sistemas actuales, etc.

Con todo, el analista intentará obtener finalmente un documento detallando los *requerimientos* y la descripción del *dominio del problema*.

Para ello debe intentar identificar el problema a resolver y elegir uno o varios *marcos de problema* que se ajusten al mismo. Debe así mismo identificar cuál es el *dominio del problema*, aquel lugar donde el sistema hará sentir sus efectos y también donde se encuentra el problema a resolver. Es importante destacar que *la máquina* o sistema no es el problema a resolver, sino el medio para solucionarlo. Los *requerimientos* justamente deben describir el problema que debe resolverse.

Cada *marco de problema* se describe en función de sus partes, por lo que se debe identificar cada una de ellas, analizando qué tipo de *dominio* es cada una de ellas para luego proceder a su descripción, como también los *fenómenos compartidos* entre los distintos *dominios*. Todo esto da lugar a una completa *descripción del dominio del problema*. Luego se debe probar que el marco ha sido bien elegido o se ajusta al problema en cuestión.

Para resolver el problema se debe diseñar una *máquina* (sistema).

Completamos entonces el modelo estático con la Tabla Concepto/Atributo/Valor.

### 6.2.1.3 Tabla Concepto/Atributo/Valor

Concepto	Atributo	Valor
Dominio	Nombre	Mundo Real, Dominio Controlado, Datos de entrada, Datos de salida, Usuario, Pieza de trabajo, Dominio de interés, Conexión.
	Tipo	Estático, Dinámico, Activo, Reactivo, Inerte, Autónomo, Delegable, Programable, Tangible, Intangible, Unidimensional, Multidimensional
	Identificado	TRUE, FALSE
	Complejidad	Alta, Baja
Descripción del Dominio del problema	- Descripción de dominios parte de un Marco de problema	Descripción de clases, descripción de atributos, descripción de acciones, descripción de relaciones.
	- Descripción de fenómenos compartidos	Descripción de estados, descripción de eventos.
	- Completado	Si - No
Fenómenos Compartidos	Tipo	Estado, Evento
	Nombre	FC_C1, FC_C2, FC_C3, FC_E1, FC_E2, FC_E3, FC_H1
	Descripción	Clases afectadas, Origen de evento, parámetros de evento, Estado inicial, final, definición del estado.
	distorsion-retardo	Si, No
	Dominio1	Dominio
	Dominio2	Dominio

Concepto	Atributo	Valor	
Marcos de Problema	Nombre	Texto describiendo el marco en el problema analizado	
	Tipo	Información(MPSI), Control (MPSC), Transformación (MPTR), Conexión (MPCO),	
	Partes	Workpieces (MPWP) Mundo Real, Consultas, Máquina, Dominio Controlado, Regla de comportamiento, Datos de entrada, Datos de salida, Mapeos, Usuario, Operaciones en dominios creados Pieza de trabajo, Correspondencias aseguibles, Dominio de interés, Conexión.	
	FC_XX	Fenómenos compartidos (FC)	
Proyecto	Nombre	Texto Nombre del proyecto	
	Objetivo	Texto describiendo el objetivo del proyecto	
	Partes	Descripción Dominio de la Aplicación, Requerimientos	
	Documento	de Path y nombre del archivo que lo contiene.	
	Requerimientos		
	Aproximación	Interior-exterior, exterior-interior, marco compuesto conocido.	
	Descomposición		
	Dificultad	Identidad, conexión, Req-flexibles.	
Marcos	Lista de marcos que describen el proyecto. Puede ser elementales o uno compuesto.		
Requerimientos	Identificación	ID de texto que identifica unívocamente a cada requerimiento.	
	Tipo	Consultas, Reglas de Comportamiento, Mapeos, Operaciones sobre el dominio creado, Correspondencias aseguibles	
	Descripción	Texto libre que describe el requerimiento.	
	Completado	Si - No	
Documento	de	Nombre de archivo	Texto representando el path y el nombre del archivo
Requerimientos		Partes	Requerimientos, Descripción del Dominio del problema.

Tabla 6.3: Concepto/Atributo/Valor

### 6.2.2 Relaciones entre conceptos

A continuación se determina el segundo componente de la terna de conceptualización consistente en las relaciones entre concepto o conjunto relacional de base.

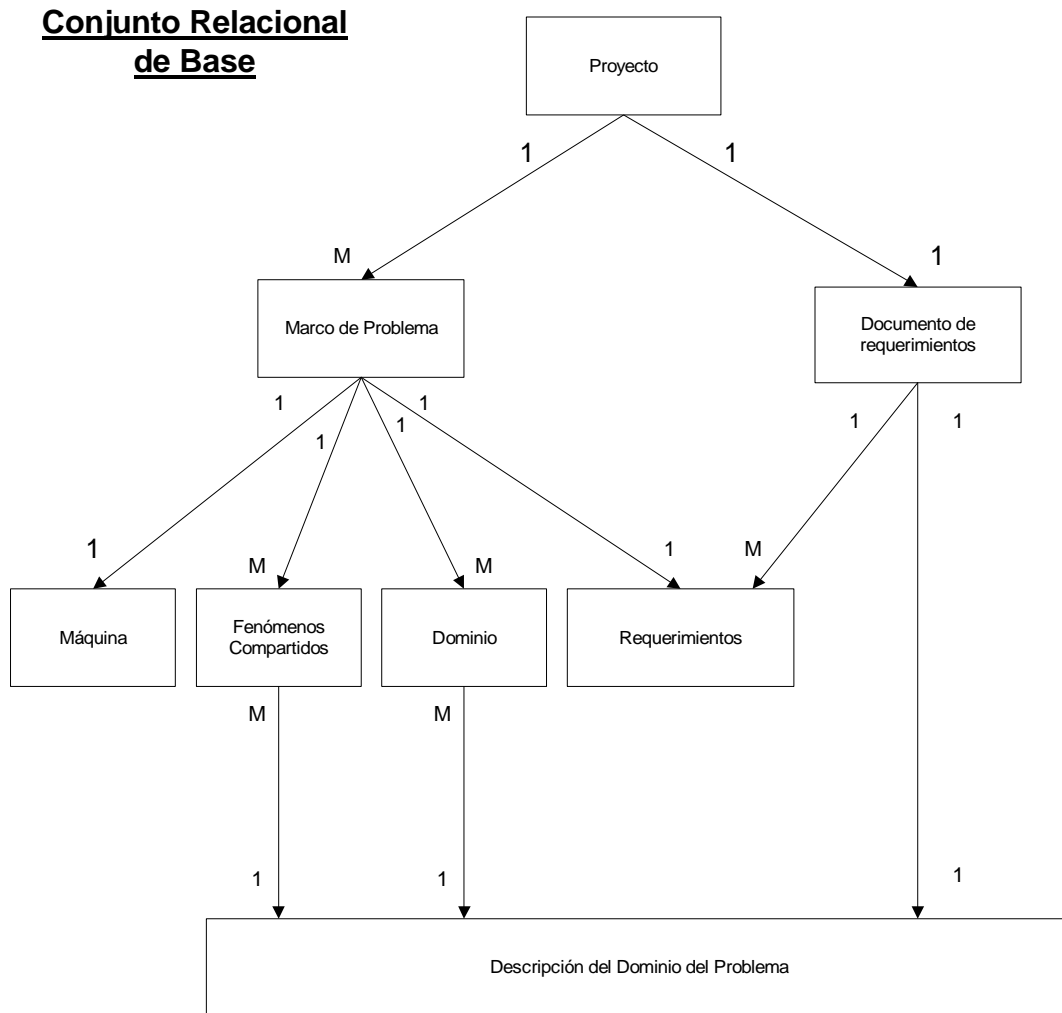


Figura 6.1: Conjunto Relacional de Base.

En el esquema podemos observar que *Proyecto* está compuesto de uno a muchos *Marcos de Problema* y un único documento de requerimientos.

A su vez, Marco de Problema posee cuatro dependientes: *Máquina* (Sistema software a construir) *Requerimientos*, *Dominios* y *Fenómenos compartidos*. Se necesitan la *Descripción del Dominio del problema* y los *Requerimientos* para conformar el *Documento de Requerimientos*.

La *Descripción del Dominio del problema* está formada por la descripción de muchos *Fenómenos Compartidos* y muchos *Dominios*.

### **6.2.3. Análisis de los conocimientos estratégicos**

Este análisis permite desarrollar una definición muy precisa de:

- a. Los pasos modulares que sigue el experto al desarrollar su tarea.
- b. El flujo de control que regirá el funcionamiento del sistema experto, de modo de que al realizar la síntesis se pueda realizar un reacomodamiento de pasos, subtareas, etc. en el caso de que resulte necesario.

En el esquema de la figura 6.2 se representa el modelo mental que siguen los expertos para ejecutar su tarea, en donde se han resaltado en trazos gruesos aquellos pasos que serán objeto del presente trabajo. El conjunto total de pasos serán descritos brevemente de modo de tener una idea general del modelo estratégico que los expertos poseen del problema.

Luego se describirán solo los pasos remarcados siguiendo la propuesta de Juan Pazos de representación, esto es, pasos de alto nivel, subpasos de la tarea y pasos de bajo nivel, mediante la descripción de la tarea, su propósito, las entradas necesarias para su concreción, el razonamiento que se aplica y las salidas obtenidas.



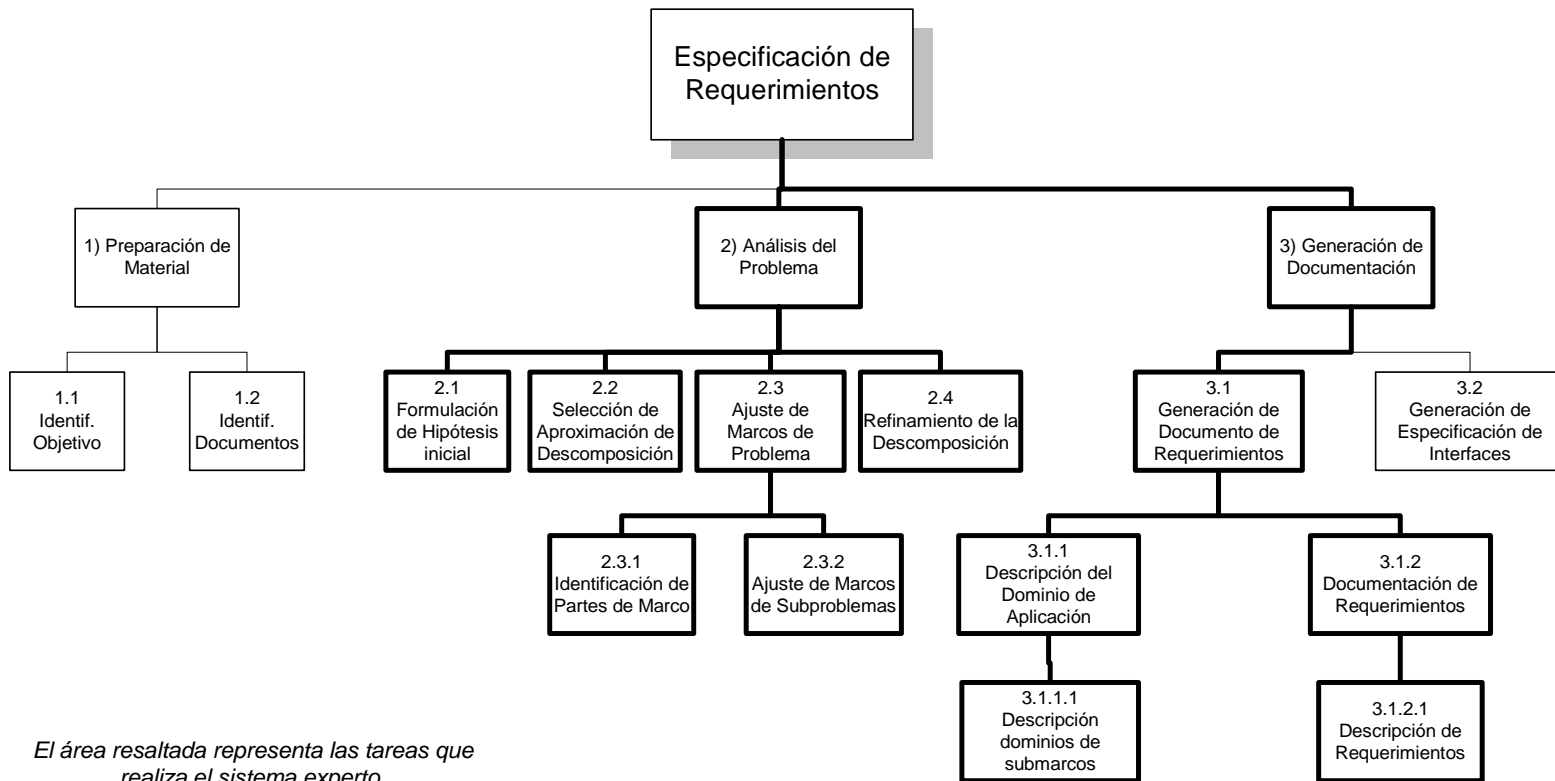


Figura 6.2 : Diagrama Jerárquico de tareas

### **6.2.3.1 Pasos de Alto nivel**

1. Preparación de material disponible
2. Análisis del Problema
3. Generación de Documentación

#### **Paso 1: Preparación de material disponible:**

##### *Propósito:*

En éste paso el experto se concentra en todo el material disponible al momento de comenzar la especificación.

##### *Subpasos:*

- 1.1: Identificación del objetivo del sistema
- 1.2: Identificación y catalogación de documentos de elicitación de requerimientos

##### *Entradas:*

La misma puede consistir en listados o reportes que el usuario desea que el sistema le provea, listados o reportes de sistemas existentes, documentos resultantes de la etapa de elicitación de requerimientos obtenidos vía técnicas como JAD, brainstorming, entrevistas, etc.

##### *Razonamiento:*

El experto revisa toda la documentación chequeando que sea lo más completa posible, y realiza en este momento una primera aproximación a la estructura general de la especificación, enfoques de especificación (por usuarios, por subsistemas, etc.), tipo de información requerida, etc.

##### *Salidas:*

Objetivo preciso del problema que el sistema a desarrollar pretende resolver, junto con los efectos principales que el sistema debe crear en el dominio donde será implantado, o con el que se relacionará.

Identificación detallada de toda la documentación disponible.

#### **Paso 2: Análisis del Problema**

##### *Propósito:*

Aquí el experto realiza una primera hipótesis acerca del problema al que se enfrenta tratando de razonar sobre el tipo básico de marco de problema que podría ajustarse. Luego tratará de hallar una de las tres aproximaciones posibles a la descomposición del problema si sucediere que no se ajusta a un marco de problema básico. Luego ,

identificará las diferentes partes del marco o de los marcos de problema involucrados, realizando un nuevo ajuste para verificar la certidumbre de la descomposición realizada.

#### Subpasos

2.1 Formulación Inicial de Hipótesis de marco de problema

2.2 Selección de la aproximación de descomposición

2.3 Ajuste de marcos de problema

2.4 Refinamiento de la descomposición

#### *Entradas:*

Descripción acabada del problema a resolver mediante software

Material clasificado de elicitación

Definición del objetivo del sistema

#### *Razonamiento:*

Mediante el estudio y la observación del problema y la ayuda de algunas heurísticas, el experto realizará la descomposición del problema en subproblemas y luego ajustará un marco elemental a cada subproblema para luego sintetizar todos los marcos en un marco general del problema.

Por ejemplo, en el caso de marco de problema de sistemas de información, debe buscar encontrar la parte del dominio que conforma el mundo real, la máquina, la información requerida y la entregada, las características del dominio que requiere la información y la función de información que vincula la información requerida, la información entregada y el mundo real, que constituye los requerimientos a especificar. Finalmente realiza la comprobación de ajuste a dicho marco.

#### *Salidas:*

Composición de las partes del marco general y de cada submarco.

### **Paso 3: Generación de la documentación**

El experto comienza entonces a analizar cada marco de subproblema, cada parte identificada dentro de dichos marcos y luego a describir las mismas según una serie de preguntas y pautas de modo que nada en el dominio de la aplicación quede sin describir y ningún requerimiento quede fuera de la documentación. Los marcos actúan como guías para que el experto aplique las técnicas de descripción adecuadas a cada

tipo de parte y se efectúe las preguntas necesarias para una correcta especificación de requerimientos.

Subpasos:

3.1 Descripción del dominio de aplicación

3.2 Documentación de los requerimientos del sistema

Razonamiento

El experto sistemáticamente comienza el recorrido por cada parte identificada, y teniendo en cuenta a que marco elemental pertenece, efectúa el análisis y la descripción de dicha parte no olvidando ningún detalle mediante la adhesión a ciertas guías de descripción de dominios. Dicha descripción del dominio consume gran parte del documento de requerimientos. Seguidamente continúa con la descripción de cada requerimiento del sistema, realizando los cuestionamientos correspondientes según el tipo de marco de subproblema al que pertenece la parte de requerimientos a describir.

Salidas

Documento de requerimientos finalizado.

### **6.2.3.2 Subpasos de la tarea**

#### **1.1: Identificación del objetivo del sistema**

*Entradas*

Documentación resultante del proceso de Elicitación.

*Razonamiento*

Aquí el experto analiza el objetivo propuesto por el usuario o cliente acerca del motivo preciso de la construcción de la solución software.

El experto debe poder identificar en primera instancia el tipo de problema a resolver para realizar su hipótesis a partir de dicho objetivo. No obstante se afianzará la misma con el análisis posterior.

No debe quedar lugar a dudas acerca del objetivo del sistema.

El tipo o clase de problema que debe resolverse con un sistema software tiene esta forma:

Configurar la Máquina M para producir el conjunto de efectos R en el dominio D

El tipo de problema debe limitarse a aquellos que al momento las personas saben cómo resolver, y cuyos efectos los desarrolladores saben como llevar a cabo.

Una cuestión que el experto se plantea previo a conocer nada acerca del dominio del problema es:

¿Qué beneficios desea el cliente que el software sea responsable de llevar a término?

Esto hace que se esté enfocado en los efectos mas importantes a ser ejecutados en el dominio del problema (informar a las personas acerca de algo, hacer que sucedan cosas de acuerdo a ciertas reglas, proveer resultados de ciertos cálculos, etc.).

#### *Salidas*

Objetivo explícito del problema a resolverse por medio de software.

### **1.2: Identificación y catalogación de documentos de elicitación de requerimientos**

#### *Entradas*

Documentación de elicitación completa

#### *Razonamiento*

El experto debe analizar toda la documentación reunida y clasificar la información que el usuario provee para poder luego ser contrastada con su análisis del dominio. Dicha clasificación se hace de acuerdo a un estándar de numeración, que servirá para la trazabilidad de los requerimientos.

#### *Salidas*

Documento índice de la documentación disponible al momento de comenzar el trabajo.

### **2.1 Formulación Inicial de Hipótesis de marco de problema:**

#### *Entradas*

Objetivo del sistema

Documentación catalogada de elicitación.

#### *Razonamiento*

Luego de la preparación del material el experto está en condiciones de realizar una hipótesis acerca de cuál marco de problema podría ajustarse al tipo de problema que está analizando. Teniendo en cuenta el objetivo primario del sistema a construir intentará identificar los principales componentes del problema, los requerimientos esbozados en el material de elicitación disponible, intentará asociar el problema a un

marco de problema elemental o bien a un marco de problema compuesto previamente resuelto por el experto. Para ello sigue los tres siguientes pasos:

**2.2 Selección de la aproximación de descomposición:** Habiendo analizado en una primera etapa el problema estará en condiciones de seleccionar una aproximación de descomposición, ya sea:

- a. De afuera hacia adentro
- b. De adentro hacia fuera
- c. A partir de un marco compuesto de un problema común o ya resuelto por el experto.

**2.3 Ajuste de marcos de problema:** Una vez seleccionado el método de descomposición el experto debe identificar marcos y partes de marcos en el problema para corroborar la metodología de descomposición requerida o bien cambiarla y finalmente compaginar las partes y marcos identificados en un único diagrama que luego será chequeado de modo de verificar si todo tiene cabida, si nada importante ha quedado fuera. Con el marco definido el experto se concentra en la parte que indica el mismo donde debe realizarse la especificación. Esto asegura que la especificación se realice sobre el problema a resolver en forma concreta y no se desvíe hacia aspectos de la máquina u otros aspectos no relevantes del dominio.

Entonces debe:

- a. Identificar partes o marcos principales
- b. Identificar marcos restantes

#### **2.4 Refinamiento de la descomposición**

Una vez analizado el problema surge que pueden quedar ciertas partes del problema sin ajustar correctamente, normalmente se corresponden con dificultades que se identificarán y ajustarán posteriormente a un marco elemental de problema, además el experto busca unificar los diferentes marcos de subproblema en un único marco para el problema global de modo que le sirva como referencia en futuros proyectos similares. En el proceso se identificarán partes comunes que juegan un papel diferentes para los diferentes marcos a los cuales pertenecen.

- a. Identificar dificultades inherentes y luego ajustarlas a marcos elementales
- b. Unificar partes comunes que juegan un papel diferente en cada marco

- c. Proceder a la síntesis de un marco general para propósitos de reutilización, si correspondiere.

### *Salidas*

Problema totalmente enmarcado, ya sea en un marco compuesto o bien en un marcos elemental si la simplicidad del problema lo permite. Esta salida puede ser una descripción del marco, o puede ser el diagrama de marco correspondiente.

### **3.1.1 Generación de documentación: Descripción del dominio de aplicación**

#### ***Entradas***

Descripción y/o diagrama de marco de problema

#### *Razonamiento*

Es entonces que el experto debe analizar el dominio de la aplicación buscando e identificando las diferentes entidades relevantes que lo componen utilizando las técnicas de abstracción, proyección y partición, los fenómenos del dominio, los atributos de dichas entidades y toda otra información que sea accesoria como por ejemplo unidades de medida de dichos atributos, formatos de salida, métodos de almacenamiento, etc., las relaciones entre las mismas, las causalidades entre entidades, eventos, estados, acciones, etc. Toda esta información servirá para que el experto la contraste con el material disponible al comienzo, con el objeto de verificar si toda la información disponible es entregada al usuario, o bien si éstos han olvidado algún dato importante del dominio para sugerirlo durante un refinamiento de la elicitación, si la información solicitada por los usuarios es coherente y asequible a partir del dominio, etc.

En la documentación de elicitación está toda la información, funcionalidades, etc. solicitadas por los usuarios. Pero puede que tras el análisis realizado sobre el dominio, existan datos, conceptos o atributos que hayan sido pasados por alto y que sean útiles. Es por eso que se debe hacer este contraste en busca de información faltante, o poco clara. Puede que el usuario solicite una información de maestro-detalle pero que en el dominio del mundo real tal relación uno-muchos no haya podido ser identificada. Esto obliga a revisar la elicitación o bien a consultar a los usuarios.

### *Salidas*

Documento con la descripción completa y revisada del Dominio de la Aplicación. Dicha descripción puede ser reutilizable para otros sistemas a construir sobre el mismo dominio.

Reporte de información a ser educada en un nuevo ciclo de elicitación de requerimientos.( En caso de ser necesario).

### **3.1.2 Generación de documentación: Documentación de requerimientos**

#### *Entradas*

Descripción y/o diagrama de marco de problema

Descripción del dominio de aplicación

#### *Razonamiento*

El experto comienza a especificar los requerimientos. El objetivo es que todos los requerimientos de los usuarios plasmados en la elicitación tengan su correlato en la especificación. Es importante que sea trazable mediante la codificación numérica de ambos documentos.

Se efectúan iteraciones (otorgando precisión y agregando nuevas cláusulas). Se agregan aquellos requerimientos surgidos a partir del análisis del dominio y se agrega precisión a las especificaciones en cuanto a formato de los datos, atributos que intervienen en relaciones, organización de la presentación de los reportes, etc. El proceso se repite hasta que no hay información adicional para agregar o cuestionar al usuario.

Se revisa finalmente el documento obtenido atendiendo no solo a la precisión y completitud de la especificación sino también a la organización, requerimientos de no comportamiento, conformidad a algún estándar solicitado por el cliente, etc.

### *Salidas*

Documento con la descripción y definición de cada uno de los requerimientos del problema.

### **6.2.3.3 Comprobación de los conocimientos estratégicos**

El modelo jerárquico ha sido validado por los expertos. Se han realizado sesiones informales en las cuales se evaluó la representación de conocimientos que intervienen en la tarea. Se continúa entonces con el análisis de los conocimientos tácticos.



#### **6.2.4. Análisis de los conocimientos tácticos**

Estos conocimientos especifican cómo el sistema experto utiliza hechos conocidos y las hipótesis actuales acerca del caso para obtener nuevos hechos e hipótesis tanto en situaciones deterministas como en condiciones de incertidumbre.

Se debe detallar cómo se realizan los diferentes pasos identificados en el razonamiento estratégico.

##### **6.2.4.1 Representaciones intermedias**

Se han utilizado diferentes representaciones intermedias en función del tipo de conocimiento que se desea detallar, de modo de otorgar un mayor entendimiento al problema.

Se ha decidido utilizar las seudorreglas como representación de la información que se desea conceptualizar en cuanto a los conocimientos tácticos adquiridos.

Como puede observarse, primeramente se han elaborado las tablas para cada regla en la etapa de conceptualización mediante la formulación externa de la regla, y luego se representará la misma una vez codificada en la herramienta en la fila llamada Formulación en la herramienta.

Las seudorreglas descritas corresponden a las siguientes decisiones tácticas:

- a. Cómo decidir el tipo de aproximación que se utilizará para descomponer el problema. (Correspondiente a tarea estratégica 2.2: Selección de la aproximación de descomposición)
- b. Cómo formular la hipótesis del marco de problema adecuado (Correspondiente a tarea estratégica 2.1: Formulación inicial de hipótesis de marco de problema)
- c. Cómo corroborar si cada marco se ajusta al problema (Correspondiente a tarea estratégica 2.3: Ajuste de marcos de problema)
- d. Cómo reconocer las dificultades en la aproximación interior-exterior de descomposición. (Correspondiente a tarea estratégica 2.4: Refinamiento de la descomposición)
- e. Cómo detectar posibles problemas de conexión (Correspondiente a tarea estratégica 2.4: Refinamiento de la descomposición)
- f. Cómo inferir el tipo de información que se necesita describir en cada parte de cada marco. Dado que cada marco de problema está compuesto por los dominios que conforman el dominio de la aplicación y también por los requerimientos, entonces estas reglas corresponden a la tarea estratégica

3.1.1: Generación de documentación: Descripción del dominio de aplicación y cómo describir cada tipo de requerimiento encontrado. (Correspondiente a tarea estratégica 3.1.2: Generación de documentación: Documentación de requerimientos)

g. Generación del documento de requerimientos

Se verá a continuación la representación de cada seudoregla.

