

Definición de un Proceso de Elicitación de Objetivos

Autor: Pablo Javier Thomas

Director: Alejandro Oliveros

**Tesis presentada a la Facultad de Informática de la
Universidad Nacional de La Plata como parte de los requisitos
para la obtención del título de Magister en Ingeniería de
Software.**

La Plata, Octubre de 2005

**Facultad de Informática
Universidad Nacional de La Plata
Argentina**

Indice

Prólogo	4
Capítulo1. Introducción	7
1.1 Ingeniería de Requerimientos.....	7
1.2 Procesos de Ingeniería de Requerimientos.....	9
1.2.1 Elicitación de Requerimientos.....	10
1.2.2 Especificación de Requerimientos.....	11
1.2.3 Validación de Requerimientos.....	13
1.3 Técnicas de Elicitación de Requerimientos.....	14
1.3.1 Técnicas de Elicitación según Loucopoulos.....	14
1.3.1.1 Originadas en el usuario.....	14
1.3.1.2 Escenarios.....	15
1.3.1.3 Análisis de Objetivos.....	16
1.3.1.4 Análisis de Formularios.....	16
1.3.1.5 Lenguaje Natural.....	16
1.3.1.6 Reuso de Requerimientos.....	17
1.3.1.7 Análisis de Tareas.....	17
1.3.2 Técnicas de Elicitación según Nuseibeh y Easterbrook....	18
1.3.2.1 Técnicas Tradicionales.....	18
1.3.2.2 Técnicas de Elicitación Grupales.....	18
1.3.2.3 Prototipos.....	18
1.3.2.4 Técnicas Orientadas por Modelos.....	19
1.3.2.5 Técnicas Cognitivas.....	19
1.3.2.6 Técnicas Contextuales.....	20
Capítulo 2. Escenarios	21
2.1 Introducción.....	21
2.2 Léxico Extendido del Lenguaje.....	21
2.3 Características de los Escenarios.....	23
Capítulo 3. Análisis de Objetivos	27

3.1	Introducción.....	27
3.2	Enfoques orientados a Objetivos.....	28
3.2.1	Enfoque de Annie Antón.....	30
3.2.1.1	Heurísticas.....	37
3.2.1.2	Resumen.....	40
3.2.2	Enfoque de Axel van Lamsweerde.....	41
3.2.3	Enfoque de Mylopoulos J., Chung L., Liao S., Wang H., Yu E.....	45
3.2.4	Enfoque de Loucopoulos P. , Karakostas V.....	48
3.2.5	Enfoque de Colette Rolland, Souveyet C., Ben Achour C.	50
3.2.6	Enfoque de Oshiro K., Watahiki K., Saeki M.....	53
3.3	Resumen comparativo.....	55
	Capítulo 4. Elicitación de Objetivos.....	58
4.1	Análisis de los enfoques presentados.....	58
4.2	Propuesta de Trabajo.....	65
	Capítulo 5. Proceso de Elicitación de Objetivos.....	68
5.1	Estrategia para obtener Objetivos.....	68
5.2	Definición de las Reglas.....	69
5.2.1	Reglas aplicadas a la Identificación de Escenarios.....	70
5.2.2	Reglas aplicadas a los Componentes de Escenarios.....	72
5.2.3	Reglas aplicadas a los Episodios de Escenarios.....	77
5.2.4	Reglas aplicadas a las bifurcaciones de Escenarios.....	81
5.3	Resumen de Reglas.....	84
5.4	Algoritmo para la aplicación de las Reglas.....	87
5.5	Análisis del caso de estudio.....	88
	Capítulo 6. Conclusiones y Trabajos Futuros.....	90
	Anexo.....	92
	Bibliografía.....	106

Prólogo

Es difícil plasmar en pocas páginas todos los detalles y consideraciones que permitieron elaborar esta tesis. Trataré de reflexionar en lo más relevante.

En primer lugar, es destacable la creciente importancia de la Ingeniería de Requerimientos en la construcción de Software. Años atrás, tanto la literatura como la industria del Software, sólo consideraban la Ingeniería de Software como un todo omitiendo la Ingeniería de Requerimientos como disciplina.

Uno de los propósitos más importantes de la Ingeniería de Requerimientos es obtener una Especificación de Requerimientos que establezca claramente "qué" se quiere resolver, minimizando los errores de interpretación acerca del dominio del problema. Gran parte de esto está relacionado con la Elicitación, etapa inicial en todo proceso de Requerimientos.

Diferentes técnicas son aplicables para obtener la Especificación de Requerimientos de Software. La técnica de Metas u Objetivos es una de la más aceptada dentro de la comunidad de Ingeniería de Requerimientos y posee una larga tradición en el campo de los Sistemas de Información [Davis87] [Kendall91].

El propósito inicial fue analizar los enfoques orientados a objetivos más importantes y establecer un denominador común (en este caso, una carencia), constituyendo así un elemento esencial sobre el cual se pudo elaborar este trabajo.

Posteriormente, se eligió un enfoque sobre el cual suplir esta carencia. Se seleccionó GBRAM, ya que es un enfoque ampliamente reconocido, referente en el tema de objetivos y con bibliografía ampliamente disponible. Pero lo más sugestivo fue que el esquema de los objetivos obtenidos con GBRAM conducían, en algún sentido, a pensar en Escenarios. Esta intuición despertó un interés especial y fue quizás una de las mayores motivaciones.

A partir del hallazgo de esta "similitud", se analizaron diversos escenarios (particularmente todos los escenarios de [Bertolami01] y [Leite96]) y desde ellos se planteó la generación de objetivos con el esquema GBRAM.

Con esa meta, se plantearon un conjunto de reglas de identificación de las componentes de los Objetivos basadas en las componentes de los Escenarios.

Esas reglas se clasificaron en término de las componentes de Escenarios y salvo excepciones, resultan sencillas de operar. Además, poseen la ventaja de ser explicitadas en términos suficientemente rigurosos para la Ingeniería de Requerimientos, a pesar de no ser formales. Se aplicaron a todos los escenarios que previamente se habían analizado y en todos los casos el resultado fue satisfactorio.

Por último, se definió un mecanismo de aplicación de reglas, eliminando cualquier duda respecto a cómo y cuándo aplicar cada una de ellas. Este mecanismo es un algoritmo en pseudocódigo, simple, pero necesario para la aplicación de reglas.

Los resultados obtenidos confirman fuertemente una *analogía semántica* entre la información contenida en los Escenarios y la información necesaria para representar Objetivos.

Es probable que pueda juzgarse que la intuición de similitud semántica entre el enfoque de Escenarios y Objetivos no sea nueva, pero

no se han hallado trabajos que muestren o comprueben fehacientemente dicha analogía, y es precisamente en esta tesis donde se intenta dar un sustento razonable a esa similitud.

En resumen, el presente trabajo refleja la posibilidad de identificar Objetivos a partir de Escenarios, estableciendo una metodología de Elicitación de Objetivos a partir de Escenarios. Por lo tanto, la contribución realizada es la definición de un ***Proceso de Elicitación de Objetivos***.

Capítulo 1

Introducción

1.1 Ingeniería de Requerimientos

La **Ingeniería de Software** comprende la aplicación de principios científicos para realizar la transformación ordenada de un problema en una solución elaborada de software, y el mantenimiento subsecuente de ese software hasta el final de su vida útil [Davis93].

La Ingeniería de Software es más que programar. El proceso de Ingeniería de Software comienza generalmente mucho antes que la escritura de líneas de código y continúa por mucho tiempo después de la finalización de la versión inicial del programa [Davis93].

Las personas y proyectos que adoptan un enfoque ingenieril para el desarrollo de software, generalmente atraviesan una serie de fases o estados.

En el recorrido de esa serie de fases o estados, comúnmente conocido como ciclo de vida del desarrollo del software, la fase inicial comprende a la Ingeniería de Requerimientos. La Ingeniería de Requerimientos tiene que ver con aquellas actividades en pos de entender exactamente las necesidades de los usuarios de un Sistema de Software y traducir tales necesidades a un conjunto de sentencias precisas, no ambiguas, las cuales serán usadas para el desarrollo del Sistema [Loucopoulos95].

El ciclo de vida de desarrollo de software es considerado como una secuencia de procesos.

Un **Proceso** es una serie de pasos que incluye actividades, restricciones y recursos para producir un determinado resultado esperado. Los procesos son importantes porque imponen consistencia y estructura sobre un conjunto de actividades, manteniendo un nivel de consistencia y calidad en los productos o servicios que son producidos [Pfleeger2002].

Cada proceso puede ser escrito en una variedad de formas, utilizando texto, gráficos o una combinación de ambos. Además, todo proceso debe ser adaptado para la situación especial en que será utilizado.

Antes de planificar un proyecto de software, se deberían establecer sus **objetivos** y su ámbito de aplicación, considerando soluciones alternativas e identificando tanto dificultades técnicas como de gestión. El desarrollador de software y el cliente deben reunirse para definir los **objetivos** del proyecto y su ámbito. En muchos casos, esta actividad comienza como parte del proceso de ingeniería del sistema y continúa como el primer paso en el análisis de los requisitos del software [Pressman98].

Los objetivos identifican las metas generales del proyecto sin considerar como se conseguirán. Una vez que se han comprendido los objetivos y el ámbito del proyecto, se consideran las soluciones alternativas [Pressman98]. Además, los objetivos establecidos permiten posteriormente formular los requerimientos del Sistema de Software.

El propósito de esta tesis es definir un proceso de elicitación de objetivos, como parte del proceso de Ingeniería de Requerimientos, en el ciclo de vida de desarrollo de software.

Esta tesis está organizada de la siguiente manera: el resto del capítulo 1 describe los procesos que abarca la Ingeniería de Requerimientos, con especial énfasis en la Elicitación de Requerimientos, principalmente en las técnicas más importantes que se utilizan para la elicitación. El capítulo 2 describe el enfoque de Escenarios para la elicitación de Requerimientos. El capítulo 3 contempla los distintos

enfoques de Objetivos para la elicitación de Requerimientos, con especial énfasis en la propuesta de Annie Antón. En el capítulo 4 se analizan los principales enfoques orientados a objetivos desde el punto de vista de Elicitación de Objetivos. En el capítulo 5 se presentan reglas para elicitar objetivos, y cómo esas reglas pueden ser aplicadas constituyendo un proceso de elicitación de objetivos. El capítulo 6 resume las conclusiones y los trabajos futuros a partir de esta propuesta.

1.2 Procesos de Ingeniería de Requerimientos

El Proceso de Ingeniería de Requerimientos sobre el que se fundamenta este trabajo de tesis es el propuesto por [Loucopoulos95]. Este modelo de proceso abarca la consideración de tres aspectos fundamentales:

- Comprender el problema
- Describir el problema
- Acordar sobre la naturaleza del problema

Además, plantea la ejecución de tres etapas para comprender, describir y acordar. Estas etapas, presentadas en la Figura 1.1, son :

- Elicitación de Requerimientos
- Especificación de Requerimientos
- Validación de Requerimientos

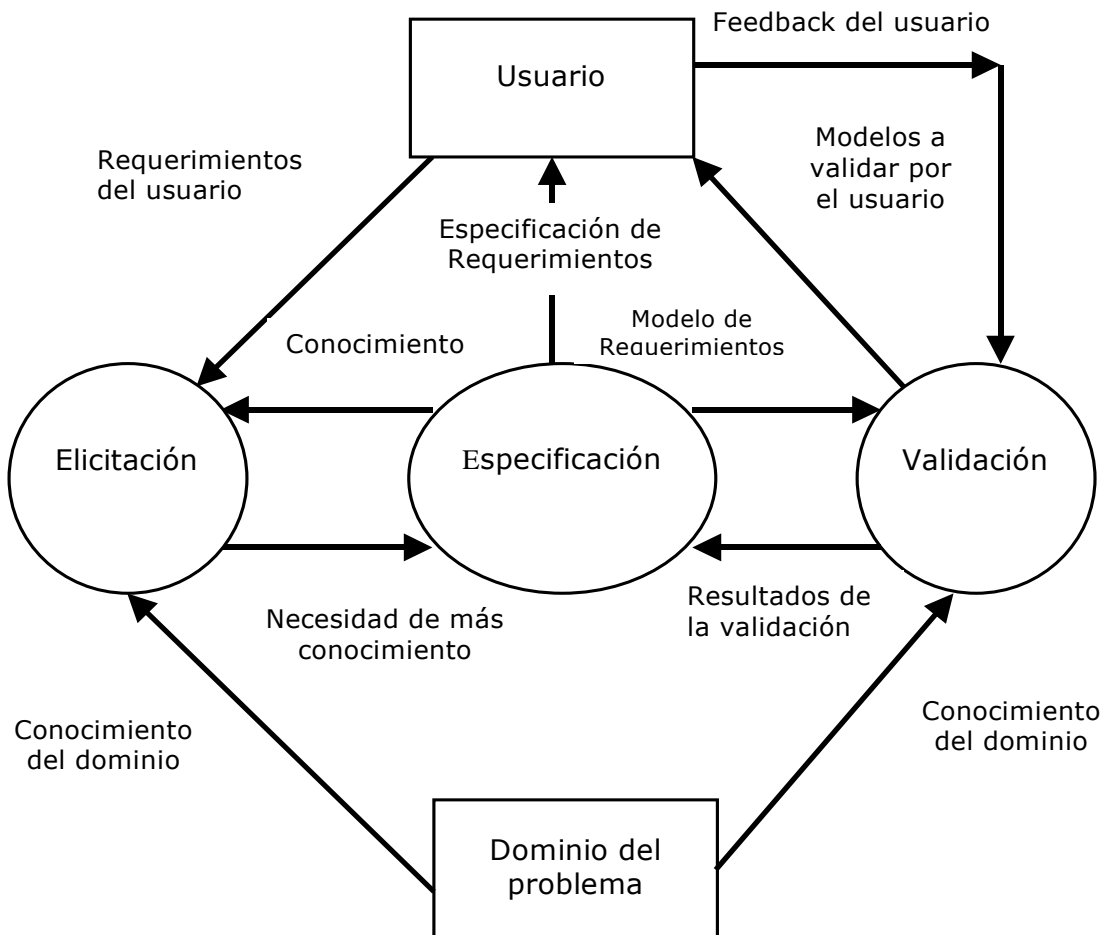
Como se ve en la Figura 1.1, cada etapa necesita de otra etapa y no necesariamente empieza y termina. Se debe iterar en las tres etapas tantas veces como sea requerido. El papel del usuario es crucial en todo este proceso, tanto para transmitir conocimiento como para certificar que el analista comprende el problema, en el marco de un dominio de problema específico.

La etapa que abarca este trabajo es la Elicitación de Requerimientos, específicamente una de las técnicas que se puede utilizar durante ese proceso.

1.2.1 Elicitación de Requerimientos

En la mayoría de los casos, al comenzar un proyecto de software, el analista conoce muy poco acerca del problema a resolver. La elicitación de requerimientos es el proceso que consiste en adquirir todo el conocimiento

Figura 1.1 – Procesos de Ingeniería de Requerimientos



relevante, necesario para producir un modelo de requerimientos (especificación) de un dominio de problema.

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

La "captura" de conocimiento es un problema en sí mismo dado que el conocimiento no siempre está disponible en un formato que pueda ser usado por el analista, y además, no es fácil elicitar conocimiento desde su fuente, especialmente cuando la fuente es un "experto" humano.

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

Mientras que algunos enfoques restringen las fuentes de elicitación sólo a personas, [Loucopoulos95] considera además de los expertos del dominio del problema, otras fuentes de conocimiento del dominio:

- literatura sobre el dominio
- software existente en el dominio
- software similar en otros dominios
- standards nacionales e internacionales
- otros "stakeholders" (personas afectadas de un modo u otro por el Sistema a construir) de la organización dónde se implantará en nuevo sistema de software.

Las actividades que abarcan las tareas del analista incluyen la identificación de todas las fuentes de conocimiento de requerimientos, adquirir conocimiento, decidir sobre la relevancia del conocimiento de un problema, y comprender su significado y cómo éste impacta sobre los requerimientos de software.

Los métodos y/o técnicas de elicitación de requerimientos más relevantes, serán detallados más adelante en este capítulo.

1.2.2 Especificación de Requerimientos

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

Es importante ver la especificación como un proceso complejo que requiere "feedback" desde el analista al usuario y viceversa. El proceso es *analítico* debido a que diferentes clases de conocimiento que el analista elicitaba de un dominio de problema, debe ser examinado y posteriormente vinculado de algún modo. También es *sintético* debido a que conocimiento heterogéneo debe ser combinado para producir una especificación de requerimientos coherente.

Por lo tanto, la especificación de requerimientos puede ser descrita en términos de dos actividades principales:

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

Se considera que la especificación de requerimientos produce una variedad de modelos, los cuales corresponden a diferentes visiones del problema. Es así como la especificación de requerimientos produce:

- Modelos orientados al usuario, especificando el comportamiento y características no funcionales del software que servirá como punto de entendimiento entre el analista, el cliente y el usuario.
- Modelos orientados al desarrollador, especificando propiedades funcionales y no funcionales del sistema de software, así como restricciones sobre recursos, restricciones sobre diseño, etc. Estos modelos son importantes de considerar en etapas de desarrollo posteriores.

Esta diferencia de modelos es muy útil, ya que notaciones de especificación claras para desarrolladores pueden ser difíciles de comprender por los usuarios.

La especificación de requerimientos es el proceso central de ingeniería de requerimientos. Actúa como medio de control para los procesos de elicitación y de validación.

Durante la especificación puede surgir la necesidad de mayor información acerca del problema. Esta necesidad dispara nuevamente el proceso de elicitación en búsqueda de información. Por otra parte, algún cambio en el dominio del problema debe producir cambios en la especificación. De este modo, puede requerirse elicitación durante la especificación.

Similar situación ocurre en la interacción con la validación, proceso que se describe a continuación.

1.2.3 Validación de Requerimientos

El proceso de validación certifica la corrección del modelo/especificación de requerimientos contra las intenciones de los usuarios (por lo que la participación de los usuarios es crucial). Validar, luego de la fase de especificación de requerimientos, puede ayudar a evitar costosas correcciones después del desarrollo.

Otra definición [Locoupoulos95] indica que la validación establece y justifica la convicción del analista y del usuario, de que el modelo de requerimientos especifica una solución de software la cual es correcta para las necesidades del usuario.

La validación de requerimientos es más una actividad de "debugging" que de probar corrección. La meta consiste en identificar y corregir errores en la fase de requerimientos y no más tarde cuando el software esté desarrollado (errores que sobreviven a la fase de

requerimientos pueden tener consecuencias catastróficas para el proyecto de software). Por lo tanto, es una actividad siempre presente en el proceso de requerimientos. Es una actividad no estructurada, por lo que no tiene una solución algorítmica.

Diferentes técnicas se proponen como herramientas que asisten al proceso de validación. Las más usuales son el prototipado de software y la realización de animaciones. Dichas técnicas deben utilizar lenguajes de modelización y ambientes en los cuales los requerimientos puedan estar más cerca de la experiencia del usuario, a través de ejecución, animación y simulación, pudiendo ser luego más fácilmente validados.

1.3 Técnicas de Elicitación de Requerimientos

La elección de la técnica de elicitación depende del tiempo y de los recursos que dispone el ingeniero de requerimientos y, por supuesto, de la clase de información que se necesita elicitar.

Se detallan en este punto la clasificación de técnicas de elicitación de requerimientos de Loucopoulos [Loucopoulos95] y la clasificación de Nuseibeh y Easterbrook [Nuseibeh00]

1.3.1 Técnicas de Elicitación según Loucopoulos

Según Loucopoulos [Loucopoulos95] las técnicas de elicitación pueden clasificarse en Originadas en el Usuario, Análisis de Objetivo y Meta, Escenarios, Análisis de Formularios, Lenguaje Natural, Reuso de Requerimientos y Análisis de Tareas.

1.3.1.1 Originadas en el usuario

Este enfoque es el más intuitivo y directo, dado que los usuarios tienen la posibilidad de expresar qué quieren. Sin embargo, en la práctica se presentan dificultades por diferentes motivos:

- Los usuarios no siempre tienen una idea clara de lo que quieren.

- Pueden presentar dificultad en expresar lo que quieren o en transmitir su conocimiento.
- Pueden tener diferencias con el analista al utilizan un vocabulario diferente.
- Pueden no desear un sistema de software o un nuevo sistema de software.

Para superar estos problemas potenciales, existen técnicas que posibilitan y facilitan la comunicación entre el analista y usuarios:

- ***Entrevistas de comienzo y final abierto:*** es la forma de interacción más simple entre analistas y usuarios. El analista simplemente permite que el usuario hable sobre sus tareas. Son apropiadas para obtener una visión global de dominio de problema, pero inadecuadas para obtener información detallada.
- ***Entrevistas estructuradas:*** direccionan al usuario hacia aspectos específicos de requerimientos a elicitar, a través de la realización de preguntas cerradas, abiertas, de sondeo y de guía. Son útiles para obtener información detallada.
- ***Brainstorming:*** el nombre "tormenta de ideas" pretende ser una traducción circunstancial de "brainstorming" y dar una imagen de esfuerzo y creatividad cooperativa con la finalidad de encontrar una trayectoria factible, consensual y efectiva para un problema planteado. Es una técnica de elicitación grupal, que permite generar numerosas alternativas gracias al esfuerzo mental, y al aplazamiento del juicio o aceptación de las ideas generadas, pues la creación de ideas es más productiva si se excluye la crítica.