**6.2.4.1.a:** Cómo decidir el tipo de aproximación que se utilizará para descomponer el problema.

*Abreviaturas utilizadas en las reglas:*

FC: Fenómenos Compartidos

MP: Marco de Problema

MPSI: Marco de Problema de Información

MPSC: Marco de Problema de Control

MPWP: Marco de Problema Worpieces

MPTR: Marco de Problema de Transformación

MPCO: Marco de Problema de Control

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>A veces el problema parece incluso no encajar ni siquiera aproximadamente en ningún marco conocido. Puede ser entonces útil descomponer el problema trabajando del exterior hacia el interior. La aproximación aquí es intentar encontrar partes o aspectos del problema reconocibles que correspondan a los marcos conocidos, y analizarlos en el contexto de esos marcos. Entonces ellos pueden considerarse como problemas resueltos, y las partes y aspectos del problema original que deben todavía ser resueltos pueden ser considerados sin la complicación adicional del subproblema ya resuelto.</i>
Formulación externa de la regla	SI Proyecto.Marcos = Marco-elemental ENTONCES Proyecto.Aproximación-descomposición = Exterior-interior
Nombre de la regla	Rdescomponer

*Tabla 6.4: Aproximación descomposición Exterior-interior*

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>A veces el problema parece encajar un marco conocido aproximadamente, pero exhibe dificultades que frustran la pura aplicación del marco. Estas dificultades dan lugar a subproblemas que puede ser reconocibles como que se ajustan a otros marcos en si mismos.</i>
Formulación externa de la regla	SI Proyecto.Marcos <> (Marco-elemental Or Marco-compuesto-conocido) ENTONCES Proyecto.Aproximación-descomposición = Interior-Exterior
Nombre de la regla	rDescomponer1

*Tabla 6.5: Ajustar Dificultades en aproximación interior-exterior.*

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>una técnica totalmente desarrollada deberá tener un juego rico de marcos compuestos. Puede esperarse que una parte sustancial de un problema real, o incluso, el problema completo, encajará en un marco compuesto conocido.</i>
Formulación externa de la regla	SI Proyecto.Marcos = Marco-compuesto-conocido ENTONCES Proyecto.aproximación-descomposición = Marco-compuesto-conocido
Nombre de la regla	RDescomponer2

*Tabla 6.6: Aproximación descomposición por reuso de problema resuelto.*

### 6.2.4.1.b: Cómo formular la hipótesis del marco de problema adecuado

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>Habiendo analizado la documentación de entrada, si se observa que los usuarios realizan preguntas, consultas, pedidos de información, sin que en ningún caso se pretenda modificar el mundo real, entonces estamos en condiciones de establecer la hipótesis de que el marco de problema adecuado es sistema de información.</i>
Formulación externa de la regla	SI Requerimientos.Tipo = consultas ENTONCES Proyecto.Marco = MP de Información
Nombre de la regla	AgregarMPIinformación

Tabla 6.7: Establecer Hipótesis de marco de problema de información

Se puede inferir entonces las otras reglas para determinar si se trata de otro tipo de marco

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>Si el problema requiere que el software controle un dispositivo haciendo que cumpla una serie de reglas de comportamiento estamos frente a un problema de control.</i>
Formulación externa de la regla	SI Requerimientos.Tipo = reglas-de-comportamiento ENTONCES Proyecto.Marco = MP Control
Nombre de la regla	AgregarMPControl

Tabla 6.8: hipótesis de marco de problema de Control

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>Habiendo analizado la documentación de entrada, si se observa que los usuarios requieren una herramienta para crear y editar artefactos intangibles tales como texto o gráficos entonces estamos en presencia de un problema Workpieces.</i>
Formulación externa de la regla	SI Requerimientos.Tipo = operaciones-entornos-creados ENTONCES Proyecto.Marco = MP Workpieces
Nombre de la regla	AgregarMPWorkpieces

*Tabla 6.9: hipótesis de marco de problema de Workpieces*

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>Habiendo analizado la documentación de entrada, si se observa que los usuarios requieren que se realicen ciertos cálculos o transformaciones en una colección de entrada para obtener una colección diferente y correspondiente a cada entrada entonces estamos en presencia de un problema de transformación.</i>
Formulación externa de la regla	SI Requerimientos.Tipo = mapeos ENTONCES Proyecto.Marco = MP Transformación
Nombre de la regla	AgregarMPTransformacion

*Tabla 6.10: hipótesis de marco de problema de transformación*

#### **6.2.4.1.c:** Cómo corroborar si cada marco se ajusta al problema

En este punto tenemos reglas para identificar las distintas partes que componen un marco de problema (tarea 2.3.1) y reglas para verificar que a partir de los elementos que se identifican, el marco seleccionado se ajusta al problema del mundo real que se está analizando.

**6.2.4.1.c.1** Reglas que se utilizan en la tarea 2.3.1

<b>Estado de la Regla</b>	<b>Texto de la Regla</b>
Palabras del experto	<i>Una vez elegido el marco de problema que se desea ajustar, y sus partes han sido identificadas, se debe verificar si existen los fenómenos compartidos necesarios para completar dicho marco. Para marcos de problemas de información hay que preguntarse: "Existen Fenómenos Compartidos entre la Máquina y el Mundo Real controlados por éste último?" "Existen Eventos Compartidos entre los dominios Máquina y Requirientes de Información controlados por éstos últimos?" "Existen Eventos entre los dominios Máquina y Respuestas controlados por la Máquina?"</i>
Formulación externa de la regla	SI Proyecto.Marco.tipo= Información ENTONCES preguntar(MPSI.FC-H1, MPSI.FC-E1, MPSI.FC-E2)
Nombre de la regla	FenómenosCompartidosPI

*Tabla 6.11: Fenómenos Compartidos en problemas de Información*

<b>Estado de la Regla</b>	<b>Texto de la Regla</b>
Palabras del experto	<i>Para marcos de problemas de control hay que preguntarse: "Existen Fenómenos Controlables en el Dominio Controlado?" "Existen Fenómenos Compartidos Controlados por la Máquina?" "Existen Fenómenos Compartidos Controlados por El Dominio Controlado?"</i>
Formulación externa de la regla	SI Proyecto.Marco.tipo = Control ENTONCES preguntar(MPSC.FC-C1, MPSC.FC-C2, MPSC.FC-C3)
Nombre de la regla	FenómenosCompartidosPC

*Tabla 6.12: Fenómenos Compartidos en problemas de Control*

<b>Estado de la Regla</b>	<b>Texto de la Regla</b>
Palabras del experto	<p><i>Para marcos de problemas de Workpieces hay que preguntarse: "Existe una Cadena de Eventos Producidos por el Dominio Operaciones en su interface con el Dominio Herramienta?"</i></p> <p><i>"Existen Eventos Generados por el dominio Herramienta en su interface con el dominio Workpieces?"</i></p> <p><i>"Existen Cambios de Estado en el Dominio Workpieces como consecuencia de eventos generados por Herramienta?"</i></p>
Formulación externa de la regla	<p>SI Proyecto.Marco.tipo = Workpieces  ENTONCES preguntar(MPWP.FC-E1,  MPWP.FC-E2, MPWP.FC-S1)</p>
Nombre de la regla	FenómenosCompartidosWP

*Tabla 6.13: Fenómenos Compartidos en problemas de Workpieces*

<b>Estado de la Regla</b>	<b>Texto de la Regla</b>
Palabras del experto	<p><i>Para marcos de problemas de Conexión hay que preguntarse:</i></p> <p><i>"Existen Fenómenos Compartidos entre la Máquina y el Dominio de Conexión?"</i></p> <p><i>"Existen Fenómenos Compartidos entre el Dominio de Conexión y el Dominio de Interés?"</i></p>
Formulación externa de la regla	<p>SI Proyecto.Marco.tipo = Conexión  ENTONCES preguntar(MPCO.FC-C1,  MPCO.FC-C2)</p>
Nombre de la regla	FenómenosCompartidosCO

*Tabla 6.14: Fenómenos Compartidos en problemas de Conexión*

**6.2.4.1.c.2** Reglas que se corresponden a la tarea 2.3.2, ajuste de los marcos de subproblemas.

<b>Estado de la Regla</b>	<b>Texto de la Regla</b>
Palabras del experto	<i>También se debe evaluar en el marco elegido, cada uno de los dominios intervinientes en cuanto a si el tipo de dominio se corresponde con el que le asigna el marco de problema al cual pertenece. En el caso de la Parte Mundo Real debe ser un dominio del tipo Autónomo o Estático.</i>
Formulación externa de la regla	SI Dominio.Nombre = MundoReal AND Dominio.Tipo = Autonomo Xor Estático ENTONCES Dominio.identificado=TRUE
Nombre de la regla	DomMundoReal

*Tabla 6.15: análisis de dominios de la parte MundoReal*

<b>Estado de la Regla</b>	<b>Texto de la Regla</b>
Palabras del experto	<i>La Parte DominioControlado debe ser un dominio del tipo Tangible y Autónomo o bien solamente Reactivo.</i>
Formulación externa de la regla	SI Dominio.Nombre = domControlado AND Dominio.Tipo = (Autonomo And Tangible) Or Reactivo ENTONCES Dominio.identificado=TRUE
Nombre de la regla	DomControlado

*Tabla 6.16: análisis de dominios de la parte Dominio Controlado*



Estado de la Regla	Texto de la Regla
Palabras del experto	<i>La Parte Solicitantes de Información debe ser un dominio del tipo Autónomo o bien Delegable o bien Programable.</i>
Formulación externa de la regla	Dominio.Nombre=SolicitantesInformación And Dominio.Tipo = Autonomo Xor Programable Xor Delegable ENTONCES Dominio.identificado=TRUE
Nombre de la regla	DomSolicitantesInformacion

*Tabla 6.17: análisis de dominios de la parte Solicitantes de Información*

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>La Parte Conjunto de datos de entrada debe ser un dominio del tipo Autónomo.</i>
Formulación externa de la regla	SI Dominio.nombre = datosentrada AND Dominio .tipo = Autonomo ENTONCES Dominio.identificado=TRUE
Nombre de la regla	DomDatosEntrada

*Tabla 6.18: análisis de dominios de la parte Datos de Entrada*

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>La Parte Conjunto de datos de salida debe ser un dominio del tipo Inerte.</i>
Formulación externa de la regla	SI Dominio.nombre = datosSalida AND Dominio.tipo = inerte ENTONCES Dominio.identificado=TRUE
Nombre de la regla	DomDatosSalida

*Tabla 6.19: análisis de dominios de la parte Datos de Salida*

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>La Parte Usuario del marco de Workpieces debe ser un dominio del tipo Delegable y Autónomo.</i>
Formulación externa de la regla	SI Dominio.nombre = Usuario and Dominio.Tipo = Autonomo And Delegable and Parte-de(Usuario,Workpieces) ENTONCES Dominio.identificado=TRUE
Nombre de la regla	DomUsuarioWP

Tabla 6.20: análisis de dominios de la parte Usuario de Workpieces

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>La Parte Usuario de cualquier otro marco de problema debe ser un dominio del tipo Autónomo.</i>
Formulación externa de la regla	SI Dominio.nombre = Usuario AND Dominio.Tipo= Autonomo ENTONCES Dominio.identificado=TRUE
Nombre de la regla	DomUsuario

Tabla 6.21: análisis de dominios de la parte Usuarios

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>La Parte Pieza de Trabajo debe ser un dominio del tipo Inerte o en su defecto Reactivo.</i>
Formulación externa de la regla	SI Dominio.nombre = Piezatrabajo AND Dominio.Tipo= Inerte Xor Reactivo ENTONCES Dominio.identificado=TRUE
Nombre de la regla	DomPiezadeTrabajo

Tabla 6.22: análisis de dominios de la parte Piezas de trabajo

**6.2.4.1.d:** Durante el refinamiento de la descomposición, si se optó por la aproximación interior-exterior, el experto busca por las dificultades que circundan el marco elegido.

Pueden ser dificultades de conexión o de identidades.

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>Por ejemplo, una forma de dificultad es una dificultad de conexión: puede ser alguna información necesitada por la máquina no está directamente disponible cuando se necesita. Puede ser entonces posible definir la dificultad como un subproblema de de información en el que la máquina original juega el papel de requiriente de información</i>
Formulación externa de la regla	SI Proyecto.Marco.Tipo = Información AND FC.distorsion-retardo = SI ENTONCES Proyecto.dificultad = conexión Proyecto.Marco = Información
Nombre de la regla	dificultad-conexión

Tabla 6.23: Análisis de dificultades de conexión.

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>Otro tipo de dificultad es una dificultad de identidades en la que la máquina comparte un juego de eventos o estados (fenómenos) con el dominio del problema pero no comparte los papeles asociados que identifican las entidades del dominio participantes. Aquí cabe la aplicación de uno o mas marcos de transformación para mapear entidades entre si. Otra dificultad es la necesidad de requerimientos flexibles, cuando un requerimiento fijo es inapropiado. Los usuarios suelen desear que una secuencia de estados sea modificable por los usuarios. Aquí cabe un marco workpieces.</i>
Formulación externa de la regla	SI Proyecto.Marco.Tipo = (Control OR Información) ENTONCES Proyecto.dificultad = identidad Proyecto.Marco = Transformación
Nombre de la regla	Dificultad-Identidad

Tabla 6.24: Análisis de dificultades de identidad.

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>Otra dificultad es la necesidad de requerimientos flexibles, cuando un requerimiento fijo es inapropiado. Los usuarios suelen desear que una secuencia de estados sea modificable por los usuarios. Aquí cabe un marco workpieces.</i>
Formulación externa de la regla	SI Requerimientos.Tipo = Operaciones-sobre-dominio-creado ENTONCES Proyecto.marco = MPWP Proyecto.dificultad = req-flexible
Nombre de la regla	Dificultad-Rflexibles

Tabla 6.25: Análisis de dificultades de requerimientos flexibles.

#### 6.2.4.1.e: Cómo detectar posibles problemas de conexión

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>Los fenómenos compartidos son la esencia de la interacción y comunicación entre dominios de cualquier clase. Si lo que se observa desde dos dominios se considera como un mismo fenómeno se deben cumplir las siguientes dos precondiciones: Primero la correspondencia entre la observación en los dos dominios debe ser confiable. Segundo si el fenómeno compartido es un evento, debe estar suficientemente sincronizado entre ambos dominios, es decir no debe existir retardo significativo. Cuando ambas precondiciones de velocidad y confiabilidad no se cumplen, se debe reconocer la presencia de otro dominio: El dominio de conexión.</i>
Formulación externa de la regla	SI FC.Dominio1 = MundoReal OR FC.Dominio1 = DomControlado AND FC.Dominio1 <> FC.Dominio2 OR (FC.Tipo = evento AND FC.Distorsion-Retardo = Si) ENTONCES Proyecto.marco = MPCO
Nombre de la regla	RIDConexión

Tabla 6.26: Detección de problemas de conexión

**6.2.4.1.f:** Cómo inferir el tipo de información que se necesita describir en cada parte de cada marco

Una vez descompuesto el problema se debe proceder a describir el dominio de la aplicación y los requerimientos. Para ello hay diferentes metodologías a seguir en función de cada marco de problema. Las reglas que se describen en este apartado contribuyen a efectuar en paralelo las tareas mostradas en el diagrama jerárquico de la figura 6.2 identificadas como 3.1.1.1 (Descripción de dominios en los submarcos) y 3.1.2.1 (Descripción de los requerimientos).

Estado de la Regla	Texto de la Regla
Palabras del experto	<p><i>Cuando el problema se ajusta a un marco de información se debe documentar lo siguiente:</i></p> <p><i>Objetos en el mundo real, sus atributos y relaciones</i></p> <p><i>Todos los eventos en el mundo real que cambian el resultado de las consultas, y todas las posibles secuencias en las cuales tales eventos pueden ocurrir.</i></p> <p><i>Consultas</i></p> <p><i>Como el sistema puede acceder a tales objetos y eventos</i></p> <p><i>Distorsiones y retardos introducidos por cualquier dominio de conexión.</i></p>
Formulación externa de la regla	<p>SI marco.tipo = MPSI ENTONCES</p> <p>ShowWindow(Clases)</p> <p>ShowWindow(Eventos)</p> <p>ShowWindow(Consultas)</p>
Nombre de la regla	DescribirMPInformación

*Tabla 6.27: Descripción de problemas de información*

<b>Estado de la Regla</b>	<b>Texto de la Regla</b>
Palabras del experto	<p><i>Cuando el problema se ajusta a un marco de control se debe documentar lo siguiente:</i></p> <p><i>Objetos en el dominio controlado (modelo de datos si existiese.</i></p> <p><i>Leyes causales del dominio controlado, incluyendo los eventos que pueden ser generados por dichos objetos.</i></p> <p><i>Reglas de comportamiento</i></p> <p><i>Acciones que la computadora es capaz de iniciar en el dominio del problema.</i></p> <p><i>Fenómenos compartidos a través de los cuales la computadora puede monitorear el dominio controlado.</i></p> <p><i>Cualquier dominio de conexión que exista.</i></p>
Formulación externa de la regla	<p>SI marco.tipo = MPSC ENTONCES</p> <p>ShowWindow(Clases)</p> <p>ShowWindow(ReglasComportamiento)</p> <p>ShowWindow(Acciones)</p>
Nombre de la regla	DescribirMPControl

*Tabla 6.28: Descripción de problemas de Control*

<b>Estado de la Regla</b>	<b>Texto de la Regla</b>
Palabras del experto	<p><i>Cuando el problema se ajusta a un marco de transformación se debe documentar lo siguiente:</i></p> <p><i>Conjuntos de entrada y de salida</i></p> <p><i>Fuente y destino de los datos.</i></p> <p><i>Mapeos entre los conjuntos de entrada y salida</i></p>
Formulación externa de la regla	<p>SI marco.tipo = MPTR ENTONCES</p> <p>ShowWindow(DataEntrada-Salida)</p> <p>ShowWindow(Mapeos)</p>
Nombre de la regla	DescribirMPTransformación

*Tabla 6.29: Descripción de problemas de Transformación*

<b>Estado de la Regla</b>	<b>Texto de la Regla</b>
Palabras del experto	<i>Cuando el problema se ajusta a un marco de Workpieces se debe documentar lo siguiente: Descripción de las piezas de trabajo Descripción de las operaciones deseadas.</i>
Formulación externa de la regla	SI marco.tipo = MPWP ENTONCES ShowWindow(Clases) ShowWindow(Operaciones)
Nombre de la regla	DescribirMPWorkpieces

*Tabla 6.30: Descripción de problemas Workpieces*

<b>Estado de la Regla</b>	<b>Texto de la Regla</b>
Palabras del experto	<i>Cuando el problema se ajusta a un marco de Conexión se debe documentar lo siguiente: Estados y eventos en el dominio de interés Redundancias en el dominio de interés Mapeos reales o deseados, entre estados y eventos en diferentes dominios Distorsiones y retardos introducidos por el dominio de conexión, reales o deseados. Reglas para determinar cual de varios dominios de conexión tienen la información más confiable.</i>
Formulación externa de la regla	SI marco.tipo = MPCO ENTONCES ShowWindow(Estados) ShowWindow(Eventos) ShowWindow(Distorsionesretardos)
Nombre de la regla	DescribirMPConexión

*Tabla 6.31: Descripción de problemas de Conexión*

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>En un problema de transformación o bien en correspondencias de dominios suele ser necesario describir estructuras o secuencias, como por ejemplo formato de archivos o cadenas de datos. Una manera de describirlos es mediante el uso de diagramas. Los diagramas de Jackson se utilizan cuando la estructura es básica y consta de jerarquías, secuencias, iteratividad y selección.</i>
Formulación externa de la regla	SI Dominio.Nombre = datos_entrada and datos_entrada.TipoEstructura = Basica ENTONCES Datos.Entrada.TipoDiagrama = Jackson
Nombre de la regla	DiagramaJackson

*Tabla 6.32: Descripción de entradas mediante diagramas de Jackson*

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>Los diagramas Sintácticos se utilizan cuando la estructura además de ser básica como la de Jackson presenta características de Recursión.</i>
Formulación externa de la regla	SI Dominio.nombre = datos_entrada And datos_entrada =TipoEstructura = Basica And Recursion ENTONCES DatosEntrada.tipoDiagrama= Sintáctico
Nombre de la regla	DiagramaSintáctico

*Tabla 6.33: Descripción de entradas mediante diagramas Sintácticos*



Estado de la Regla	Texto de la Regla
Palabras del experto	<i>Los diagramas Warnier-Orr se utilizan cuando la estructura además de ser básica y recursiva presenta concurrencia.</i>
Formulación externa de la regla	SI Dominio.nombre = datos_entrada and datos_entrada.TipoEstructura = Basica And Recursion and Concurrencia ENTONCES DatosEntrada.RipoDiagrama= Warnier-Orr
Nombre de la regla	DiagramaWarnier-Orr

*Tabla 6.34: Descripción de entradas mediante diagramas de Warnier-Orr*

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>Los diagramas de Flujo se utilizan cuando la estructura además de ser básica, no presenta muchas ramificaciones y posee un flujo de pocos ciclos.</i>
Formulación externa de la regla	SI Dominio.nombre = datos_entrada and datos_entrada.TipoEstructura = Basica And Simple ENTONCES DatosEntrada.TipoDiagrama= FlowChart
Nombre de la regla	DiagramaFlowChart

*Tabla 6.35: Descripción de entradas mediante diagramas de Flujo*

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>Los diagramas Adhoc se utilizan cuando la estructura no presenta características que se ajusten a los tipos de diagramas de estructura conocidos pero que se puede representar exactamente con un diagrama a medida.</i>
Formulación externa de la regla	SI Dominio.nombre = datos_entrada and datos_entrada.TipoEstructura = SuiGeneris ENTONCES DatosEntrada.TipoDiagrama = Adhoc
Nombre de la regla	DiagramaAdhoc

*Tabla 6.36: Descripción de entradas mediante diagramas Adhoc*

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>En un problema de transformación o bien en correspondencias de dominios suele ser necesario describir estructuras o secuencias, como por ejemplo formato de archivos o cadenas de datos. Una manera de describirlos es mediante el uso de diagramas. Los diagramas de Jackson se utilizan cuando la estructura es básica y consta de jerarquías, secuencias, iteratividad y selección.</i>
Formulación externa de la regla	SI Dominio.nombre = datos_Salida and datos_salida.TipoEstructura = Basica ENTONCES Datos.Salida.TipoDiagrama = Jackson
Nombre de la regla	DiagramaJackson1

Tabla 6.37: Descripción de Salidas mediante diagramas de Jackson

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>Los diagramas Sintácticos se utilizan cuando la estructura además de ser básica como la de Jackson presenta características de Recursión.</i>
Formulación externa de la regla	SI Dominio.nombre = datos_Salida and datos_salida.TipoEstructura = Basica And Recursion ENTONCES Datos Salida.tipoDiagrama= Sintáctico
Nombre de la regla	DiagramaSintáctico1

Tabla 6.38: Descripción de Salidas mediante diagramas sintácticos

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>Los diagramas Warnier-Orr se utilizan cuando la estructura además de ser básica y recursiva presenta concurrencia.</i>
Formulación externa de la regla	SI Dominio.nombre = datos_Salida and datos_salida.TipoEstructura = Basica And Recursion and Concurrencia ENTONCES DatosSalida.TipoDiagrama= Warnier-Orr
Nombre de la regla	DiagramaWarnier-Orr1

*Tabla 6.39: Descripción de Salidas mediante diagramas de Warnier-Orr*

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>Los diagramas de Flujo se utilizan cuando la estructura además de ser básica, no presenta muchas ramificaciones y posee un flujo de pocos ciclos.</i>
Formulación externa de la regla	SI Dominio.nombre = datos_Salida and datos_salida.TipoEstructura = Basica And Simple ENTONCES DatosSalida.TipoDiagrama= FlowChart
Nombre de la regla	DiagramaFlowChart1

*Tabla 6.40: Descripción de Salidas mediante diagramas de flujo*

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>Los diagramas Adhoc se utilizan cuando la estructura no presenta características que se ajusten a los tipos de diagramas de estructura conocidos pero que se puede representar exactamente con un diagrama a medida.</i>
Formulación externa de la regla	SI Dominio.nombre = datos_Salida and datos_salida.TipoEstructura = SuiGeneris ENTONCES DatosSalida.TipoDiagrama = Adhoc
Nombre de la regla	DiagramaAdhoc1

*Tabla 6.41: Descripción de Salidas mediante diagramas Adhoc*

#### **6.2.4.1.g** Generación del documento de requerimientos

Esta regla se aplica cuando se han completado las descripciones correspondientes al dominio de la aplicación y los requerimientos.

Estado de la Regla	Texto de la Regla
Palabras del experto	<i>Si se han descrito todos las partes del tipo dominio en cada marco de problema identificado y los fenómenos compartidos asociados, y los requerimientos de cada marco de problema han sido correctamente identificados y descritos, estamos en condiciones de generar el documento formal de requerimientos en su primera versión.</i>
Formulación externa de la regla	SI Descripción-Dominio-del-Problema.completado = Si AND Requerimientos.completado = SI ENTONCES Proyecto.Documento-de-Requerimientos = Documento-de-requerimientos.nombre-Archivo Generar-Documento(Proyecto.Documento-de-requerimientos)
Nombre de la regla	Generar-Documento

Tabla 6.42: Descripción de Salidas mediante diagramas Adhoc

### Metaconocimientos tácticos

En ocasiones el experto observa que los fenomenos compartidos entre dos dominios del marco de problema bajo análisis, sufren de retardos y/o distorsiones. Esto implica que existe un problema de conexión, situación ante la cual, el experto elige diferente camino en función de la importancia que le asigna a los dominios afectados por dichos retardos y/distorsiones. Generalmente la causa es la presencia de un dominio de conexión entre ambos dominios analizados. Según el grado de distorsión y la magnitud del retardo introducidos por el dominio de conexión, sumado a la complejidad del mismo y del dominio de interés, decide por alguna de las opciones que se indican en el árbol de decisión de la figura 6.3.

A partir de dicho árbol de decisión surgen las reglas "rAgregar-MPCO" de la tabla 6.42 y la regla "rAgregar-Dominio-conexión" de la tabla 6.43.

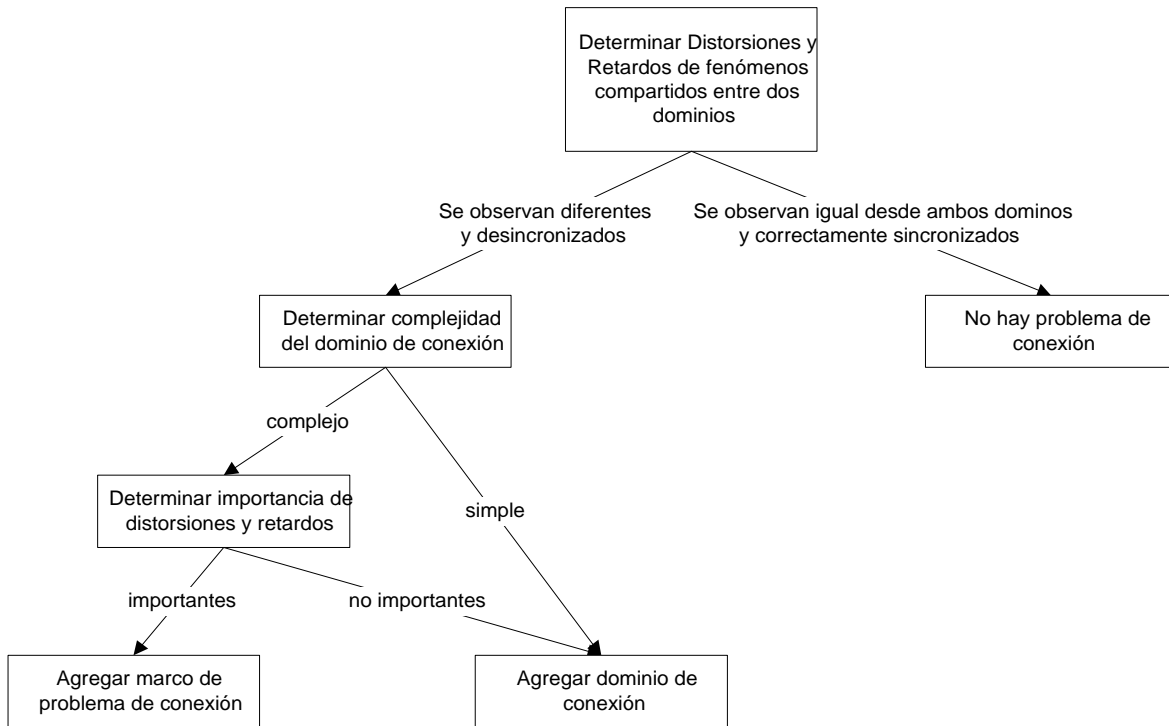


Figura 6.3: Arbol de decisión de retardos y/o distorsiones

Estado de la Regla	Texto de la Regla
Palabras del experto	1. Para proceder con un problema de conexión usted puede aceptar que el dominio de conexión y el dominio al otro lado del mismo son ambos importantes. Su interrelación es importante de modo de considerarla como un marco de problema. Utilizar entonces el marco de problema de conexión.
Formulación externa de la regla	SI FC.Distorsion-retardo = Si AND (FC.dominio1.tipo= conexión) ENTONCES Proyecto.marco = MPCO
Nombre de la regla	rAgregar-MPCO

Tabla 6.43: Agregado de marco de problema de conexión

Estado de la Regla	Texto de la Regla
Palabras del experto	<p><i>Para proceder con un problema de conexión usted puede:</i></p> <p><i>1. Ignorar el dominio de conexión y tratar los fenómenos de cada lado como compartidos, en desmedro de conocer que no lo son del todo. La desventaja es que se pierde la dimensión de lo que está sucediendo en el mundo real.</i></p> <p><i>2. Tratar el dominio de conexión como si fuera el dominio de interés. De ésta forma se ignora la existencia del mundo real</i></p>
Formulación externa de la regla	<p>SI FC.distorsion-y-retardo = Si  AND Dominio.tipo=conexion And  Dominio.complejidad=Alta ENTONCES  Proyecto.marco.parte = Conexion</p>
Nombre de la regla	rAgregar-Dominio-conexión

Tabla 6.44: Agregado de dominio de conexión.

### 6.2.5 Análisis de conocimientos fácticos

Los conocimientos fácticos del experto contienen información que el SE conocerá a priori acerca del área de la aplicación, así como información que el sistema obtendrá acerca del caso específico al ejecutar su tarea.

Los atributos generales los usa el SE como datos de entrada, conclusiones o resultados de salida.

<b>Información</b>	<b>Descripción</b>
<b>Atributo</b>	Dominio.Tipo
<b>Definición</b>	Es la característica que define a un dominio según su comportamiento en el tiempo, su dimensión, su naturaleza, su actividad.
<b>Tipo de Valor</b>	Cadena de caracteres
<b>Rango de valores</b>	estático, dinámico, activo, reactivo, inerte, unidimensional,multidimensional,tangible, intangible, autónomo, delegable, programable.
<b>Número de valores por caso</b>	Varios. En cada análisis se estudian varios dominios.
<b>Uso</b>	Para categorizar cada parte del marco de problema. Con ello se consigue: a) Verificar si el tipo de dominio de la parte del marco en cuestión es el correcto, de lo contrario el análisis no ha sido correcto, o bien falta información. b) Realizar la documentación del dominio utilizando las metodologías correspondientes a su tipo y marco de problema.
<b>Formato resultados de salida</b>	Texto
<b>Material de soporte</b>	Sesión de educación mediante emparrillado.

*Tabla 6.45: Descripción del atributo Tipo de dominio.*

Información	Descripción
<b>Atributo</b>	Dominio.Nombre
<b>Definición</b>	Es el nombre genérico que se le asigna a un dominio que forma parte de un marco de problema
<b>Tipo de Valor</b>	Cadena de caracteres
<b>Rango de valores</b>	Mundo Real, Dominio Controlado, Datos de entrada, Datos de salida, Usuario, Pieza de trabajo, Dominio de interés, Conexión.
<b>Número de valores por caso</b>	uno.
<b>Uso</b>	Se debe distinguir cada dominio por su nombre.
<b>Formato resultados de salida</b>	Texto
<b>Material de soporte</b>	Sesión de extracción #4 y #5

*Tabla 6.46: Descripción del atributo dominio.nombre*

Información	Descripción
<b>Atributo</b>	Dominio.Identificado
<b>Definición</b>	ES el valor que indica si el dominio ha sido debidamente identificado como parte del marco de problema mediante el test de ajuste de marcos.
<b>Tipo de Valor</b>	Booleano
<b>Rango de valores</b>	TRUE, FALSE
<b>Número de valores por caso</b>	uno.
<b>Uso</b>	Si todos los dominios de un marco de problema estan identificados como TRUE, entonces el marco esta ajustado correctamente.
<b>Formato resultados de salida</b>	Texto
<b>Material de soporte</b>	Sesión de extracción #5

*Tabla 6.47: Descripción del atributo dominio.identificado*



<b>Información</b>	<b>Descripción</b>
<b>Atributo</b>	Descripción Dominios de MP
<b>Definición</b>	Es la descripción de cada uno de los dominios que forman parte de los marcos de problema del proyecto y que constituye la descripción del dominio del problema.
<b>Tipo de Valor</b>	Cadena de caracteres
<b>Rango de valores</b>	[Descripción de Clases, Descripción de Atributos, Descripción de Relaciones, Descripción de Acciones]
<b>Número de valores por caso</b>	Varios. Puede haber múltiples dominios a describir en un problema.
<b>Uso</b>	La descripción sistemática de cada dominio es luego recolectada y formateada para conformar la "Descripción del dominio del problema", parte del documento de requerimientos.
<b>Formato resultados salida</b>	Texto
<b>Material de soporte</b>	Sesión de extracción #3

*Tabla 6.48: Descripción del atributo Descripción dominios de un MP*

<b>Información</b>	<b>Descripción</b>
<b>Atributo</b>	Descripción de Fenómenos Compartidos
<b>Definición</b>	Es la descripción de cada uno de los Fenómenos compartidos que forman parte de los marcos de problema del proyecto y que constituye la descripción del dominio del problema.
<b>Tipo de Valor</b>	Cadena de caracteres
<b>Rango de valores</b>	[Descripción de Eventos, Descripción de Estados, Distorsiones y retardos]
<b>Número de valores por caso</b>	Varios. Puede haber múltiples FC a describir en un problema.

Información	Descripción
<b>Uso</b>	La descripción sistemática de cada FC es luego recolectada y formateada para conformar la "Descripción del dominio del problema", parte del documento de requerimientos.
<b>Formato resultados de salida</b>	Texto
<b>Material de soporte</b>	Sesión de extracción #3

*Tabla 6.49: Descripción del atributo "Descripción de Fenómenos Compartidos"*

Información	Descripción
<b>Atributo</b>	Tipo de Marco de Problema
<b>Definición</b>	Es la característica que define la naturaleza de un marco de problema, de modo de representar un problema real bajo un modelo que permite analizarlo metodológicamente y separar los requerimientos del dominio de la aplicación y del software a construir para solucionar dicho problema.
<b>Tipo de Valor</b>	Cadena de caracteres
<b>Rango de valores</b>	[Información, Control, Conexión, Workpieces, Transformación]
<b>Número de valores por caso</b>	uno. Cada marco de problema tiene un único tipo.
<b>Uso</b>	Para representar el problema del mundo real, donde el sistema a construir debe hacer sentir sus efectos. Con el marco de problema identificado se procede a separar las partes del dominio de los requerimientos y a documentar cada uno de ellos mediante metodologías y cuestionamientos propios de cada marco.
<b>Formato resultados de salida</b>	Texto
<b>Material de soporte</b>	Sesión de extracción #4 y #5

*Tabla 6.50: Descripción del atributo Tipo Marco de Problema*

<b>Información</b>	<b>Descripción</b>
<b>Atributo</b>	Partes de Marco de Problema
<b>Definición</b>	Son los componentes, actores o individuos que forman un marco de problema. Corresponden a uno de los tres siguientes tipos: Requerimientos, Dominio de la aplicación o máquina.
<b>Tipo de Valor</b>	Cadena de caracteres
<b>Rango de valores</b>	[ Consultas, Regla de comportamiento, Mapeos, Operaciones en dominios creados, Correspondencias asequibles, Mundo Real, Dominio Controlado, Datos de entrada, Datos de salida, Usuario, Pieza de trabajo, Dominio de interés, Conexión, Respuesta, Máquina, Fenómeno Compartido]
<b>Número de valores por caso</b>	Uno. Cada parte pertenece a un único tipo
<b>Uso</b>	Permite identificar el tipo de marco de problema, y verificar que se ajuste al problema real.
<b>Formato resultados de salida</b>	Texto
<b>Material de soporte</b>	Sesión de extracción #5 y #6

*Tabla 6.51: Descripción del atributo Partes de marco de problema.*

Información	Descripción
<b>Atributo</b>	Nombre de Marco de Problema
<b>Definición</b>	Es un identificador que se le asigna a un marco de problema de acuerdo al subproblema que pretende reflejar.
<b>Tipo de Valor</b>	Cadena de caracteres
<b>Rango de valores</b>	Cadena libre de 20 caracteres
<b>Número de valores por caso</b>	Uno. Cada marco tiene un nombre
<b>Uso</b>	Permite identificar el marco de problema en el análisis y la descomposición del problema real.
<b>Formato resultados de salida</b>	Texto
<b>Material de soporte</b>	Sesión de extracción #5 y #6

*Tabla 6.52: Descripción del atributo Nombre de marco de problema.*

Información	Descripción
<b>Atributo</b>	Tipo de Requerimiento
<b>Definición</b>	Es la naturaleza común que tienen los efectos que los usuarios desean se produzcan en el dominio de la aplicación.
<b>Tipo de Valor</b>	Cadena de Caracteres
<b>Rango de valores</b>	[Consultas, Reglas de Comportamiento, Mapeos, Operaciones sobre el dominio, Correspondencias asequibles]
<b>Número de valores por caso</b>	Uno. Cada requerimiento es de un solo tipo.
<b>Uso</b>	Tiene dos usos: a) Para identificar primariamente el marco de problema de cada subproblema y b) Para documentar los requerimientos con los cuestionamientos adecuados a cada tipo.
<b>Formato resultados de salida</b>	Texto
<b>Material de soporte</b>	Sesión de extracción #5 y #6

*Tabla 6.53: Descripción del atributo Tipo de requerimiento.*

Información	Descripción
<b>Atributo</b>	Identificación de Requerimiento
<b>Definición</b>	Es el código de identificación (ID) para cada requerimiento de modo de lograr la trazabilidad de requerimientos en el proyecto de desarrollo.
<b>Tipo de Valor</b>	Cadena de Caracteres
<b>Rango de valores</b>	R#, donde # indica un numero entero secuencial.
<b>Número de valores por caso</b>	Uno. Cada requerimiento tiene un único ID.
<b>Uso</b>	Para documentar los requerimientos unívocamente en el documento de requerimientos.
<b>Formato resultados de salida</b>	Texto
<b>Material de soporte</b>	Sesión de extracción #5 y #6

*Tabla 6.54: Descripción del atributo Identificación de requerimiento.*

Información	Descripción
<b>Atributo</b>	Descripción de Requerimiento
<b>Definición</b>	Es una sentencia que define el requerimiento en términos del problema, como un efecto que la máquina debe causar en el dominio.
<b>Tipo de Valor</b>	Cadena de Caracteres
<b>Rango de valores</b>	Texto libre describiendo el requerimiento.
<b>Número de valores por caso</b>	Uno. Cada requerimiento tiene una descripción
<b>Uso</b>	Para documentar los requerimientos con los cuestionamientos adecuados a cada tipo.
<b>Formato resultados de salida</b>	Texto
<b>Material de soporte</b>	Sesión de extracción #5 y #6

*Tabla 6.55: Descripción del atributo Descripción de requerimiento.*

<b>Información</b>	<b>Descripción</b>
<b>Atributo</b>	Nombre de Archivo (Documento de requerimientos)
<b>Definición</b>	Nombre en formato DOS (8.3) del nombre de archivo con extensión HTM, donde se guardará el documento resultante del SE.
<b>Tipo de Valor</b>	Cadena de Caracteres (8 máx.)
<b>Rango de valores</b>	Cualquier combinación respetando la nomenclatura DOS.
<b>Número de valores por caso</b>	Uno.
<b>Fuente</b>	Provisto por el usuario.

*Tabla 6.56: Descripción del atributo Nombre de archivo del documento de requerimientos.*

<b>Información</b>	<b>Descripción</b>
<b>Atributo</b>	Partes (Documento de requerimientos)
<b>Definición</b>	Descripciones que conforman el documento de requerimientos
<b>Tipo de Valor</b>	Cadena de Caracteres
<b>Rango de valores</b>	[Descripción dominio del problema, Descripción de requerimientos]
<b>Número de valores por caso</b>	Uno.
<b>Fuente</b>	Obtenido a partir de los atributos descripción de dominio y descripción de requerimientos.

*Tabla 6.57: Descripción del atributo Partes del documento de requerimientos*

Información	Descripción
<b>Atributo</b>	Aproximación de descomposición (Concepto Proyecto)
<b>Definición</b>	Indica el tipo de descomposición que se realizará sobre el problema real para identificar los marcos de problema constitutivos.
<b>Tipo de Valor</b>	Cadena de Caracteres
<b>Rango de valores</b>	[Exterior-interior, Interior-Exterior, Compuesto-conocido]
<b>Número de valores por caso</b>	Uno.
<b>Fuente</b>	Provisto por el usuario luego de analizar con ayuda de ciertas heurísticas el problema.

*Tabla 6.58: Descripción del atributo aproximación de descomposición del concepto proyecto*

Información	Descripción
<b>Atributo</b>	Dificultad (Concepto Proyecto)
<b>Definición</b>	Cuando se selecciona la aproximación Interior-Exterior, pueden presentarse dificultades para la aplicación de un marco elemental. El tipo de dificultad se puede sortear mediante un marco apropiado. Este atributo lo define.
<b>Tipo de Valor</b>	Cadena de Caracteres
<b>Rango de valores</b>	[Identidad, Conexión, Requerimientos flexibles]
<b>Número de valores por caso</b>	Varios. Un proyecto puede exhibir mas de una dificultad para aplicar un marco elemental.
<b>Fuente</b>	Provisto por el usuario una vez que seleccionó la aproximación de descomposición Interior-Exterior.

*Tabla 6.59: Descripción del atributo Dificultad del concepto Proyecto*

Información	Descripción
<b>Atributo</b>	Nombre (Concepto Proyecto)
<b>Definición</b>	Nombre que se asigna en el proceso de desarrollo al proyecto.
<b>Tipo de Valor</b>	Cadena de Caracteres
<b>Rango de valores</b>	Texto libre
<b>Número de valores por caso</b>	Uno.
<b>Fuente</b>	Provisto por el usuario.

*Tabla 6.60: Descripción del atributo Nombre del Proyecto*

Información	Descripción
<b>Atributo</b>	Objetivo (Concepto Proyecto)
<b>Definición</b>	Texto que describe someramente el objetivo perseguido al intentar resolver el problema. Delimita claramente el contexto y el alcance de la solución.
<b>Tipo de Valor</b>	Cadena de Caracteres
<b>Rango de valores</b>	Texto libre
<b>Número de valores por caso</b>	Uno.
<b>Fuente</b>	Provisto por el usuario.

*Tabla 6.61: Descripción del atributo Objetivo del Proyecto*

Información	Descripción
<b>Atributo</b>	Tipo (Concepto Fenómenos Compartidos)
<b>Definición</b>	Tipo de FC que vincula a dos dominios en un marco de problema.
<b>Tipo de Valor</b>	Cadena de Caracteres
<b>Rango de valores</b>	[Estado, Evento]
<b>Número de valores por caso</b>	Uno.
<b>Fuente</b>	Provisto por el usuario.

*Tabla 6.62: Descripción del atributo Tipo de Fenomeno Compartido*



Información	Descripción
<b>Atributo</b>	Dominio1/2 (Concepto Fenómenos Compartidos)
<b>Definición</b>	Primer/Segundo Dominio que interviene en el análisis de fenómenos compartidos para detectar la presencia de un dominio o marco de conexión.
<b>Tipo de Valor</b>	Cadena de Caracteres
<b>Rango de valores</b>	[Mundo Real, Dominio Controlado, Máquina]
<b>Número de valores por caso</b>	Uno.
<b>Fuente</b>	Provisto por el usuario.

Tabla 6.63: Descripción del atributo Dominio1/2 del concepto Fenómenos Compartidos

### 6.2.6. Modelo Dinámico o de Proceso

Una vez finalizado el análisis de los distintos conocimientos, estratégicos, tácticos y fácticos, que componen la etapa de análisis de la conceptualización, se debe dar comienzo a la etapa de Síntesis.

El estudio de las tareas obtenidas en la etapa de análisis permiten elaborar *el Modelo dinámico o de proceso* que lleva a cabo el experto y comprobar que no hay demasiados errores ni olvidos. Para ello se recuerda cada tarea estudiada durante la etapa de análisis de conocimientos estratégicos y luego se define una jerarquía entre las tareas. Posteriormente derivar todos los formalismos necesarios para la ejecución de la tarea, por ejemplo, metas, atributos para el comienzo de cada tarea.

Finalmente el proceso debe establecerse, completa y coherentemente, usando una representación estándar como se muestra en las figuras siguientes.

La jerarquía de tareas obtenida a partir del análisis estratégico es la siguiente:

## Diagrama Jerárquico de Tareas

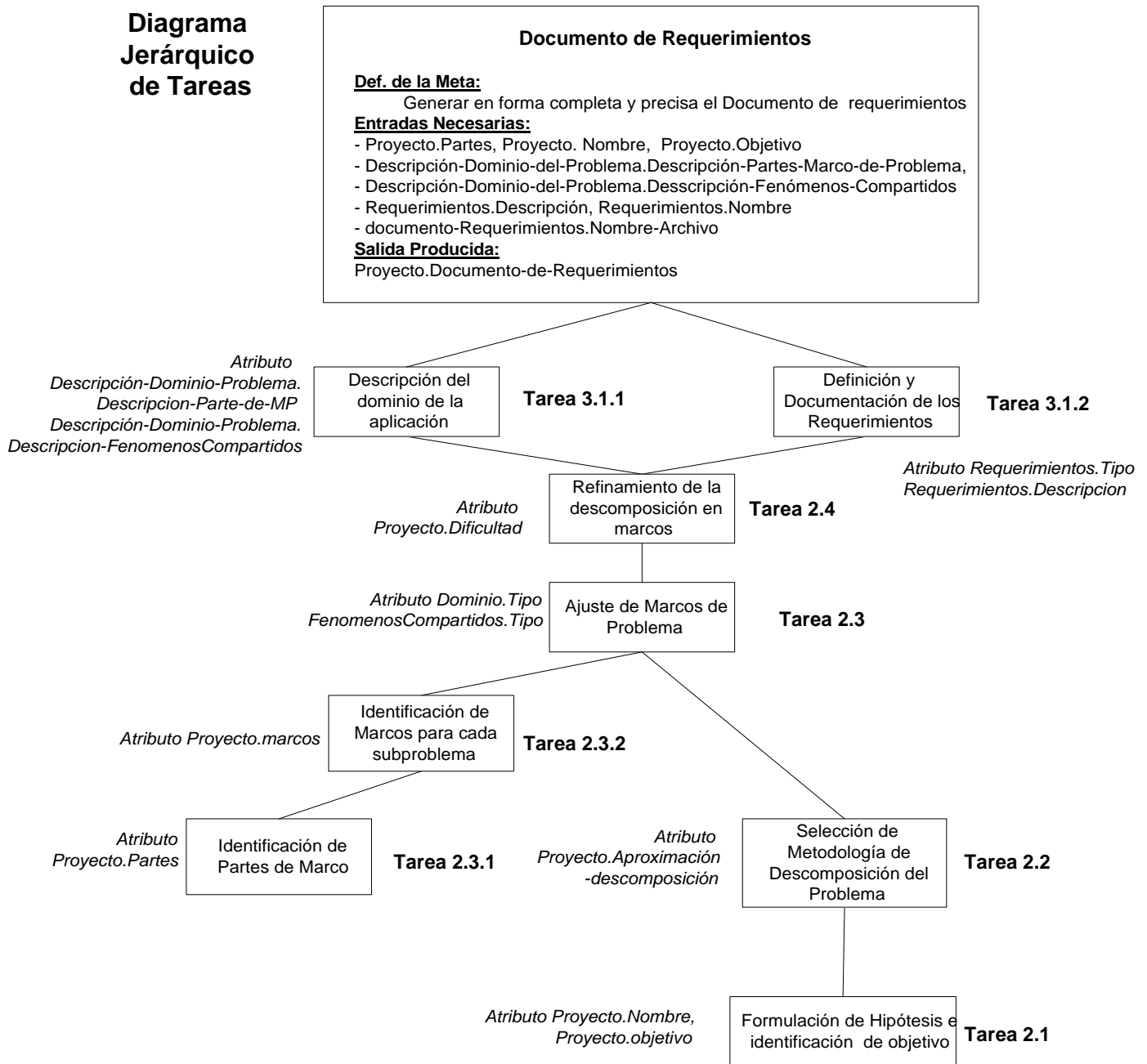


Figura 6.4: Diagrama Jerárquico de tareas.

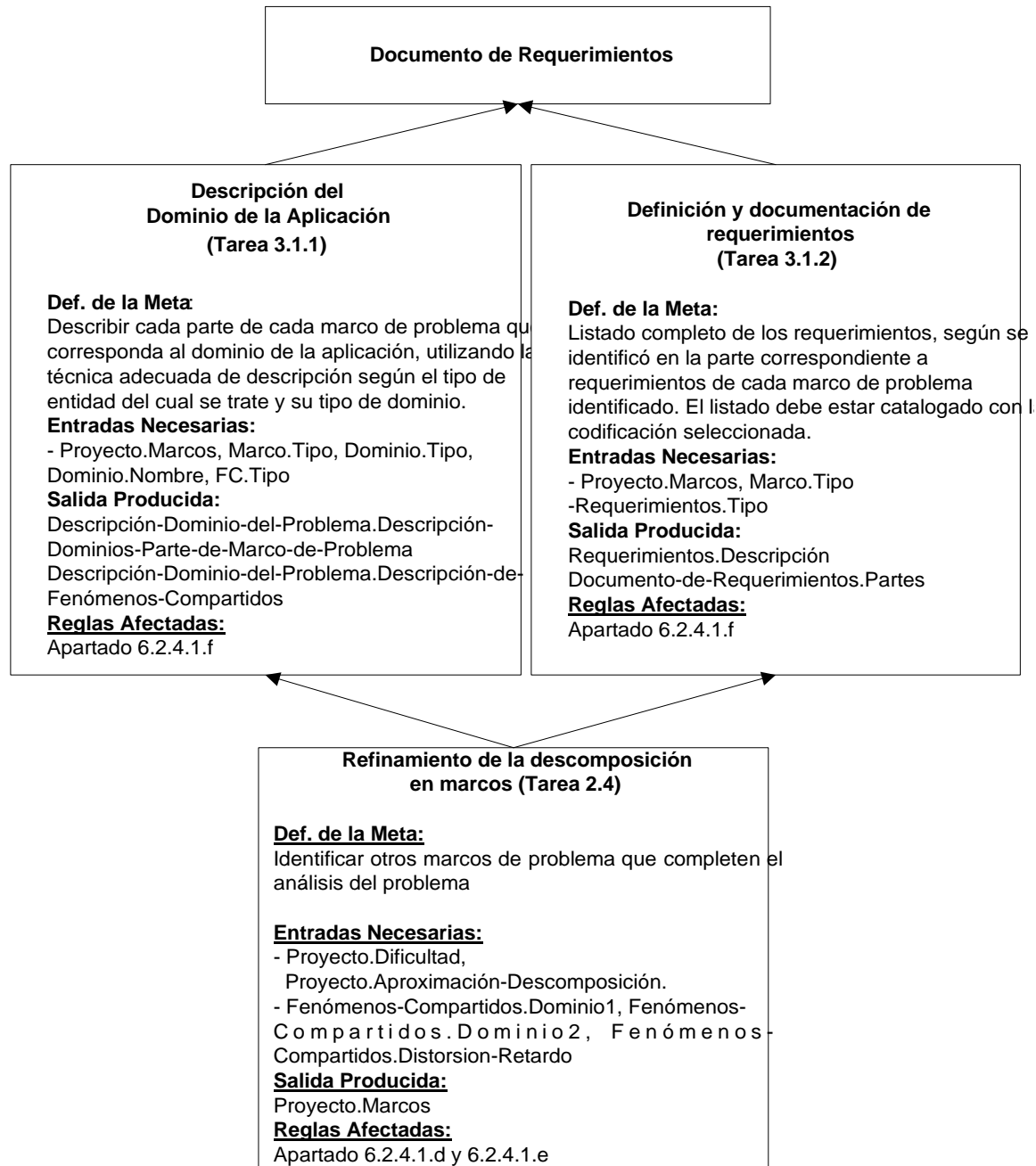
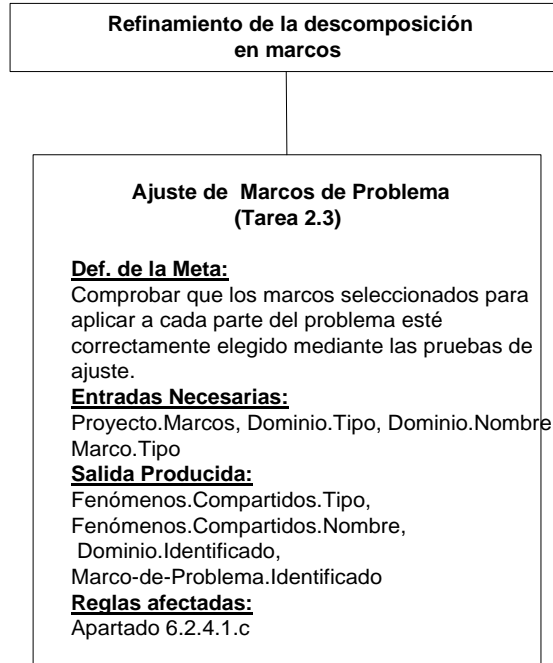


figura 6.5: Jerarquía de tareas del documento de requerimientos.



*Figura 6.6: Jerarquía de tareas del refinamiento de la descomposición.*

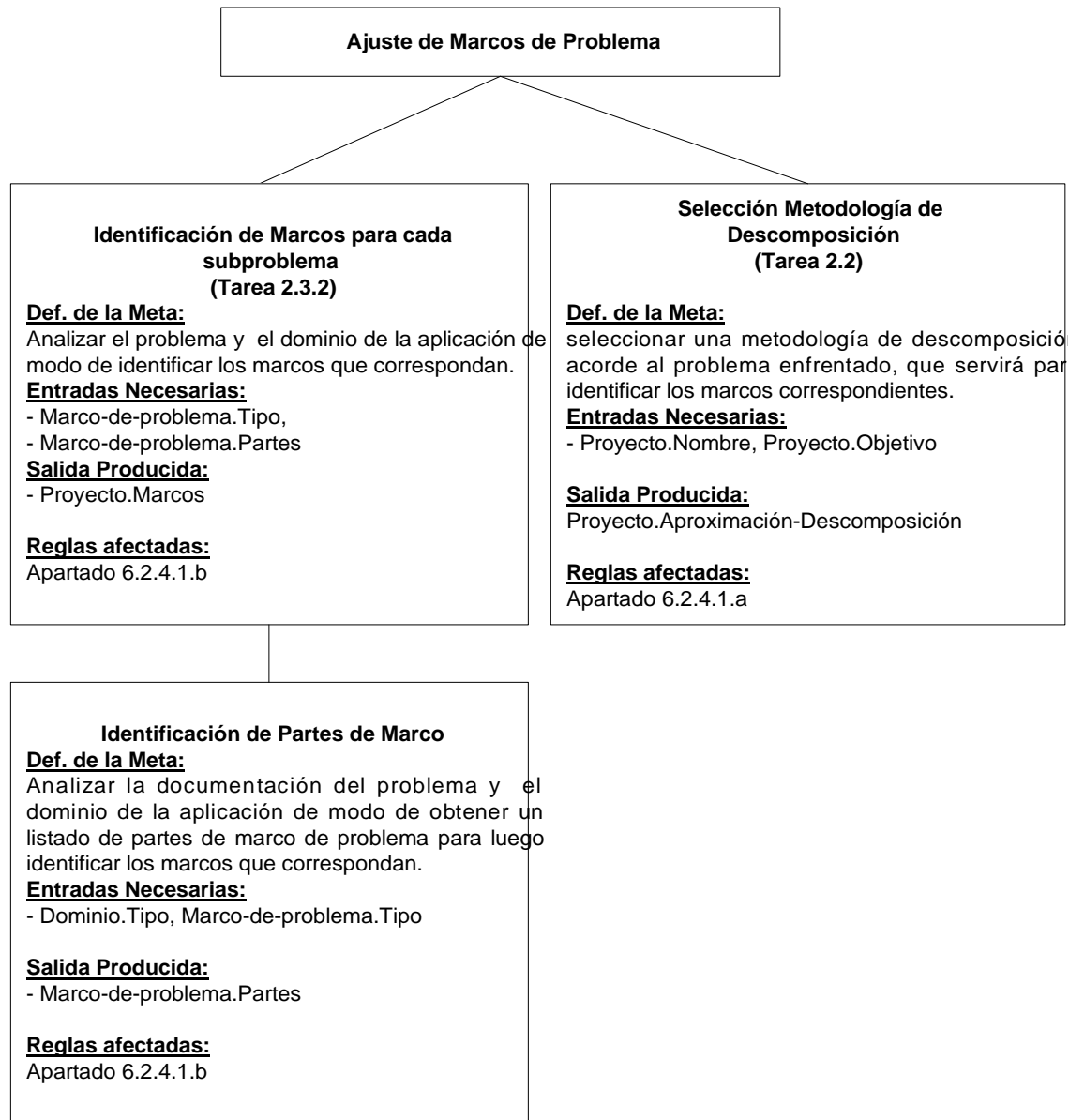


Figura 6.7: Jerarquía de tareas de la identificación y ajuste de marcos.