1.3.1.2 Escenarios

Los Escenarios son descripciones parciales del comportamiento del Sistema, que permiten asegurar la comunicación entre usuarios y analistas, facilitando la captura de requerimientos. La técnica de Escenarios es un tema importante de este trabajo de tesis, y se describe en detalle en el capítulo 2.

1.3.1.3 Análisis de Objetivos

Los fundamentos de un sistema de software están establecidos por los objetivos de la organización donde el software funcionará. Usualmente estos objetivos son definidos como las metas a ser cumplidas por el sistema y su entorno, aunque algunos autores distinguen los objetivos del sistema de los objetivos de la organización. La técnica de Análisis de Objetivos es tema central de este trabajo, y se describe en detalle en el capítulo 3.

1.3.1.4 Análisis de Formularios

La metodología de Análisis de Formularios no considera al usuario como fuente primaria de conocimiento acerca de un dominio de problema dado. Un formulario es una colección estructurada de variables que está formateada para dar soporte al ingreso de datos y su recuperación.

Es una fuente importante pues es un modelo formal, ya que no tiene las inconsistencias que posee la expresión del mismo conocimiento en lenguaje natural. A menudo, contiene información sobre la organización, y su análisis puede automatizarse.

1.3.1.5 Lenguaje Natural

El lenguaje natural es una fuente importante de información, debido a que en la mayoría de los dominios es el modo más común de representación de conocimiento. Existen dos categorías: interacción directa con el usuario utilizando lenguaje natural y elicitación de requerimientos desde un documento en lenguaje natural.

El mayor atractivo del lenguaje natural reside en su vocabulario preexistente, informalidad y sintaxis. La existencia de un vocabulario de miles de palabras predefinidas usadas para describir cualquier concepto posible, hace que el lenguaje natural sea un medio de comunicación eficiente. Es familiar tanto para el usuario como para el analista y no requiere tiempo de aprendizaje.

No obstante, existen dos claras limitaciones: el lenguaje natural es muy complejo y es a la vez ambiguo.

1.3.1.6 Reuso de Requerimientos

La técnica de Reuso de Requerimientos parte de la idea de que los requerimientos que ya han sido capturados para alguna aplicación, pueden ser reusados en la especificación de otra aplicación similar.

Entre las razones que consideran atrayente a esta metodología se encuentran: el ahorro de tiempo con la consecuente mejora de productividad, el nivel significativo de similitud entre sistemas que pertenecen a una misma área de aplicación, y la potencial obtención de mejoras (calidad).

No obstante, estas características atrayentes contrastan con una serie de cuestiones prácticas de aplicabilidad. La primera de ellas es que los documentos de requerimientos de sistemas existentes no siempre se encuentran fácilmente disponibles. Otro inconveniente es la dificultad aparente de adecuar un requerimiento antiguo en el contexto de especificación de un nuevo sistema. Por lo tanto, para que la idea de reusabilidad de requerimientos sea posible, existen algunos prerrequisitos de aplicación:

- los requerimientos de sistemas existentes deben ser fácilmente accesibles,
- la selección, testeo y adecuación de un viejo requerimiento en el contexto de un nuevo modelo de requerimientos no debe ser una tarea compleja
- el "costo" debe ser menor que obtener los requerimientos desde "cero"

1.3.1.7 Análisis de Tareas

El Análisis de Tareas es una técnica efectiva para elicitar requerimientos de usuarios, en particular aquellos requerimientos que reflejan la interacción hombre-máquina. El término "Análisis de Tareas" se

refiere a un conjunto de métodos que analizan y describen el modo en que los usuarios realizan sus trabajos en términos de:

- actividades que ejecutan y cómo están estructuradas,
- el conocimiento requerido para ejecutar esas actividades

El análisis de tareas es una herramienta valiosa para el proceso de elicitación de requerimientos. Sin embargo, no produce directamente los requerimientos para un nuevo sistema debido a que se refiere a tareas del sistema existente (no del sistema deseado), y por lo tanto incluye muchos elementos que no formarán parte del sistema de software futuro. Pero por otra parte, puede ser considerado como base para el futuro sistema.

1.3.2 Técnicas de Elicitación según Nuseibeh y Easterbrook

Nuseibeh y Easterbrook [Nuseibeh00] clasifican las técnicas de elicitación de requerimientos en Tradicionales, de Elicitación Grupales, Prototipos, Orientadas por Modelos, Cognitivas y Contextuales.

1.3.2.1 Técnicas Tradicionales

Las técnicas tradicionales abarcan una amplia clase de métodos de recolección de datos genéricos, incluyendo el uso de cuestionarios, entrevistas (de comienzo y final abierto, estructuradas), y el análisis de documentación existente (diagramas organizacionales, modelos o normas de procesos y otros manuales de sistemas existentes).

1.3.2.2 Técnicas de Elicitación Grupales

Las técnicas de elicitación grupales proponen promover un acuerdo con los *stakeholders*, mientras explotan la dinámica de grupo para elicitar una mejor comprensión de las necesidades. Estas incluyen la "tormenta de ideas" (Brainstorming) como así también RAD/JAD.

1.3.2.3 Prototipos

Un prototipo es un modelo del producto final de software, que permite efectuar una evaluación sobre determinados atributos del mismo, sin necesidad de que esté disponible. Se trata, simplemente, de testear haciendo uso del modelo.

Se utiliza cuando existe incertidumbre acerca de los requerimientos, o cuando es necesaria una respuesta temprana de los stakeholders.

La técnica de prototipación se puede combinar con otras técnicas (base para un grupo de discusión) o como base de un cuestionario o análisis de un protocolo.

1.3.2.4 Técnicas Orientadas por Modelos

Las técnicas orientadas por modelos proveen un modelo específico del tipo de información a capturar, y utilizan este modelo para orientar el proceso de elicitación. Abarcan los métodos basados en escenarios y los métodos basados en objetivos, detallados posteriormente en los capítulos 2 y 3 respectivamente.

1.3.2.5 Técnicas Cognitivas

Las técnicas cognitivas incluyen un conjunto de técnicas originariamente desarrolladas para la adquisición de conocimiento destinado a sistemas basados en conocimiento.

Dichas técnicas abarcan

- *análisis de protocolo* (en el cual el experto piensa en voz alta mientras realiza su tarea, para así proveer al observador de perspectivas en los sistemas cognitivos que se utilizan para realizar la tarea),
- *“laddering”* (se realizan investigaciones para elicitar la estructura y el contenido de conocimiento de los stakeholders),
- *la distribución de tarjetas* (se pide que los *stakeholders* distribuyan las tarjetas en grupos, cada una de las cuales tiene el nombre de alguna entidad de dominio), y

- *grillas de repertorio* (se construye una matriz de atributos para las entidades, solicitándole a los *stakeholders* los atributos aplicables a las entidades y los valores para las celdas en cada entidad).

1.3.2.6 Técnicas Contextuales

Las técnicas contextuales surgieron en la década de 1990 como una alternativa tanto de las técnicas tradicionales como de las cognitivas. Incluyen el uso de métodos etnográficos (observación de los participantes). También incluyen la etnometodología y el análisis de conversación, las cuales aplican un análisis minucioso para identificar los patrones en la conversación e interacción.

Capítulo 2

Escenarios

2.1 Introducción

En diferentes disciplinas, incluyendo ciencias de la administración, ingeniería de software e interacción hombre-máquina, los escenarios han emergido como una herramienta clave en la búsqueda de cambios efectivos de administración, con el uso de tecnología de información [Jarke98].

Pero recientemente, la comunidad de Sistemas de Información ha dedicado especial atención a las posibilidades que los escenarios proveen para facilitar la comprensión de tareas relacionadas y mejorar la comunicación entre stakeholders [Leite00].

2.2 Léxico Extendido del Lenguaje

La comunicación es un elemento complejo pero fundamental en cualquier relación humana. Permite que dos o más personas, manteniendo sus propios intereses, puedan entenderse. Durante todo el proceso de desarrollo del software, la comunicación es un elemento indispensable que todos los actores tienen que manejar a diario. Cuando el ingeniero de software se comunica con el usuario utiliza su propio vocabulario. Paralelamente, el usuario utiliza el lenguaje habitual de su medio ambiente. Una comunicación basada en diferentes léxicos tiene una mayor probabilidad de fracasar [Hadad99].

Que el usuario y el desarrollador compartan el mismo lenguaje asegura la comunicación entre ambos, en particular el uso del lenguaje propio del usuario mejora considerablemente esta comunicación.

En el proceso de Ingeniería de Requerimientos la validación de los diferentes productos requiere una fuerte interacción con el usuario [Loucopoulos95], la que se ve facilitada por el vocabulario común usuario-desarrollador.

Las distintas representaciones que se construyen en el proceso de desarrollo de software encuentran en el vocabulario del usuario un marco referencial que permite mantener la continuidad del vocabulario, lo que a su vez facilita el acceso a la documentación por parte de todos los participantes en el desarrollo.

La Especificación de Requerimientos de Software es consistente si no hay dos subconjuntos de afirmaciones contradictorios entre sí [Thayer90].

El Léxico Extendido del Lenguaje (LEL) ataca el Universo del Discurso (UdeD) del Sistema de Información. El UdeD incluye todas las fuentes de información y todas las personas relacionadas al software, que son también conocidas como los actores de ese universo [Leite97].

El LEL está compuesto por un conjunto de símbolos que identifican el lenguaje de la aplicación [Leite97]. Los símbolos son, en general, las palabras o frases utilizadas por el usuario y que repite con más frecuencia. También se incluyen aquellas palabras o frases que son relevantes para el dominio del problema más allá de la frecuencia de repetición. Los símbolos se adquieren, por ejemplo, de entrevistas, observaciones, lectura de documentos, entre otros.

Se genera una lista con todos los símbolos reconocidos. Durante el proceso de recolección, el ingeniero de software procura entender el significado de cada símbolo. La semántica de cada símbolo se representa con una o más nociones y uno o más impactos. El impacto puede no existir. La noción indica qué es el símbolo y el impacto cómo repercute en

el sistema. Por lo tanto, cada símbolo tiene un "nombre" que lo identifica, una "noción" y un "impacto" que lo describen [Leite97].

El conjunto de símbolos forma una red que permite representar al LEL en un hipertexto [Leite97] y navegar en él para conocer todo el vocabulario del dominio.

En la descripción de las nociones e impactos existen dos reglas básicas [Leite97] que se deben cumplir simultáneamente: una tiende a maximizar el uso de símbolos en el significado de otros símbolos. Esto se denomina "principio de circularidad". La otra requiere que se minimice el uso de símbolos externos al lenguaje de la aplicación. Este principio se denomina "principio del vocabulario mínimo".

2.3 Características de los Escenarios

Los Escenarios son descripciones parciales del funcionamiento del sistema, que se concentran en un momento específico de la aplicación. Los Escenarios no son formales, y se los puede representar con una variedad de recursos: narrativa textual, storyboards, videos mock-ups, prototipos escritos o situaciones físicas [Hadad99].

Si bien cada Escenario es una descripción parcial del comportamiento de la aplicación, ningún Escenario es totalmente independiente del resto de los Escenarios. Cada Escenario tiene una relación semántica con otros Escenarios.

El enfoque de [Leite96], [Leite97], [Leite00] incluye el uso de lenguaje natural para la elicitación y construcción de Escenarios. Esto implica utilizar un medio fácil de comprender por el usuario, pero al mismo tiempo con riesgos de inclusión involuntaria de ambigüedades o inconsistencias, que se reducen considerablemente con el empleo de un vocabulario bien definido del Universo de Discurso: el Léxico Extendido del Lenguaje (LEL).

La particularidad de este enfoque es que la construcción de Escenarios se basa exclusivamente en el LEL, posibilitando comprender el vocabulario específico del dominio del problema, y esa comprensión se refuerza posteriormente con nuevas entrevistas a los stakeholders.

En la Tabla 2.1 se reproduce el esquema de representación del enfoque de escenarios de [Leite00].

Tabla 2.1. Esquema de escenario

Componente	Descripción
Nombre	Identifica al escenario (en el caso de un sub-escenario, el título es el mismo que la sentencia episodio, sin las restricciones y/o excepciones.)
Objetivo	Establece la finalidad del escenario, a ser alcanzada en el contexto del problema. El escenario describe el logro del objetivo
Contexto	Describe las acciones previas necesarias para iniciar el escenario, las precondiciones, la ubicación física y temporal.
Recursos	Identifican los objetos pasivos con los cuales los actores trabajan.
Actores	Detalla las entidades que se involucran activamente en el escenario, es decir aquellas personas o estructuras organizacionales que tienen un papel en el escenario.
Set de episodios	Cada episodio representa una acción realizada por un actor, donde participan otros actores y se utilizan recursos. Los episodios se ejecutan secuencialmente. Un episodio también puede referenciar a un escenario. Su ocurrencia puede estar sujeta a condiciones. Se incluyen las <i>restricciones</i> del escenario o de cada episodio según corresponda.
Casos alternativos	Menciona los casos de excepción, que pueden corresponder a otros escenarios.

Los escenarios se pueden clasificar de la siguiente manera:

1. Escenarios particulares, son aquellos que describen un momento específico de la aplicación. Usualmente los episodios de estos escenarios corresponden a simples acciones.

2. Escenarios generales, son aquellos que representan las funciones fundamentales de la aplicación. Usualmente los episodios de estos escenarios corresponden a otros escenarios.

En el proceso de construcción de Escenarios se emplea el punto de vista de la aplicación. En la Tabla 2.2 se muestra un ejemplo, en el contexto del problema de administrar la recepción de un hotel [Bertolami01].

Muchas ventajas hacen que esta técnica sea elegida como medio para elicitación y especificación de requerimientos. Los usuarios finales y otros stakeholders de sistemas encuentran más fácil relacionar las funciones provistas por un sistema con ejemplos de la vida real que con descripciones abstractas. Por ello, resulta muy útil desarrollar un conjunto de escenarios con el objeto de utilizarlos para modelar el comportamiento del sistema. Además el conjunto de escenarios brinda una visión más amplia del comportamiento del macro sistema.

Es así como se logra un mayor compromiso del usuario y stakeholders con el desarrollo del sistema y se facilita la validación de los requerimientos. También se garantiza el seguimiento ("traceability") de los requerimientos durante todo el ciclo de vida del software.

Otro uso muy difundido de los escenarios está orientado a modelar el comportamiento del sistema en una etapa posterior al diseño. En este caso, suelen confundirse con los casos de uso. Éstos están destinados a representar las funciones del sistema para el caso general. Los escenarios, en cambio, ejemplifican el uso del sistema. En la práctica, sin embargo, la distinción entre ambos es menos clara y los términos suelen ser usados como sinónimos [Ridao2001].

En cualquier caso, el propósito del uso de los escenarios es asegurar un buen entendimiento y una mayor colaboración entre todos los participantes del proceso de definición de requerimientos. Los analistas entenderán, modelarán y analizarán el dominio de la aplicación donde el software se utilizará y los clientes/usuarios validarán si la visión de los analistas es correcta o no.

Tabla 2.2 Ejemplo de escenario

Nombre	Cancelación de la Reserva
Objetivo	Dar de baja una solicitud de reserva de un pasajero / huésped / pax
Contexto	Se realiza en la Recepción del Hotel. Existe una solicitud de reserva para un pasajero / huésped / pax.
Recursos	Planilla de reservas Planilla de ocupación de habitaciones Teléfono / Fax / E-mail
Actores	Recepcionista Agencia Otro hotel Pasajero / huésped / pax
Set de episodios	If el recepcionista recibe el pedido de anulación de una solicitud de reserva o el pasajero / huésped / pax no se presenta en el período comprendido entre las 12 hs del día de ingreso establecido en la solicitud de reserva y las 06 hs. del día siguiente then El recepcionista elimina la solicitud de reserva de la planilla de reservas. El recepcionista actualiza la disponibilidad de habitaciones en la planilla de ocupación de habitaciones.
Casos alternativos	El teléfono, el fax o el e-mail no funcionan

Capítulo 3

Análisis de Objetivos

3.1 Introducción

La Ingeniería de Requerimientos de Sistemas de Software, o Ingeniería de Requerimientos, busca producir una descripción detallada de un problema, con el fin de construir un Sistema de Software, que satisfaga las “necesidades y objetivos” de la organización donde funcionará dicho sistema.

En la comunidad de Ingeniería de Requerimientos, estos objetivos constituyen el fundamento del sistema, y son usualmente definidos como las metas a ser cumplidas por el sistema y su entorno, aunque algunos autores distinguen los objetivos del sistema de los objetivos de la organización [Thomas03].

Debido a que los documentos de especificación de requerimientos cumplen también una función contractual entre analistas y stakeholders, es importante para los stakeholders que el contenido de la especificación de requerimientos esté en un lenguaje entendible con el cual ellos puedan interactuar activamente.

Los stakeholders son aquellas personas afectadas de un modo u otro por el sistema a construir, es decir, tienen alguna clase de interés en el proyecto [Robertson02].

Mediante la concentración sobre Metas/Objetivos más que en requerimientos específicos, los analistas se comunican con los stakeholders usando un lenguaje basado en conceptos (ej. Objetivos) con los que se sienten familiarizados y confortables.

A los stakeholders puede resultarle más fácil comprender los Objetivos a cumplir que la funcionalidad que se exhibiría en el sistema deseado. Los requerimientos a menudo son difíciles de entender, pero ellos pueden ser justificados y explicados a través de la discusión de Objetivos. Debe notarse que los Objetivos de la empresa y los Objetivos del sistema son más estables (a través del tiempo de vida de la empresa) que los requerimientos definidos en un momento dado [Anton97].

Un factor crítico en el éxito de los proyectos es que los desarrolladores no solamente entiendan qué tienen que desarrollar, sino por qué están desarrollando un sistema dado. El enfoque de Objetivos se concentra en por qué el sistema es construido, proveyendo motivación y fundamentos para justificar los requerimientos de software [Mylopoulos94].

Actualmente existen diferentes enfoques para formular los requerimientos del sistema de software a partir de los objetivos, cada uno de ellos con características diferentes pero con un propósito en común: disponer de las Metas u Objetivos que deberá cumplir el sistema. En lo que resta de este capítulo se presentan esos enfoques, y un cuadro comparativo que resume sus características permitiendo compararlos.

3.2 Enfoques orientados a Objetivos

Diferentes autores del ámbito de la Ingeniería de Requerimientos consideran esencial disponer de los Objetivos del sistema para poder arribar con éxito a la construcción del mismo. Todos coinciden en que el análisis de Objetivos sirve como un "puente" de comunicación entre los analistas y stakeholders, proveyendo un lenguaje más entendible entre ellos, y a la vez produciendo requerimientos más fáciles de validar [Thomas03].

A continuación se presentan seis enfoques orientados a objetivos que poseen un importante reconocimiento en la comunidad de Ingeniería de Requerimientos de Sistemas de Software:

- a) **GBRAM (Goal-Based Requirements Analysis Method)**: es un método no formal basado en heurísticas, propuesto por Annie Antón [Anton96] [Anton97] [Anton98].
- b) **KAOS (Knowledge Acquisition in autOmated Specification)**: es un método formal basado en lógica temporal y técnicas de refinamiento de Inteligencia Artificial, donde todos los términos son consistente y rigurosamente definidos. Este método fue desarrollado por Axel van Lamsweerde [Lamsweerde93] [Lamsweerde96] [Lamsweerde00].
- c) Un método que da relevancia a los requerimientos no funcionales (NFR), basado en softgoals más que en "hard" goals, desarrollado por Mylopoulos J., Chung L., Liao S., Wang H. y Yu E., [Mylopoulos01].
- d) La visión de Loucopoulos y Karakostas [Loucopoulos95], bibliografía de base en el estudio de Ingeniería de Requerimientos.
- e) El enfoque de Colette Rolland, Souveyet C. y Ben Achour C., [Rolland98] [Rolland99] desarrollado como parte del proyecto CREWS (Cooperative Requirements Engineering With Scenarios) ESPRIT.
- f) La visión de Oshiro K., Watahiki K. y Saeki M. [Oshiro03] que propone la utilización de métodos como brainstorming entre stakeholders para generar ideas, y luego esas ideas plantearlas como posibles sub-metas de objetivos ya existentes.

En la mayoría de los enfoques, los objetivos se representan en un árbol n-ario utilizando descomposiciones AND/OR. Esta estructura es utilizada para mostrar cómo cada objetivo M puede ser descompuesto en objetivos M_1, M_2, \dots, M_n de menor nivel de abstracción. El resultado es una jerarquía de objetivos con diferentes niveles de abstracción, donde a cada objetivo de menor nivel en la jerarquía se le pueden rastrear sus orígenes.

Una descomposición AND significa que para cumplir el objetivo M deben cumplirse todos los objetivos M_1, M_2, \dots, M_n . En una descomposición OR para cumplir el objetivo G debe cumplirse alguno de los objetivos M_1, M_2, \dots, M_n .

Se omite la descripción de visiones más genéricas, como la visión de Gil Regev y Alain Wegmann [Regev01], quienes establecen que la definición de objetivos debería ser enfocada con el propósito de mejorar la organización, en vez de especificar los requerimientos para un Sistema de Información. Consideran más importante proveer los conceptos necesarios para definir las Metas del sistema a ser construido, pero comenzando sin asumir la existencia o el deseo de existencia de tal sistema.

Todos los enfoques enunciados se describen para comprender cómo es el tratamiento de objetivos en cada uno de ellos. En cada caso se pretende brindar un resumen y no una descripción pormenorizada, a excepción de GBRAM del cual se ofrece un mayor nivel de detalle por ser el enfoque utilizado para esta tesis.