**Identificación de Objetivos  
(Tarea 2.1)**

**Def. de la Meta:**

Obtener una definición del objetivo del sistema a construir, determinando que tipo de máquina se desea configurar, que efectos se desea que dicha configuración produzca y cuales son las propiedades del mundo real que la máquina puede explotar para producir tales efectos.

**Entradas Necesarias:**

- Información de elicitación.

**Salida Producida:**

- Proyecto.Nombre, Proyecto.Objetivo

*Figura 6.8: Tareas de identificación de objetivos.*

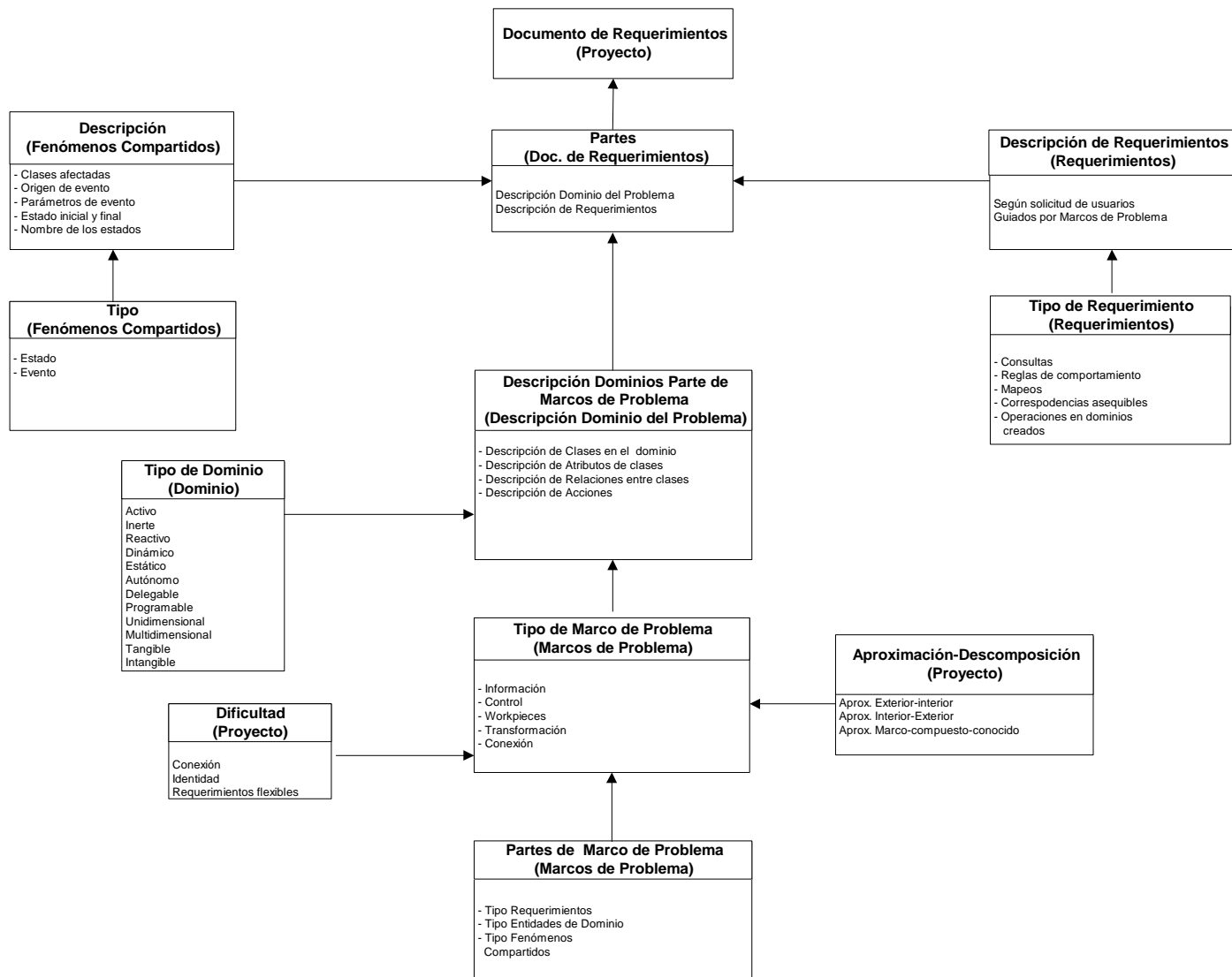


Figura 6.9: Mapa de Conocimientos

### **6.2.7 El Mapa de Conocimientos: Comprobación del MC**

Una vez finalizado el mapa de conocimiento, mostrado en la figura 6.9, se hacen un conjunto de comprobaciones para eliminar subjetividades, considerar condiciones desconocidas o incertidumbre y verificar la completud y consistencia del modelo.

Estas consideraciones son:

a. Contrastar respuestas para eliminar subjetividades

En la periferia están los siguientes atributos:

- Dificultad
- Aproximación de descomposición
- Tipo de Fenómenos compartidos
- Tipo de requerimientos
- Tipo de dominio

En la aproximación de descomposición es inevitable la subjetividad del usuario ya que la identificación de los subproblemas depende de cómo percibe el dominio el usuario y de su experiencia en el mismo.

Una manera de eliminar las subjetividades es la inclusión de librerías de patrones para contrastar y guiar al usuario. Además se debe recomendar al usuario realizar el análisis en conjunto con los usuarios del sistema a construir.

En la introducción de los valores de atributos de dominio, fenómenos compartidos, requerimientos y dificultad fno hay lugar a subjetividades.

b. Examinar condiciones desconocidas o por defecto

No existen condiciones desconocidas. Si el usuario percibe su existencia, esto indica que la elicitación de requerimientos es incompleta y debe revisarse.

c. Confrontar la incertidumbre

No se advierten incertidumbres.

d. Verificar la completud y consistencia

No se observan hasta el momento valores que no sean utilizados.

e. En todas las situaciones se conocen los valores de los atributos.

f. Todos los atributos están en la tabla Concepto/Atributo/Valor. Los atributos no periféricos pueden ser calculados o inferidos.



# Capítulo 7

## Formalización de Conocimientos

## Capítulo 7

### Formalización de Conocimientos

#### 7.1 Introducción

La etapa de formalización tiene como objetivo expresar los conocimientos sobre el problema y su resolución en estructuras que puedan ser utilizadas por una computadora.

Las distintas estructuras que permiten expresar formalmente los conocimientos de un dominio reciben el nombre de formalismos de representación. A los mecanismos de razonamiento de propósito general, independiente del dominio, que llevan adelante su razonamiento con dichas estructuras para resolver el problema se las llama **motores de inferencia**.

La **estrategia de control** se encuadra dentro del componente de resolución de problemas, e incorpora una estrategia de resolución específicamente seleccionada para el tipo de problemas que el sistema basado en conocimiento resolverá, además determina la secuencia de tareas, evalúa alternativas, evalúa por qué se ha llegado a ciertas decisiones y verifica si las mismas son las adecuadas.

Todos estos elementos se combinan en un modelo formalizado, que es una representación semi-formal o semi-computable de los conocimientos y condiciones del experto. Para que un modelo formalizado sea totalmente computable necesita de un modelo implementado, formado por una base de conocimientos, un motor de inferencia y una estrategia de control.

#### 7.2 Selección de formalismos

El formalismo de representación elegido para el presente proyecto es un sistema híbrido de marcos y sistemas de producción.

Los sistemas de producción están formados por una base de hechos, una base de reglas y la estrategia de control. La base de hechos o memoria de trabajo almacena información sobre la tarea y las metas a alcanzar. La base de reglas está formada por un conjunto de reglas o producciones que presentan la forma SI-ENTONCES. La parte SI se llama antecedente y representa una lista de conocimientos a verificar. La parte ENTONCES se llama consecuente y representa las acciones a realizar si los

antecedentes son ciertos. Las acciones incluyen o excluyen datos de la base de hechos, solicitan datos al usuario o bien se los proporcionan.

Los marcos son estructuras de datos que representan situaciones estereotipadas construidas sobre situaciones similares ocurridas anteriormente, permitiendo así aplicar a situaciones nuevas el conocimiento de situaciones, eventos y conceptos previos [Minsky, 75]. El conocimiento que se expresa en la estructura de datos es el conocimiento declarativo del dominio. Dentro del marco existe conocimiento procedimental que se refiere a como utilizar el marco, que se espera que suceda durante su uso, así cómo las acciones que se deben llevar a cabo si las expectativas se cumplen o no.

A través de formalismos de marcos se representarán los conceptos y sus atributos asociados estudiados en la etapa de conceptualización.

### **7.3 Representación de los conocimientos en marcos**

Se utilizarán marcos para representar conceptos, relaciones para expresar dependencias entre conceptos, propiedades para describir cada concepto y facetes para expresar de variadas formas los valores que puede tomar cada propiedad.

Los **marcos clase** se utilizan para representar conceptos, clases o situaciones genéricas dadas por un conjunto de propiedades, unas con valores y otras sin valores asignados, que son comunes al concepto.

Los marcos clase que se representarán son:

*Proyecto, Documento de Requerimientos, Marco de problema, Requerimientos, Dominio, Descripción Dominio de Problema, Fenómenos Compartidos.*

Luego de haber elaborado el modelo formalizado se concluyó que era necesario agregar marcos subclase para los conceptos anteriores pero especializados.

Se agregaron entonces los siguientes marcos subclase:

Para el concepto Marco de problema se tiene: *Marco de problema de Información, Marco de problema de Control, Marco de problema de Transformación, Marco de problema para Piezas de trabajo (Workpieces), Marco de problema de Conexión.*

Para el concepto Dominio se tiene: *Mundo Real, DominioControlado, Pieza de Trabajo, Datos Entrada / Salida, Dominio de Interés.*

Para el concepto Requerimientos se tiene: *Consulta, Regla de Comportamiento, Mapeo, Operaciones sobre dominios creados, Correspondencias.*

Para el concepto Descripción del dominio del problema se tiene: *Clases, Atributos, Eventos, Estados, Acciones.*

#### **7.4 Relaciones entre conceptos.**

El formalismo de marcos permite representar las relaciones del dominio, con relaciones entre marcos clase, entre marcos instancia y entre marcos clase y marcos instancia, formando un sistema basado en marcos (SBM). A continuación se explican las relaciones existentes entre ellos.

- a. La relación "**subclase-de**" definida entre los marcos clase *Marco de problema* y *Marco de Problema de Información-Control-Transformación-Pieza de trabajo-Conexión* representa la jerarquía de marcos clase del SBM desde los marcos generales a los especializados según el tipo de problema que se quiere analizar.
- b. La relación "**compuesto-de**" definida entre el marco *Documento de Requerimientos* y *Requerimientos*, establece que el documento objetivo del sistema incluye requerimientos como uno de sus componentes.
- c. La relación "**compuesto-de**" definida entre el marco *Documento de Requerimientos* y *Descripción del Dominio del Problema* permite establecer que el documento también incluye la descripción del dominio del problema.
- d. La relación "**descrito-por**" definida entre el marco *Descripción del Dominio del Problema* y *Clases, Atributos, Acciones, Estados y Eventos* permite establecer cómo construir la descripción que es parte del documento de requerimientos completando cada uno de dichos marcos descriptores.
- e. La relación **Compuesto-de** definida entre los marcos clase *Documento de Requerimientos* y *Proyecto* establece que el documento corresponde a un proyecto determinado.
- f. La relación **Compuesto-de** establecida entre el marco clase *Proyecto* y el marco clase *Marco de problema* establece que el proyecto puede estar formado por uno o más marcos de problema.

- g. La relación **Parte-de** establecida entre el marco clase *Marco de Problema* y los marcos clase *Dominios* y *Requerimientos* establece que todo marco de problema esta formado por partes tales como *Requerimientos* y *Dominios*.
- h. La relación **es-un** establecida entre el marco clase *Parte de Marco de Problema* y el marco clase *Dom. de Conexión, Mundo Real, Dom. Controlado, Datos Entrada-Salida, Workpieces* establece que el dichos marcos son Partes del Dominio con su tipología característica que define el marco de problema.
- i. La relación **es-un** establecida entre el marco clase *Requerimientos* y los marco clase *Operaciones, Correspondencias, Consultas, Reglas de Comportamiento, Mapeos* establece que el dichos marcos son subclases del marco requerimientos con su forma de descripción y características que permiten describir los requerimientos de manera especial según el marco.

**Esquema del SBM del Asistente para generar el Documento de Requerimientos**

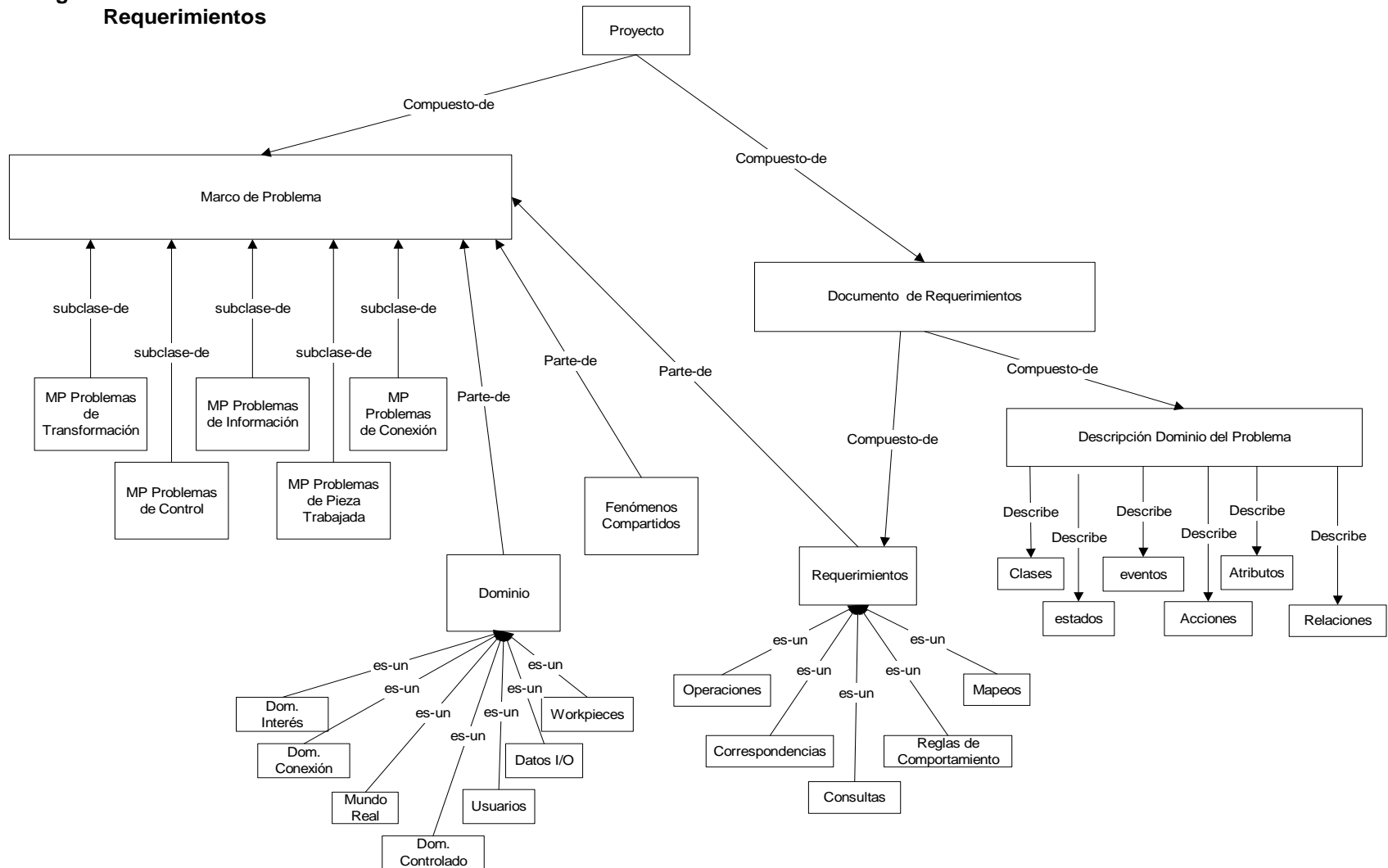


Figura 7.1: Representación de Relaciones entre las jerarquías de Marcos

## **7.5 Propiedades de los conceptos**

Se utilizan dos tipos de propiedades: De clase y de instancia.

Las propiedades de clase las definimos y completamos en el marco clase, por lo que toman el mismo valor en todas las instancias del mismo.

Las propiedades de instancia, aunque se definen en el marco clase y son comunes a todas las instancias del marco clase, se completan en cada instancia con valores concretos que dependen del elemento de la clase que esté representado.

En la siguiente figura se muestran los marcos con sus correspondientes propiedades indicadas como de clase y de instancia.

Nota: Las propiedades precedidas del símbolo (\*) son propiedades de instancia.

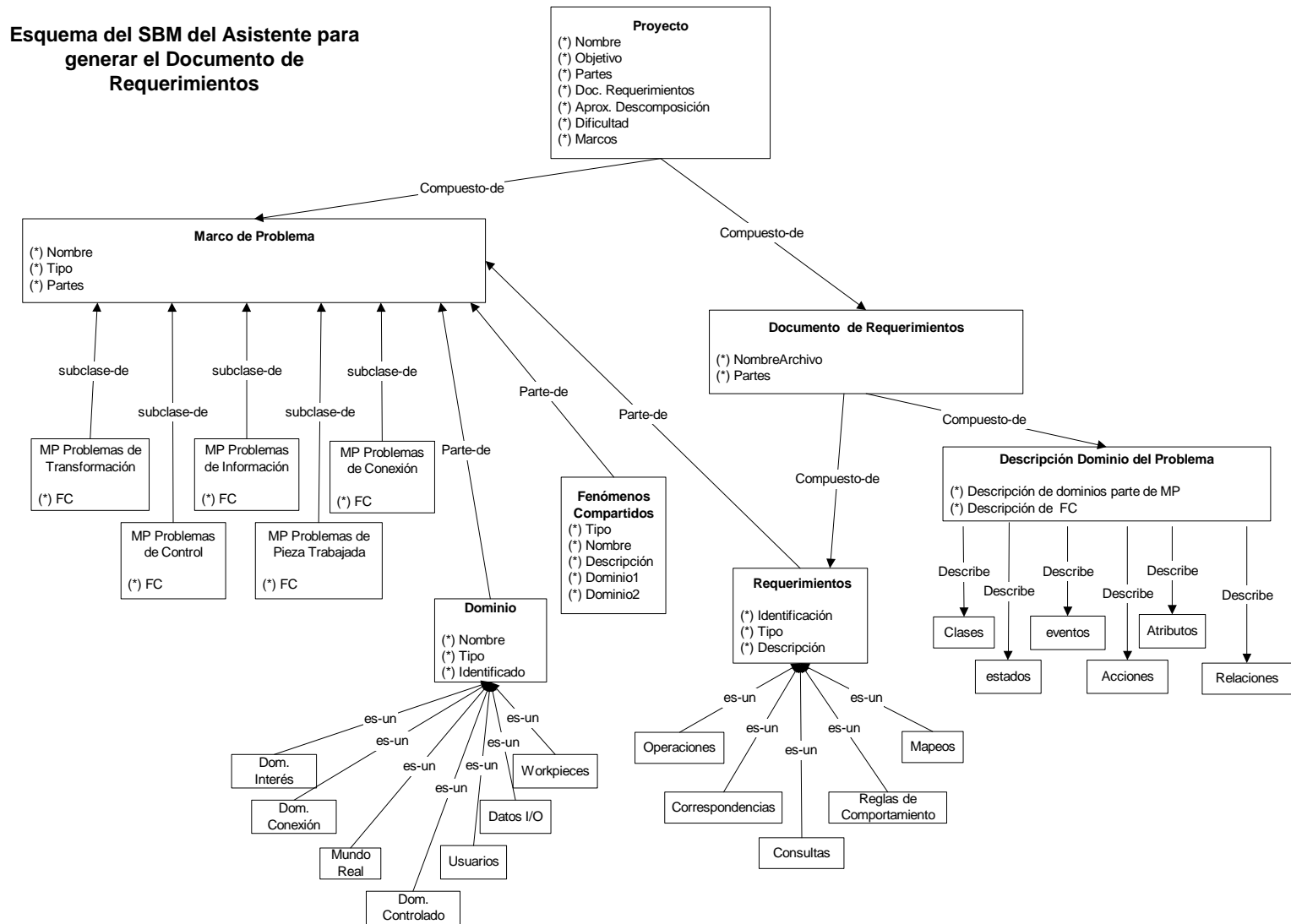


Figura 7.2: Representación de la distribución de propiedades en los diferentes marcos.



### **7.5.1 Facetas de propiedades**

Luego de haber identificado primero las relaciones y posteriormente las distintas propiedades en los marcos clase, se debe continuar con la descripción de las facetas de cada propiedad.

Las facetas son las características de las propiedades y las relaciones entre marcos clase. Se definen a través de un puntero, un tipo de dato (booleano, cadena de caracteres, número, o bien un rango), un procedimiento o una regla.

Se pueden clasificar en tres categorías:

#### ***7.5.1.1 Facetas que definen propiedades de clase, de instancia y de relación.***

- Tipo Ranura: Si se rellena con un valor perteneciente a un tipo de dato, esta propiedad especifica el tipo correspondiente.

Si en cambio se trata de relaciones, se identifica con el nombre de la relación y se carga con el nombre del marco destino.

- Cardinalidad mínima y cardinalidad máxima: Establecen el número mínimo y máximo de valores con que se carga esta propiedad.

- Multivaluada: Informa si la propiedad puede o no tener más de un valor.

#### ***7.5.1.2 Facetas que definen propiedades de clase y de relaciones.***

- Las propiedades de clase que se han definido en la faceta tipo ranura como un tipo de datos, cargan la faceta propiedad general con los valores que toma la propiedad en el marco clase.

- Las propiedades de clase definidas como marco y las relaciones cargan esta faceta como un puntero a un marco clase.

#### ***7.5.1.3 Facetas que definen propiedades de instancia***

Para cada propiedad de instancia definida en un marco clase de debe definir las siguientes facetas:

- Valores permitidos: Conjunto de valores válidos que puede tomar la propiedad de instancia.

- Valores por omisión: Define los valores que debe tomar la propiedad de instancia en un marco instanciado si no se ha indicado otro valor particularmente.

- Si Necesito: Almacena un procedimiento o regla que se ejecuta al solicitar el valor de una propiedad de instancia en un marco instanciado o si dicho valor se desconoce. Estos procedimientos se utilizan para: Requerir el valor de una propiedad de instancia, en un marco instanciado mediante una pregunta al usuario, mantener la integridad de la Base de Conocimientos, gestionar valores dinámicamente y determinar dinámicamente el rango de valores de una propiedad.

i. Se requiere el valor de la propiedad al momento desconocido.

Si el valor de la propiedad en el marco instanciado es desconocido, se puede crear un procedimiento o método para preguntar el valor al usuario, o calcular el valor a partir de otros conocidos.

ii. Mantener la integridad semántica de la Base de Conocimientos.

Se utilizan para impedir que una propiedad de instancia de un marco instanciado se cargue con un valor que es imposible en el dominio de la aplicación. Por ejemplo valores por encima del máximo definido.

iii. Gestión dinámica de valores.

Se utilizan para obtener dinámicamente el valor de una propiedad de un marco instanciado si el conjunto de valores con los que se carga la propiedad en la instancia se puede obtener a partir de los valores almacenados en otras propiedades y relaciones.

- Si Modifico: Almacena el procedimiento a ejecutar antes de modificar un valor de una propiedad en un marco instanciado. La ejecución de este procedimiento puede añadir, modificar o borrar valores en otras ranuras, disparando así sus procedimientos asociados.

- Si Borro: Almacena el procedimiento a ejecutar si se borra el valor de una propiedad de un marco instanciado. Su ejecución puede añadir, modificar o borrar el contenido de otras ranuras, disparando así sus procedimientos asociados.

Las facetas son utilizadas por el motor de inferencia para mantener la integridad semántica de los datos, o sea para comprobar que los valores introducidos en las propiedades se corresponden con el tipo especificado, en la obtención de valores y en la asignación de valores por defecto.

### **7.6 Los marcos, sus propiedades y facetas**

Se define en el siguiente conjunto de tablas todos los marcos clase definidos en el sistema basado en marcos a saber:

*Proyecto, Documento de Requerimientos, Marco de problema, Parte de Marco de Problema, Dominio, Requerimientos, Descripción Dominio de Problema, Fenómenos Compartidos.*

.

En el apéndice D se describen todos los marcos subclase según se detalla en el apartado 7.3.

Marco Clase: Proyecto											
Nombre	Cadena de Caracteres	1	1	No	-	String	-	PedirUsuario(\$NombreSistema)			
Ranura	Tipo Ranura	Card. Min.	Card. Max.	Multiv.	Valor Omisión	Valores Permitidos	Prop. General	Si Necesito	Si Añado	Si Modifico	Si Borro
Objetivo	Cadena de Caracteres	1	1	No	-	String	-	PedirUsuario(\$Objetivo)			
Marcos	Marco	1	5	SI	-	-	^Marco de Problema	PedirTipoDescomposicion(\$TipoDesc)	-	-	-
Ajustado	Boolean	1	1	No	FALSE	TRUE/FALSE	-	-	-	-	-
DocumentoReq	Marco	1	1	No	-	-	-	GenerarDoc(\$NombreDoc)	-	-	-
Partes	Marco	1	1	No			^Partes MarcoProblema				
AproxDesc	Cadena de Caracteres	1	1	No		Int-Ext Ext-Int M. Comp.					
Dificultad	Cadena de Caracteres	1	1	No		Identidad Conexión Req-Flex					

Tabla 7.1: Marco Clase Proyecto

<b>Marco Clase: Documento de Requerimientos</b>											
<b>Ranura</b>	<b>Tipo Ranura</b>	<b>Card. Min.</b>	<b>Card. Max.</b>	<b>Multiv.</b>	<b>Valor Omisión</b>	<b>Valores Permitidos</b>	<b>Prop. General</b>	<b>Si Necesito</b>	<b>Si Añado</b>	<b>Si Modifico</b>	<b>Si Borro</b>
NombreDoc	Cadena de Caracteres	1	1	No	-	String	-	-	-	-	-
PathArchivo	Cadena de Caracteres	1	1	No	-	String	-	-	-	-	-
Estado	Cadena de Caracteres	1	1	No	Inicial	Inicial Creando Terminada	-	-	-	-	-
Requerimiento	Marco	1	1	No			^Requerimientos	-	-	-	-
Descripción Dom. Problema	Marco	1	1	No			^DescDominoProblema	-	-	-	-

*Tabla 7.2: Marco Clase Documento de Requerimientos*

<b>Marco Clase: Marco de Problema</b>											
<b>Ranura</b>	<b>Tipo Ranura</b>	<b>Card. Min.</b>	<b>Card. Max.</b>	<b>Multiv .</b>	<b>Valor Omisión</b>	<b>Valores Permitidos</b>	<b>Prop. General</b>	<b>Si Necesito</b>	<b>Si Añado</b>	<b>Si Modifico</b>	<b>Si Borro</b>
Ajustado	Boolean	1	1	No	FALSE	TRUE/FALSE					
Descripcion	Cadena de Caracteres	1	1	No	-	-					
Grafico	Cadena de Caracteres	1	1	No	-	-					
Nombre	Cadena de Caracteres	1	1	No	-	-		ProcIdentif Partes			
Partes	Marco	1	5	SI	-	Nota1					
Tipo	Cadena de caracteres	1	1	No							

*Tabla 7.3: Marco Clase Marco de Problema*

NOTA1: Los valores permitidos en la ranura Partes de la clase de problema son: Mundo\_Real, Dominio\_Controlado, Consultas, Mapeos, Operaciones\_sobre\_dominios\_creados, Máquina, Reglas\_Comportamiento, Usuario, Workpieces, Entradas, Salidas, Dominio\_Conexión, Dominio\_Interes, Solicitante\_Información, Correspondencias.

Marco Clase: Parte de Marco de Problema											
Ranura	Tipo Ranura	Card. Min.	Card. Max.	Multiv.	Valor Omisión	Valores Permitidos	Prop. General	Si Necesito	Si Añado	Si Modifico	Si Borro
Nombre	Cadena de Caracteres	1	1	No							
Descripcion	Cadena de Caracteres	1	1	No							
Dominio	Marco	1	-	Si		^Marco Dominio					
Identificado	Booleano	1	1	No	False	TRUE/FALSE					
Tipo	Cadena de Caracteres	1	1	No	-	Máquina/ Dominio_Aplicación/Requerimiento					
Parte-de	Marco	1	1	No	-	^Marco Problema					

Tabla 7.4 Marco Clase Parte de Marco de Problema

Marco Clase Requerimientos											
Ranura	Tipo Ranura	Card. Min.	Card. Max.	Multiv.	Valor Omisión	Valores Permitidos	Prop. General	Si Necesito	Si Añado	Si Modifico	Si Borro
Identificación	Cadena Caracteres	1	1	No	-	Cadena Caracteres	-	-	-	-	-
Prioridad	Cadena Caracteres	1	1	No		Deseable/Esencial/ FuturaVersión	-	Preguntar Prioridad			
Fuente	Cadena Caracteres	1	1	No		Cadena Caracteres	-				
Tipo	Cadena Caracteres	1	1	No		Consultas/Regla_C omportamiento/Mapo/OperacDomCreados/CorrespondenciaAsequible					

Tabla 7.5: Marco Clase Requerimientos

<b>Marco Clase: Dominio</b>											
<b>Ranura</b>	<b>Tipo Ranura</b>	<b>Card. Min.</b>	<b>Card. Max.</b>	<b>Multiv.</b>	<b>Valor Omisión</b>	<b>Valores Permitidos</b>	<b>Prop. General</b>	<b>Si Necesito</b>	<b>Si Añado</b>	<b>Si Modifico</b>	<b>Si Borro</b>
Nombre	Cadena Caracteres	1	1	No	-	Cadena Caracteres					
Descripción	Cadena Caracteres	1	1	No	-	Cadena Caracteres					
Tipo	Cadena Caracteres	1	-	Si	-	Cadena Caracteres	Nota 2				

*Tabla 7.6: Marco Clase Dominio*

Nota 2: Los valores permitidos para marco de dominio son: Estático, Dinámico, Unidimensional, Multidimensional, Tangible, Intangible, Activo, reactivo, Inerte, Activo-Autónomo, Activo-Programable, Activo-Delegable.



Marco Clase Descripción Dominio del Problema											
Ranura	Tipo Ranura	Card. Min.	Card. Max.	Multiv.	Valor Omisión	Valores Permitidos	Prop. General	Si Necesito	Si Añado	Si Modifico	Si Borro
Identificacion	Cadena Caracteres	1	1	No	-	Cadena Caracteres					
Partes_Descripciones	Marco	1	-	Si		^Partes Marco Problema					
Desc_Fin	Boolean	1	1	No	FALSE	TRUE_FALSE		Preguntar(\$Desc_Fin)			
Descripciones	Cadena Caracteres	1	-	Si	-	Cadena Caracteres					
Req_Def	Boolean	1	1	No	FALSE	TRUE-FALSE		Preguntar(\$Req_Def)			
Tareas	Cadena Caracteres	1	-	Si		Cadena Caracteres					
Tareas_Descripciones	Cadena Caracteres	1	-	Si		Cadena Caracteres					

Tabla 7.7: Marco Clase Descripción Dominio del Problema (Advisor)

<b>Marco Clase: Fenómenos Compartidos</b>											
<b>Ranura</b>	<b>Tipo Ranura</b>	<b>Card. Min.</b>	<b>Card. Max.</b>	<b>Multiv.</b>	<b>Valor Omisión</b>	<b>Valores Permitidos</b>	<b>Prop. General</b>	<b>Si Necesito</b>	<b>Si Añado</b>	<b>Si Modifico</b>	<b>Si Borro</b>
Nombre	Cadena Caracteres	1	1	No	-	Cadena Caracteres	Nota 3				
Descripción	Cadena Caracteres	1	1	No	-	Cadena Caracteres					
Tipo	Cadena Caracteres	1	-	Si	-	Cadena Caracteres					
Dominio1	Cadena Caracteres	1	-	No							
Dominio2	Cadena Caracteres	1	-	No		Cadena Caracteres					

*Tabla 7.8: Marco Clase Fenómenos Compartidos*

Nota 3: Son estados o eventos que dependen y están caracterizados según el marco de problema al que pertenezcan. Pueden ser: FC\_C1, FC\_C2, FC\_C3, FC\_E1, FC\_E2, FC\_E3, FC\_H1.

## **7.7 El Sistema de Producción**

La arquitectura de un sistema de producción está formada por tres elementos: La base de Hechos (BH), la base de Reglas (BR) y la Estrategia de Control (EC).

La base de conocimientos del sistema (BC) está formada por la base de hechos y la base de reglas.

$$BC = BH + BR$$

### **7.7.1 La Base de Hechos (BH)**

El estado actual del problema y las metas a alcanzar se localiza en la BH.

La BH del asistente de requerimientos está compuesta por conocimientos expresados en hechos y datos que muestran la información elicitada por el analista sobre el dominio de la aplicación que se intenta construir, mas los requerimientos que los usuarios pretenden sobre el sistema, y finalmente posee la descomposición del problema realizada por el analista.

Los estados de la BH se pueden resumir en los tres siguientes:

a. Estado Inicial: No se conoce la descomposición del problema, los elementos que componen el dominio sin ninguna descripción del mismo, al igual que los requerimientos.

b. Estados intermedios: Son estados que se utilizan para representar, en cada instante, el estado del problema siendo analizado o siendo descrito.

El primer estado intermedio se logra cuando el usuario logró identificar los distintos marcos de problema en los que se descompone el problema analizado, habiéndose justificado y ajustado cada uno de ellos. Luego, el siguiente estado intermedio consiste en describir los detalles de cada uno de los componentes del dominio de la aplicación y definir con precisión los distintos requerimientos deseados.

Existen otros estados intermedios dependiendo de otras submetas como por ejemplo la descripción de las conexiones que se identifican entre los distintos dominios y la máquina.

c. Estados Meta : Se logra cuando el sistema logra su objetivo que consiste en la generación del documento de requerimientos.

La estrategia de control es el mecanismo que examina en cada paso del asistente cuáles son los hechos y datos que se tienen en la BH y determina el paso siguiente, o la regla a aplicar. Dicha aplicación de una determinada regla puede modificar la BH al agregar o quitar hechos.

### **7.7.2 La Base de Reglas (BR)**

En ella se encuentran almacenados los conocimientos acerca de la solución al problema. Dicha representación consiste en un conjunto de reglas o producciones de la forma SI --- ENTONCES.

El lado izquierdo de la regla, denominado condición o antecedente, representa una lista de elementos a verificar en la BH, y el lado derecho, o consecuente, un conjunto de acciones a llevar a cabo sobre la BH, toda vez que los antecedentes resulten ciertos.

En el apéndice D se describen las reglas utilizadas en el sistema híbrido.

### **7.7.3 Conocimiento de Control**

La estructura de control o *motor de inferencia*, examina a su debido tiempo la Base de Hechos y selecciona la regla a disparar, encadenando las mismas en lo que se denomina el ciclo de resolución.

Para que un SE imite el comportamiento de un experto es necesario que utilice un módulo de control que determine cuáles acciones debe ejecutar el sistema en cada momento del proceso de resolución, cuáles son los objetivos a alcanzar y los efectos colaterales que se producirán. Dicho módulo está formado por las estrategias de control que guían en forma flexible el funcionamiento del sistema, mediante un plan que se modifica según las características del problema actual.

- Determinar el valor de hechos automáticamente a partir de otros conocidos:

Es el caso de cuando se conoce cuáles son los marcos de problema que forman el proyecto, automáticamente el sistema establecerá cuáles son los tipos de requerimientos que deben describirse.

- Resolver Conflictos: Es el caso que ocurre cuando se intenta identificar un marco de problema pero existen dificultades en el dominio que impiden la aplicación directa del mismo. El sistema busca resolver dichas dificultades mediante la aplicación de marcos extra según el tipo de dificultad encontrada.

El conocimiento de control utilizado es de dos tipos. General, que utiliza estrategias generales de resolución de problemas y específico, que es intrínseco al dominio del problema de análisis de requerimientos.

Las heurísticas proveen conocimiento sobre cómo explorar el espacio de búsqueda. Para ello se han generado grupos de reglas que actúan en función del estado de avance del proceso de análisis y de la descripción realizada y se han programado metarreglas para seleccionar y aplicar los grupos de reglas según el estado en que se encuentre el asistente de Requerimientos.

Por ejemplo:

Reglas de Ajuste (AjusteMarcoInformación, AjusteMarcoControl, AjusteMarcoTransformación, AjusteMarcoConexion, AjusteMarcoWorkpieces)

Metarregla: rAjusteRules

Si Proyecto.Marcos <> NULL

ENTONCES Aplicar(rAjusteRules)

Reglas de Descomposición (AgregarMarcoInformación, AgregarMarcoControl, AgregarMarcoTransformación, AgregarMarcoConexion, AgregarMarcoWorkpieces)

## **7.8 Descripción del Funcionamiento del SBM**

El sistema debe comportarse de la siguiente manera:

1. Pide al usuario la siguiente información para comenzar el análisis:
  - Nombre del Sistema
  - Objetivo: Somera descripción del propósito de la construcción del sistema. Si es reemplazo de un sistema antiguo y sus problemas, o bien si es un sistema nuevo y el problema que intenta resolver.
2. Pide al usuario que luego de observar ciertas características del problema y en base a las condiciones generales que presenta seleccione una forma de descomposición que puede ser: Elegir de una biblioteca de problemas ya resueltos, Descomponer de adentro hacia afuera o de afuera hacia adentro.

3. Según la descomposición elegida el sistema pide al usuario los diferentes marcos que componen el problema, excepto para la aproximación de problema anteriormente resuelto.

Para el caso de exterior a interior el sistema pregunta al usuario el/los tipos de partes que va encontrando para deducir un marco, luego identificar el resto de partes de dicho marco y eliminar la parte del problema resuelto para continuar con el resto.

Para el caso de interior a exterior el sistema pregunta al usuario el marco de problema elemental que mas se asemeja al problema enfrentado y luego pide que identifique sus partes de acuerdo al problema. Finalmente lo guía en la resolución de las dificultades que impiden el ajuste preciso de dicho marco al problema. Las mismas pueden ser dificultades de conexión y de identidad. El sistema brinda ejemplos posibles y gráficos en cada caso.

4. Luego se recorren todos los marcos para identificar cada parte de cada uno mediante la indicación del tipo de dominio que conforma, la existencia de diferentes fenómenos compartidos entre dominios para luego proceder a verificar que los marcos elegidos para la descomposición se ajusten correctamente al problema. En caso de resultado positivo continua adelante con la descripción de cada dominio y de los requerimientos.

En caso negativo vuelve a atrás recomendando una descomposición diferente o bien un marco distinto en caso de que alguno de ellos no esté ajustado.

5. El siguiente paso es resolver las conexiones entre dominios cuando no se cumplen las condiciones de velocidad y confiabilidad entre ellos; fundamentalmente en el caso de marcos de problema de información y de control. Allí las alternativas son 3:

- a. Ignorar la existencia de un dominio de conexión y tratar los distintos dominios como si estuvieran directamente conectados.
- b. Tratar al dominio de conexión como un ente intermediario entre ambos dominios y solamente describir dicho dominio.
- c. Reconocer la existencia de un nuevo marco llamado marco de problema de conexión y agregar el mismo a la lista de marcos del proyecto.

6. Se procede entonces a recorrer cada marco de problema y describir cada una de las partes del mismo que conforman el dominio del problema según reglas de descripción que dependen del tipo de dominio del que se trate, utilizando técnicas de descripción adecuadas.

Una vez descritos los dominios se procederá a definir en cada marco los requerimientos correspondientes donde el sistema según el caso pide diferentes datos para conformar el requerimiento.

7. Finalmente, cuando el analista lo crea conveniente, el sistema generará el documento con la especificación preliminar de requerimientos según una plantilla estándar y lo dejará en formato HTML (HyperText Markup Language) para su posterior manipulación.

# Capítulo 8

## Selección de la Herramienta e Implementación del Sistema



## Capítulo 8

### Selección de la Herramienta e Implementación del Sistema

#### 8.1 Introducción

Se describe aquí la selección de la herramienta a utilizar en la implementación del Sistema Experto Asistente en la Elaboración del Documento de Requerimientos.

A partir del modelo formal obtenido se seleccionó la herramienta que permite trasladar el modelo formal del comportamiento del experto a un modelo que pueda ser interpretado por una computadora, de modo de representar adecuadamente los modelos formales.

La implementación de un sistema basado en conocimiento exige una herramienta de desarrollo que proporcione los formalismos de representación en los cuales puede codificarse la base de conocimientos, así como los mecanismos de inferencia y control que pueden interpretar la base de conocimientos para ejecutar la tarea deseada.

Una vez seleccionado el entorno, el modelo conceptual guía la implementación del Sistema Experto y permite crear un diseño de implementación al aplicar el modelo conceptual sobre las capacidades de la herramienta. Mientras que el modelo conceptual especifica QUÉ hará el sistema experto, el diseño de la implementación especifica CÓMO puede alcanzarse el comportamiento deseado, dados los formalismos de representación y los mecanismos de inferencias y control del entorno de desarrollo.

El diseño de implementación debe mezclar los requisitos del modelo conceptual con las restricciones del entorno operativo y las necesidades de integración.

#### 8.2. Criterio de Selección de la Herramienta

El proceso de evaluación y selección de una herramienta consta de cinco pasos:

1. Identificar las capacidades o criterios requeridos a la herramienta.
2. Asignar pesos a los criterios según su importancia.
3. Identificar las herramientas candidatas.

4. Evaluar cada candidato basándose en los criterios.
5. Seleccionar el candidato con mayor resultado.

La herramienta escogida para el desarrollo del prototipo ha sido Kappa PC de INTELLICORP, Inc. la cual posee amplias características compatibles con la metodología que se ha aplicado en el desarrollo del sistema. Además guarda estrecha relación con el modelo de formalización elegido para la representación computable del modelo conceptual que se posee de la tarea que desempeñan los expertos.

Kappa PC brinda un entorno de desarrollo que permite prototipado rápido, de modo de llevar a cabo un ciclo de vida en donde en cada iteración se incrementen los conocimientos y, así lograr consistencia con el desarrollo propuesto por la Metodología I.D.E.A.L.

Posee además un ambiente de desarrollo totalmente gráfico, contando con una amplia gama de objetos que permite modelar en tiempo y forma la interfaz de usuario del sistema.

La tabla 8.1 resume los aspectos observados en la selección de la herramienta.

Característica	Subcaracterística		Evaluación
Representación de Conocimientos	Dominio		Marcos Reglas
	Incierto		No posee
	Control		Agendas Prioridades de Reglas
Motor de Inferencias			Herencia Múltiple Demonios Encadenamiento hacia delante Encadenamiento hacia atrás Métodos
Interfaces con humanos	IC	BC	Integridad Relacional Gráficos Barras de menú Librerías Ayuda No Recompilación Interfaz gráfica ampliable
		Depuración	Navegador Traza Paso a Paso Puntos de ruptura Detecta y evita bucles infinitos Validación
	Usuario		Creación de interfaz gráfica con ventanas Gráficos Ejecución interactiva Herramienta ampliable Adaptable al cliente
Interfaces con otro software	Lenguaje		Ansi C C++ Fortran
	BBDD		Sistemas con SQL
Requisitos del Sistema	Hardware		PC
	Sistema Operativo		Unix X-Windows Windows DLL MVS
Otras características Técnicas			Modular En Capas Integración con otro software sencilla Mecanismos de hoja de cálculo Prototipado rápido Orientado a objetos Reuso Independiente del sistema operativo Capacidad de Retroceso

Tabla 8.1: Características de Kappa-PC

Además de lo expresado, Kappa se soporta en computadoras del tipo PC, las cuales eran las únicas accesibles para llevar adelante el proyecto de sistema experto, permite la implementación de marcos y reglas de producción, facilitando la implementación dado que ofrece recursos de programación visual para la implementación del modelo, y de la interfaz de usuario y mecanismos de depuración.

Posee el lenguaje de programación llamado KAL que se ejecuta interpretado. Además al realizarse la programación visual, automáticamente la herramienta genera el código KAL correspondiente, pudiéndose luego refinar dicho código, o bien agregar mediante código aquellas rutinas necesarias que no se generen automáticamente.

En el presente proyecto, por su naturaleza, hubiera sido muy útil que la herramienta realizase equiparación de marcos, algo no soportado en Kappa-PC. No obstante, mediante programación se salvó la carencia.

### **8.3. Plan de Implementación del Asistente**

Una vez familiarizado con el entorno de desarrollo que ofrecía Kappa PC, se ha dado comienzo a la implementación del prototipo. Como lo establece la Metodología I.D.E.A.L., el ciclo de vida del sistema se basa en prototipado incremental. Lo cual implica, que para haber logrado el prototipo, se desarrollan una serie de prototipos hasta lograr aquel que satisface los requisitos determinados por los expertos.

#### *a) Implementación de Marcos.*

La Base de Conocimientos formalizada en marcos, se ha representado de la siguiente manera:

- Para representar cada marco clase se utilizaron los Objetos Clase de Kappa y, para representar los marcos instancia se han utilizado instancias de objetos. Por ejemplo se utilizó el objeto *Elemental* para implementar el Marco Clase *Marco de problema*.
- También se utilizó la jerarquía de objetos brindada por Kappa PC para implementar las subclases de marcos de problema como por ejemplo el objeto *información* para representar la subclase *Marco de Problema de información*.
- los objetos instancia para los objetos *marcos de problema de información* se crean en forma dinámica cuando se produce la ejecución del asistente de requerimientos.

Una vez implementada la jerarquía de marcos, se prosiguió con la definición de las propiedades de clase de los diferentes marcos, utilizando las ranuras o (slots) que se puede definir en cada objeto, y se ha descrito el valor que debería poseer durante la ejecución del sistema. Las propiedades de instancia también se definen en Kappa PC en los Objetos Clase, conservándose invariable su valor durante la ejecución del sistema.

Definido el conocimiento declarativo, se implementa el conocimiento procedimental definido en las propiedades de los marcos instanciados, mediante la declaración de demonios que corresponden a las facetas Si Necesito, Si Modifico y Si Borro.

Los demonios consiguen su representación en Kappa PC a través de métodos, que una vez definidos en cada objeto, deben ser activados de acuerdo a la función que poseen asignada.

#### *b) Implementación de Reglas.*

Habiendo implementado el Modelo de Marcos, el siguiente paso es la implementación de las reglas que conforman el sistema de producción.

Para comenzar se han implementado aquellas reglas que permiten seleccionar las aproximaciones de descomposición del problema ya sea *Interior-Exterior*, *Exterior-interior* o *Marco-compuesto-conocido*. Luego de verificar su funcionamiento adecuado, se procedió con las reglas de selección, identificación y ajuste de los diferentes marcos que conforman el problema software que enfrenta el analista.

Luego se continuó con las reglas correspondientes a la solución de las dificultades para la aproximación *interior-exterior* y finalmente las reglas para la descripción del dominio de aplicación y requerimientos según los marcos seleccionados.

El siguiente paso ha sido la incorporación de los procedimientos adecuados que procuran la activación de los mecanismos de razonamiento.

#### *c) Implementación del Conocimiento de Control.*

Una vez codificada la información del dominio a través de su implementación en marcos y reglas, se han codificado los conocimientos de control. Estos conocimientos corresponden a las metarreglas que gobiernan la aplicación de los

conjuntos de reglas en forma heurística según la fase de avance del proceso de análisis con el asistente.

Una vez terminada la implementación de todos los conocimientos se ha llevado a cabo la evaluación integral del funcionamiento del sistema, verificando y evaluando las posibles alternativas en el proceso, con avance y retroceso en el análisis de los requerimientos y con iteraciones a través de la realización de nuevas sesiones de elicitación con los usuarios. A su vez se ha refinado de la misma forma la interfaz de usuario de modo de darle un tinte de asistente guía con elementos que aporten a la visibilidad del proceso. Existen ventanas que indican el grado de avance del proceso (GAP) de modo que el analista se sienta cómodo en la navegación multidireccional observada frecuentemente en el uso del sistema. La descripción del dominio se vá realizando en múltiples frentes a medida que el analista vá profundizando en el dominio e interactuando con los futuros usuarios del producto software a construir.

#### **8.4 La Interfaz de Usuario**

La interfaz de usuario es muy importante para que el sistema sea realmente utilizado y aceptado por los usuarios.

Kappa PC suministra un conjunto de clases prefabricadas para crear la interfaz de usuario y además un mecanismo de programación visual que facilita esa creación mediante el clásico ciclo de "prueba y error".

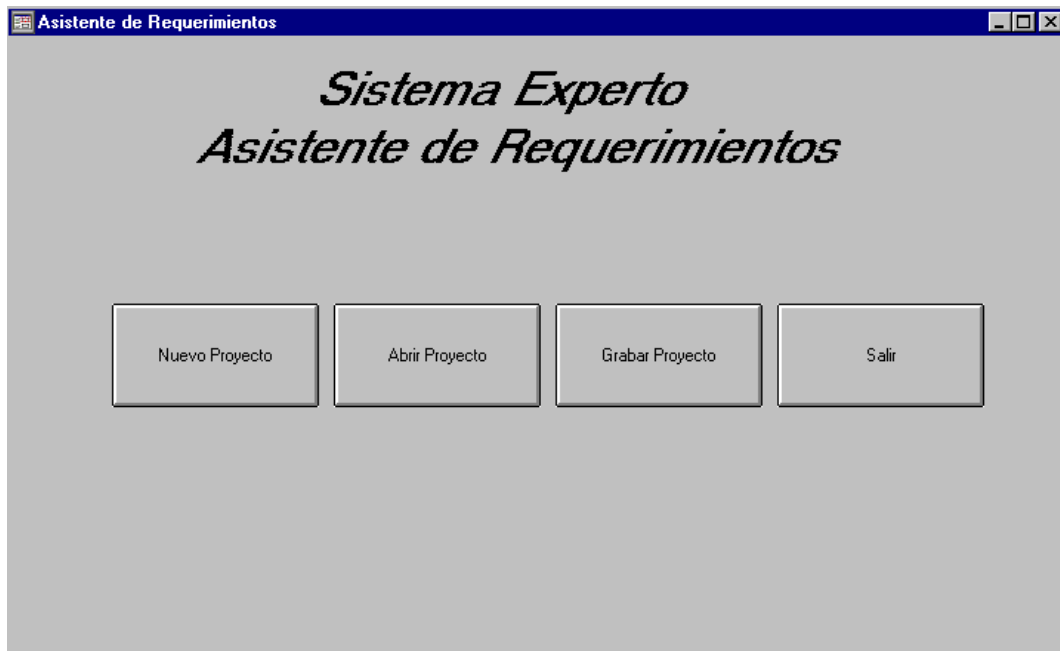
En el presente trabajo se empleó una gran cantidad de tiempo en la implementación de la interfaz de usuario ya que se debía mantener una interfaz amigable, que mantenga el proceso visible, guiando al analista en cada etapa. Considerando que los datos a introducir son muchos, la interfaz debió ser reelaborada varias veces con el aporte de los expertos de modo de lograr un proceso sencillo pero completo.

Además como menciona Natalia Juristo<sup>1</sup> hay que diseñar correctamente la interfaz de usuario, debido a que, *"existen grandes diferencias de rendimiento cuando los Sistemas Expertos son operados por sus desarrolladores, que cuando estos son manipulados por los usuarios"*. El diseño inadecuado de la interfaz de usuario puede condenar el sistema al fracaso.

---

<sup>1</sup> Juristo, N. *"Introducción a los Sistemas Expertos"*, Máster en Ingeniería del Software, Universidad Politécnica de Madrid / Instituto Tecnológico de Buenos Aires.

En las siguientes figuras se observan las diferentes interfaces logradas para tal fin.



*Figura 8.1: Pantalla inicial del asistente*

La figura 8.1 representa la primera pantalla en la que se observa las operaciones básicas al iniciar el asistente: Se puede crear un nuevo proyecto, se puede abrir un proyecto terminado o bien un proyecto en curso siendo analizado, o por último salir del sistema.

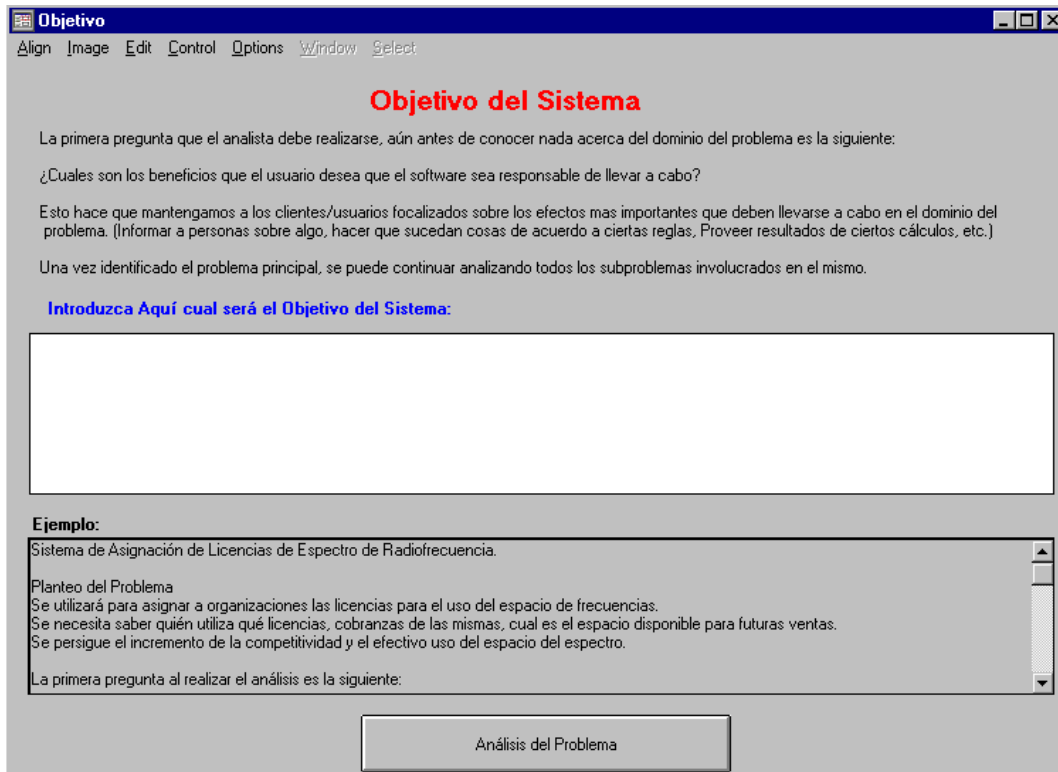


Figura 8.2

Si el usuario opta por crear un nuevo proyecto, el sistema inmediatamente preguntará por el nombre del sistema, producto de un demonio que necesita dicho dato. Una vez introducido el mismo pasa a otra pantalla, representada por la figura 8.2 en la que a través de otro demonio se pide al analista que ingrese el objetivo para el cual se construye el software. Este es un punto muy importante dado que definirá en la mayoría de los casos el marco de problema básico. Por ejemplo supongamos que estamos construyendo un sistema para administrar los números de IP estáticos de una red TCP/IP con muchas subredes. El sistema servirá de soporte al administrador para realizar las asignaciones estáticas de direcciones. Puede que suceda dos alternativas:

- a) Que el sistema simplemente informe el estado de las asignaciones realizadas y permita que el administrador decida qué número nuevo asignar cada vez. En éste caso se trata de un problema de información.
- b) Que el sistema decida qué números de IP asignar cada vez que se le pide la operación de alta de una nueva estación. En éste caso el problema es de Piezas de trabajo.