3.2.1 Enfoque de Annie Antón

Este enfoque ataca los problemas que surgen en el proceso de elicitación y especificación de requerimientos (ej: complejidad del dominio del problema, volatilidad de los requerimientos, dependencia de la intuición, ambigüedad de las formulaciones, desacuerdos entre stakeholders, imprecisiones). Propone resolver estos problemas mediante la utilización de un método para identificar, clasificar y refinar los objetivos

que el sistema debe alcanzar, con la subsecuente transformación de esos objetivos en requerimientos operacionales.

Normalmente, a medida que progresa la formalización de los requerimientos, decrece el entendimiento por parte de los stakeholders, debido a la complejidad de las representaciones resultantes.

GBRAM (Goal-Based Requirements Analysis Method) provee mecanismos de representación adecuados a la comprensión de los stakeholders, facilitando la comunicación con los analistas, proveyendo un lenguaje más entendible, y produciendo requerimientos más fáciles de validar [Anton96] [Anton97] [Anton98]. Además, como metodología basada en Objetivos, se concentra en los fundamentos que justifican los requerimientos de software.

El propósito de este enfoque es desarrollar, validar y afirmar un método basado en metas u objetivos, el cual provee soporte procedural para la identificación, elaboración, refinamiento, y organización de objetivos, en la especificación de Sistemas de Información basados en Software. Metas y objetivos son utilizados como sinónimos.

Los requerimientos usualmente son difíciles de entender por parte de los stakeholders, pero ellos pueden ser justificados y explicados a través de la discusión de objetivos. Para GBRAM [Anton97] los objetivos son el sustento de la construcción del sistema, proveyendo motivación y fundamentos para justificar los requerimientos de software y guiar la toma de decisiones dentro de la organización donde el sistema funcionará. El esquema de representación de una meta u objetivo utilizado por este método se presenta más adelante en la tabla 3.1.

Las actividades que forman parte de GBRAM están integradas en un modelo Entrada → Proceso → Salida, donde el Proceso se divide en Análisis y Refinamiento, tal como se presenta en la Figura 3.1. Estas actividades en las cuales está involucrado el analista, si bien no necesariamente se realizan en orden secuencial, son las siguientes:

a) Identificar Metas

- Explorar la documentación existente
- Identificar agentes y sus responsabilidades
- Identificación de stakeholders

b) Organizar y clasificar metas

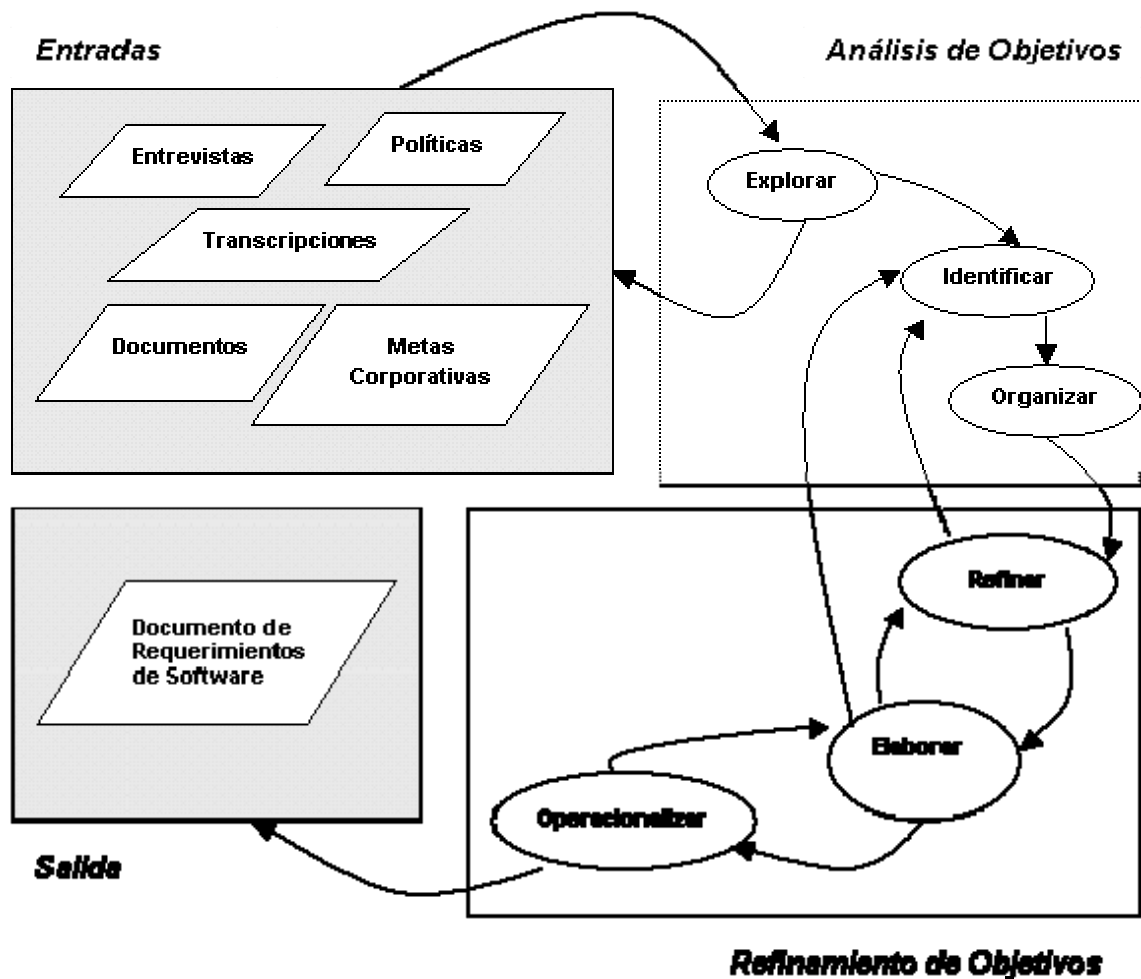
- Eliminar redundancias y reconciliar metas sinónimo
- Clasificar metas
- Especificar dependencias entre metas
- Construir una jerarquía de metas

c) Refinar y elaborar Metas

- Especificar los obstáculos para cada meta
- Construir escenarios
- Identificar restricciones

d) Operacionalizar Metas en Requerimientos

Figura 3.1 Resumen de las actividades GBRAM [Anton97]



a) Identificar Metas

Las actividades de exploración exigen examinar toda la información disponible a partir de transcripciones de entrevistas, metas corporativas, políticas de la organización, requerimientos existentes, entre otros. Para identificar metas, cada "fragmento" de información es analizada preguntando "¿qué meta/s ejemplifica este fragmento?" y/o ¿ qué metas son obstruidas o bloqueadas por este fragmento?"

Como regla general se plantea que todas aquellas afirmaciones que conducen a decisiones de diseño en varios niveles dentro del sistema u organización, apuntan a ser consideradas posibles metas. Los objetivos también pueden ser identificados a través de la búsqueda de palabras de acción, orientadas a algún estado que es o puede ser alcanzado dentro del sistema una vez que la acción es completada. Ciertos tipos de verbos como por ejemplo "completar", "obtener", "satisfacer"; sugieren posibles metas.

Para identificar stakeholders, no se considera a un simple usuario de sistema (en un sentido clásico), sino que se determina quién o quienes tienen interés en cada meta u objetivo.

Los agentes son identificados como los responsables del cumplimiento y/o satisfacción de una meta u objetivo, dentro de una organización o sistema.

b) Organizar y clasificar metas

La organización de metas implica eliminar redundancias y reconciliar metas sinónimos, mientras que la clasificación de metas implica diferenciar las metas u objetivos de acuerdo a sus propósitos.

Las metas son consideradas sinónimos si significan lo mismo para diferentes stakeholders, quienes simplemente expresan las metas usando diferente terminología.

La identificación de redundancias y metas sinónimos se realiza más fácilmente después de que han ordenado los objetivos de acuerdo a sus relaciones de precedencia.

Las metas según su propósito se clasifican en "*Mantenimiento*" ("Maintenance"), generalmente son requerimientos de seguridad y se identifican a través de palabras clave como "mantener", "asegurar", "evitar", "seguir", entre otras; y "*Logro*" ("Achievement"), las cuales reflejan acciones que ocurren en la organización y se pueden identificar a través de las palabras claves como "obtener", "lograr", "hacer", "mejorar", "incrementar", entre otras.

Las relaciones de dependencia se establecen entre pares de metas u objetivos, posibilitando una mejor comprensión de las relaciones entre ellas. Las dependencias entre metas se especifican de modo tal, que una jerarquía de objetivos puede ser construida basada en esas relaciones de dependencia. Tres tipos de dependencia son establecidas en GBRAM:

- Relación de precedencia
- Dependencia contractual
- Dependencia entre agentes

El único tipo de dependencia necesario para organizar las metas es la relación de precedencia. Dadas las metas M1 y M2, M1 precede a M2 (expresada como $M1 < M2$), si M1 se completa antes que ocurra M2. Las relaciones de precedencia se identifican para cada objetivo a través de la realización de dos preguntas:

- ¿ Qué metas son prerrequisitos para este objetivo ?
- ¿ Qué metas deben seguir a este objetivo ?

Para el caso de la dependencia contractual, dadas dos metas M1 y M2, M1 tiene una dependencia de contrato con M2 (expresada como $M1 \rightarrow M2$), cuando M2 debe ser obtenida si M1 ocurre.

Si un agente debe completar una meta u objetivo, y para ello previamente otro agente debe completar otra meta u objetivo, en ese

caso se establece que existe una dependencia entre esos agentes, originada por las metas sobre las cuales cada uno es responsable.

Una vez establecidas las relaciones de dependencia, se construye un estructura de organización de las metas, denominada *Topografía de Metas*. En general, una *Topografía* es una representación gráfica de las características físicas de un lugar. Para el caso de GBRAM, una *Topografía de Metas* es una representación de las características de un Documento de Especificación de Requerimientos (conocido como SRD), expresado en forma de jerarquía.

En GBRAM la jerarquía tiene dos representaciones posibles, presentadas en las Figuras 3.2. y 3.3.

Figura 3.2 Topografía de metas (Sistema de Administración de Cursos)

Representación 1

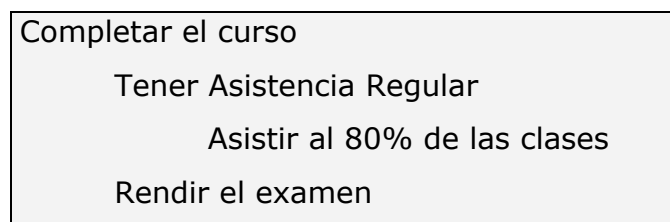
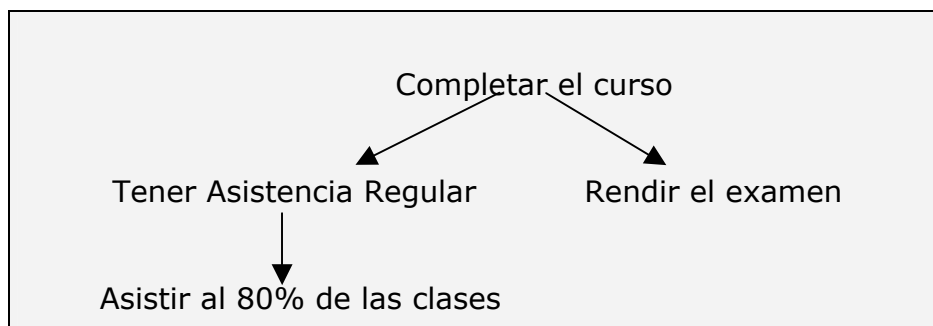


Figura 3.3 Topografía de metas (Sistema de Administración de Cursos)

Representación 2



La organización de la información sobre requerimientos en un SRD según la topografía de objetivos facilita la administración de cambios en los requerimientos.

Debido a que las relaciones entre objetivos pueden ser rápidamente identificadas, surge la posibilidad de disponer rastreabilidad (traceability) para la localización de objetivos afectados por el cambio de un objetivo específico.

Una vez que la topografía de objetivos es construida, el analista debe elaborar y refinar sistemáticamente el conjunto de objetivos.

c) Refinar y elaborar Metas

Las actividades de refinamiento implican "podar" el conjunto de metas u objetivos obtenidos, eliminando nuevamente las redundancias y reconciliando metas sinónimos.

Se analiza el conjunto de metas considerando para cada una de ellas los posibles obstáculos. Los obstáculos representan situaciones en las cuales una meta u objetivo no se puede completar.

La tarea de identificación de posibles obstáculos requiere la inventiva de los analistas ya que deben establecer y construir los obstáculos de las metas a partir de las fuentes de información disponibles. Una vez que los obstáculos han sido especificados, son analizados a través de los Escenarios posibles.

Para Antón [Anton97] los Escenarios ofrecen un modo natural para describir circunstancias excepcionales o especiales, asistiendo al analista en la adquisición y validación de requerimientos.

El análisis de Escenarios permite la consideración de posibles operacionalizaciones alternativas de metas, ofreciendo soluciones más razonables. También denotan circunstancias concretas bajo las cuáles una meta puede fallar, ayudando a los analistas a descubrir metas ocultas, que de otro modos serían omitidas. Cuando las prioridades de metas cambian, los Escenarios facilitan la evaluación de esas nuevas prioridades.

La identificación y construcción de Escenarios provee, por lo tanto, un modo sistemático para encontrar casos anormales tal que las excepciones puedan ser especificadas.

La última tarea dentro de la elaboración de metas es la identificación de restricciones. Las restricciones proveen información considerando circunstancias que deben existir, o condiciones que deben ser satisfechas, para que una meta sea alcanzada. Como regla general, se pueden obtener "observando" las relaciones de dependencia y buscando conectores temporales como "durante", "antes" o "después".

d) Operacionalizar Metas en Requerimientos

Esta actividad propone traducir metas en requerimientos operacionales, utilizando una sintaxis para representar cada una de ellas. Esta sintaxis se reproduce en la tabla 3.1 y se ejemplifica en la tabla 3.2.

La definición operacional de cada meta se muestra en la tabla 3.3, donde se representa la acción. En ambos esquemas puede haber instancias en las cuales algún componente no esté especificado.

3.2.1.1 Heurísticas

GBRAM además provee un conjunto de heurísticas, 51 en total. Las heurísticas son pautas que guían y ayudan al analista en las actividades previamente indicadas, con la intención de evitar mayores esfuerzos. Las heurísticas provistas se clasifican en cuatro grupos:

- Heurísticas de Identificación
- Heurísticas de Clasificación
- Heurísticas de Refinamiento
- Heurísticas de Elaboración

Heurísticas de Identificación

Asisten al analista para identificar metas, stakeholders, agentes y restricciones desde múltiples fuentes de información. GBRAM provee 25 heurísticas de identificación.

Tabla 3.1 Esquema de objetivo de GBRAM

Componente	Descripción
Nombre de Meta	Es el identificador único para cada Objetivo. Indica un estado deseado o alcanzado.
Tipo	Las metas son clasificadas de acuerdo al comportamiento requerido: obtener algún estado ("Achievement") o mantener alguna condición o estado ("Maintenance").
Descripción	Es un texto informal que describe una meta u objetivo.
Acción	Es el nombre que se le otorga a la operacionalización de una meta. Describe el comportamiento necesario para satisfacer el objetivo.
Agente	Es el responsable de completar o cumplir un objetivo.
Stakeholders	Son las personas interesadas en que una meta u objetivo sea cumplido.
Restricciones	Son las limitaciones bajo las cuales un objetivo debe cumplirse. Una restricción especifica algún requerimiento o condición que debe cumplirse para lograr un objetivo. Tiene vigencia hasta que se logre el objetivo.
Obstáculos	Se representan las circunstancias que puedan impedir que un objetivo sea cumplido.
Precondiciones	Es la condición que debe existir para posibilitar el logro de un objetivo.
Post-condiciones	Es la condición a la que se arriba luego de obtener o completar un objetivo.
Sub-Metas	Cada sub-meta debe mapear a una acción. Si mapea a varias acciones, debería ser descompuesta y refinada.

Heurísticas de Clasificación

El propósito de estas heurísticas es asistir al analista en determinar el tipo de meta u objetivo ("Achievement" o "Maintenance"), en el momento que cada meta u objetivo es identificado, dado que generalmente es aconsejable realizar la tarea de clasificación junto a la identificación. GBRAM provee 6 heurísticas de clasificación.

Tabla 3.2 Ejemplo de objetivo de GBRAM

Componente	Descripción
Nombre	Curso completo.
Tipo	“Logro”.
Descripción	Un empleado deben realizar un curso para mejorar sus capacidades y habilidades. Ello mejora su calificación.
Acción	Completar Curso.
Agente	Empleado.
Stakeholders	Empleado.
Restricciones	La asistencia durante el curso debe ser verificada.
Obstáculos	El empleado no asiste al curso. El empleado rinde mal el examen.
Precondiciones	Los empleados están notificados para realizar el curso.
Post- condiciones	Capacidades y habilidades de los empleados mejoradas.
Sub-Metas	Curso Asistido Curso Aprobado

Tabla 3.3 Sintaxis de la definición operacional de cada meta

Componente	Descripción
Acción	Es el nombre de la función que el sistema permite al usuario invocar y/o ejecutar. Coincide con el nombre de la acción en el esquema del objetivo.
Agente	Responsable de invocar la función.
Entradas	Son las fuentes e información requerida por el agente para ejecutar la función.
Cambios	Son las fuentes o información asociada con la función, en términos de modificaciones que producen en el sistema.
Suposiciones	Son las precondiciones que deben existir para que la función sea completada.
Resultados	Son las condiciones a las que se arriba después que la función sea completada.

Heurísticas de Refinamiento

Estas heurísticas permiten que el analista pueda “podar” o refinar el conjunto de objetivos, con el propósito no sólo de disminuir la cantidad sino también proporcionar mayor claridad. Existen tres conjuntos de heurísticas de refinamiento, 8 en total: para eliminación de redundancias, para reconciliar metas sinónimos , y para refinar metas específicas de sistema.

Heurísticas de Elaboración

El propósito de estas heurísticas es obtener información más detallada. Para ello se proporcionan tres conjuntos de heurísticas, 12 en total: para examinar dependencias entre metas, para guiar el análisis de obstáculos y para guiar el análisis de escenarios.

3.2.1.2 Resumen

La principal contribución del enfoque de Antón [Anton96] [Anton97] [Anton98] es la introducción de un método de análisis de requerimientos basado en objetivos, denominado GBRAM, para la identificación y elaboración de metas u objetivos, y su refinamiento hasta convertirlos en requerimientos. Si bien no soporta razonamiento formal, su informalidad lo hace comprensible y fácil de utilizar.

Este método es asistido por un conjunto de heurísticas, provistas para evitar mayores esfuerzos del analista en las actividades.

Las metas elaboradas ofrecen una rica estructura para organizar la información de requerimientos. Esa estructura es jerárquica y permite organizar la información de objetivos, así como “rastrear” objetivos de mayor o menor nivel de abstracción a partir de una meta dada.

Si bien no soporta razonamiento formal, su informalidad lo hace comprensible y fácil de utilizar. Los objetivos elaborados proveen una rica estructura de representación del conocimiento adquirido de un dominio de problema.

3.2.2 Enfoque de Axel van Lamsweerde

El enfoque de Axel van Lamsweerde [Lamsweerde93] [Lamsweerde96] [Lamsweerde00] consiste en un framework formal basado en lógica temporal y técnicas de refinamiento de Inteligencia Artificial, donde todos los términos son consistente y rigurosamente definidos. El principal énfasis de **KAOS** (**K**nowledge **A**cquisition in aut**O**mated **S**pecification) está en la prueba formal que los requerimientos "satisfacen" las metas/objetivos definidas para el sistema previsto.

Formaliza la representación de metas, objetos, agentes y la operacionalización de Metas/Objetivos en Requerimientos sobre objetos/operaciones.

KAOS es una metodología de elaboración de requerimientos [Lamsweerde93] que provee un lenguaje de especificación multiparadigma y un método de elaboración orientado a Metas u Objetivos [Lamsweerde96].

El lenguaje combina redes semánticas para modelar objetivos, restricciones, agentes, objetos y operaciones en el sistema; lógica temporal para la especificación de Objetivos, restricciones y objetos, y especificaciones basadas en estados para la especificación de operaciones [Lamsweerde96].

Para Axel van Lamsweerde [Lamsweerde93] [Lamsweerde96] [Lamsweerde00] una meta es un objetivo que el sistema debería lograr a través de la cooperación de agentes utilizando el sistema en un entorno definido.

El proceso resumido de elaboración orientado a Metas u Objetivos consiste en:

- a. Elaborar la estructura de Objetivos*
- b. Identificar los objetos involucrados*

- c. *Identificar los agentes y operaciones*
- d. *Operacionalizar Objetivos en requerimientos sobre objetos y operaciones*
- e. *Identificar responsabilidades y asignar operaciones*

a. Elaborar la estructura de Objetivos

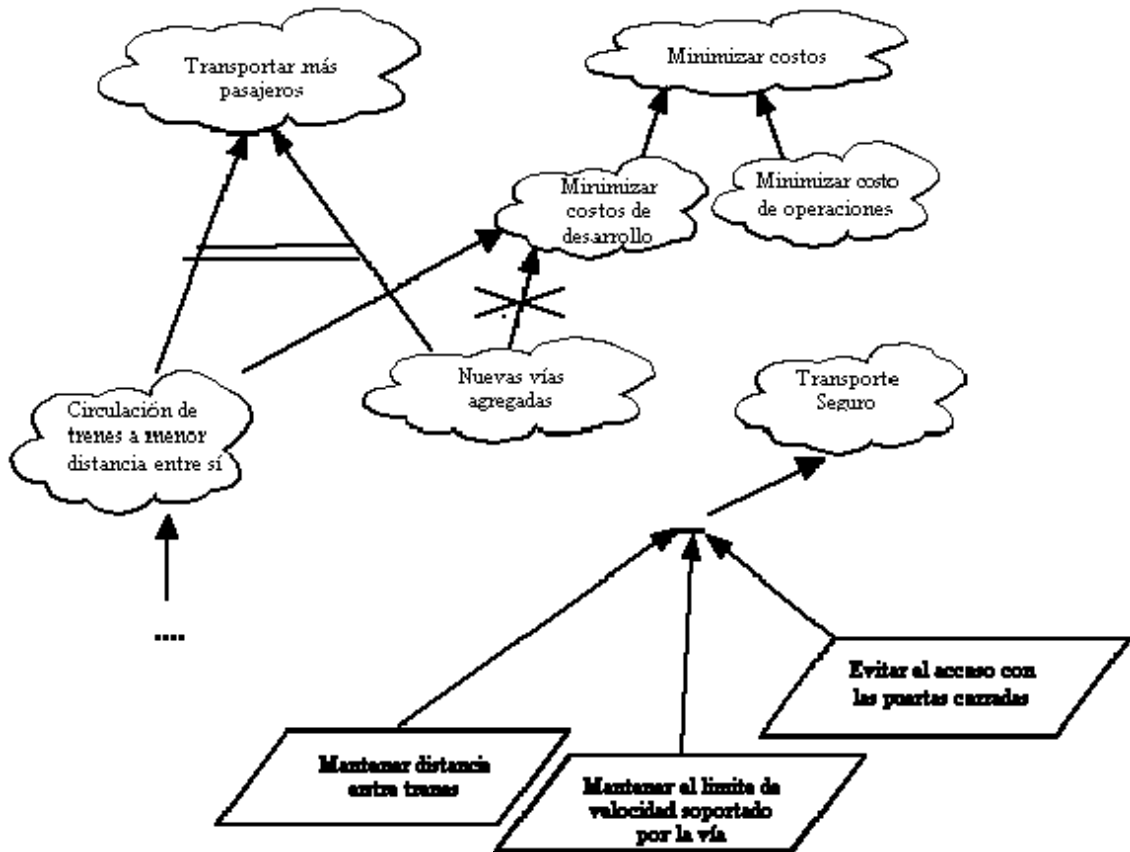
Un primer conjunto de Objetivos es identificado a partir de una lectura inicial de los documentos de información disponibles, buscando palabras claves tales como “objetivo”, “propósito”, “intención”, “concerniente”, etc. [Lamsweerde00]. Este conjunto es refinado siendo la estructura resultante una jerarquía de softgoals.

El concepto de “*Softgoals*” utilizado por KAOS es el propuesto por John Mylopoulos [Mylopoulos01], presentado más adelante en este capítulo, el cual expresa que las softgoals son metas sin un criterio claro de satisfacción.

Las metas descendientes (o hijas) son identificadas preguntando “¿cómo?”, mientras que las metas padre son identificadas preguntado “¿por qué?”, quedando definida de este modo la estrategia de refinamiento. El refinamiento continúa hasta que las softgoals sean asignables a agentes individuales.

Un ejemplo de la representación gráfica aplicado a un *Controlador Avanzado y Automático de Trenes* desarrollado por el *San Francisco Bay Area Rapid Transit (BART) system* [Lamsweerde00] se reproduce parcialmente en la Figura 3.4, donde las “nubes” denotan softgoals y los paralelogramos indican metas de bajo nivel que se podrían formalizar. Las flechas muestran vínculos entre metas de bajo nivel y/o softgoals. Una doble línea vinculando flechas indica una alternativa OR de refinamiento y las líneas “cruzadas” denotan un conflicto.

Figura 3.4 Controlador Avanzado y Automático de Trenes [Lamsweerde00]



b. Identificar los objetos involucrados

La modelización de objetos comienza cuando los objetivos son lo suficientemente precisos (los de más bajo nivel en la jerarquía representados por paralelogramos). El propósito es identificar objetos, relaciones y atributos a partir de la especificación de objetivos.

Ejemplo para el mismo sistema enunciado previamente:

Objetivo *Mantener el límite de velocidad soportado por la vía*

Definición informal:

Un **tren** debería viajar a una velocidad menor a la máxima que puede soportar la **vía**.

Definición formal:

$\forall tr : Tren, v : Via :$

$Circula(tr, v) \Rightarrow tr.Vel \leq v.Limite de Velocidad$

Objetivo *Mantener distancia entre trenes*

Definición Informal: Un **tren** nunca debería acercarse tanto al tren de adelante, tal que si el tren de adelante se detiene repentinamente, entonces el próximo tren no lo chocará.

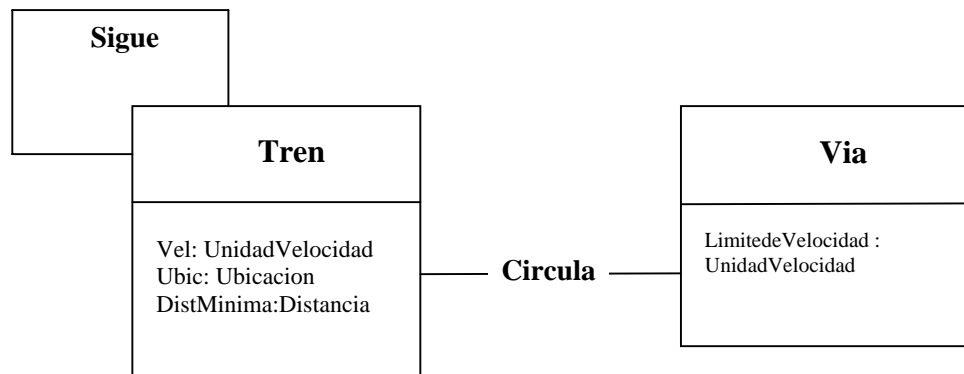
Definición formal:

$\forall tr1, tr2 : Tren :$

$Sigue(tr1, tr2) \Rightarrow tr1.Ubic - tr2.Ubic > tr1.DistMínima$

De estas definiciones de objetivos surgen objetos, atributos y relaciones entre objetos, que derivan en la parte del modelo de objetos presentado en la Figura 3.5.

Figura 3.5 Modelo de objetos parcial



c. Identificar Agentes y Operaciones

Los Objetivos se refieren a transiciones de estado específicos, y por cada transición, una operación que la causa es identificada y preliminarmente definida por las pre y pos condiciones del dominio que captura la transición de estado. Los agentes se identifican como los posibles responsables de cada operación.

d. Operacionalizar Objetivos en requerimientos sobre objetos y operaciones

El propósito es consolidar las condiciones del dominio tal que se asegure la satisfacción de los diferentes Objetivos vinculados a cada operación. Se aplican reglas preliminares de derivación definidas en [Lamsweerde00].

e. Identificar responsabilidades y asignar operaciones

Se vinculan los agentes a cada una de las operaciones definidas.

Las ventajas de esta propuesta según [Lamsweerde93] [Lamsweerde96] [Lamsweerde00] se resumen en:

- El modelo de objetos y el de requerimientos se deriva sistemáticamente a partir de objetivos.
- Los objetivos aportan el fundamento lógico a los requerimientos.
- El grafo de objetivos provee trazabilidad desde aspectos estratégicos de alto nivel, hasta detalles técnicos de muy bajo nivel, y permite evolucionar las versiones del sistema.
- Los grafos de objetivos AND/OR proveen el nivel correcto de abstracción.
- La estructura de refinamiento de objetivos provee una estructura comprensible para los documentos de requerimientos.
- La formalización de objetivos permite que los refinamientos sean probados de manera correcta y completa.

Por otra parte, aplicar este método requiere un alto conocimiento de lógica matemática y sistemas formales, por el analista y el stakeholder o usuario que valide la especificación obtenida. Consecuentemente, su costo de aplicación es mayor y es utilizado entonces para las partes más importantes de sistemas críticos, donde no deben ocurrir errores (al menos derivados de los requerimientos).

3.2.3 Enfoque de Mylopoulos J., Chung L., Liao S., Wang H., Yu E.

El enfoque de Mylopoulos J., Chung L., Liao S., Wang H., Yu E. [Mylopoulos01] plantea que el análisis orientado por Objetivos, complementa y fortalece las técnicas tradicionales de análisis de requerimientos. Además, afirma que la necesidad de explorar y evaluar alternativas con respecto a los objetivos del negocio ha estimulado el análisis orientado por Objetivos.

Los objetivos se representan en un árbol n-ario utilizando descomposiciones AND/OR, constituyendo una jerarquía para explorar soluciones alternativas. Cada alternativa refleja un plan potencial para satisfacer un objetivo donde el doble arco en el arco significa OR y el arco simple AND.

Generalmente, los diagramas AND/OR permiten sistematizar la exploración de alternativas dentro de un espacio que puede ser muy "grande", y a través de un simple algoritmo se puede establecer si el objetivo que origina el diagrama puede ser cumplido.

Dado que no todas las Metas u Objetivos pueden ser delineados claramente (como los vinculados a requerimientos no funcionales), se plantea la necesidad de tener una noción aunque sea vaga de esas Metas, y un conjunto de relaciones que indiquen que una Meta soporta o impide el cumplimiento de otra Meta, sin limitarse solamente a las estrictas relaciones AND/OR. Así surgen las "*Softgoals*", para representar Metas u Objetivos poco claros y sus interdependencias.

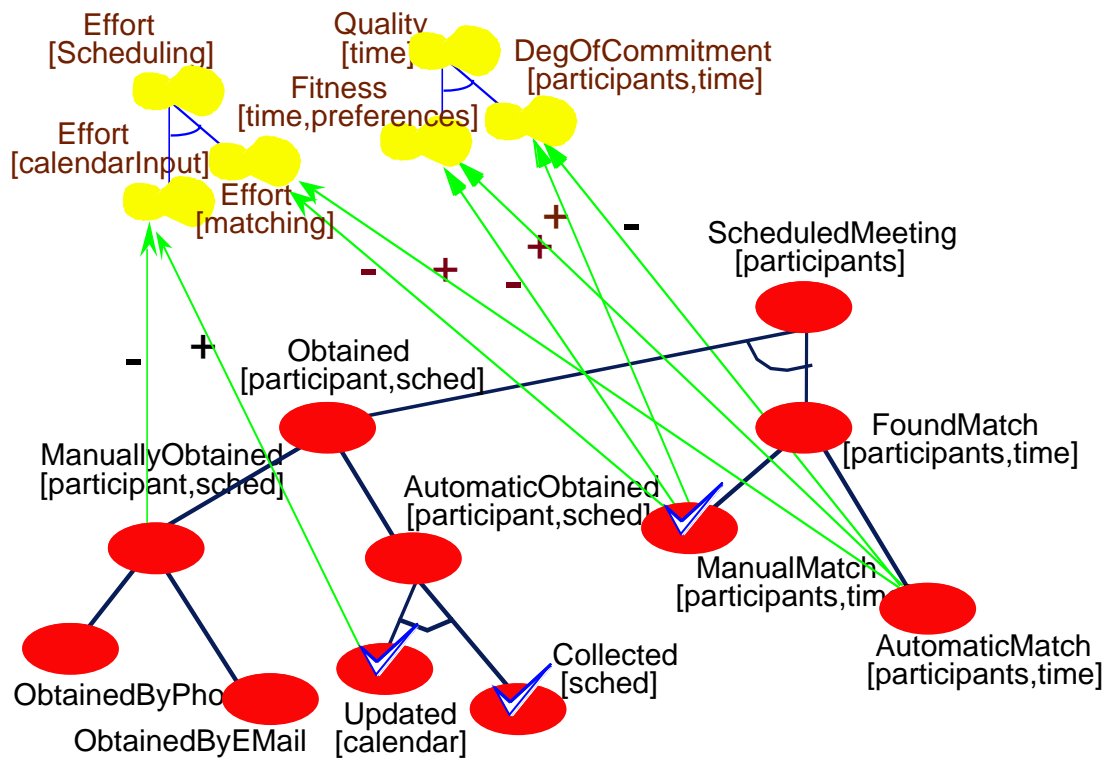
Se establece que una *Softgoal* se satisface cuando hay suficiente evidencia positiva y poca evidencia negativa contra ella, y no se satisface cuando hay suficiente evidencia negativa y poca evidencia positiva contra ella. Para su representación, también se arma una jerarquía utilizando un árbol n-ario con descomposiciones AND/OR, donde además los arcos se etiquetan con + y con - según la influencia positiva o negativa del hijo hacia el padre.

Debido a que las *Softgoals* pueden ser conflictivas unos con otras, se establece un análisis de correlación como medio de ayuda para descubrir relaciones "laterales" positivas o negativas entre esas *Softgoals*. Esas correlaciones pueden descubrirse usando reglas genéricas que establecen condiciones bajo las cuales las *Softgoals* de un tipo pueden influenciar positivamente o negativamente sobre *Softgoals* de otro tipo.

Por último, la propuesta final de este enfoque consiste en utilizar las jerarquías de *Softgoals* como medio de evaluación de las descomposiciones de metas u objetivos funcionales. Para esto desde la jerarquía de Metas/Objetivos, se establecen arcos etiquetados con + ó - hacia los nodos de las jerarquías de *Softgoals*. Esta se considera la principal ventaja que ofrece la distinción de metas u objetivos y *Softgoals*.

Se representa en la Figura 3.6 un ejemplo de las jerarquías de metas u objetivos y *Softgoals* y sus correlaciones.

Figura 3.6 – Ejemplo del Meeting Scheduler [Mylopoulos01]



El proceso propuesto por [Milopoulos01] se resume del siguiente modo:

Entrada: conjunto de Metas/Objetivos funcionales

a) Descomponer cada Meta/Objetivo funcional en una jerarquía AND/OR.

b) Descomponer cada rasgo de calidad dado en una jerarquía de Softgoals.

c) Identificar correlaciones entre Softgoals.

d) Identificar correlaciones entre Goals y Softgoals. Seleccionar un conjunto de "hojas" Softgoals que mejor "satisfagan" todos los rasgos de calidad pedidos.

e) Seleccionar un conjunto de Goals y Softgoals, que "satisfagan" todas las metas funcionales dadas y "mejor satisfagan" todos los rasgos de calidad pedidos.

Salida: La salida es un conjunto de funciones a ser ejecutadas por el sistema que colectivamente cumplirán las metas funcionales definidas.

El método propuesto permite fundamentar la elección de una solución particular para desarrollo de un sistema de software, y posibilita vincular decisiones de diseño con los objetivos organizacionales y técnicos que las originaron.

3.2.4 Enfoque de Loucopoulos P. y Karakostas V.

El enfoque de Loucopoulos y Karakostas [Loucopulos95] se basa en comprender el concepto de la *Visión Teleológica de los Sistemas*. De acuerdo a esta visión, un sistema (organización, máquina, persona, etc.) tiene un conjunto de metas las cuales deben ser obtenidas. De este modo, la Visión Teleológica intenta explicar el comportamiento de un sistema en términos de sus metas.

Se concibe a una meta como un estado definido del sistema. Dado que un estado se describe en términos de valores de un número de parámetros, una meta puede ser alternativamente definida como un conjunto de valores deseados para un número de parámetros (ejemplo en la meta "Tener una ganancia de un millón en el próximo año", el parámetro de la meta es "ganancia" y el valor deseado "un millón").

Las metas pueden variar en su grado de especificidad o abstracción. El grado de especificidad de las metas tiene que ver con la organización

jerárquica de las personas. Por ejemplo, el trabajo de quien administra una empresa es tomar decisiones sobre la estrategia general de la organización. Esto hace que sus metas sean necesariamente más abstractas que las metas en los bajos niveles de decisión en la jerarquía de la organización. Por lo tanto, las metas pueden ser estratégicas (las más abstractas), tácticas u operacionales (las menos abstractas).

De este modo, este enfoque ve al dominio del problema como una jerarquía de Objetivos, Metas y Sub-Metas. Las metas más abstractas son frecuentemente llamadas "Objetivos". Los "Objetivos" usualmente no especifican "cuándo", "cuánto" ni "cómo". Un ejemplo de objetivo es "aumentar la utilidad".

Un objetivo se descompone en una jerarquía AND/OR de objetivos menos abstractos. Las metas con más bajo nivel en la jerarquía, representan el modo en el cual un objetivo más abstracto puede ser obtenido. Dos metas que aparecen en el mismo nivel en la jerarquía, pueden ser mutuamente conflictivas o de apoyo mutuo (la satisfacción de una de ellas afecta positivamente la obtención de la otra). Por ejemplo las metas "Incrementar la automatización" y "Aumentar inversión en tecnología" serían de apoyo mutuo, e "Incrementar la automatización" y "Reducir inversión en tecnología" serían mutuamente conflictivas.

Una restricción es definida como una limitación que impide la satisfacción de un objetivo, y puede ser generada por la organización o su entorno (clientes, factores externos). Más aún, en el caso de sistemas de software existen restricciones inherentes a la tecnología.

El proceso de análisis de Objetivos/Metas de este enfoque es resumido del siguiente modo:

- a) Analizar la organización y el ambiente externo con la cual ella interactúa en términos de objetivos, metas y restricciones.***

- b) Crear una jerarquía metas-submetas consistente en: objetivos organizacionales, metas y restricciones, y sus interrelaciones (soporte, conflicto, restricción)*
- c) Validar el modelo y obtener el consenso entre stakeholder acerca de él.*
- d) Identificar la parte de la jerarquía de meta-submeta que modeliza la parte de procesamiento de la información de la organización.*
- e) Eliminar los casos de conflictos negociando y pactando con los stakeholders.*
- f) Seleccionar tareas (requerimientos) por eliminación de alternativas en la jerarquía.*

Esta visión de Objetivos/Metas aborda satisfactoriamente el problema de elicitar requerimientos dado que el analista tiene una clara comprensión del dominio del problema, incluyendo la distinción entre lo que pertenece al sistema de software y lo que pertenece a su contexto donde será insertado. Además, un número de soluciones potenciales podría ser considerado y comparativamente evaluado[Loucopulos95].

La utilización de jerarquía de objetivos permite identificar los requerimientos de software en el contexto del dominio del problema y además posibilita el vínculo o "mapeo" de los requerimientos hasta los objetivos de alto nivel del sistema [Loucopulos95].

3.2.5 Enfoque de Colette Rolland, Souveyet C. y Ben Achour C.

El enfoque CREWS-L'Écritoire desarrollado [Rolland98] [Rolland99] como parte del proyecto CREWS (Cooperative Requirements Engineering With Scenarios) ESPRIT, se centraliza en un acoplamiento bidireccional entre Metas/Objetivos y Escenarios.

Propone y define el uso de "Requirement Chunk" (RC) como un par <G,Sc> donde G es una Meta/Objetivo y Sc un Escenario. Dado que una

Meta/Objetivo es intencional y un Escenario de naturaleza operacional, un RC es un posible camino de alcanzar la Meta/Objetivo.

Un objetivo se define como algo que algún stakeholder espera alcanzar en el futuro. En este enfoque una meta/objetivo se expresa como una cláusula con un verbo principal y varios parámetros, donde cada parámetro tiene un papel diferente respecto al verbo. Ejemplo:

*Proveer*_{verbo} *(eficientemente)*_{calidad} *(electricidad)*_{objeto} *(desde el productor de energía)*_{fFuente} *(para nuestros clientes)*_{beneficiario} *(usando la red de energía)*_{medio} *(de una manera normal)*_{modo}

Un *Escenario* está compuesto por una o más acciones, en el cual una acción es una interacción entre agentes. La combinación de escenarios expresa la mayoría del comportamiento que es necesario para el propósito de modelización de objetivos basado en escenarios.

Un Escenario está caracterizado por un estado inicial y un estado final. El estado inicial define una precondition a cumplirse antes de su comienzo y el estado final define un estado a ser alcanzado al finalizar el escenario. Se hace una distinción entre Escenarios normales y excepcionales. Los primeros conducen a alcanzar sus Metas/Objetivos asociados, mientras que los últimos fallan en la satisfacción de Metas/Objetivos.

Se introducen además tres niveles de abstracción: *contextual*, *funcional* y *físico*. El nivel contextual identifica el servicio que un sistema debería proveer a una organización y sus fundamentos. El funcional se centraliza en la interacción entre el sistema y sus usuarios, para alcanzar los servicios necesitados. El nivel físico trata con la real performance de las interacciones. Cada nivel corresponde a un tipo de RC. Como resultado, se organiza a los requerimientos en una jerarquía de abstracción de tres niveles.

Existen tres tipos de relaciones entre RC llamadas relaciones de composición, alternativa y refinamiento. Las relaciones de composición y

alternativa conducen a una estructura AND/OR entre RCs (extensiones de las relaciones AND/OR convencionales entre metas/objetivos). La relación de refinamiento vincula RCs de diferentes niveles de abstracción, estableciendo un enlace vertical entre ellos.

El descubrimiento de Metas/Objetivos de este enfoque está focalizado en Metas menos abstractas (sub-metas) y no en Metas más abstractas (super-metas) o de más alto nivel. Estas últimas, consideradas como metas/objetivos iniciales en el proceso, son obtenidas a través de un adecuado estudio de documentos iniciales y sesiones de brainstorming.