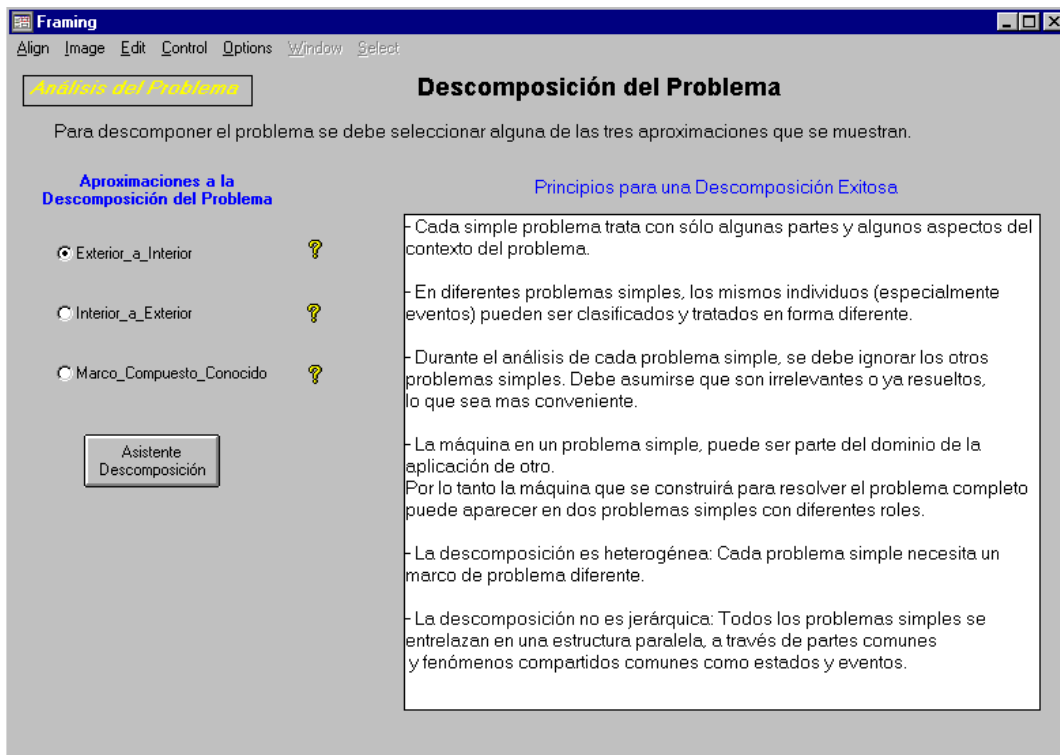


Figura 8.3

En la mayoría de los casos de problemas de software de la realidad, su complejidad hace que deban ser representados por mas de un marco de problema elemental, o bien por un marco de problema elemental con elementos adicionales que dificultan su simple aplicación.

Entonces se debe proceder a la descomposición del problema real en diferentes marcos de problema elementales para su correcto análisis.

Una vez introducido el objetivo del sistema a construir, el asistente propone tres aproximaciones diferentes para la descomposición: Exterior a Interior, Interior a Exterior y similitud con un marco compuesto de un problema anteriormente resuelto.

Provee para ello de íconos con el signo de pregunta que asisten al analista sobre cual tipo de aproximación elegir. Además brinda consejos sobre los principios para lograr una correcta descomposición.

Una vez elegida la aproximación de descomposición continúa proponiendo el siguiente paso que dependerá justamente de la aproximación elegida:

a) Exterior-Interior: En éste caso sugiere analizar el tipo de requerimientos observado en la documentación obtenida en la etapa de elicitación. Según los diferentes tipos aconsejará un marco diferente y solicitará que el analista ingrese los distintos componentes según el marco del que se trate.

En la figura 8.4 se observa que además el asistente brinda información sobre el tipo de marco apropiado según los requerimientos que se observen, además pautas para identificar correctamente cada tipo de requerimiento.

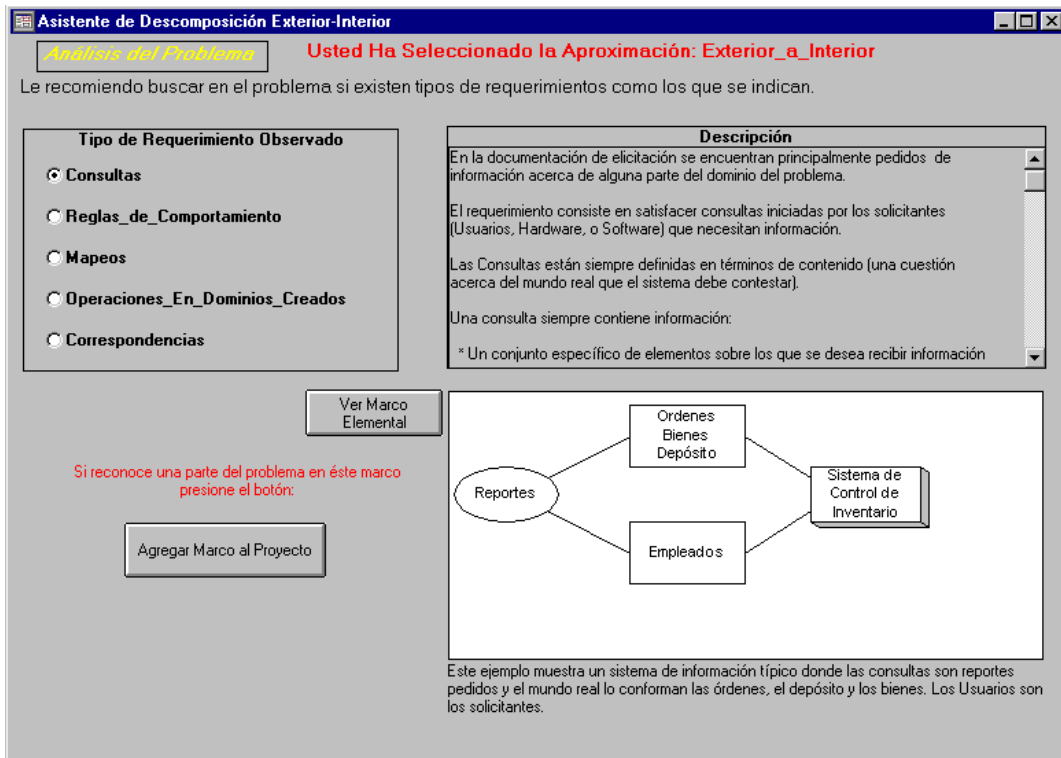


Figura 8.4

b) Interior-Exterior: Con esta aproximación el asistente pregunta mediante el disparo de un demonio cuál es el tipo de marco de problema central que mas se ajusta al problema en cuestión, luego solicita la identificación de cada una de las partes que componen dicho marco. A posteriori guía al analista en la tarea de resolver las distintas dificultades que aún impiden tener el problema totalmente resuelto como se observa en la figura 8.5.

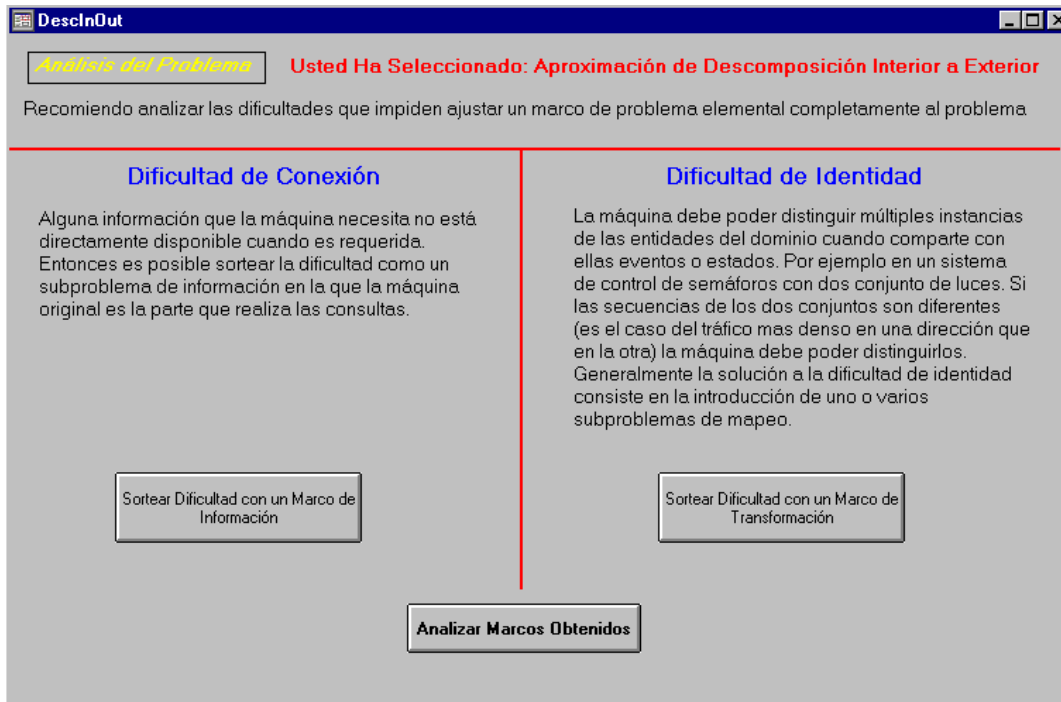


Figura 8.5

Según sea el tipo de dificultad que pudiere plantearse el sistema plantea el problema e invita al analista a solucionarlo mediante la aplicación de un marco adicional, como puede observarse en la figura 8.5.

c) Marcos compuestos conocidos: Esta opción permite al analista observar una biblioteca de problemas resueltos anteriormente mediante un navegador que muestra los distintos marcos compuestos y sus correspondientes marcos elementales.

En la figura 8.6 y 8.7 puede observarse ambas pantallas.

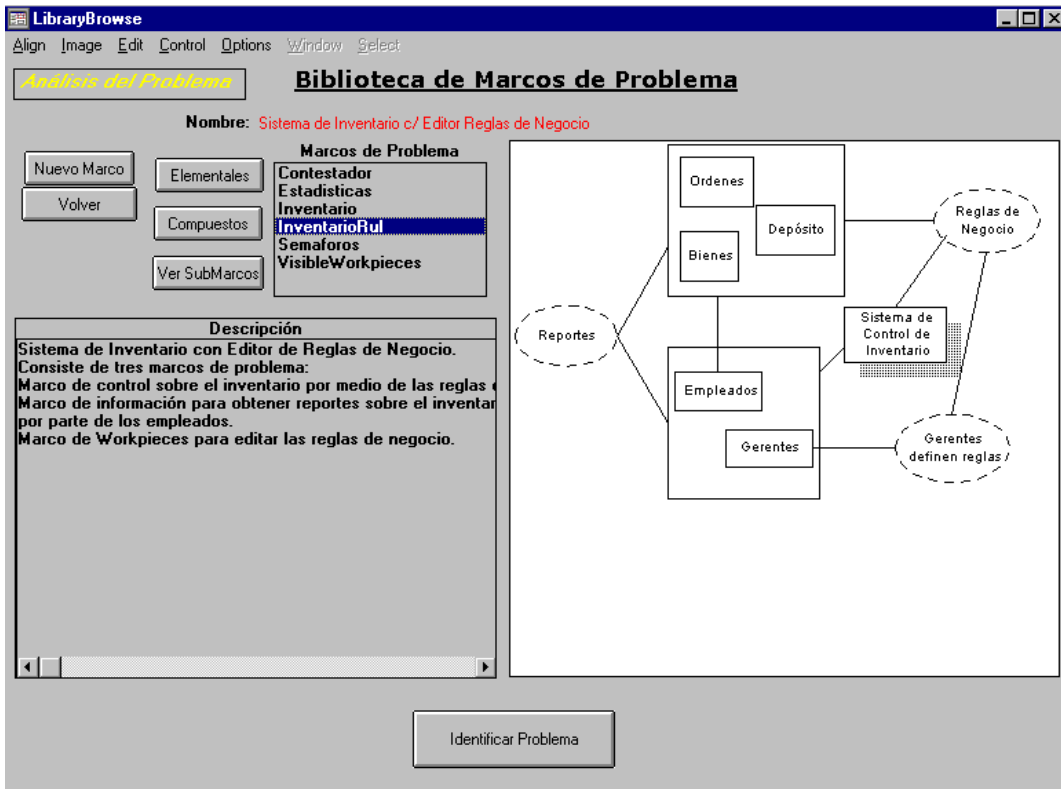


Figura 8.6

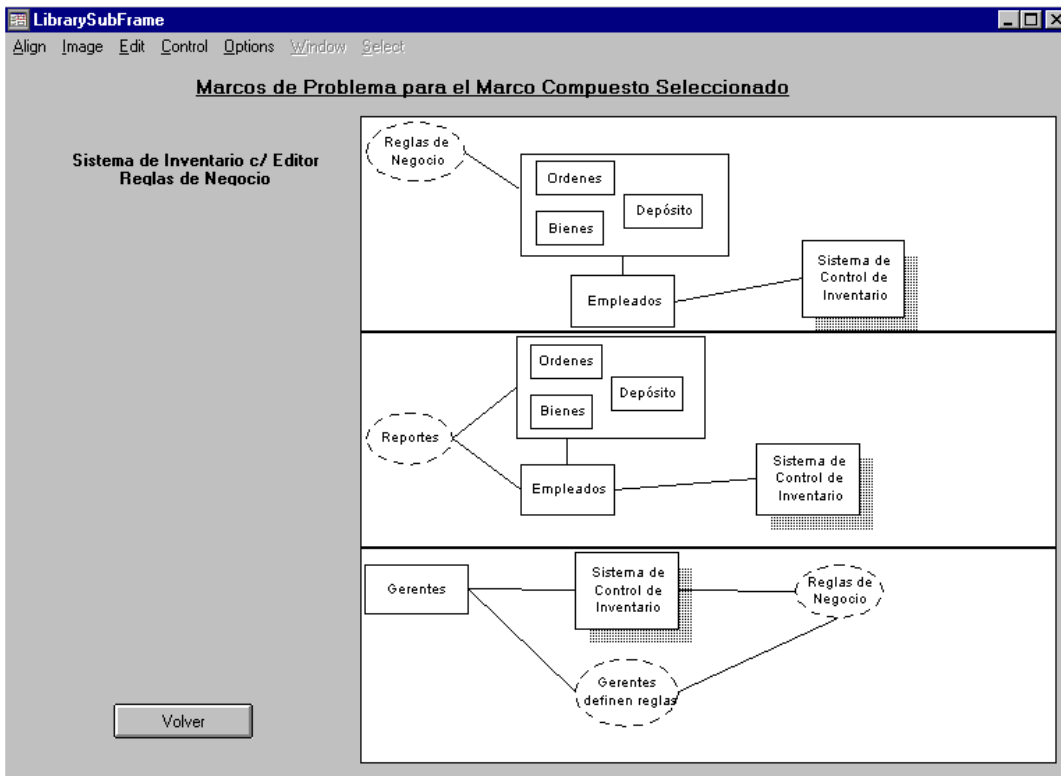


Figura 8.7

Luego de haber descompuesto el problema en diferentes marcos elementales, el sistema provee la asistencia necesaria para verificar si cada uno de ellos se ajusta al problema y está correctamente delineado. Para ello presenta una pantalla que se puede apreciar en la figura 8.8:

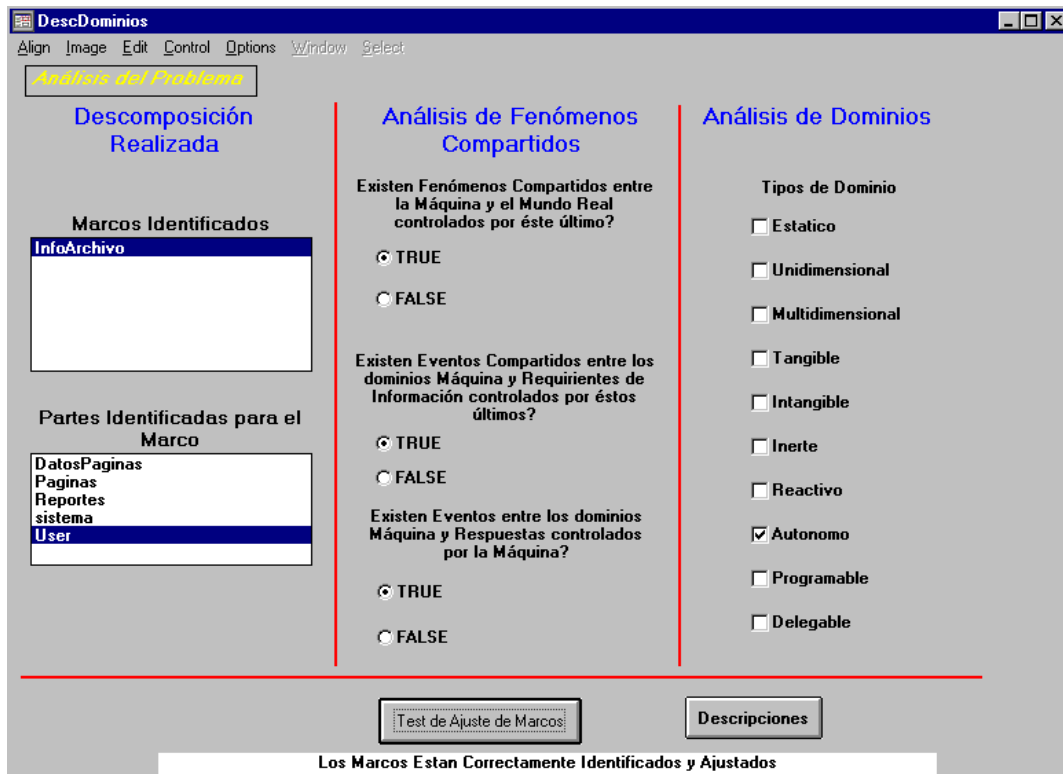


Figura 8.8

En la misma se observan los siguientes elementos:

- Una lista con los marcos elementales identificados con el nombre asignado.
- Para cada uno, una lista con las partes identificadas de dicho marco.
- Una serie de cuestiones acerca de los fenómenos compartidos que deberían observarse en cada marco, donde el analista debe reflexionar acerca de los mismos, sobre los eventos, estados, objetos y su interrelación. Esto ayuda a verificar que el marco de problema haya sido correctamente aplicado en la descomposición.
- Para cada parte del marco que sea un dominio que forma parte del dominio del problema, una lista de tipos de dominio donde el analista debe indicar qué tipo corresponde a la parte seleccionada.

El analista debe seleccionar cada parte y completar los cuestionamientos. Cuando haya finalizado debe presionar el botón "Test de Ajuste de marcos" y el sistema empleando las reglas de producción verificará el correcto ajuste. En caso contrario indicará cuales partes tienen problemas.

A su vez, si se hace click con el botón derecho sobre cada cuestión o sobre cada tipo de dominio, el sistema proveerá una ayuda rápida para su mejor análisis.

Si el/los marcos están correctamente ajustados, lo indicará en la parte baja y luego el analista puede continuar con las descripciones del dominio del problema y los requerimientos, pulsando el botón correspondiente.

La figura 8.9 muestra la pantalla de disparo de las diferentes descripciones que deben realizarse.

Según el marco y el tipo de parte seleccionado el sistema pedirá distintos tipos de descripciones, aplicando distintos criterios y técnicas de descripción.

Cabe acotar que el presente trabajo no constituye una herramienta CASE, sino que es un sistema basado en conocimiento, por lo tanto no se explotan las técnicas gráficas de descripción en si mismas, sino que el sistema recomienda la técnica más conveniente y brinda ejemplos gráficos de la misma.

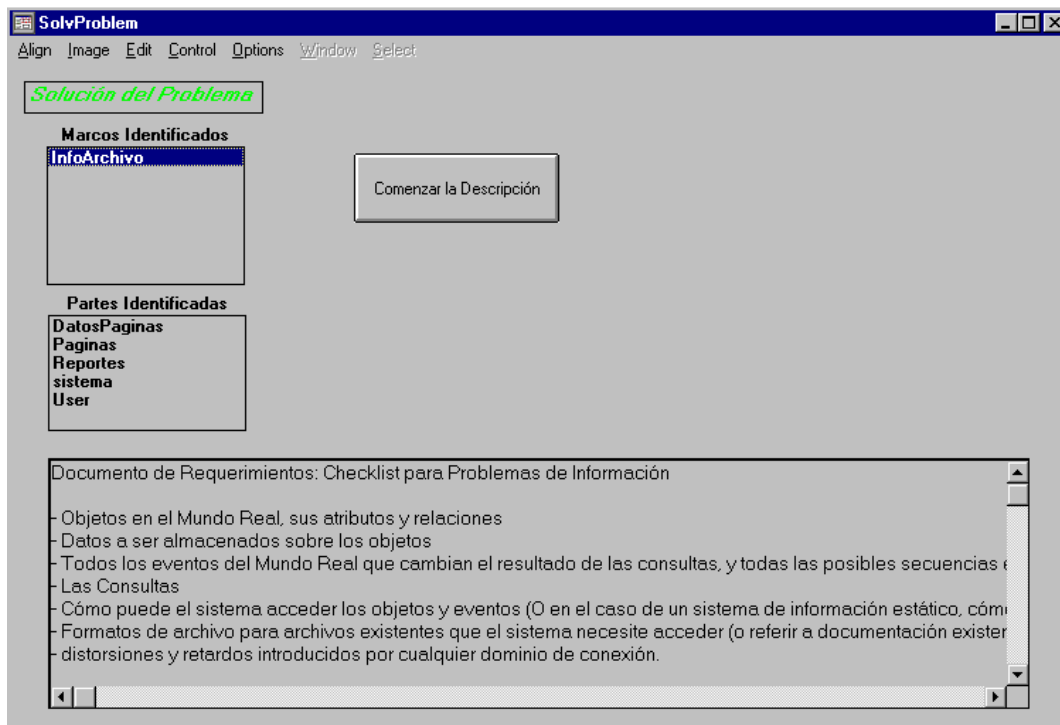


Figura 8.9

En la figura se observa que además el sistema provee de un checklist a modo de guía sobre las distintas descripciones que debe realizarse para el marco de problema elemental seleccionado.

Pulsando el botón "Comenzar la descripción" el sistema pedirá las descripciones necesarias como se indicó anteriormente.

A continuación las figuras 8.10 a 8.16 muestran las diferentes pantallas para describir los distintos dominios, donde en cada una se puede describir:

- Clases (Objetos en el dominio de la aplicación) con sus atributos, etc.
- Eventos
- Estados
- Acciones
- Estructuras y secuencias utilizadas para describir conjuntos de datos.

Requerimientos tales como:

- Consultas
- Mapeos
- Reglas de comportamiento
- Operaciones sobre dominios creados
- Correspondencias entre dominios