Por cada Meta/Objetivo elicitado, se crea un Escenario, el cual es analizado para evaluar la Meta/Objetivo. La secuencia de pasos a cumplir por el proceso es:

Identificación inicial de Metas/Objetivos

Repetir

Análisis de Metas/Objetivos

Creación de Escenarios

Elicitación de Metas/Objetivos a través del análisis de los Escenarios creados

Hasta que todas Metas/Objetivos han sido elicitados

La elicitación de metas/objetivos y creación de escenarios son pasos complementarios, y las Metas/Requerimientos son "identificadas" incrementalmente mediante la repetición de *Análisis de Metas/Objetivos*, *Creación de Escenarios* y *Elicitación de Metas/Objetivos a través del análisis de los Escenarios creados*.

Para lograr la elicitación de Metas/Objetivos a través del análisis de Escenarios, se definen 3 estrategias de identificación:

- Estrategia de Refinamiento: descubre Metas/Objetivos de menor nivel de abstracción que una Meta/Objetivo dada
- Estrategia de Composición: descubre Metas/Objetivos cuya conjunción compone la Meta/Objetivo original

- Estrategia de Alternativa: descubre Metas/Objetivos cuya disyunción compone la Meta/Objetivo original

Cada uno de los 3 pasos del ciclo es soportado por un mecanismo que guía la ejecución de cada paso. El mecanismo que guía el análisis de metas/objetivos está basado sobre un análisis lingüístico de la declaración de metas/objetivos. Esto ayuda a reformular una declaración narrativa de una meta/objetivo como una plantilla mostrada en el ejemplo de meta/objetivo previamente dado.

En resumen, se plantea que el uso de escenarios ayuda a encontrar las metas/objetivos correctos, y que el acoplamiento bidireccional entre metas/objetivos y escenarios facilita los cambios de niveles de abstracción así como el descubrimiento de la funcionalidad completa del sistema.

3.2.6. Enfoque de Oshiro K., Watahiki K. y Saeki M.

La visión de Oshiro K., Watahiki K. y Saeki M. [Oshiro03] enfatiza las debilidades de otros enfoques orientados por objetivos que utilizan jerarquías AND/OR, respecto a la carencia de un mecanismo que esencialmente:

- a) Facilite las actividades de refinamiento y descomposición de objetivos en objetivos más concretos o sub-objetivos.
- b) Facilite las actividades colaborativas entre stakeholders para elicitación de objetivos y construir la jerarquía de objetivos.

En particular, considera a los stakeholders como fuente de conocimiento que juega un papel importante en la elicitación de requerimientos de alta calidad, y que deberían participar en las actividades de elicitación de requerimientos.

El método propuesto basado en generación de ideas, donde se combina Brainstorming con análisis orientado por objetivos, debe ser usado por stakeholders para identificar sub-objetivos. Las ideas que los

stakeholders llevan a su mente usando el método, se consideran sub-objetivos candidatos.

Los stakeholders aplican esta estrategia en reuniones "cara a cara", y para hacer eficientes esos encuentros hay un coordinador, usualmente un analista experimentado, que controla el progreso de las reuniones.

Los pasos que sigue el método son:

a) Incluir las necesidades de los clientes como objetivos iniciales

Repetir

b) Seleccionar un objetivo de la jerarquía de objetivos

c) Generar ideas vinculadas al objetivo seleccionado

d) Agrupar las ideas generadas

e) Buscar asociaciones entre los grupos de ideas

f) Se agregan los grupos de ideas y sus asociaciones al grafo de objetivos

Hasta obtener "suficiente" objetivos en el grafo

a) Incluir las necesidades de los clientes como objetivos iniciales

Con los objetivos iniciales se generan una jerarquía.

b) Seleccionar un objetivo de la jerarquía de objetivos

El coordinador elige un objetivo de los ya elicitados, para que sea descompuesto y refinado.

c) Generar ideas vinculadas al objetivo seleccionado

Cada stakeholder genera ideas vinculadas al objetivo elegido, las escriben en tarjetas que luego se explican al resto de los participantes, y se disponen en un pizarrón.

d) Agrupar las ideas generadas

Las ideas que semánticamente se relacionan con otras son agrupadas en el pizarrón y encerradas en círculos formando conjuntos o grupos de

ideas. Cada conjunto o grupo de ideas se describe con otra tarjeta que se ubica sobre el borde del conjunto.

e) Buscar asociaciones entre los grupos de ideas

Se buscan asociaciones entre los grupos y/o las ideas que los forman, y se dibujan esas asociaciones en el pizarrón.

f) Se agregan los grupos de ideas y sus asociaciones al grafo de objetivos

De acuerdo a las asociaciones identificadas, los participantes seleccionan los candidatos a sub-objetivos del objetivo inicialmente elegido, a partir de los grupos de ideas y sus elementos. Intuitivamente, los grupos de ideas y sus elementos tienen un vínculo con el objetivo inicialmente elegido y pueden ser entonces sub-objetivos.

Esta metodología prioriza la participación de stakeholders y sus ideas se reflejan directamente. Muchas de esas ideas generadas, conducen a una rápida identificación de objetivos [Oshiro03].

Los objetivos obtenidos a partir de ideas son de "alta calidad" debido a que están basados en el consenso de los stakeholders, aunque la eficiencia de las reuniones depende de la capacidad del coordinador [Oshiro03].

3.3 Resumen comparativo

En la tabla 3.3 se reproduce una comparación entre los enfoques presentados. Para cada enfoque se indica cuál es su comportamiento respecto a un conjunto de variables. Dichas variables son:

- **Formal / No Formal:** se refiere a la forma de expresión. La utilización de un lenguaje de especificación rigurosamente definido, consistente y no ambiguo corresponde a un lenguaje formal. En caso contrario el lenguaje es informal (Ej: lenguaje natural).

- **Jerarquía AND-OR:** es una forma de organizar el conjunto de objetivos, generalmente utilizando un árbol n-ario, que posibilita la expresión de los objetivos en diferentes niveles de abstracción, tal como se explicó al principio de este capítulo.
- **Softgoal:** es el concepto expresado por Mylopoulos J., Chung L., Liao S., Wang H. y Yu E. [Mylopoulos01], expresado en el punto 3.2.3 de este capítulo.
- **Método definido:** se refiere a la secuencia de actividades definidas para el tratamiento de objetivos.
- **Identificación de conflictos:** este concepto está vinculado a la identificación de objetivos que representen ideas de stakeholders que se contraponen respecto a una situación o tema particular.
- **Complemento con otra técnica de elicitación:** se refiere al uso de otra técnica de elicitación como soporte dentro de las actividades propuestas en el enfoque.
- **Esquema de representación:** se refiere a la existencia de una representación claramente definida para un objetivo.
- **Distinción entre metas y objetivos:** corresponde a la concepción explícita diferente de metas y objetivos.
- **Heurísticas definidas:** corresponde al uso de pautas que asisten al analista en la aplicación del método.
- **Operacionalización de objetivos en requerimientos:** se refiere a la transformación explícita de objetivos en requerimientos de un sistema de software.
- **Incluir objetivos de calidad:** se refiere a la inclusión de objetivos a partir de los cuales luego se deriven requerimientos no funcionales.
- **Taxonomía de objetivos:** este concepto se vincula al uso de una clasificación de objetivos, o a una distinción explícita entre objetivos (utilizando nombres diferentes) de acuerdo a su nivel de abstracción.

Tabla 3.3 – Cuadro Comparativo de enfoques de Objetivos

CARACTERISTICAS	G B R A M	K A O S	M Y L O P O U L O S	L O U C O P O U L O S	R O L L A N D	O S H I R O
Formal		√				
No Formal	√		√	√	√	√
Genera Jerarquía AND-OR		√	√	√	√	√
Usa Softgoal		√	√			
Tiene un método definido	√	√	√	√	√	√
Identifica conflictos	√	√	√	√	√	√
Se complementa con otras técnicas de elicitación	√				√	√
Tiene esquema de representación	√	√			√	
Distingue metas de objetivos				√		
Tiene heurísticas definidas	√					
Operacionaliza objetivos en requerimientos	√	√				
Incluye objetivos de calidad	√	√	√	√		√
Taxonomía de objetivos	√	√	√	√		

Capítulo 4

Elicitación de Objetivos

4.1 Análisis de los enfoques presentados

Los enfoques orientados a objetivos en el capítulo anterior ofrecen distintas perspectivas de concepto, modelización y uso de objetivos. Esas diferencias permiten disponer de una herramienta de elicitación de requerimientos orientada hacia una técnica específica, con características particulares de aplicación. No obstante, hay similitudes que merecen ser destacadas. Un ejemplo de esas similitudes es la visión y clasificación de objetivos en diferentes niveles de abstracción, así como su representación en jerarquías con estructuras no lineales (árboles o grafos). Además, todos están orientados a elaborar o construir Metas u Objetivos, a partir de los cuales se obtienen los requerimientos del sistema de software, o son una parte decisiva de esos requerimientos. Se dispone entonces de un amplio espectro de elección dentro de esta técnica de elicitación de requerimientos.

A pesar de que podrían analizarse con más profundidad las ventajas y desventajas de cada enfoque, no es propósito de esta tesis. Otros trabajos ya lo ha realizado [Kavakli99] [Kaiva02].

Sin embargo, existe un aspecto en el cual no se ha puesto mayor énfasis y que es pertinente considerarlo. Ese aspecto está relacionado con el *proceso de obtención del conocimiento necesario para identificar los objetivos de un sistema de software y sus características*, es decir, con el **Proceso de Elicitación de Objetivos**.

Todos los enfoques estudiados se concentran en metas/objetivos, pero cabe preguntarse para cada uno de ellos:

- ¿Tiene un camino específico para la obtención de objetivos?

- ¿Define un “proceso” de elicitación de estos objetivos?

A continuación, se analiza este aspecto en cada uno de ellos.

GBRAM [Anton96] [Anton97] [Anton98] propone identificar y extraer metas/objetivos a través de la exploración de las fuentes de información existentes.

Cada “porción” de información explorada debe ser analizada a través de la realización de preguntas. Un ejemplo simple de esas preguntas es: ¿Qué objetivo ejemplifica el fragmento de información explorado?

Como regla general, plantea que las afirmaciones que conducen a decisiones de diseño sugieren metas/objetivos posibles. También considera que los objetivos pueden ser identificados a través de la búsqueda de ciertos tipos de verbo (tales como “obtener”, “lograr”, “incrementar”, entre otros) que denoten algún estado que puede ser alcanzado dentro del sistema, una vez que la acción está finalizada. Todo esto, lo detalla en un conjunto de Heurísticas.

Las heurísticas en relación con esta tesis están vinculadas a la identificación de Metas/Objetivos, Stakeholders y Agentes. Existen 21 heurísticas de este tipo.

Ejemplos de tales heurísticas, entre otras, son:

- “Las metas son nombradas utilizando un subconjunto estandarizado del lenguaje natural, en el cual la primer palabra es un verbo que describe la clase de objetivo mencionado”
- “Palabras de acción clave tales como: seguir, monitorear, proveer, otorgar, evitar, asegurar, etc., apuntan a metas/objetivos candidatos”
- “Los stakeholders tienden a expresar sus requerimientos en término de operaciones y acciones, más que objetivos. Así,

cuando se dispone de la transcripción de una entrevista, se beneficia la estrategia de búsqueda de palabras de acción para extraer metas/objetivos, a partir de las descripciones de stakeholders.”

- “Los objetivos son también identificados considerando los posibles obstáculos de objetivos previamente especificados”
- “Los objetivos pueden ser identificados mediante la consideración de restricciones”

Evidentemente, existe un mecanismo de identificación de objetivos. No obstante, surgen una serie de interrogantes que sin dejar de valorar lo propuesto, no se pueden soslayar:

- ¿El mecanismo garantiza la identificación de objetivos?
- ¿Cómo se identifican las afirmaciones y los verbos?
- ¿Quién provee las fuentes de información para explorar y obtener estas afirmaciones y verbos?
- Si no se proveen estas fuentes de información, ¿cuáles son las técnicas y herramientas y cómo deben utilizarse para obtener las fuentes de información necesarias?

Estas son algunas de las preguntas que no tienen respuesta ni se deducen en GBRAM, y que un proceso de elicitación debe contemplar.

De este modo, es posible afirmar que GBRAM si bien posee pautas para elicitación de objetivos, detalladas en un conjunto de heurísticas, esas heurísticas no conforman un proceso de Elicitación de Objetivos.

KAOS [Lamsweerde93] [Lamsweerde96] [Lamsweerde00] para identificar metas/objetivos, propone la búsqueda de palabras claves (tales como objetivo, intención, propósito, entre otras) en los documentos de información disponibles.

A partir de allí, comienza a elaborar la jerarquía de metas/objetivos, que se completa a través de la realización de preguntas sobre las primeras metas/objetivos identificados.

Lamentablemente, este enfoque se limita a obtener el conocimiento de la documentación escrita (en parte coincidente con GBRAM), y deja de lado una porción demasiado importante del conocimiento necesario para formular los requerimientos de un sistema de software.

Además, la ausencia de palabras clave en los documentos disponibles de información no significa la ausencia de objetivos, con lo cual este mecanismo, sin dejar de ser valioso, no garantiza la identificación de metas/objetivos.

Para este caso valen las mismas preguntas que las realizadas para GBRAM, es decir, ¿quién provee los documentos de información? Si no hay documentos de información disponible, ¿cual es el método de recolección?

Es posible afirmar entonces que KAOS, aunque provee indicaciones para la identificación de objetivos, carece de un proceso de Elicitación de Objetivos.

El enfoque de **Mylopoulos J., Chung L., Liao S., Wang H. y Yu E.** [Mylopoulos01] omite la identificación de metas/objetivos. Directamente, considera que la entrada al proceso de análisis de objetivos por ellos establecido, es un conjunto de Metas/Objetivos funcionales, sin detallar cuando ni cómo se obtiene ese conjunto.

En este caso, claramente se ignora cualquier pauta o sugerencia que asista al analista en la identificación de objetivos, pudiendo afirmarse una vez más que se carece de un proceso de Elicitación de Objetivos.

Pericles Loucopoulos y Vassilios Karakostas [Loucopoulos95] proveen pautas de identificación de objetivos organizacionales. Su propuesta consiste en analizar la organización y el ambiente externo con la cual ella interactúa, en término de objetivos.

Este análisis de la organización remite a las técnicas de investigación de las empresas, es decir a recursos de indudable utilidad para la obtención de conocimiento del problema, pero que no agotan los objetivos que debe satisfacer un sistema de software. Además, este análisis de la organización no es un análisis guiado ni con el soporte de una metodología.

Por otra parte, no establecen la forma de "derivar" los objetivos del sistema de software a partir de estos objetivos organizacionales. Con lo cual, estas indicaciones se caracterizan más por su generalidad que por su precisión.

Se concluye entonces que la propuesta de Pericles Loucopoulos y Vassilios Karakostas no dispone de un proceso de Elicitación de Objetivos.

CREWS-L'Ecritoire [Rolland98] [Rolland99] distingue los objetivos iniciales del resto de los objetivos del sistema.

A partir de esta diferencia, provee una metodología para obtener objetivos, pero a partir del conjunto de objetivos iniciales. Sin embargo, no hay mayores indicaciones para identificar las metas/objetivos iniciales que luego serán quienes darán origen a nuevas metas/objetivos. Solo establece que esos objetivos se pueden obtener a través de un estudio de la documentación existente y sesiones de Brainstorming.

Esto constituye una indicación pertinente para la elicitación, pero no basta con una técnica, pues reduce la aplicación del enfoque a los ámbitos en los que el Brainstorming puede ser aplicado.

CREWS-L'Ecritoire al proveer una metodología para obtener objetivos, conforma un antecedente de la motivación de esta tesis. Se establece la necesidad de contar con un proceso de elicitación de objetivos, lo que es omitido por otros enfoques. Sin embargo, este criterio sistemático no se hace extensivo a los objetivos iniciales, a los que no les aplica esta metodología.

El enfoque concentrado en la generación de **Ideas** [Oshiro03], tiene como propósito asistir a los otros métodos basados en metas/objetivos en las tareas de elicitación, refinamiento y descomposición de metas/objetivos.

La propuesta afirma que estas tareas, especialmente la elicitación, no tienen un mecanismo que las facilite. Esta afirmación concuerda con la motivación de esta tesis, resumiendo en cierto aspecto las características enunciadas de los enfoques analizados, desde el punto de vista de la elicitación de metas/objetivos.

Como mecanismo de elicitación para asistir y facilitar esas tareas, propone reunir a los stakeholders y en función de una meta/objetivo previamente establecido, ellos deben generar ideas como base de identificación de nuevas metas/objetivos. Esto coincide con el propósito del mecanismo de Brainstorming.

De este aspecto se destacan claramente dos puntos. En primer lugar, se toman objetivos previamente establecidos, aunque no se plantea ¿cómo y de donde se obtienen esos objetivos? Directamente se establece que los objetivos iniciales son las necesidades de los clientes. Es evidente que semejante simplificación contrasta con un criterio metodológico para obtener esos objetivos.

El segundo punto a mencionar, es que mediante la utilización de sesiones de Brainstorming para generar nuevos objetivos, se está limitando la aplicación sólo a ámbitos donde sea propicia su aplicación, objeción que también se puede realizar a la propuesta de Rolland.

En resumen, este enfoque propone algunas indicaciones para la identificación de metas/objetivos, pero no un proceso de Elicitación de Objetivos.

Se pueden resumir las principales características de estos enfoques con respecto a la elicitación de metas/objetivos, en los siguientes puntos (detallados también en la Tabla 4.1):

- A. Algunos no proponen nada o sólo alguna indicación general.
- B. Otros proveen indicaciones más específicas, aunque acotadas y sin un criterio metodológico.
- C. Algunos establecen una o varias fuentes de elicitación de objetivos.
- D. No se dispone de pautas lo suficientemente precisas para obtener los objetivos.

Existe un enfoque denominado **AGORA** (Attributed Goal-Oriented Requirements Analysis Method) [Kaiva02] que está orientado a la resolución de las debilidades de los distintos enfoques de metas/objetivos.

Esto constituye un valioso aporte para la metodología de metas/objetivos. Sin embargo, en el análisis de esas debilidades, AGORA omite que los enfoques de metas/objetivos carecen de un Proceso de Elicitación. Este aspecto, refuerza la poca consideración que existe respecto a la elicitación de metas/objetivos.

Tabla 4.1 – Resumen Enfoques de Objetivos

	GBRAM	KAOS	Mylopoulos	Loucopoulos	Rolland	Ideas	AGORA
A			√	√		√	√
B	√				√		
C	√	√					
D	√	√	√	√	√	√	√

Se puede concluir entonces, que ninguno de los enfoques estudiados se concentra en la generación inicial de metas/objetivos. Por lo tanto, se carece de un Proceso de Elicitación de Metas/Objetivos, afirmación ya establecida por el autor de esta tesis en [Thomas03].

4.2 Propuesta de trabajo

El propósito de este trabajo a partir de establecer la carencia de un Proceso de Elicitación de Objetivos, es proponer un Proceso de Elicitación de Metas u Objetivos a partir de Escenarios.

Cabe mencionar, que la descripción del *"Proceso"* es utilizada en un sentido menos riguroso que el habitual en la Ingeniería Software [Humphrey89] aunque sí con un criterio sistemático y metodológico.

La Ingeniería de Requerimientos orientada a Objetivos y la Ingeniería de Requerimientos basada en Escenarios ubican a la Elicitación de Requerimientos en el contexto en el que el Sistema de Software operará [Rolland99].

Ambas técnicas han demostrado su utilidad, aunque poseen limitaciones una respecto de la otra. Precisamente, para superar esas limitaciones y deficiencias, se han propuesto enfoques de uso integrado de Objetivos y Escenarios [Kavakli96] [Rolland98] [Rolland99] [Liu01] [Kim03] [Uchitel04].

Este trabajo también asume en parte el mismo argumento: para superar la limitación de que ningún enfoque orientado a Objetivos plantea un proceso o guía metodológica para la elicitación de Objetivos, se propone la integración de Objetivos con Escenarios.

Los Escenarios se obtienen más rigurosamente que los Objetivos, debido a que tienen definido un procedimiento de elicitación a partir del Léxico Extendido del Lenguaje, tal como se describió en el capítulo 2. Los Objetivos en cambio, sólo pueden hacerse explícitos después de entender con más profundidad el sistema a desarrollar [Rolland99].

Los Escenarios utilizan un vocabulario que captura el lenguaje usado por los expertos del dominio y además describen el comportamiento del

sistema. Por lo tanto, en los escenarios está el conocimiento del dominio a partir del cual se puede obtener objetivos.

Adicionalmente, se considera que el uso combinado de más de una técnica de elicitación para obtener requerimientos, ayuda al analista en sus tareas de especificación de requerimientos.

Para comenzar a desarrollar la propuesta de este trabajo, el punto de partida consistió en seleccionar una metodología basada en Objetivos para el cual se elabore un "Proceso de Elicitación".

La metodología elegida fue GBRAM. Las razones de su elección son las siguientes:

- No es formal, por lo que no se requiere experiencia en lenguajes formales. Consecuentemente, los objetivos de ese enfoque son más fáciles de entender y validar, lo que permite y promueve la participación activa de los stakeholders.
- Las metas/objetivos a obtener tienen un esquema expresivo, formado por una serie de atributos que no sólo permiten su comprensión sino además comprender su ámbito de aplicación.
- Existe amplia documentación disponible de esta metodología.
- Es ampliamente aceptado en la comunidad de Ingeniería de Requerimientos y es considerado como uno de los principales referentes de los enfoques orientados a metas/objetivos.