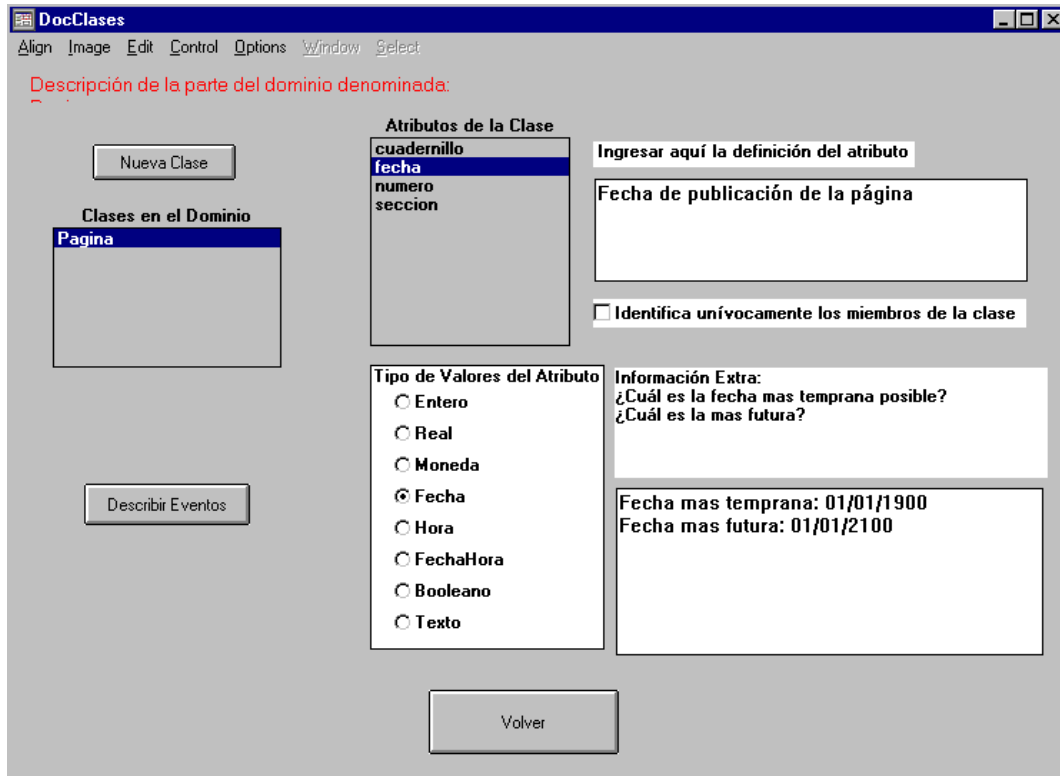


Figura 8.10

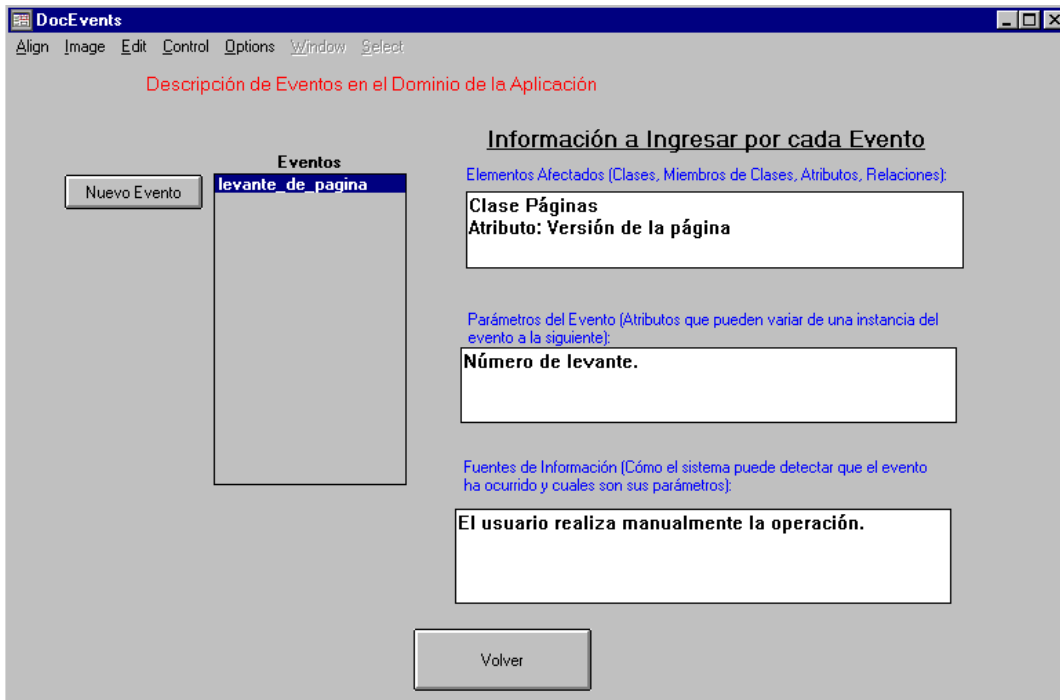


Figura 8.11

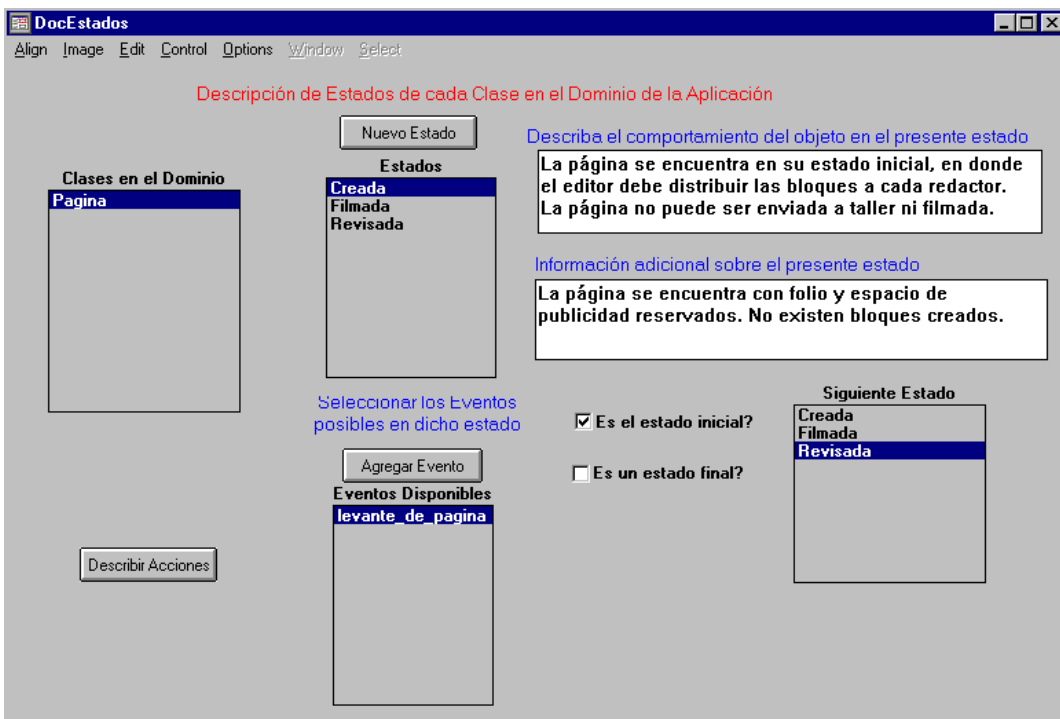


Figura 8.12



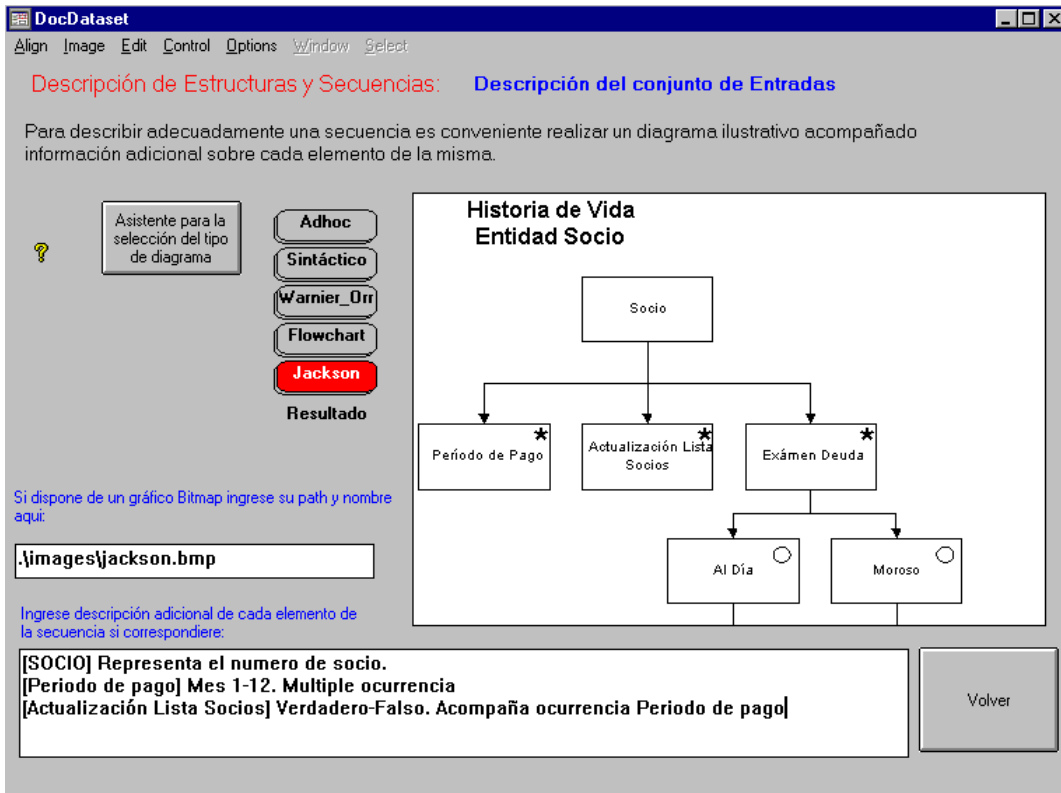


Figura 8.13

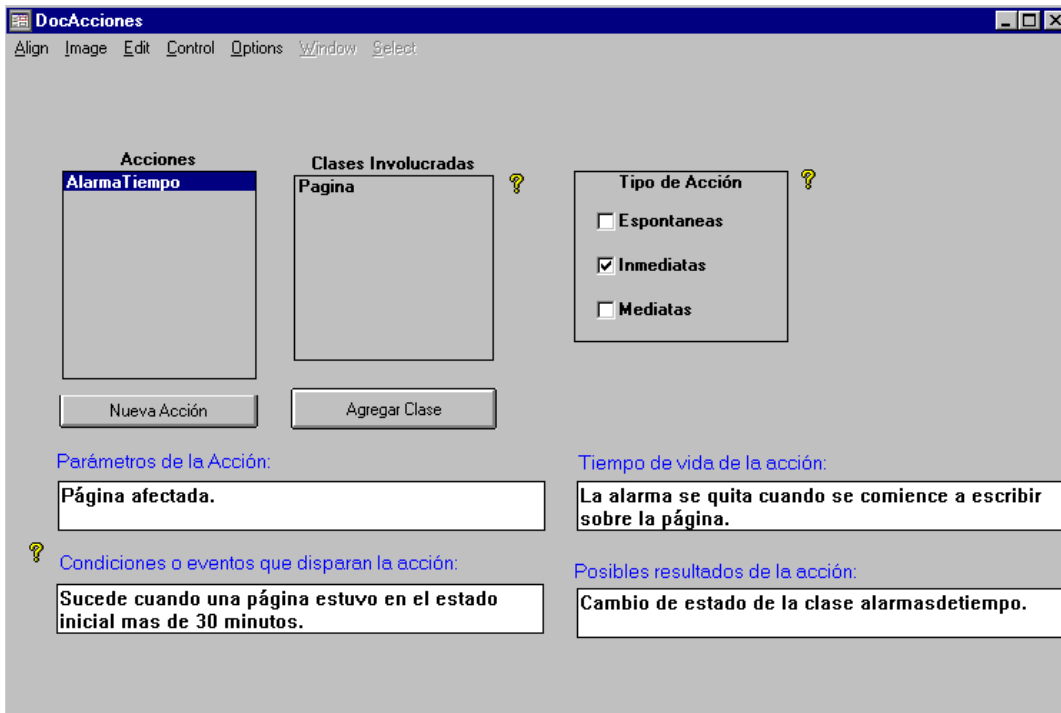


Figura 8.14

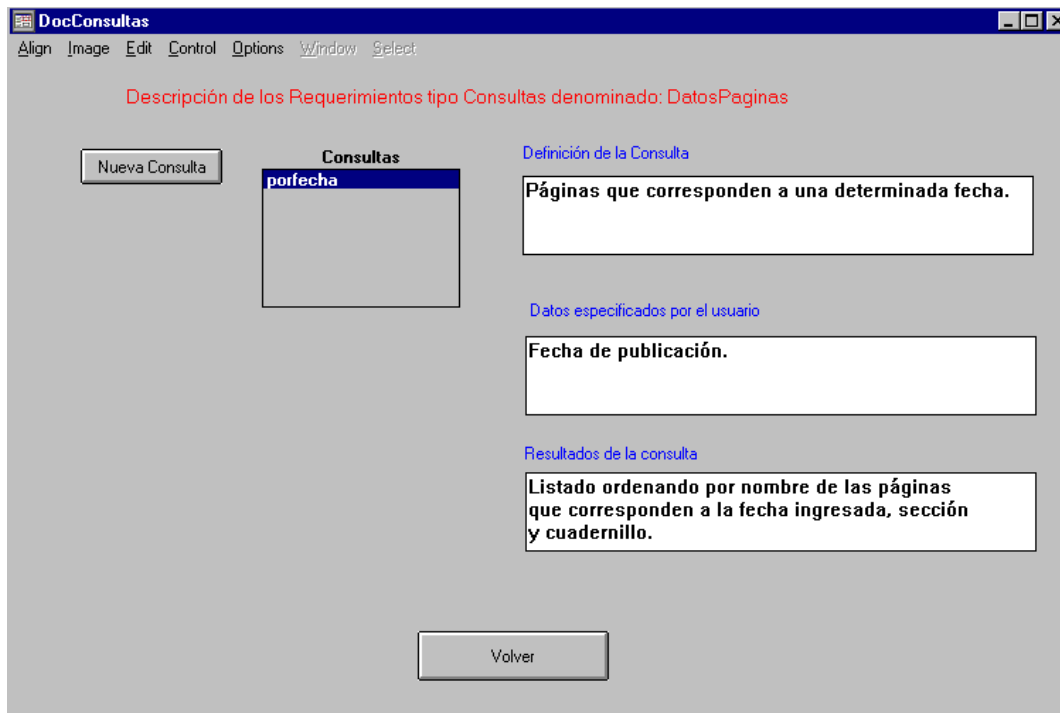


Figura 8.15

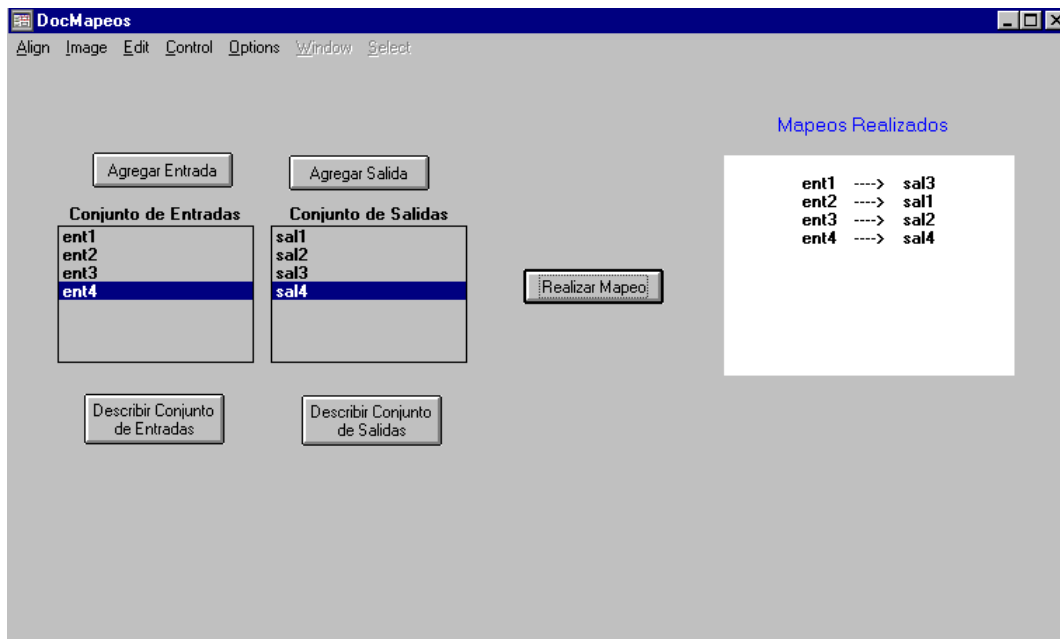


Figura 8.16

El sistema experto permite navegar por estas pantallas de modo de guiar al usuario en las descripciones.

En todo momento se puede controlar mediante una pantalla que oficia de navegador la situación de la descripción, elementos que falta describir, etc. Cuando el analista considera que está todo finalizado, presiona el botón en el navegador para generar el documento de requerimientos en formato HTML.

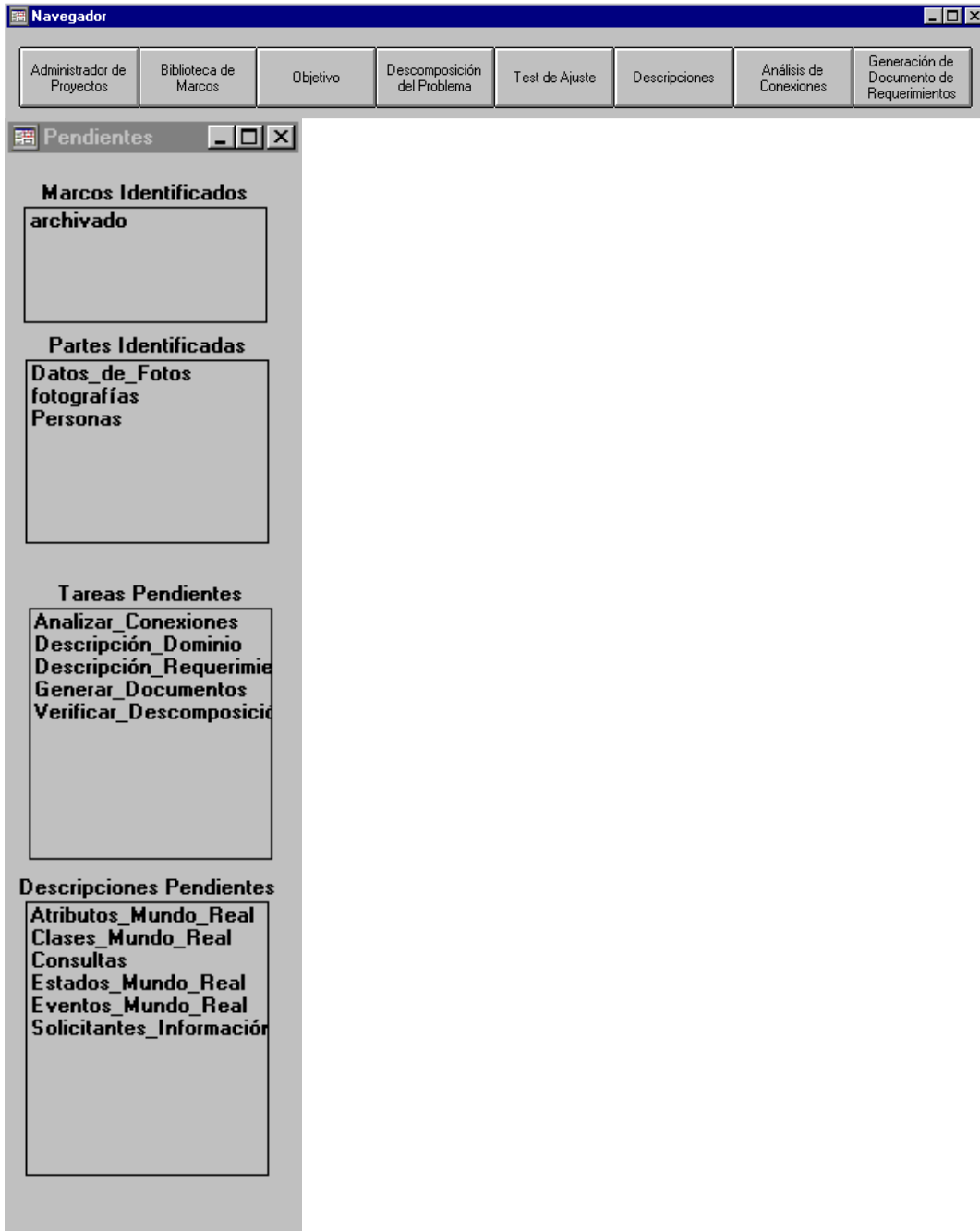


Figura 8.17

El documento de requerimientos resultante puede apreciarse en la siguiente figura (Sólo el comienzo):

# Compañía de Software XYZ

## Proyecto: Sistema de archivo

### Documento de Requerimientos

---

#### **1. OBJETIVO DEL SISTEMA**

El sistema tiene como principal objetivo proveer de información acerca de las publicaciones efectuadas en los últimos 3 años en la editorial.

Debe proveer información sobre textos publicados de forma que puedan ser reutilizados además de una imagen gráfica del mismo.

#### **2. ANALISIS DEL PROBLEMA**

Como resultado del Análisis del Problema se han identificado correctamente los siguientes marcos de problema:

*Figura 8.18*

De ésta forma se ha presentado el funcionamiento global del sistema. Se adjunta al presente trabajo un disco CD-ROM con el código fuente completo en Kappa-PC.

# Capítulo 9

## Evaluación del Sistema Experto

## Capítulo 9

### Evaluación del Sistema Experto

#### 9.1. Introducción.

La evaluación de Sistemas Basados en Conocimiento enfrenta una serie de dificultades que los diferencian de los sistemas convencionales. En primer lugar, ya que los sistemas expertos tratan de resolver problemas que normalmente lo resuelven humanos expertos, los criterios para medir su éxito suelen no estar definidos. En segundo término, los problemas que afrontan los sistemas basados en conocimiento implican grandes espacios de búsqueda que nos son susceptibles de modelización, al menos fácilmente.

Entre las diferencias más importantes con los sistemas convencionales podemos citar:

- a. Los sistemas basados en conocimiento no son completamente objetivos.
- b. Toleran una cierta cantidad de incertidumbre.
- c. No pueden ser fácilmente verificados en el laboratorio.
- d. Ya que modelan conocimientos de los expertos acerca de un dominio, existen variaciones de opinión entre los distintos expertos.

Es por esto que se necesitan diferentes métodos de evaluación para asegurar la confiabilidad de un sistema experto, que si bien no están completamente normalizados ni universalmente aceptados, es necesario llevar adelante la evaluación además de ser una tarea crítica de la que depende en gran parte el éxito del proyecto.

En el proceso de desarrollo de un sistema experto se deben efectuar múltiples evaluaciones<sup>1</sup>, cada una de las cuales con su propio objetivo.

Siempre se deben llevar a cabo, al menos, evaluaciones con los siguientes objetivos:

---

<sup>1</sup> Juristo, N. "Evaluación de Sistemas Expertos", Máster en Ingeniería del Software, Madrid 1996.

- Comprobar que una parte conceptualizada de los objetivos es válida.
- Comprobar que una parte formalizada de los objetivos es válida y correcta.
- Comprobar que un prototipo implementado conteniendo una parte de los conocimientos, es correcto y válido.
- Comprobar que el modelo conceptual de la totalidad de los conocimientos del sistema es válido.
- Comprobar que el modelo formal de todos los conocimientos del sistema es válido y correcto.
- Comprobar que el modelo implementado del sistema es válido y correcto.
- Comprobar que al usuario le satisface la interacción con el sistema.
- Comprobar que el sistema es útil para la organización en la que se ha implantado.

Dada la complejidad de los Sistemas Expertos existen múltiples criterios por los que se pueden evaluar.

No obstante hay cuatro grandes aspectos del sistema que deben evaluarse:

- La corrección del modelo formal y computable.
- La validez del modelo conceptual, formal y computable.
- La usabilidad del sistema, que debe satisfacer al usuario en su interacción con el sistema.
- La utilidad del sistema por haber alcanzado la organización las metas perseguidas con el desarrollo del sistema.

*Un modelo es **correcto** si posee una sintaxis adecuada.*

La corrección se corresponde con estar conforme a las reglas sintácticas del formalismo en que está expresado el modelo. Lo cual hace pensar que solo se puede hablar de la corrección del modelo formal y del modelo computable.

*Un modelo es **válido** si corresponde a una semántica adecuada.*

La semántica está relacionada con el significado de las cosas. Sin embargo, las cosas no tienen un significado absoluto, por el contrario, el significado de las cosas depende del contexto.

Los Sistemas Expertos son inválidos cuando cometen fallos en la semántica, cuando las respuestas que dan no se corresponden con las que darían los expertos.

Además se puede señalar que otro elemento a evaluar es la satisfacción de la necesidad para la que se desarrolla el sistema, dicho de otra manera, que el sistema construido se corresponda con aquel que se quería desarrollar.

Un sistema es **usable** si al usuario le resulta agradable la interacción con el mismo, bajo lo cual se pretende evaluar la relación usuario-sistema

Un sistema experto se puede considerar **útil** si una vez que está en uso rutinario, cumple las expectativas que se tenían de él, aportando las mejoras esperadas.

## **9.2. Verificación del Sistema Experto.**

La verificación del sistema experto consiste en la evaluación de los modelos formales y de los modelos computables con respecto a su sintaxis. Los errores sintácticos de los modelos no se producen por problemas en la comunicación entre los IC y los expertos, si no que se producen, aún habiendo entendido perfectamente los conocimientos del experto, y dominando la sintaxis del formalismo, el IC no puede evitar cometer errores a la hora de transcribir el Modelo Conceptual educido del experto al Modelo Formal y de éste al computable, debido a que no existen reglas formales que permitan pasar de un modelo a otro. Es decir la transformación del Modelo Conceptual al Modelo Formal no puede ser evaluada formalmente, por la carencia antes mencionada.

El proceso de verificación se ha llevado a cabo como sigue:

Primero se ha formalizado e implementado el conocimiento obtenido sobre los Marcos de Problema.

Inmediatamente se realizó la prueba de los diferentes marcos de problema elementales, su identificación de partes, dominios y su ajuste al problema analizado.

Luego se procedió a la formalización e implementación de los conocimientos que se requerían para la descomposición de un problema en marcos de problema elementales. Se introdujeron casos conocidos para luego verificar la adecuación del sistema en la asistencia del análisis.



Juristo también propone un conjunto de criterios para verificar un sistema basado en reglas, los mismos han sido utilizados para verificar el presente sistema. Cabe acotar que obedecen a criterios de redundancia, incompletud e Inconsistencia.

En todos estos casos la valoración consiste en detectar si existen o no dichos elementos, es decir, de forma booleana mediante la inspección del sistema. Es un aspecto *estático*.

Se encontraron algunas reglas que resultaron redundantes, dado que se intentaba ajustar un marco con diferentes reglas que en esencia chequeaban los mismos parámetros del marco.

Finalmente se procedió a formalizar e implementar los conocimientos sobre la resolución de conexiones y descripciones de dominios y requerimientos. Se comprobaron las reglas para chequear la resolución de diferentes tipos de conexiones.

### **9.3. Validación del Sistema Experto.**

La Validación de conocimientos tiende a evaluar los errores semánticos que puedan haber sido introducidos por el IC cuando desarrolló la Base de Conocimientos.

La validez del sistema se juzga con criterios *dinámicos* a saber:

- Adecuación del sistema a los requisitos
- Capacidad para tratar los casos de prueba
- Exactitud en las respuestas

Para desarrollar la validación se trabajó con un entorno controlado. Es decir, se han de reducir todas las posibles situaciones con que se puede enfrentar el sistema experto, a un número de casos de prueba representativo de tales situaciones.

Los casos escogidos representan una serie de problemas paradigmáticos en el mundo del desarrollo de software.

Se pidió a los expertos que realicen el análisis del problema, la descripción del dominio y los requerimientos necesarios sin la ayuda del sistema experto. Luego se pidió a los mismos expertos que realicen nuevamente el análisis y documentación guiados por el sistema experto, comparando finalmente los resultados.

Casos seleccionados:

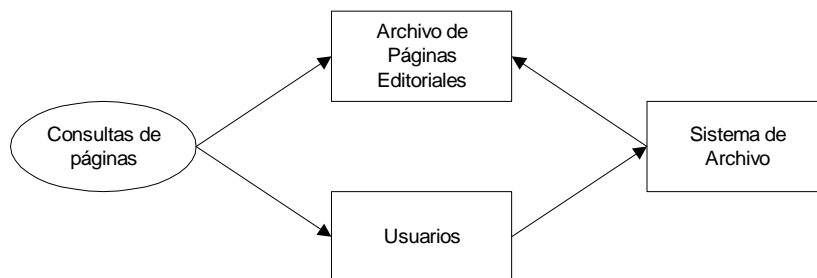
- Sistema de archivo digital de páginas editoriales.
- Sistema de control de semáforos con editor de secuencias.
- Sistema para realizar Estadísticas.
- Sistema de control de inventario.
- Sistema de ruteo automático de paquetes.

Caso 1: Sistema de archivo digital de páginas editoriales.

Este sistema consiste en la construcción de un software que permita brindar información acerca de páginas editoriales publicadas en medios de comunicación gráficos. Se desea obtener información acerca de personajes y temas publicados obteniendo imágenes de la página, texto reutilizable, las diferentes publicaciones donde aparece un determinado tema y/personaje, todo limitado a rangos de fechas. El formato de las páginas será PDF.

En este caso, claramente se observa que se corresponde con un problema de información.

Usuarios actúan como "solicitantes de información" sobre un mundo real modelizado por una representación electrónica de páginas almacenadas en un archivo. La máquina debe recibir esas consultas, interrogar al modelo y luego entregar al usuario la información adecuada. El dominio mundo real modelizado es estático dado que sólo ocurren cambios cuando el sistema actualiza la información allí contenida. No se considera en el problema las dificultades de capturar la información.



*Figura 9.1: Marco de problema de información para las páginas editoriales*

No obstante hay un elemento que no permite aplicar directamente el marco y es la información de texto reutilizable que pide el problema. Aparece aquí un nuevo

marco de problema de Transformación. LA información obtenida en la consulta debe ser transformada desde formato PDF a formato ASCII.

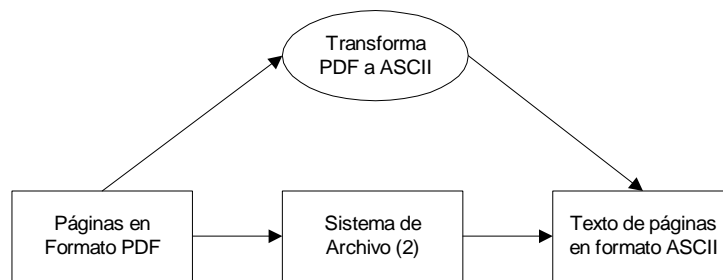


Figura 9.2: Marco de problema de transformación para ;as páginas editoriales

Caso 2: Sistema de control de semáforos con editor de secuencias.

Este sistema debe poder resolver el problema del secuenciamiento de un cruce semaforizado de 2 avenidas, el cual debe poder ser reprogramado para ajustarlo a diferentes ciclos de funcionamiento dependiendo de el día de la semana, estación, temporada y/o eventos especiales.

A primera vista es un problema de control, en el que el dominio controlado son los semáforos y el conjunto de electrónica de control.

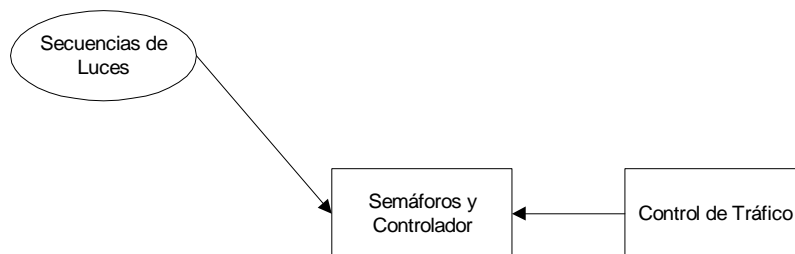


Figura 9.3: Marco de control para los semáforos.

No obstante el problema pide que las secuencias puedan ser ajustadas según varios factores. Esto hace que deba agregarse un problema de Workpieces, que actúa como un editor de las secuencias.

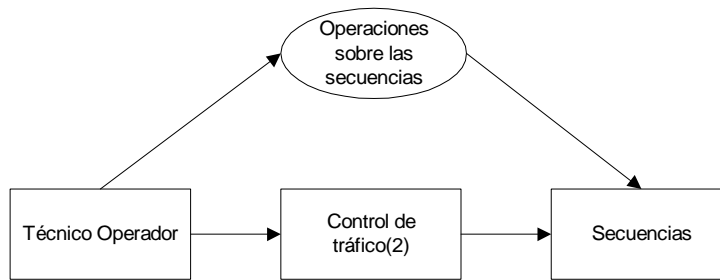


Figura 9.4: Marco de workpieces para los semáforos.