En el caso de los Escenarios, se seleccionó el enfoque de Julio Leite por los siguientes motivos:

- El esquema de cada escenario es simple y fácil de entender.
- Existe amplia documentación disponible.
- Existen casos de estudio completos (como lo son los Escenarios para la obtención de Pasaporte y la Administración de la Recepción de un Hotel).
- La "aparente" similitud semántica con las metas/objetivos de GBRAM.
- Tiene definido un proceso de elicitación a partir del LEL.

Finalmente, es así como se propone la Definición de un Proceso de Elicitación de Objetivos con formato GBRAM [Anton96] [Anton97] [Anton98], a partir de Escenarios con el esquema de Julio Leite [Leite97] [Leite00].

Capítulo 5

Proceso de Elicitación de Objetivos

5.1 Estrategia para obtener Objetivos

En los casos “*Léxico extendido del lenguaje y Escenarios del sistema nacional para la obtención de pasaportes*” [Leite96] y “*Administración de la Recepción de un Hotel*” [Bertolami01] se ha estudiado el vínculo existente entre el conocimiento del problema que contiene un Escenario [Leite97] [Leite00] y la información necesaria para representar una Meta u Objetivo con el esquema de GBRAM [Anton96] [Anton97] [Anton98]. Para ello, en cada caso se analizaron los componentes/elementos de los escenarios y se buscaron eventuales analogías semánticas con la representación de objetivos. Se controlaron las relaciones encontradas entre distintos escenarios y los posibles objetivos “derivados” a partir de ellos.

En todos los casos se arribó a resultados similares, lo que permitió la definición de una estrategia que utiliza un conjunto de *reglas*, con un modelo de representación uniforme. Una versión inicial de algunas de estas reglas fue anticipada con un formato preliminar en [Thomas04], y la versión definitiva de todas en [Thomas05].

Básicamente, la estrategia consiste en aplicar reglas a cada componente de un Escenario, obteniéndose de esta forma los componentes de un Objetivo. En GBRAM este luego se clasifica en términos de Logro (“Achievement”) o “Mantenimiento” (“Maintenance”).

Dado que el enfoque de Escenarios contempla la eliminación de redundancias y, consecuentemente, su depuración; la derivación de objetivos tendrá las mismas características, es decir, sin necesidad de depuración, considerando que de cada Escenario se deriva un Objetivo con

todos sus componentes. Esto garantiza que no son necesarias las actividades de GBRAM para con los Objetivos derivados. Por lo tanto, la propuesta consiste sólo en obtener Objetivos con formato GBRAM a partir de Escenarios.

El mecanismo se basa en establecer una asociación entre Escenarios y Objetivos, y a partir de cada Escenario identificar el Objetivo correspondiente.

5.2 Definición de reglas

La relación muy directa establecida entre los escenarios y los objetivos, permitió utilizar un formato de representación de las reglas que permite una aplicación muy simplificada de las reglas. Se utilizó un modelo entrada-proceso-salida que se describe en la Tabla 5.1.

Tabla 5.1 - Esquema de las reglas

Nº de Regla	
ENTRADA	Elemento del Escenario
Transformación	Es la acción que se ejecuta para convertir el elemento del Escenario en un Elemento del Objetivo
SALIDA	Elemento del Objetivo

Este modelo recibe como entrada un componente de escenario y produce como salida un componente de objetivo. La relación entre estos elementos no es unívoca como se verá más adelante, es decir, que a partir de un componente de un escenario se puede identificar más de un elemento de un objetivo. El nivel de entrada de todas las reglas es el de un escenario que cumple con los requisitos definidos para los escenarios.

El propósito fue disponer de un conjunto de reglas tal que aplicadas a un escenario, permitan identificar al menos un objetivo con todos sus elementos, de acuerdo al formato de representación de GBRAM.

Se establecieron 12 reglas que, según el alcance de su aplicación, se agruparon en cuatro clases (entre paréntesis se indica la cantidad reglas de cada clase):

1. Reglas aplicadas a la Identificación de Escenarios (2)
2. Reglas aplicadas a los Componentes de Escenarios (5)
3. Reglas aplicadas a los Episodios de Escenarios (3)
4. Reglas aplicadas a las Bifurcaciones de Escenarios (2)

Si bien cada elemento de un escenario se denomina componente, se determinó esta clasificación agrupando estos elementos por su significado, a partir del cual se obtuvieron 4 grandes grupos. El primero está vinculado al nombre del escenario dado que es quien identifica al escenario; el segundo al objetivo, contexto y actor; el tercero a los episodios y el cuarto a los casos alternativos.

Los recursos de un escenario permiten identificar los objetos pasivos con los cuales los actores trabajan. Dado que estos objetos no son necesarios en la representación de un objetivo con formato GBRAM, son obviados en la definición de reglas. Por lo tanto, hay información de los escenarios que no es utilizada.

Los ejemplificación de las reglas se basa en el escenario titulado "*Cancelación de la Reserva*" para el problema de la Administración de la Recepción de un Hotel [Bertolami01].

5.2.1 Reglas aplicadas a la Identificación de Escenarios

La **Regla 1** establece el nombre del objetivo. El nombre o título de un escenario identifica el escenario. Su sintaxis incluye un verbo que indica una acción a ejecutar. El nombre de un objetivo es el único identificador del objetivo. Los objetivos se establecen para indicar un estado deseado, y no deberían identificarse directamente como una acción [Anton97].

Si se toma el nombre del escenario y se indica un estado deseado a partir de él, se obtiene un objetivo. Consecuentemente, se considera el estado alcanzado al cumplir un escenario (indicado por su nombre) como un objetivo a satisfacer, identificando ese objetivo. Por lo tanto, ese estado alcanzado representa el nombre de un objetivo. En la Tabla 5.2 se reproduce la regla.

Tabla 5.2 - Regla 1

Regla 1	
ENTRADA	Nombre del Escenario
Transformación	Incorporar el estado alcanzado a partir del cumplimiento del escenario analizado como el nombre de un objetivo.
SALIDA	Nombre del Objetivo

Ejemplo: el escenario titulado *“Cancelación de la Reserva”* en el contexto de la Administración de Recepción de un Hotel, incluye el verbo “cancelar”, y el estado deseado en ese caso es que la reserva esté cancelada, generando de esa forma el objetivo *“Reserva Cancelada”*. Por lo tanto, se establece que *“Reserva Cancelada”* es el nombre del objetivo derivado del escenario *“Cancelación de la Reserva”*.

La **Regla 2** toma como entrada el nombre del escenario. El nombre del escenario con el verbo en infinitivo es la acción que resume todo el escenario. La acción de un objetivo es el nombre de la operacionalización del objetivo. Indica el comportamiento que debe estar presente para llevar a cabo el objetivo [Anton97].

Existe una analogía entre la acción indicada por el escenario y la acción que se debe cumplir para satisfacer un objetivo derivado de ese escenario.

Por lo tanto, se incorpora el nombre del escenario con el verbo en infinitivo como la acción del objetivo o meta derivada de ese escenario. Si

el nombre del escenario no tuviera un verbo que represente una acción del lado del "sistema", se toma como entrada el verbo del objetivo del escenario. En la Tabla 5.3 se reproduce la regla.

Tabla 5.3 - Regla 2

Regla 2	
ENTRADA	Nombre del Escenario + {Objetivo del Escenario}
Transformación	Incorporar el nombre del escenario con el verbo (que integra el nombre del escenario o el objetivo del escenario) en infinitivo, como la acción del objetivo o meta derivada de ese escenario.
SALIDA	Acción del Objetivo

Ejemplo: el escenario *"Cancelación de la Reserva"* incluye la acción de cancelar, siendo *"Cancelar la Reserva"* la acción que operacionaliza el objetivo *"Reserva Cancelada"*. Por lo tanto, se establece que *"Cancelar la Reserva"* es la acción del objetivo *"Reserva Cancelada"* derivado del escenario *"Cancelación de la Reserva"*.

El escenario *"Pago"* incluye la acción recibir en el objetivo, siendo *"Recibir Pago"* la acción que operacionaliza el objetivo *"Pago recibido"*. Por lo tanto se establece que *"Recibir Pago"* es la acción del objetivo *"Pago Recibido"* derivado del escenario *"Pago"*

5.2.2 Reglas aplicadas a las Componentes de Escenarios

La **Regla 3** tiene como entrada el objetivo del escenario. El objetivo de un escenario establece su propósito o finalidad [Leite97] [Leite00], permitiendo que su comprensión sea más clara. Representa una descripción resumida de lo que se pretende realizar. La descripción de un objetivo en GBRAM [Anton97] es un texto informal aclaratorio que también ayuda a la comprensión del objetivo. Por lo tanto, se considera el

objetivo del escenario como una descripción del objetivo o meta derivado del escenario. En la Tabla 5.4 se reproduce la regla.

Tabla 5.4 - Regla 3

Regla 3	
ENTRADA	Objetivo del Escenario
Transformación	Incorporar el objetivo del escenario como parte de la descripción del objetivo o meta derivado de ese escenario.
SALIDA	Descripción del Objetivo

Ejemplo: el objetivo del escenario *“Cancelación de la Reserva”* es *“dar de baja una solicitud de reserva de un pasajero / huésped / pax”*. Esta acción es exactamente una descripción resumida de lo que se pretende en el objetivo *“Reserva Cancelada”* derivado del escenario *“Cancelación de la Reserva”*. Por lo tanto, se concluye que *“Dar de baja una solicitud de reserva de un pasajero / huésped / pax”*, objetivo del escenario *“Cancelación de la Reserva”*, es descripción del objetivo *“Reserva Cancelada”* derivado del escenario *“Cancelación de la Reserva”*.

La **Regla 4** tiene como entrada el contexto del escenario. El contexto de un escenario describe las acciones previas necesarias para iniciar el escenario, las precondiciones, la ubicación física y temporal [Leite97] [Leite00]. Esta información permite comprender el ámbito de desarrollo del escenario, es decir, dónde y cuándo transcurre el escenario y bajo qué condiciones. Se plantea entonces que para un objetivo derivado de un escenario, el contexto del escenario ayuda (aunque sea redundante) a la comprensión del objetivo. Consecuentemente se incorpora el contexto (parcial o total) del escenario como parte de la descripción del objetivo o meta derivado de ese escenario. En la Tabla 5.5 se reproduce la regla.

Ejemplo: el contexto del escenario *“Cancelación de la Reserva”* es *“Se realiza en la Recepción del Hotel. Existe una solicitud de reserva para un pasajero / huésped / pax”*. Esta información permite comprender el

ámbito y condiciones de la cancelación. En el objetivo *“Reserva Cancelada”* esa misma información aclara o describe el objetivo, y se puede incorporar (parcial o totalmente) como parte de la descripción del objetivo. Por lo tanto, se incorpora *“Se realiza en la Recepción del Hotel”*, ubicación física en el contexto del escenario *“Cancelación de la Reserva”*, como descripción del Objetivo *“Reserva Cancelada”* derivado del escenario *“Cancelación de la Reserva”*, dejando el resto del contexto del escenario como información de entrada para la aplicación de las Regla 5.

Tabla 5.5 - Regla 4

Regla 4	
ENTRADA	Contexto del Escenario
Transformación	Incorporar el contexto (parcial o total) del escenario como parte de la descripción del objetivo o meta derivado de ese escenario.
SALIDA	Descripción del Objetivo

La **Regla 5** también tiene como entrada el contexto del escenario, y como se describió en la regla anterior, el contexto puede incluir explícitamente una precondition. La precondition en un escenario indica que su desarrollo está sujeto a una condición necesariamente existente, marcando el estado inicial del escenario. Una precondition debe existir para que la satisfacción de un objetivo sea posible. La precondition se define como una condición que incluye un predicado (lista de cláusulas), y el predicado es falso si una de las cláusulas es falsa [Anton97]. Si no se declara una precondition, entonces el objetivo no depende de ningún agente, entidad u otro objetivo. Existe entonces una analogía entre lo que significa una precondition para un escenario y lo que significa para un objetivo.

Dado que el contexto del escenario incluye o puede incluir precondiciones, se considera la precondition incluida en el contexto del escenario como una precondition del objetivo o meta derivado de ese escenario. En la Tabla 5.6 se reproduce la regla.

Tabla 5.6 - Regla 5

Regla 5	
ENTRADA	Contexto del Escenario
Transformación	Incorporar la precondition incluida en el contexto del escenario como una precondition del objetivo o meta derivado de ese escenario.
SALIDA	Precondición del Objetivo

Ejemplo: El contexto del escenario *"Cancelación de la Reserva"* incluye la precondition *"Existe una solicitud de reserva para un pasajero / huésped / pax."*. Esa condición es exactamente una precondition del objetivo *"Reserva Cancelada"*, dado que la reserva no puede llegar al estado de cancelada sin la existencia previa de una solicitud de reserva. Por lo tanto, se incorpora *"Existe una solicitud de reserva para un pasajero / huésped / pax."*, precondition en el contexto del escenario *"Cancelación de la Reserva"*, como precondition del objetivo *"Reserva Cancelada"* derivado del escenario *"Cancelación de la Reserva"*.

La **Regla 6** toma a los actores como entrada. Un actor es una persona, dispositivo o estructura de la organización que tiene un papel en el escenario [Leite97] [Leite00]. El papel dentro del escenario responsabiliza al actor en el desarrollo del escenario.

Para un objetivo de [Anton97], un agente es el responsable de completar o satisfacer el objetivo, identificándose a través de un nombre. Compartiendo el criterio de responsabilidad, un actor es al escenario como un agente al objetivo. Por lo tanto cada actor del escenario que representa una persona física (del "lado" del sistema) es un agente potencial del objetivo o meta derivado de ese escenario. En la Tabla 5.7 se reproduce la regla.

Ejemplo: en el escenario *"Cancelación de la Reserva"* el actor *"recepcionista"* es el responsable de realizar la cancelación, es decir que tiene la responsabilidad de que la reserva se cancele. De ese modo, es el responsable de que se complete el objetivo *"Reserva Cancelada"*. Por lo

tanto, se considera el *“repcionista”*, actor del escenario *“Cancelación de la Reserva”*, como agente del objetivo *“Reserva Cancelada”* derivado del escenario *“Cancelación de la Reserva”*.

Tabla 5.7 - Regla 6

Regla 6	
ENTRADA	Actor del Escenario
Transformación	Incorporar cada actor del escenario que represente una persona física (del “lado” del sistema) como agente del objetivo o meta, derivado de ese escenario.
SALIDA	Agente del Objetivo

La **Regla 7** también toma al actor como su entrada. Como parte interviniente del escenario, generalmente la participación del actor es parte interesada en el desarrollo del escenario. Los stakeholders son aquellos participantes que tienen interés en que un objetivo se complete, y se identifican a través de un nombre [Anton97]. Existe entonces una similitud entre un actor del escenario con un stakeholder del objetivo, y se puede establecer que son la misma entidad. En la Tabla 5.8 se reproduce la regla.

Tabla 5.8 - Regla 7

Regla 7	
ENTRADA	Actor del Escenario
Transformación	Incorporar el actor del escenario (del “lado” del sistema) como STK del objetivo o meta, derivado de ese escenario
SALIDA	Stakeholder del Objetivo

Ejemplo: en el escenario *“Cancelación de la Reserva”* el *“repcionista”* y la *“agencia”* participan con interés en la cancelación. Es así como se establece que son parte interesada del objetivo *“Reserva Cancelada”*. Por lo tanto, se incorporan el *“repcionista”* y la *“agencia”*, actores del escenario *“Cancelación de la Reserva”*, como stakeholders del

objetivo *“Reserva Cancelada”* derivado del escenario *“Cancelación de la Reserva”*.

5.2.3 Reglas aplicadas a los Episodios de Escenarios

La **Regla 8** tiene a los episodios como entrada. Los episodios representan acciones realizadas por un actor donde participan otros actores y se utilizan recursos, detallando y otorgando comportamiento al escenario. Un episodio también puede hacer referencia a otro escenario [Leite97] [Leite00]. Los episodios se ejecutan en forma secuencial garantizando de ese modo el cumplimiento o desarrollo del escenario que integran. Del mismo modo, la satisfacción de un objetivo puede depender del cumplimiento de objetivos más específicos llamados sub-objetivos o sub-metas [Anton97]. Consecuentemente, se puede realizar un mapeo directo entre los episodios de un escenario, con las sub-metas de un objetivo originado por ese escenario, de acuerdo a la Regla 1. La sintaxis de un episodio incluye un verbo que indica una acción a ejecutar. El nombre de una sub-meta es su único identificador. Las sub-metas también son establecidas para indicar un estado deseado (más específico que el objetivo que integran), y no deberían ser identificadas directamente como una acción.

Por lo tanto, se considera cada estado alcanzado cumplido cada uno de los episodios del escenario analizado, como sub-meta del objetivo derivado del escenario que incluye esos episodios. En la Tabla 5.9 se reproduce la regla.

Ejemplo: en el escenario *“Cancelación de la Reserva”* se incluyen dos episodios: *“El recepcionista elimina la solicitud de reserva de la planilla de reservas”* y *“El recepcionista actualiza la disponibilidad de habitaciones en la planilla de ocupación de habitaciones”*. Estos episodios generan dos sub-metas, como estado deseado y no como acción: *“Solicitud de reserva eliminada por el recepcionista de la planilla de reservas”* y *“Disponibilidad de habitaciones actualizada por el recepcionista en la planilla de ocupación de habitaciones”*. El cumplimiento o satisfacción

de estas dos sub-metas, involucra el cumplimiento o satisfacción del objetivo *“Reserva Cancelada”*.

Por lo tanto, se establece que *“Solicitud de reserva eliminada por el recepcionista de la planilla de reservas”* y *“Disponibilidad de habitaciones actualizada por el recepcionista en la planilla de ocupación de habitaciones”*, episodios del escenario *“Cancelación de la Reserva”*, son sub-metas del objetivo *“Reserva Cancelada”* derivado del escenario *“Cancelación de la Reserva”*.

Tabla 5.9 - Regla 8

Regla 8	
ENTRADA	Episodio del Escenario
Transformación	Incorporar cada estado alcanzado cumplido cada uno de los episodios del escenario analizado, como sub-meta del objetivo derivado del escenario que incluye esos episodios.
SALIDA	Sub-Meta del Objetivo

Para la **Regla 9** se toma como entrada el último episodio del escenario. Como se enunció previamente, los episodios se ejecutan secuencialmente garantizando de ese modo el cumplimiento o desarrollo del escenario que integran.

Una poscondición de un objetivo indica un estado o circunstancia a la que se arriba una vez que el objetivo ha sido obtenido o completado, y se relaciona con la precondition [Anton97].

El último episodio del escenario es la última acción realizada en el escenario. Como consecuencia de esta acción, se genera un estado o circunstancia producto del cumplimiento del escenario. Puede establecerse entonces, que es una poscondición del escenario, aunque el esquema de [Leite97] [Leite00] no la incluya explícitamente.

Por lo tanto, se incorpora la consecuencia del último episodio de un escenario (o el episodio directamente), como una poscondición del objetivo o meta derivado de ese escenario.

De acuerdo a la regla 8, cada episodio del escenario genera una sub-meta del objetivo. Es así como el último episodio del escenario genera una sub-meta del objetivo. Entonces, puede haber coincidencia entre la última sub-meta del objetivo y la poscondición del objetivo. En la Tabla 5.10 se reproduce la regla.

Tabla 5.10 - Regla 9

Regla 9	
ENTRADA	Ultimo Episodio del Escenario
Transformación	Incorporar la consecuencia del último episodio (o el episodio directamente) de un escenario como una poscondición del objetivo o meta derivado de ese escenario.
SALIDA	Poscondición del Objetivo

Ejemplo: el último episodio del escenario *“Cancelación de la Reserva”* es *“el recepcionista actualiza la disponibilidad de habitaciones en la planilla de ocupación de habitaciones”*. La consecuencia de esta acción es *“las habitaciones están disponibles y la planilla de ocupación actualizada”*, siendo a la vez consecuencia del cumplimiento del escenario *“Cancelación de la Reserva”*. El objetivo *“Reserva Cancelada”* tiene como última sub-meta *“Disponibilidad de habitaciones actualizada por el recepcionista en la planilla de ocupación de habitaciones”*, derivada del último episodio del escenario del escenario *“Cancelación de la Reserva”*.

Por lo tanto, se establece que *“las habitaciones están disponibles y la planilla de ocupación actualizada”*, consecuencia del último episodio del escenario *“Cancelación de la Reserva”*, es una poscondición del objetivo *“Reserva Cancelada”* derivado del escenario *“Cancelación de la Reserva”*.

La **Regla 10** tiene en cuenta las condiciones de los episodios. Esas acciones pueden depender del cumplimiento de una condición previa o precondition. Para que un escenario se complete, deben ocurrir cada uno de sus episodios. Consecuentemente, si un episodio tiene una precondition, la misma debe ser verdadera para que el episodio ocurra y así se pueda completar el escenario. Es así como la precondition de un episodio afecta a todo el escenario.

Si se considera un objetivo derivado de un escenario, es evidente que el cumplimiento del objetivo está ligado a la posibilidad de que el escenario se complete.

Por lo tanto, se incorpora la precondition de un episodio de un escenario, como precondition del objetivo o meta derivado del mismo escenario. Además, si la precondition generada incluye una precondition ya existente para el mismo objetivo, se considerará como precondition solamente a aquella que abarque a la/s otra/s. En la Tabla 5.11 se reproduce la regla.