### Caso 3: Sistema para Realizar Estadísticas.

Analizar el problema de un sistema que debe poder realizar cálculos estadísticos complejos y definibles por el usuario.

En éste caso rápidamente se reconoce un problema de transformación, responsable de realizar los cálculos. Sin embargo el usuario debe poder definir las fórmulas utilizadas en dichos cálculos. Esto se logra a través de un marco de Workpieces. Se observa en el siguiente diagrama el marco compuesto detectado por el experto para todo el problema.

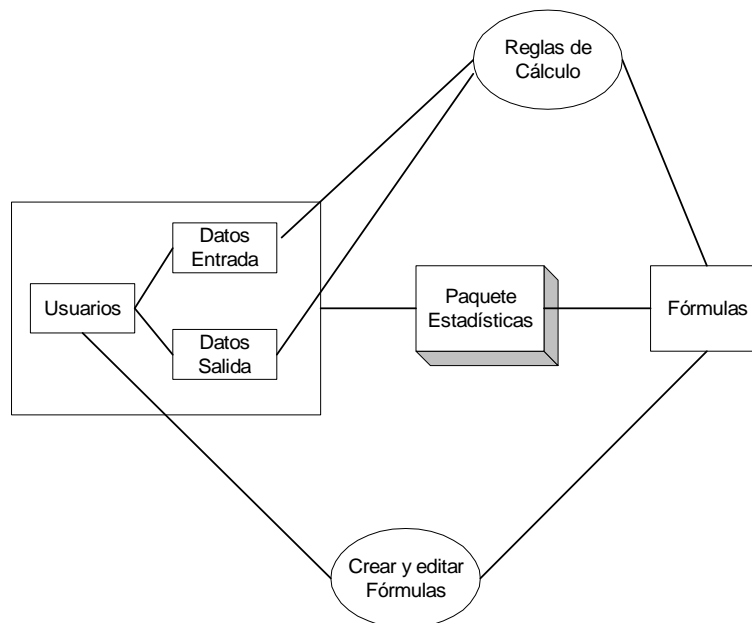


Figura 9.5: Marco compuesto para el paquete de estadísticas.

Los usuarios mediante las reglas de cálculos desea transformar los datos de entrada en datos de salida resultantes llamados estadísticas. No obstante las fórmulas serán creadas también por los usuarios mediante las operaciones se definen en el óvalo correspondiente. Veamos como quedan los marcos resultantes:

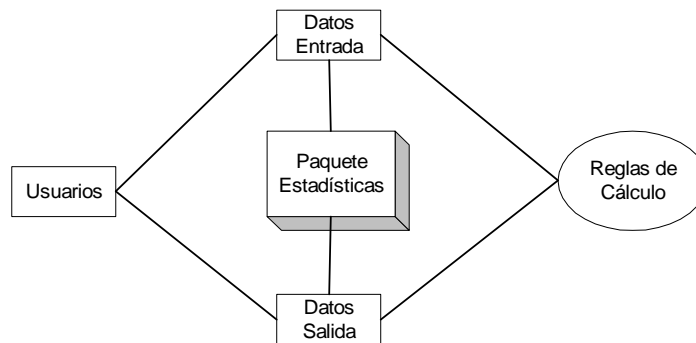


Figura 9.6: Marco de transformación:

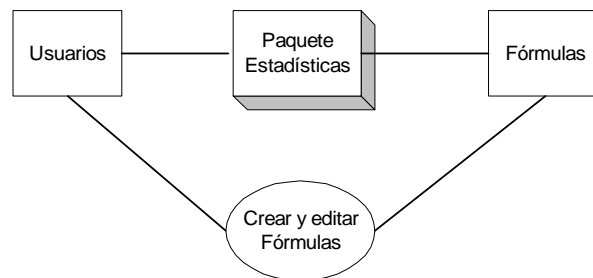


Figura 9.7: Marco Workpieces:

#### Caso 4: Sistema de control de inventario.

Este sistema debe resolver el problema del control de inventario clásico, con movimiento desde y hacia el depósito de bienes.

El principal trabajo de un sistema de control de inventario es el guiado del transporte de bienes desde y hacia un depósito. Debe dirigir a los empleados para que almacenen los bienes a medida que llegan, y dirige los empleados para que busquen los bienes cuando se reciben las ordenes y los envíen a los clientes.

Otra tarea del sistema es reportar sobre la actividad del depósito: Su estado actual, y el flujo pasado de dinero y bienes. Por lo tanto se necesitan dos marcos de problema: Uno de control y otro de información.

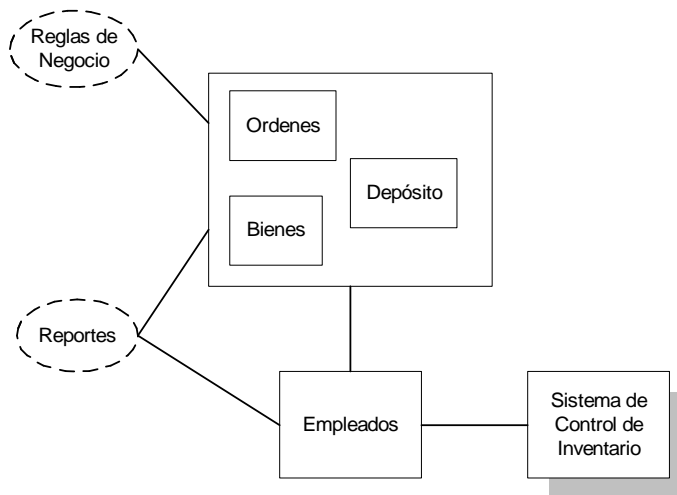


Figura 9.8: marco compuesto del sistema de inventario.

Los empleados son la única conexión directa al sistema. Las reglas de negocio, que definen el flujo de bienes dentro del depósito en respuesta a las órdenes pertenecen sólo a los bienes, depósito y órdenes; no a los empleados. Los empleados son requeridos por el sistema para llevar a cabo las reglas de negocio.

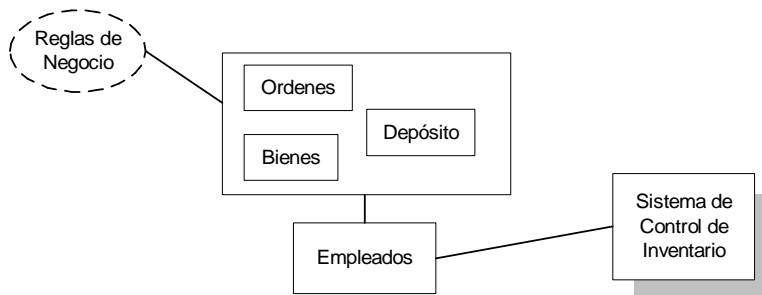


Figura 9.9: Marco de control de inventario

El siguiente diagrama muestra el problema de información. Se debe generar reportes sobre los bienes, depósito y órdenes en respuesta a requerimientos de los empleados. Ellos otra vez son un dominio de conexión. Además de requerir información deben suministrar al sistema la información sobre los bienes, órdenes y depósito.

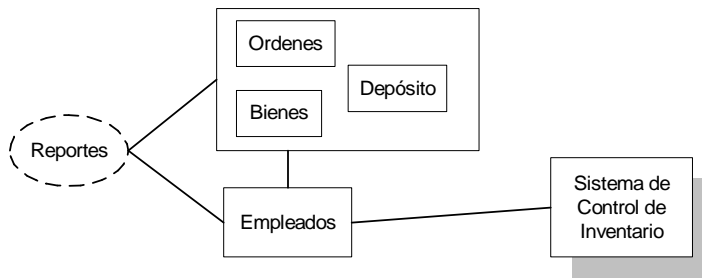


Figura 9.10: Marco de información de inventario

#### Caso 5: Sistema de ruteo automático de cargas.

*"El Router de cargas es un sistema para distribuir bultos en diferentes bandejas destino. Los mismos arriban a una estación de distribución que está conectada a las bandejas a través de tubos. Un único tubo sale de la estación. Los tubos están conectados entre sí por puertas de dos posiciones. Una puerta permite que un paquete que se desliza por el tubo de entrada, sea dirigido a cualquiera de sus dos tubos de salida. Hay un único camino entre la estación de distribución y cualquier bandeja.*

*Los paquetes que arriban a la estación de distribución se escanean por lectores de barra quienes determinan una bandeja de destino para el mismo. El paquete se deja caer luego por el tubo que deja la estación. El sistema debe configurar todas las puertas previo a cada paquete que se desliza a través de los tubos de modo que cada uno sea ruteado hasta la bandeja determinada por la estación.*

*Luego que se ha determinado el destino, se demora el paquete un tiempo fijo antes de ser liberado en el primer tubo. Esto previene que un paquete siga a otro muy de cerca y que las puertas no tengan tiempo de ser modificadas entre paquetes sucesivos cuando sea necesario. Sin embargo si el destino de un paquete es el mismo que el previo, no es demorado dado que no es necesario modificar las puertas entre los dos paquetes.*

*Generalmente habrá varios paquetes deslizando por los tubos al mismo tiempo. Así mismo se deslizan a diferentes e impredecibles velocidades, por lo que es imposible calcular cuando un paquete dado alcanzará una puerta. Sin embargo, las puertas contienen sensores estratégicamente ubicados en su entrada y salida para detectar los paquetes.*

*Los sensores se ubican de manera que la puerta pueda ser cambiada de estado si y solo si no hay paquetes entre su entrada y salida.*

*Debido a las características impredecibles del deslizamiento, es posible que los paquetes estén tan cerca entre sí que no sea posible restablecer una puerta a tiempo para rutear un paquete apropiadamente. Los paquetes mal ruteados pueden ser ruteados a cualquier bandeja, pero no deben causar el mal ruteo de otros paquetes. Las bandejas también tienen sensores en su entrada y en el arribo de un paquete mal ruteado, el sistema debe indicar el la bandeja de destino requerida y la realmente alcanzada."*

Inicialmente este problema aparenta ser esencialmente un problema de control. La maquina debe abrir y cerrar las puertas de modo que los paquetes lleguen a sus

destinos adecuados. La puerta debe ser cambiada de estado cuando un paquete atraviesa el sensor al final del tubo que ingresa a la misma.

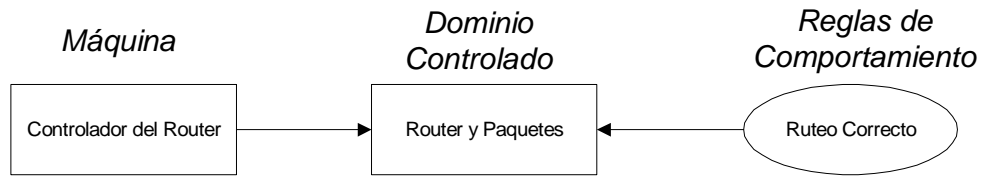


Figura 9.11: Marco de control del router.

La interface entre el controlador de ruteo y el dominio de Router y paquetes es como sigue:

- El dominio Router y paquetes controla los eventos de Lectura en el cual se lee el código de barras de un paquete.
- El dominio Router y paquetes también controla los eventos de pulsado de sensores por parte de los paquetes.
- El dominio Router y paquetes controla el estado que determina la posición física (izquierda, derecha) de las puertas.
- La máquina controladora del router controla los eventos de movimiento de las puertas.

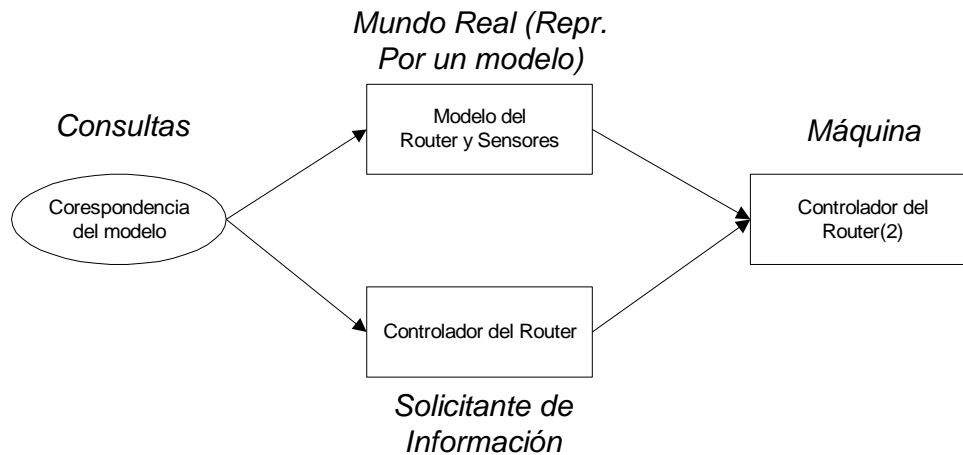
El óvalo de Requerimientos tiene los siguientes fenómenos:

- El evento "arribar" en el cual un paquete llega a una bandeja.
- Tablas de verdad que asocian paquetes a un código de barra de destino, y códigos de barra de destino con su bandeja física correspondiente.

Observando el problema puede advertirse que existe una dificultad de conexión. El controlador del router puede detectar cuando un sensor ha sido tocado, pero no puede saber cual paquete fue.

La dificultad puede ser subsanada por un marco de información.





*Figura 9.12: Marco de información del router*

Se puede considerar los paquetes como si formaran un conjunto de pilas: En cada evento de lectura en la estación se agrega un paquete a la cola de la pila del tubo superior, y en el evento de pulsado de un sensor en el final de un tubo, se mueve desde la cabeza de una pila a la cola de otra. El mundo real representado por un modelo de los "Paquetes y sensores" contiene una representación de esas pilas y un atributo "destino" para cada elemento de la pila. Dicho atributo "destino" se asigna cuando se lee el código de barras del destino del paquete en la estación de lectura.

El modelo puede ser interrogado por el controlador del Router en el problema de control de modo que responda a la siguiente consulta: "Cual es el destino en código de barras del paquete que participó en el mas reciente evento de pulsado en el cuál este sensor particular ha participado?" La respuesta a esta consulta resuelve la dificultad de conexión: Cuando un paquete arriba al sensor de una puerta su destino es conocido.

El problema ofrece también dos ejemplos claros de dificultades de identidad. La primera involucra los sensores y las puertas. Cada sensor y puerta esta conectado a un puerto específico de la máquina. Cuando la máquina detecta un "hit" lo hace en un puerto determinado, pero la identidad del sensor no es explícita. Una dificultad similar ocurre con las puertas.

La segunda dificultad de identidad ocurre con los destinos representados en códigos de barra y las bandejas. Cada bandeja puede ser asociada uno-a-uno con el sensor registrando su entrada, pero la máquina no tiene acceso al mapeo entre los sensores de las bandejas y el string del código. Efectivamente no hay una manera de determinar cual es la bandeja destino para un código de barra determinado.

Ambas dificultades se resuelven de la manera estándar para dificultades de identidad: El mapeo debe hacerse explícito y hacerlo accesible por la máquina. Crear el mapeo es quizás un problema de Workpieces. El mapeo es un problema de Transformación. El mapeo es luego utilizado por el controlador del Router para identificar el destino de cada paquete con el sensor de la bandeja objetivo.

Existe aún otra dificultad de conexión: Cuando un paquete arriba al sensor que precede a una puerta, la máquina debe accionar la misma o nó de acuerdo a la ruta requerida para el paquete. Pero la máquina no tiene acceso a la información de ruteo necesaria: Esto es, no hay manera de determinar cuando una bandeja determinada puede ser alcanzada desde una puerta particular y a su vez cuando por la derecha o por la izquierda.

La solución a esta dificultad es otro modelo, mediante un problema de información.

Este modelo provee toda la información necesaria sobre la topología del router. Cuales son los sensores que están al final de cuales tubos. Cuales tubos preceden a cuales puertas. Cuales tubos siguen a cuales puertas. Cual tubo sale de la estación. A partir de esta información el Controlador del Router puede determinar la ruta de cada paquete hasta su bandeja de destino.

### **9.3.1 Validación con usuarios no expertos.**

Para completar el proceso de validación se realizó un experimento para obtener alguna medida de la experticidad del sistema.

Para ello se tomaron los casos anteriormente descritos y se seleccionaron algunas personas no expertas para que los resuelvan, previo aprendizaje de los marcos de problema, primero sin el sistema y luego utilizando el sistema experto. Con esto lo que se pretendió es que el sistema fuera utilizado por otros usuarios distintos a los expertos que ayudaron a construirlo.

En todos los casos se obtuvieron mejores resultados mediante el uso del sistema. En algunos casos no se describió completamente el dominio de la aplicación. En otros se falló en la identificación de los distintos tipos de requerimientos, se definieron en forma mezclada y como consecuencia faltaban requerimientos.

### **9.4. Usabilidad del Sistema Experto.**

La usabilidad pretende evaluar al sistema desde el punto de vista de la relación entre el usuario y el software. Es obviamente un criterio subjetivo.

En nuestro caso se invitó a utilizar el sistema a un grupo de estudiantes avanzados del último curso de ingeniería de sistemas observando cuán cómodos se sentían utilizando el mismo. Se recibieron importantes críticas en varios aspectos a saber:

- Mayor asistencia en el aprendizaje del modelo de marcos de problema.
- Incorporación de mecanismos de chequeo de entidades en el dominio que permita verificar la unicidad de nombres.
- Incorporación de bibliotecas de modelo de datos y eventos comunes en diferentes dominios típicos de aplicación.

Este último punto estaría implementado en cierta manera en la biblioteca de marcos compuestos resueltos en los que se puede utilizar proyectos anteriores viendo cómo se resolvió la descripción. El sistema actúa como agente de archivo de casos con descripciones incluidas.

### **9.5. Utilidad del Sistema Experto asistente de requerimientos.**

Según Juristo, la utilidad de un Sistema Experto tiene que ver con la relación del nuevo sistema, formado por el usuario, el SE, y la organización a la que pertenece. Al igual que ocurre con la usabilidad no es posible determinar en estos momentos, basándose en su uso práctico, el grado de utilidad que brindará el sistema a la organización.

No obstante es importante mencionar que el hecho de disponer de descripciones de dominios tiene una gran utilidad para el desarrollo de otros productos software sobre el mismo dominio de aplicación. Se evita una etapa costosa al tener disponible la descripción del dominio de la aplicación para su reuso.

# Capítulo 10

Conclusiones y  
Futuras Líneas  
de investigación

## Capítulo 10

### Conclusiones y futuras líneas de investigación

#### 10.1 Introducción

La tesis de Máster en Ingeniería de Software es el corolario del aprendizaje recibido durante el posgrado. En él se ha intentado plasmar todos los conocimientos recibidos tanto en la etapa de ingeniería del software como en la de ingeniería de conocimiento. Su desarrollo ha permitido poner en práctica aspectos de gestión de proyectos, estimaciones, evolución de prototipos, diseño, prueba, codificación, análisis de riesgos, etc. A su vez se pudo cristalizar la concreción de un trabajo puramente profesional, con los detalles y particularidades que ello encierra, tales como relaciones sociales con usuarios, expertos, incertidumbres, errores de estimación y de apreciación de conceptos, entre otros.

A continuación se exponen las diferentes conclusiones y seguidamente un propuesta sobre futuras líneas investigativas para evolucionar el sistema experto desarrollado.

#### 10.2 Conclusiones del trabajo de tesis

La utilización de una metodología que promueve el desarrollo incremental de prototipos, como es la metodología I.D.E.A.L., procura que la culminación del proyecto se dé en tiempos y costos acordes a los estimados.

Si bien el dominio de la aplicación del sistema experto es complejo, se pudo establecer mediante el test de viabilidad que su concreción era posible.

La libertad de elección de diferentes técnicas de educación que promueve la metodología ha permitido capturar satisfactoriamente las tareas que desarrolla el experto, sus razonamientos y heurísticas.

La etapa de formalización ha permitido acercar el modelo conceptual al modelo computacional de una manera muy analítica, lo que redundó en un sistema utilizable sin mayores complicaciones.

La herramienta Kappa-PC resultó adecuada para el tipo de sistema a construir, sobre todo en la modelización de las interfaces de usuario prototipadas en forma evolutiva con los expertos. Además la herramienta se adecuó perfectamente a la información recibida de la etapa de formalización.

El tema seleccionado para el presente trabajo implicó la unión de las dos grandes áreas temáticas que comprende el programa de máster: Realizar un Sistema Basado en Conocimiento cuyo dominio de aplicación existe en una de las áreas mas importantes de la Ingeniería del Software: Los Requerimientos de sistemas software convencionales. Esta unión fue muy productiva para el autor, dado que permitió profundizar en el conocimiento de la Ingeniería de Requisitos a la vez que se ponía en práctica la construcción de un sistema experto.

Durante la evaluación, se citó también a personas no expertas para que lo utilizaran en algunos de los casos citados en el capítulo 9. como resultado, el sistema ayudó a descubrir aspectos del problema analizado que de otro modo hubieran quedado sin tratar por parte de los usuarios.

### **10.3 Futuras líneas de investigación y desarrollo**

Habiendo completado el desarrollo propuesto del sistema experto que asiste en la tarea de la elaboración del documento de requerimientos de un sistema software, quedan abiertas las siguientes líneas investigativas y de desarrollo:

- Obtención de nuevas y mejores heurísticas en el área de descomposición del problema.
- Investigar y detectar nuevos tipos de marcos de problema, de modo de incorporarlos en el sistema actual.
- Trabajar sobre la segunda etapa del sistema, en la que se realizan las descripciones, mediante la incorporación de una biblioteca de patrones y modelos de dominios de aplicación. El sistema entonces asiste al usuario mediante preguntas sobre el dominio y luego razonando sobre dicha biblioteca de modo de proveer al analista de la información necesaria para que realice una completa descripción sin olvidar detalles.
- Esta biblioteca equivale a incorporar conocimiento propio del dominio de aplicación. Esta es una de las partes más importantes que debe contener un

sistema experto sobre requerimientos, tal como lo señala Alan Davis en su libro [Davis 1].

- Completar la etapa de descripciones mediante la asistencia en la creación de la especificación de requisitos. Los marcos de problema sirven además como modelo y guía para la confección del documento especificación de requisitos.

# **Bibliografía**



## Bibliografía

[Bjorner 1] Dines Bjorner, Souleymane Koussoube, Roger Noussi and Gueorgui Satchok, Jackson's Problem Frames: Domains, Requirements and Design; UNU/IIST Report No. 102, United Nations University, may 1997.

[Brooks 1] Brooks Jr., Frederick P., The Mythical Man-Month, Addison-Wesley, 1995.

[Burg] Analyzing Informal Requirements Specifications: A First Step towards Conceptual Modeling. J.F.M. Burg and R.P. van de Riet. Applications of Natural Language to Information Systems. IOS PRes, 1996.

[Cauvet] Alecsi: An Expert System for Requirements Engineering. C. Cauvet, c. Proix, C. Rolland. Université Paris, 1 IAE.

[Coad] Object Models. Strategies, Patterns, and Applications. Peter Coad. 528 pages. Feb. 1995.

[Davis 1] Davis, Alan. M, Software Requirements, Objects, functions & states, Prentice-Hall, 1993.

[Fickas] Critiquing Software Specifications. Stephen Fickas and P. Nagarajan. IEEE Software, November 1998, Pg. 37.

[Haumer 1] Haumer, Peter, Requirements Elicitation and Validation with Real World Scenes, IEEE Transactions on Software Engineering, Vol. 24 No. 12, December 1998.

[Hofmann] Requirements Engineering, A Survey of Methods and Tools. Hubert F. Hofmann. IFI: Institut Fur Informatik der Universitat Zurich. Nr. 93.05, Marz, 1993.

[Hooks 1] Why Johnny can't Write Requirements. Ivy Hooks, Paper at AIAA conference, 1990.  
(<http://www.complianceautomation.com/whyjohnny.html>)

[Hooks 2] Writing Good Requirements. Paper given at the 4th INCOSE Symposium. Ivy Hooks, 1994.  
(<http://www.complianceautomation.com/writingreqs.html>)

[IEEE 830] IEEE std 830-1998, IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, Sponsored by the Software Engineering Standards Committee. October 1998.

[Ip] A Knowledge-Based Requirement Engineering Assistant. Saimond Ip, Louis Cheung, Tony Holden. CASE: Current Practice, Future Prospects. Cap. 14. 1992, John Wiley & Sons, Ltd.

[Jackson 1] Jackson, Michael, Software Requirement & Specification, Addison-Wesley, ACM Press, 1995.

[Jackson 2] Jackson, Michael, Problem Analysis Using Small Problem Frames, South African Computer Journal 22; Special Issue on WOFACS'98, pp47-60, 1999

[Jackson 3] Jackson, Michael, Problem Analysis and Structure, Keynote Talk at ITG/SEV Symposium, Zürich, 29 September 1999

[Jackson 4] Jackson Michael And Jackson Daniel, Problem Decomposition for Reuse, Enero 1995.

[Jarke] Metamodels for REquirements Engineering. Matthias Jarke, Nature Team. InformatikV,RWTH,Aachen.  
(<http://ksi.cpcs.ucalgary.ca/KAW/KAW96/jarke/jarke.html>)

[Jones] Interfacing Requirements Management Tools In The Requirements Managements Process - A First Look. David A. Jones, Pradip Kar. Proceedings of the 7th annual International Symposium of the INCOSE. August 1997.

[Kappa 1] Kappa PC, Quick Start, Intellicorp, Inc. 1992.

[Kappa 2] Kappa PC, User's Guide, Intellicorp, Inc. 1992.

[Kappa 3] Kappa PC, Advanced Topics, Intellicorp, Inc. 1992.

[Kar] Characteristics of Good Requirements. Paper given ath the 6th INCOSE Symposium. Pradip Kar and Michelle Bailey, 1996.  
(<http://www.complianceautomation.com/incose.html>)

[Kovitz 1] Kovitz, Benjamin L., Practical Software Requirements, Manning, 1999.

[Lamsweerde] Managing Conflicts in Goal-Driven Requirements Engineering, Axel Van Lamsweerde, IEEE. Transactions on Software Engineering, Vol. 24, No. 11, November 1998.

[Minsky, 75] Minsky, Marvin. "A framework for representing knowledge". En P. Winston. "The Psicology of Computer Vision", Mc graw Hill. Nueva York (Estados Unidos). 1975.

[Pazos 1] Pazos, J. "Análisis de Viabilidad en Sistemas Basados en Conocimiento", Máster en Ingeniería de software, Universidad Politécnica de Madrid, Instituto Tecnológico de Buenos Aires, 1997.

[Reubenstein] The Requirements Apprentice: Automated Assistance for Requirements Acquisition. Howard B. Reubenstein and Richard Waters. IEEE Transactions on Software Engineering, Vol 17, No 3, March 1991.

[Robertson] Volere: Requirements Specification Template. Edition 6.0. James & Suzanne Robertson. 1998, Atlantic Systems Guild.

[Shaw 1] Shaw, Mary, Problem Definition with Problem Frames, Carnegie Mellon Computer Science school. 1998.

[Sommerville 1] Sommerville, Ian, Software Engineering, Addison-Wesley, 1996.

[Sutcliffe 1] Sutcliffe, Alistair and Maiden, Neil, The Domain Theory for Requirement Engineering, IEEE Transactions on Software Engineering, Vol. 24 No. 3, March 1998.

[Sutcliffe 2] Sutcliffe, Alistair and Maiden, Neil, Supporting Scenario-Based Requirement Engineering, IEEE Transactions on Software Engineering, Vol. 24 No. 12, December 1998.

[Thayer] Software Requirements Engineering, Second Edition. Richard Thayer and Merlin Dorfman, IEEE Computer Society, 1997.

[Wiegiers 1] Wiegiers, Karl E., Software Requirements, Microsoft Press, Best Practices, 1999.

[Zave 1] *Zave, Pamela and* Jackson, Michael, FOUR DARK CORNERS OF REQUIREMENTS ENGINEERING, 16 July 1996.

[Zehtab] Knowledge Representation for Software Development. Peyman Zehtab. Submission to USCCS'97, Sweden.