Tabla 5.11 - Regla 10

Regla 10	
ENTRADA	Condición de un episodio del Episodio del Escenario
Transformación	Incorporar la precondition de un episodio de un escenario como precondition del objetivo o meta derivado del mismo escenario.
SALIDA	Precondición del Objetivo

Ejemplo: el primer episodio del escenario *“Cancelación de la Reserva”* es
“if el recepcionista recibe el pedido de anulación de una solicitud de reserva o el pasajero/huésped/pax no se presenta en el período comprendido entre las 12 hs del día de ingreso establecido en la solicitud de reserva y las 06 hs. del día siguiente
then

el recepcionista elimina la solicitud de reserva de la planilla de reservas”.

En este caso, el episodio tiene una disyunción como precondition y para que se pueda cumplir el episodio la precondition debe ser verdadera. Para que la reserva sea cancelada, es claro entonces que debe existir un pedido expreso del pasajero, o el pasajero debe estar ausente en el lapso de tiempo establecido para ocupar una habitación, condicionando de este modo la cancelación.

Por lo tanto, *“el recepcionista recibe el pedido de anulación de una solicitud de reserva ó el pasajero/huésped/pax no se presenta en el período comprendido entre las 12 hs del día de ingreso establecido en la solicitud de reserva y las 06 hs. del día siguiente”,* precondition del episodio *“el recepcionista elimina la solicitud de reserva de la planilla de reservas”* dentro del escenario *“Cancelación de la Reserva”,* es una precondition del objetivo *“Reserva Cancelada”* derivado del escenario *“Cancelación de la Reserva”.*

5.2.4 Reglas aplicadas a las bifurcaciones de Escenarios

La **Regla 11** toma como entrada los casos alternativos de un escenario. Un caso alternativo menciona los casos de excepción, que pueden corresponder a otros escenarios. Usualmente, un caso alternativo refleja la falta o mal funcionamiento de un recurso necesario para el desarrollo del escenario [Leite97] [Leite00]. Esto provoca una dificultad para cumplir el objetivo del escenario, obstaculizando de este modo el desarrollo del escenario.

Un obstáculo en el enfoque de objetivos de [Anton96] [Anton97] [Anton98] es una circunstancia que impide que se cumpla un objetivo. Puede verse entonces que tanto el caso alternativo de un escenario como un obstáculo de un objetivo, alteran o impiden su cumplimiento. Es más, si se considera un objetivo derivado de un escenario, se puede establecer

que el caso alternativo del escenario impedirá la satisfacción del objetivo, convirtiéndose de ese modo en su obstáculo.

Por lo tanto, se incorpora la consecuencia ocasionada por la ocurrencia de un caso alternativo, o directamente el caso alternativo mismo (según el caso), como obstáculo del objetivo o meta derivado del escenario. La Tabla 5.12 muestra la regla.

Tabla 5.12 - Regla 11

Regla 11	
ENTRADA	Caso Alternativo del Escenario
Transformación	Incorporar la consecuencia ocasionada por la ocurrencia de un caso alternativo, o el caso alternativo mismo, como obstáculo del objetivo o meta derivado de ese escenario.
SALIDA	Obstáculo del Objetivo

Ejemplo: en el escenario *“Cancelación de la Reserva”* el caso alternativo es *“el teléfono, fax o e-mail no funcionan”*. Este mal funcionamiento puede impedir la cancelación de la reserva, o sea, obstaculizar el objetivo *“Reserva Cancelada”*.

Por lo tanto, se establece que *“el teléfono, fax o e-mail no funcionan”*, caso alternativo del escenario *“Cancelación de la Reserva”*, es un obstáculo del objetivo *“Reserva Cancelada”* derivado del escenario *“Cancelación de la Reserva”*.

La **Regla 12** también toma como entrada los casos alternativos. Como se mencionó en la regla anterior un caso alternativo menciona los casos de excepción que reflejan la falta o mal funcionamiento de un recurso necesario para el desarrollo del escenario.

En un objetivo de GBRAM, la restricción refleja una limitación bajo las cuales un objetivo debe cumplirse. Especifica algún requerimiento o

condición que debe ocurrir para lograr que un objetivo se complete. Esa limitación estará presente en todo el desarrollo del objetivo.

El caso alternativo de un escenario marca una falta que dificulta o impide su desarrollo. La negación del caso alternativo quita esa falta, convirtiéndose así en un soporte del escenario. La restricción de un objetivo exige la presencia de una circunstancia que permita completar el objetivo.

Para el caso de un objetivo derivado de un escenario, si la negación del caso alternativo es un soporte para el cumplimiento del escenario, también lo es para el objetivo, transformándose así en una restricción para que el objetivo se complete.

Por lo tanto, se incorpora la negación de cada caso alternativo de un escenario como una restricción del objetivo o meta derivado de ese escenario. La Tabla 5.13 muestra la regla.

Tabla 5.13 - Regla 12

Regla 12	
ENTRADA	Caso Alternativo del Escenario
Transformación	Incorporar la negación de cada caso alternativo de un escenario como una restricción del objetivo o meta derivado de ese escenario.
SALIDA	Restricción del Objetivo

Ejemplo: en el escenario *“Cancelación de la Reserva”* el caso alternativo es *“el teléfono, fax o e-mail no funcionan”*. Su negación es *“el teléfono, fax y e-mail funcionan”*. Esta es una restricción necesaria para poder completar el objetivo *“Reserva Cancelada”*, ya que los medios de comunicación deben funcionar para poder recibir un pedido de cancelación de reserva.

Por lo tanto, se establece que *“el teléfono, fax y e-mail funcionan”* negación del caso alternativo del escenario *“Cancelación de la Reserva”*, es

una restricción del objetivo “Reserva Cancelada” derivado del escenario “Cancelación de la Reserva”.

5.3 Resumen de reglas

La Tabla 5.14 resume las entradas y salidas de todas las reglas. Puede notarse que no existe una correspondencia biunívoca entre los componentes de escenarios y los componentes de objetivos, tal como se enunció previamente. No obstante, es importante remarcar que cada escenario permite la generación de un objetivo con formato GBRAM con todos sus elementos.

Tabla 5.14 - Resumen de Reglas

Nº Regla	Entrada	Salida
1	Nombre del Escenario	Nombre del Objetivo
2	Nombre del Escenario	Acción del Objetivo
3	Objetivo del Escenario	Descripción del Objetivo
4	Contexto del Escenario	Descripción del Objetivo
5	Contexto del Escenario	Precondición del Objetivo
6	Actor del Escenario	Agente del Objetivo
7	Actor del Escenario	Stakeholder del Objetivo
8	Episodio del Escenario	Sub-Meta del Objetivo
9	Ultimo Episodio del Escenario	Poscondición del Objetivo
10	Condición de un episodio del Escenario	Precondición del Objetivo
11	Caso Alternativo del Escenario	Obstáculo del Objetivo
12	Caso Alternativo del Escenario	Restricción del Objetivo

La Tabla 5.15 muestra la aplicación de las 12 reglas enunciadas al Escenario “Cancelación de la Reserva”, para obtener el Objetivo “Reserva Cancelada” con el formato establecido por GBRAM.

En la Tabla 5.16 se representan las relaciones que las reglas enunciadas establecen entre los componentes de los Escenarios (columna 1) con los de los Objetivos (fila 1). En la intersección de la fila y columna de la tabla se indica la regla mediante la que, aplicada a la componente del escenario, se obtiene la componente del objetivo.

En la mayoría de los casos un componente del Escenario es utilizado para obtener diferentes componentes de los Objetivos. Sólo en tres casos un componente del Objetivo necesita más de un componente del Escenario para ser generado.

Tabla 5.15 Ejemplo de aplicación de las Reglas

Nombre	Cancelación de la reserva.	Nombre de Meta	Reserva Cancelada.
Objetivo	Dar de baja una solicitud de reserva de un pasajero / huésped / pax.	Tipo	Achievement.
Contexto	Se realiza en la Recepción del Hotel. Existe una solicitud de reserva para un pasajero/huésped /pax.	Descripción	Dar de baja una solicitud de reserva de un pasajero / huésped / pax. Se realiza en la Recepción del Hotel.
Recursos	Planilla de reservas. Planilla de ocupación de habitaciones. Teléfono/Fax/E-mail.	Acción	Cancelar la Reserva.
Actores	Recepcionista / Agencia. Otro hotel. Pasajero/huésped/pax.	Agente	Recepcionista.
Set de episodios	If el recepcionista recibe el pedido de anulación de una solicitud de reserva o el pasajero/huésped/pax no se presenta en el período comprendido entre las 12 hs del día de ingreso establecido en la solicitud de reserva y las 06 hs. del día siguiente then El recepcionista elimina la solicitud de reserva de la planilla de reservas. El recepcionista actualiza la disponibilidad de habitaciones en la planilla de ocupación de habitaciones.	Stakeholders	Recepcionista / Agencia. Otro hotel. Pasajero / huésped / pax.
Casos alternativos	El teléfono, el fax o el e-mail no funcionan.	Restricciones	El teléfono, fax y e-mail funcionan.
		Obstáculos	el teléfono, fax o e-mail no funcionan.
		Precondiciones	El recepcionista recibe el pedido de anulación de una solicitud de reserva o el pasajero / huésped / pax no se presenta en el período comprendido entre las 12 hs del día de ingreso establecido en la solicitud de reserva y las 06 hs. del día siguiente.
		Poscondiciones	Las habitaciones están disponibles y la planilla de ocupación actualizada.
		Sub-Metas	"Solicitud de reserva eliminada por el recepcionista de la planilla de reservas". "Disponibilidad de habitaciones actualizada por el recepcionista en la planilla de ocupación de habitaciones".

Tabla 5.16. Mapa de Reglas para derivar Objetivos GBRAM

OBJETIVO	NOMBRE	DESCRIPCION	ACCION	AGENTES	STAKEHOLDERS	RESTRICCIONES	OBSTACULOS	PRECONDICIONES	POSCONDICIONES	SUBMETAS
ESCENARIO										
Nombre	R1		R2							
Objetivo		R3	R2							
Contexto		R4						R5		
Recursos										
Actores				R6	R7					
Set de episodios								R10	R9	R8
Casos alternativos						R12	R11			

5.4 Algoritmo para la aplicación de las Reglas

Las reglas definidas muestran una equivalencia semántica de dos modelos de representación de requerimientos, pero es necesario un procedimiento que especifique y ordene la aplicación de esas reglas.

El procedimiento es muy simple ya que a cada escenario del conjunto de escenarios disponibles, le aplica las 12 reglas definidas y obtiene un objetivo, el cual se debe clasificar y agregar a la lista de objetivos disponibles. De ese modo, a partir de un conjunto de escenarios se obtiene el conjunto de objetivos respectivo.

El propósito del procedimiento no es disponer de un código ejecutable por máquina, sino especificar el orden y forma de aplicación de las reglas. Por lo tanto, se plantea un algoritmo en pseudocódigo y no en un lenguaje de programación específico.

El procedimiento mencionado está representado de la siguiente forma:

```
cantidad de Escenarios disponibles o definidos -> (nCantEsc)
Para i := 1 a nCantEsc hacer
    Tomar Escenario i (Ei)
    Inicializar el Objetivo i a elicitar como vacío -> (Oi)
    Numerar los elementos que componen Oi de 1 a 10 (exceptuando el tipo)
    Inicializar set reglas NO aplicadas con todas las reglas -> (ReglasNOAp)
    Inicializar set de reglas aplicadas en vacío -> (ReglasAp)
    Para k:=1 a 10 hacer
        Tomar el elemento k de Oi -> (Oik)
        Repetir
            Tomar regla de ReglasNOAp que tenga como SALIDA a Oik -> (ReAplicar)
            Tomar la ENTRADA de ReAplicar -> (Entra)
            Buscar el contenido de Entra en Ei -> (Contenido)
            Aplicar la TRANSFORMACIÓN de RaAplicar a Contenido -> (Resultado)
            Asignar Resultado a Oik
            Quitar RaAplicar de ReglasNOAp
            Agregar RaAplicar a ReglasAp
        Hasta (ninguna regla de ReglasNOAp tiene como SALIDA a Oik) ó
        (ReglasNOAp=vacío)
    Fin Para
    Clasificar el tipo de objetivo elicitado como Achievement o Maintenance
    Incorporar Oi a la lista de Objetivos elicitados
Fin Para
```

5.5 Análisis del caso de estudio

En el anexo se presentan los escenarios y objetivos identificados para el problema de la Administración de la Recepción de un Hotel [Bertolami01]. Se presentan 10 escenarios y para cada uno de ellos el objetivo generado.

Utilizando el proceso definido en este capítulo, de cada escenario se generó un objetivo. Dado que todas las componentes del escenario son utilizadas para generar un objetivo, la semántica del escenario se preserva

en el objetivo, es decir, no hay pérdida de información. Por lo tanto, si el enfoque de escenarios permite reflejar completitud respecto al conocimiento adquirido de un dominio de problema dado, los objetivos generados también tienen ese conocimiento y muestran la misma completitud respecto a ese dominio.

Con el mismo fundamento, también pueden aplicarse a los objetivos generados desde escenarios, las ventajas del uso de escenarios enunciadas en el capítulo 2.

En la propuesta de [Leite97] [Leite00] un componente de un escenario puede ser vacío (Casos Alternativos). Esa misma situación ocurre con los componentes de objetivos [Anton96] [Anton97] [Anton98] pero para más de un componente.

Los escenarios del anexo sin casos alternativos, no permiten aplicar las reglas donde éstos son su entrada (Regla 11 y 12). Por lo tanto, los objetivos generados a partir de esos escenarios no incluyen restricciones ni obstáculos.

Por último, las sub-metas de objetivos originadas por episodios condicionales de escenarios, se podrían refinar y representar con el esquema de [Anton96] [Anton97] [Anton98], y tendrían como precondition en su esquema la condición del episodio del escenario.

Capítulo 6

Conclusiones y Trabajos Futuros

El resultado del trabajo realizado para esta tesis se puede resumir en las siguientes conclusiones:

- a) Se estableció la importancia de contar con un Proceso de Elicitación de Objetivos.
- b) Se identificó la carencia de tal Proceso, en los enfoques orientados a objetivos con un importante reconocimiento de la comunidad de Ingeniería de Requerimientos de Sistemas de Software.
- c) Se definió un Proceso de Elicitación de Objetivos con formato GBRAM [Anton96] [Anton97] [Anton98], a partir de Escenarios con el esquema de Julio Leite [Leite97] [Leite00].
- d) Se aplicó exitosamente el proceso en los casos "*Léxico extendido del lenguaje y Escenarios del sistema nacional para la obtención de pasaportes*" [Leite96] y "*Administración de la Recepción de un Hotel*" [Bertolami01]

No obstante, el cuestionamiento hasta quizás obvio, que podría plantearse es:

- ¿Se justifica la construcción de escenarios para poder elicitar objetivos?
- ¿No es suficiente disponer de los escenarios para comprender un problema?

La respuesta a estos interrogante tiene 3 aspectos esenciales:

- 1) Se ha definido un proceso de elicitación hasta ahora inexistente para una técnica de elicitación como lo es la técnica de objetivos, independientemente del esfuerzo que signifique su aplicación.
- 2) Si alguna metodología de desarrollo de software necesitara o se basara en objetivos con formato GBRAM, se ha contribuido con un proceso de elicitación de esos objetivos.
- 3) Aunque no sea una respuesta directa, se ha contribuido con la muestra de una analogía semántica entre los objetivos de GBRAM [Anton96] [Anton97] [Anton98] y los escenarios de Julio Leite [Leite97] [Leite00].

Respecto a la continuación de esta línea de investigación se mencionan las posibles trabajos a realizar:

- a) Herramientas
 - Extender las existentes para LEL y Escenarios.
 - Construir una nueva que genere objetivos GBRAM a partir de documentos de texto que representen escenarios.
- b) Extender el Proceso de Elicitación de Objetivos para que sea aplicable a otros enfoques orientados a objetivos.
- c) Desarrollar un Proceso de Elicitación de Objetivos a partir del LEL.

La Ingeniería de Requerimientos es un terreno fértil que aún nos da mucho para explorar...

Anexo

Se presenta en este anexo, el ejemplo completo producto de aplicar el proceso definido en el Capítulo 5, a los escenarios de *“Administración de la Recepción de un Hotel”* [Bertolami01].

En primer lugar, la tabla A.1 muestra la totalidad de escenarios y los objetivos o metas derivados de acuerdo a las reglas. En esta tabla solo se reproducen los nombres a modo de resumen. Posteriormente, se presentan en forma alternada las tablas con los escenarios junto a los objetivos o metas elicitados.

Es importante notar que algunos escenarios carecen de casos alternativos y en su contexto no incluyen una precondición. Debido a esta situación, los objetivos o metas elicitados desde esos escenarios no tienen obstáculos, restricciones, ni precondiciones (según el caso). Pero esto no contradice el esquema de Objetivos/Metas de GBRAM [Anton97], ya que este esquema contempla la posibilidad de componentes vacíos.

Tabla A.1 Escenarios y Objetivos

Escenario	Objetivo
Cancelación de la reserva	Reserva cancelada
Check in	Check in realizado
Check out	Check out realizado
Pago	Pago recibido
Pedido de alojamiento	Pedido de alojamiento atendido
Pedido de servicio extra	Pedido de servicio extra recibido
Recepción del hotel	Recepción del hotel atendido
Reposición de insumos	Insumos repuestos
Servicio de despertador	Servicio de despertador cumplido
Solicitud de reserva	Solicitud de reserva atendida

Escenario "Cancelación de la Reserva" → Objetivo "Reserva Cancelada"

Nombre	Cancelación de la Reserva.
Objetivo	Dar de baja una solicitud de reserva de un pasajero / huésped / pax.
Contexto	Se realiza en la Recepción del Hotel. Existe una solicitud de reserva para un pasajero / huésped / pax.
Recursos	Planilla de reservas. Planilla de ocupación de habitaciones. Teléfono. Fax. E-mail.
Actores	Recepcionista. Agencia. Otro hotel. Pasajero / huésped / pax.
Set de episodios	<ul style="list-style-type: none"> • if el recepcionista recibe el pedido de anulación de una solicitud de reserva o el pasajero / huésped / pax no se presenta en el período comprendido entre las 12 hs del día de ingreso establecido en la solicitud de reserva y las 06 hs. del día siguiente then el recepcionista elimina la solicitud de reserva de la planilla de reservas. • El recepcionista actualiza la disponibilidad de habitaciones en la planilla de ocupación de habitaciones.
Casos alternativos	El teléfono, el fax o el e-mail no funcionan.

Nombre de Meta	Reserva Cancelada.
Tipo	Achievement.
Descripción	Dar de baja una solicitud de reserva de un pasajero / huésped / pax. Se realiza en la Recepción del Hotel.
Acción	Cancelar la Reserva.
Agente	Recepcionista.
Stakeholders	Recepcionista. Agencia. Otro hotel.
Restricciones	El teléfono, fax y e-mail funcionan.
Obstáculos	el teléfono, fax o e-mail no funcionan.
Precondiciones	El recepcionista recibe el pedido de anulación de una solicitud de reserva ó el pasajero / huésped / pax no se presenta en el período comprendido entre las 12 hs del día de ingreso establecido en la solicitud de reserva y las 06 hs. del día siguiente.
Poscondiciones	Las habitaciones están disponibles y la planilla de ocupación actualizada.
Sub-Metas	<ul style="list-style-type: none"> • Solicitud de reserva eliminada por el recepcionista de la planilla de reservas. • Disponibilidad de habitaciones actualizada por el recepcionista en la planilla de ocupación de habitaciones.

Escenario "Check in" → Objetivo "Check in realizado"

Nombre	Check in.
Objetivo	Efectuar el ingreso del pasajero / huésped / pax al hotel.
Contexto	Se realiza en la Recepción del Hotel el día de ingreso que figura en la planilla de reservas. El voucher cumple con los requisitos.
Recursos	Voucher. Planilla del pasajero. Planilla de reservas. Planilla de ocupación de habitaciones. Fichero de pasajeros. Tarjeta codificada de la habitación. Llave del minibar / frigobar. Llave de la caja de seguridad. Control remoto TV.
Actores	Recepcionista. Pasajero / huésped / pax.
Set de episodios	<ul style="list-style-type: none"> • if el pasajero / huésped / pax presenta un voucher then el recepcionista registra los datos incluidos en el mismo y lo almacena en la carpeta de la habitación. • El recepcionista busca el nombre del pasajero / huésped / pax en la planilla de reservas. • if el pasajero / huésped / pax se presenta por primera vez al Hotel then el recepcionista le provee la planilla del pasajero que debe completar con sus datos y firmar. El recepcionista controla que la planilla del pasajero esté completa y la almacena en el fichero de pasajeros. • If el pasajero / huésped / pax ya estuvo en el Hotel then el recepcionista busca los datos personales por nombre en el fichero de pasajeros. • If el pasajero / huésped / pax tiene acompañantes then cada uno debe registrarse en forma individual. • El recepcionista asigna el número de habitación al pasajero / huésped / pax. • Al actualizar el recepcionista la disponibilidad de habitaciones, se actualiza la planilla de ocupación de habitaciones. • El recepcionista agrega el número de habitación asignado al pasajero / huésped / pax en la planilla de reservas. • El recepcionista hace una tarjeta codificada para el número de habitación. • El recepcionista le provee al pasajero / huésped / pax la tarjeta codificada de la habitación y el control remoto TV. • If el pasajero / huésped / pax lo solicita then el recepcionista le provee la llave del minibar / frigobar y/o de la caja de seguridad.
Casos alternativos	Faltan tarjetas magnéticas para codificar. La máquina generadora de tarjetas codificadas no funciona.

Nombre de Meta	Check In realizado.
Tipo	Achievement.
Descripción	Efectuar el ingreso del pasajero / huésped / pax al Hotel. Se realiza en la Recepción del Hotel el día de ingreso que figura en la planilla de reservas.
Acción	Realizar el check in.
Agente	Recepcionista.
Stakeholders	Recepcionista.
Restricciones	No Faltan tarjetas magnéticas para codificar. La máquina generadora de tarjetas codificadas funciona.
Obstáculos	Faltan tarjetas magnéticas para codificar. La máquina generadora de tarjetas codificadas no funciona.
Precondiciones	El voucher cumple con los requisitos.
Poscondiciones	Elementos de la habitación entregados al pasajero / huésped / pax.
Sub-Metas	<ul style="list-style-type: none"> • Datos del voucher registrados por el recepcionista y almacenados en la carpeta de la habitación. • Nombre del pasajero / huésped / pax buscado por el recepcionista en la planilla de reservas. • Planilla del pasajero completada y firmada. • Planilla del pasajero controlada que esté completa por el recepcionista y almacenada en el fichero de pasajeros. • Datos personales buscados por el recepcionista en el fichero de pasajeros. • Acompañantes registrados, cada uno en forma individual. • Número de habitación asignado por el recepcionista al pasajero / huésped / pax. • Disponibilidad de habitaciones, actualizada por el recepcionista. • Planilla de ocupación de habitaciones actualizada por el recepcionista. • Número de habitación asignado al pasajero / huésped / pax agregado por el recepcionista en la planilla de reservas. • Tarjeta codificada hecha por el recepcionista para el número de habitación. • El pasajero / huésped / pax es provisto de tarjeta codificada de la habitación y el control remoto TV, por parte del recepcionista. • Pasajero / huésped / pax provisto de la llave del minibar / frigobar y/o de la caja de seguridad, por el recepcionista.

Escenario "Check out" → Objetivo "Check out realizado"

Nombre	Check out.
Objetivo	Efectuar el egreso del pasajero / huésped / pax del Hotel.
Contexto	El pasajero / huésped / pax se presenta en la Recepción del Hotel el día de egreso acordado manifestando su deseo de retirarse. La Administración hizo la facturación.
Recursos	Factura. Tarjeta codificada de la habitación. Llave del minibar / frigobar. Llave de la caja de seguridad. Control Remoto TV. Sello de Pagado. Planilla de Ocupación de Habitaciones.
Actores	Recepcionista. Pasajero / huésped / pax.
Set de episodios	<ul style="list-style-type: none"> • El recepcionista entrega al pasajero / huésped / pax junto con la factura: comprobante de gastos de lavandería, comprobante de consumos de cafetería y/o restaurant, comprobante de consumos del minibar y comprobante de llamados telefónicos. • Realización de pago. • El recepcionista agrega el sello de Pagado en la factura. • El recepcionista envía la copia de la factura del pasajero / huésped / pax al Departamento Administración. • El recepcionista recibe la tarjeta codificada y el control remoto TV y la llave del minibar / frigobar y/o caja de seguridad si estuviera en poder del pasajero / huésped / pax. • Al actualizar el recepcionista la disponibilidad de habitaciones, se actualiza la planilla de ocupación de habitaciones.
Casos alternativos	El pasajero / huésped / pax no cuenta con los medios necesarios para pagar.

Nombre de Meta	Check out realizado.
Tipo	Achievement.
Descripción	Efectuar el egreso del pasajero / huésped / pax del Hotel. El pasajero / huésped / pax se presenta en la Recepción del Hotel el día de egreso acordado manifestando su deseo de retirarse.
Acción	Realizar Check out.
Agente	Recepcionista.
Stakeholders	Recepcionista.
Restricciones	El pasajero / huésped / pax cuenta con los medios necesarios para pagar.
Obstáculos	El pasajero / huésped / pax no cuenta con los medios necesarios para pagar.
Precondiciones	La Administración hizo la facturación.
Poscondiciones	Habitaciones disponibles y planilla de ocupación actualizada.
Sub-Metas	<ul style="list-style-type: none"> • Factura entregada por el recepcionista al pasajero / huésped / pax. • Comprobante de gastos de lavandería, comprobante de consumos de cafetería y/o restaurant, comprobante de consumos del minibar y comprobante de llamados telefónicos entregados por el recepcionista al pasajero / huésped / pax. • Pago realizado. • Sello de Pagado agregado en la factura por el recepcionista. • Copia de la factura del pasajero / huésped / pax enviada por el recepcionista al Departamento Administración. • Tarjeta codificada, control remoto TV y la llave del minibar / frigobar y/o caja de seguridad devuelta por el pasajero / huésped / pax. al Recepcionista. • Disponibilidad de habitaciones actualizada por el Recepcionista. • Planilla de ocupación de habitaciones actualizada por el Recepcionista.

Escenario "Pago" → Objetivo "Pago recibido"

Nombre	Pago.
Objetivo	Recibir el pago del pasajero / huésped / pax.
Contexto	Se realiza en la Recepción del Hotel durante el check out. La factura está preparada.
Recursos	Dinero en efectivo. Tarjeta de crédito. Cheque.
Actores	Recepcionista. Pasajero / huésped / pax.
Set de episodios	<ul style="list-style-type: none"> • If el pasajero / huésped / pax paga con dinero en efectivo o con cheque then el recepcionista recibe el dinero o el cheque y lo guarda en la caja de la Recepción. • If el pasajero / huésped / pax paga con tarjeta de crédito then el recepcionista solicita la autorización al emisor de la tarjeta de crédito. if el pasajero / huésped / pax está habilitado then el recepcionista hace el cupón con el importe correspondiente, el pasajero / huésped / pax lo firma, se guarda en la Recepción y se le provee la copia. • If el pasajero / huésped / pax pertenece a una empresa que tiene cuenta corporativa en el Hotel then firma la factura y se envía a la Administración. • El recepcionista coloca el sello de pagado al original de la factura. • El recepcionista entrega el original al pasajero / huésped / pax. • El recepcionista envía la copia de la factura a la Administración.
Casos alternativos	El pasajero / huésped / pax no cuenta con los medios necesarios para pagar.

Nombre de Meta	Pago recibido.
Tipo	Achievement.
Descripción	Recibir el pago del pasajero / huésped / pax. Se realiza en la Recepción del Hotel durante el check out.
Acción	Recibir pago.
Agente	Recepcionista.
Stakeholders	Recepcionista.
Restricciones	El pasajero / huésped / pax cuenta con los medios necesarios para pagar.
Obstáculos	El pasajero / huésped / pax no cuenta con los medios necesarios para pagar.
Precondiciones	La factura está preparada.
Poscondiciones	Trámite de pago cumplimentado.
Sub-Metas	<ul style="list-style-type: none"> • Dinero o cheque recibido por el recepcionista y guardado en la caja de la Recepción • Autorización al emisor de tarjeta de crédito solicitada por el recepcionista • Cupón de tarjeta realizado por recepcionista • Cupón de tarjeta firmado por pasajero / huésped / pax • Cupón de tarjeta firmado guardado por el recepcionista. • Copia del cupón de tarjeta firmado entregado al pasajero / huésped / pax. • Factura firmada por el pasajero / huésped / pax • Factura firmada enviada a la Administración. • Factura original sellada como pagada por el recepcionista. • Factura original entregada al pasajero / huésped / pax. • Copia de la factura enviada a la Administración.

Escenario "Pedido de Alojamiento" → Objetivo "Pedido de Alojamiento Atendido"

Nombre	Pedido de Alojamiento.
Objetivo	Atender un pedido de alojamiento realizada por una persona.
Contexto	Se realiza en la Recepción del Hotel. Una persona se presenta para un pedido de alojamiento.
Recursos	Planilla de ocupación de habitaciones. Planilla de reservas.
Actores	Recepcionista. Pasajero / huésped / pax.
Set de episodios	<ul style="list-style-type: none"> • El recepcionista consulta la planilla de ocupación de habitaciones para conocer la disponibilidad de habitaciones. • If existe disponibilidad de habitaciones y la persona estuviera de acuerdo then el recepcionista registra los datos personales del pasajero / huésped / pax en la planilla de reservas. • Check in.
Casos alternativos	

Nombre de Meta	Pedido de Alojamiento atendido.
Tipo	Achievement.
Descripción	Atender un pedido de alojamiento realizada por una persona. Se realiza en la Recepción del Hotel.
Acción	Atender pedido de Alojamiento.
Agente	Recepcionista.
Stakeholders	Recepcionista.
Restricciones	
Obstáculos	
Precondiciones	Una persona se presenta para un pedido de alojamiento.
Poscondiciones	Check in realizado.
Sub-Metas	<ul style="list-style-type: none"> • Información de disponibilidad de Habitaciones obtenida. • Datos personales del pasajero / huésped / pax registrados en la planilla de reservas por el recepcionista. • Check in realizado.

Escenario "Pedido de servicio extra" → Objetivo "Pedido de servicio extra atendido"

Nombre	Pedido de servicio extra.
Objetivo	Atender pedido de limpieza extra y pedido de mantenimiento extra.
Contexto	Se realiza en la Recepción del Hotel. Existe un pasajero / huésped / pax que solicita un servicio para su habitación.
Recursos	Lista de pedido de mantenimiento extra. Lista de pedido de limpieza extra.
Actores	Recepcionista. Pasajero / huésped / pax. Departamento Mantenimiento. Departamento Mucamas.
Set de episodios	<ul style="list-style-type: none"> • El recepcionista registra un pedido de limpieza extra o pedido de mantenimiento extra que hace el pasajero / huésped / pax para su habitación. • El recepcionista registra el número de habitación y el tipo de servicio a efectuar. • El recepcionista hace la lista de los pedido de limpieza extra o pedido de mantenimiento extra . • El recepcionista envía la lista al Departamento Mantenimiento o Mucamas.
Casos alternativos	

Nombre de Meta	Pedido de servicio extra atendido.
Tipo	Achievement.
Descripción	Atender pedido de limpieza extra y pedido de mantenimiento extra. Se realiza en la Recepción del Hotel.
Acción	Atender pedido de servicio extra.
Agente	Recepcionista.
Stakeholders	Recepcionista.
Restricciones	
Obstáculos	
Precondiciones	Existe un pasajero / huésped / pax que solicita un servicio para su habitación.
Poscondiciones	Departamento de Mantenimiento o de Mucamas notificado de un pedido de servicio extra.
Sub-Metas	<ul style="list-style-type: none"> • Pedido de limpieza extra o pedido de mantenimiento extra que hace el pasajero / huésped / pax para su habitación registrado por el recepcionista. • Número de habitación y el tipo de servicio a efectuar registrado por el recepcionista. • Lista de los pedido de limpieza extra o pedido de mantenimiento extra confeccionada por el recepcionista. • Lista de los pedido de limpieza extra o pedido de mantenimiento extra enviada al Departamento Mantenimiento o Mucamas.

Escenario "Recepción del hotel" → Objetivo "Recepción del hotel atendida"

Nombre	Recepción del hotel.
Objetivo	Atender las tareas referentes al movimiento de los pasajeros.
Contexto	Se realiza en la Recepción del Hotel. Funciona las 24 horas del día.
Recursos	Planilla de reservas. Planilla de ocupación de habitaciones. Fichero de pasajeros.
Actores	Recepcionista. Pasajero / huésped / pax.
Set de episodios	<ul style="list-style-type: none"> • Solicitud de reserva o pedido de alojamiento. • Check in. • Si el pasajero / huésped / pax lo solicita entonces pedido de servicio extra. • Si el pasajero / huésped / pax lo solicita entonces servicio de despertador. • La carpeta de la habitación se envía a la Administración junto con el día de ingreso y el día de egreso , para la emisión de la facturación. • Check out.
Casos alternativos	

Nombre de Meta	Recepción del hotel atendida.
Tipo	Achievement.
Descripción	Atender las tareas referentes al movimiento de los pasajeros. Se realiza en la Recepción del Hotel. Funciona las 24 horas del día.
Acción	Atender la recepción del Hotel.
Agente	Recepcionista.
Stakeholders	Recepcionista.
Restricciones	
Obstáculos	
Precondiciones	
Poscondiciones	Pasajero / huésped / pax atendido.
Sub-Metas	<ul style="list-style-type: none"> • Solicitud de reserva o pedido de alojamiento recibido. • Check in realizado. • Pedido de servicio extra atendido. • Servicio de despertador atendido. • Carpeta de la habitación enviada a la Administración junto con el día de ingreso y el día de egreso , para la emisión de la facturación. • Check out realizado.

Escenario "Reposición de insumos" → Objetivo "Insumos repuestos"

Nombre	Reposición de insumos.
Objetivo	Reponer los elementos faltantes para el funcionamiento de la Recepción.
Contexto	Se realiza en la Recepción del Hotel. El recepcionista tiene registradas las necesidades de insumos.
Recursos	Lista de pedidos de insumos. Insumos. Formularios de recibido.
Actores	Recepcionista.
Set de episodios	<ul style="list-style-type: none"> • El recepcionista hace una lista de insumos faltantes. • El recepcionista envía el pedido al Departamento compras. • El recepcionista recibe el pedido que envía el Departamento compras. • El recepcionista firma el conforme en el recibo.
Casos alternativos	

Nombre de Meta	Insumos repuestos.
Tipo	Achievement.
Descripción	Reponer los elementos faltantes para el funcionamiento de la Recepción. Se realiza en la Recepción del Hotel.
Acción	Reponer insumos.
Agente	Recepcionista.
Stakeholders	Recepcionista.
Restricciones	
Obstáculos	
Precondiciones	El recepcionista tiene registradas las necesidades de insumos.
Poscondiciones	Pedido de insumos realizado.
Sub-Metas	<ul style="list-style-type: none"> • Lista de insumos faltantes confeccionada por el recepcionista. • Pedido de insumos enviado al Departamento compras. • Pedido enviado desde el Departamento compras al recepcionista. • Recibo de conformidad de pedido de insumos firmado por el recepcionista.

Escenario "Servicio de despertador" → Objetivo "Servicio de despertador registrado y cumplido"

Nombre	Servicio de despertador.
Objetivo	Registrar el pedido del servicio de despertador y controlar que se cumpla.
Contexto	Existe un pasajero / huésped / pax que requiere ser despertado a una hora determinada cuyo pedido se registra en la Recepción.
Recursos	Teléfono. Reloj. Lista de pedido del servicio de despertador.
Actores	Recepcionista. Pasajero / huésped / pax.
Set de episodios	<ul style="list-style-type: none"> • El recepcionista registra el número de habitación y la hora solicitada por el pasajero / huésped / pax. • El recepcionista llama por teléfono al pasajero / huésped / pax cuando llega la hora.
Casos alternativos	

Nombre de Meta	Servicio de despertador cumplido.
Tipo	Achievement.
Descripción	Registrar el pedido del servicio de despertador y controlar que se cumpla.
Acción	Registrar y controlar el cumplimiento del servicio de despertador.
Agente	Recepcionista.
Stakeholders	Recepcionista.
Restricciones	
Obstáculos	
Precondiciones	Existe un pasajero / huésped / pax que requiere ser despertado a una hora determinada cuyo pedido se registró en la Recepción.
Poscondiciones	Pasajero / huésped / pax despertado.
Sub-Metas	<ul style="list-style-type: none"> • Número de habitación y la hora solicitada por el pasajero / huésped / pax. registrada por el recepcionista. • Pasajero / huésped / pax despertado por el recepcionista cuando llegó la hora.

Escenario "Solicitud de reserva" → Objetivo "Solicitud de reserva atendida"

Nombre	Solicitud de reserva.
Objetivo	Atender una solicitud de reserva realizada por una persona, agencia u otro hotel.
Contexto	Se realiza en la Recepción del Hotel. Existe una solicitud de reserva.
Recursos	Lista de precios. Planilla de reservas. Planilla de ocupación de habitaciones. Teléfono. Fax. E-Mail.
Actores	Recepcionista. Agencia. Otro hotel. Pasajero / huésped / pax.
Set de episodios	<ul style="list-style-type: none"> • El recepcionista verifica en la planilla de ocupación de habitaciones la disponibilidad de habitaciones comprendida entre el día de ingreso y el día de egreso solicitados, y si hubiera, informa la tarifa y solicita la aprobación de la persona, agencia o de otro hotel. • El recepcionista registra el nombre del pasajero / huésped / pax, cantidad de pasajeros, tipo de habitación, día de ingreso, día de egreso y tarifa en la planilla de reservas. • El recepcionista actualiza la disponibilidad de habitaciones en la planilla de ocupación de habitaciones.
Casos alternativos	El teléfono, el fax o el e-mail no funcionan.

Nombre de Meta	Solicitud de reserva atendida.
Tipo	Achievement.
Descripción	Atender una solicitud de reserva realizada por una persona, agencia u otro hotel. Se realiza en la Recepción del Hotel.
Acción	Atender una solicitud de reserva.
Agente	Recepcionista
Stakeholders	Recepcionista, Agencia.
Restricciones	El teléfono, el fax y el e-mail funcionan.
Obstáculos	El teléfono, el fax o el e-mail no funcionan.
Precondiciones	Existe una solicitud de reserva.
Poscondiciones	Disponibilidad de habitaciones actualizada.
Sub-Metas	<ul style="list-style-type: none"> • Planilla de ocupación de habitaciones consultada por el recepcionista verificando la disponibilidad de habitaciones comprendida entre el día de ingreso y el día de egreso. • Tarifa informada a la persona, agencia o u otro hotel. • Nombre del pasajero / huésped / pax, cantidad de pasajeros, tipo de habitación, día de ingreso, día de egreso y tarifa en la planilla de reservas registrada por el recepcionista. • Disponibilidad de habitaciones actualizada en la planilla de ocupación de habitaciones.

Bibliografía

- [Anton96] Antón A., **Goal-Based Requirements Analysis**, IEEE Software, 1996.
- [Anton97] Antón A., **Goal Identification and Refinement in the Specification of Software-Based Information Systems**, Ph.D. Thesis, Georgia Institute of Technology, 1997.
- [Anton98] Antón Annie I., **The Use of Goals to surface Requirements for Evolving System**, IEEE Software, 1998.
- [Bertolami01] Bertolami Mabel, Centeno M. Elena, **LEL y Escenarios de la Administración de Recepción del Hotel**. Caso de estudio desarrollado en el marco del Magister en Ingeniería del Software, 2001, UNLP.
- [Davis87] Davis G.B., Olson M.H., **Sistemas de Información Gerencial**, McGrawHill, Bogotá, 1987.
- [Davis93] Davis Alan M., **Software Requirements, Objetos, Funciones y Estados**, Editorial Prentice Hall, ISBN 0-13-805763-X, 1993.
- [Hadad99] Hadad Graciela, et. al. , **Integración de Escenarios con el Léxico Extendido del Lenguaje en la Elicitación de Requerimientos: aplicación a un caso real**, Revista de Informática Teórica e Aplicada - RITA, Vol 6, Número1, Julio 1999, Porto Alegre, Brasil.

- [Humphrey89] Humphrey W., **Managing the Software Process**, Addison-Wesley, 1989.
- [Jarke98] Jarke Matthias, Kurki-Suonio Reino, **Special issue on scenario management**, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 24, NO. 12, DECEMBER 1998.
- [Kaiva02] Kaiya H., Horai H. , and Saeki M., **AGORA: Attributed Goal-Oriented Requirements Analysis Method**, Proc. of the 10th IEEE International Requirements Engineering Conference (RE'02), pages 13–22, 2002.
- [Kavakli96] Kavakli E., Loucopoulos P., Filippidou D., **Using Scenarios to Systematically Support Goal-Directed Elaboration for Information System Requirements**, IEEE Symposium and Workshop on Engineering of Computer Based Systems (ECBS'96), 1996, GERMANY.
- [Kavakli99] Kavakli E., **Goal-Driven Requirements Engineering:Modelling and Guidance**, A thesis submitted to the University of Manchester Institute of Science and Technology for the degree of Doctor of Philosophy, 1999.
- [Kendall91] Kendall K.E., Kendall J.E., **Análisis y Diseño de Sistemas**, Prentice-Hall, México, 1991.
- [Kim03] Kim M., Yang H., et. al: **A Domain Analysis Method for Software Product Lines Based on Scenarios, Goal and Features**, Proceedings of the Tenth Asia Pacific Software Engineering Conference (ASPEC'03). 2003

- [Lamsweerde93]**Lamsweerde Axel van, et. al., **Goal-Directed Requirements Acquisition**, Science of Computer Programming, Vol. 20, pp. 3-50, 1993.
- [Lamsweerde96]**Lamsweerde Axel van, Darimont R., **Formal Refinement Patterns for Goal-Driven Requirements Elaboration**, SIGSOFT, 1996.
- [Lamsweerde00]**Lamsweerde Axel van, **Requirements Engineering in the Year 00: A Research Perspective**, 22nd International Conference on Software Engineering, 2000.
- [Leite96]** Leite, J.C.S.P., et. al., **Léxico extendido del lenguaje y escenarios del sistema nacional para la obtención de pasaportes**. Documento de trabajo, 1996, Universidad de Belgrano, Buenos Aires.
- [Leite97]** Leite, J.C.S.P., et. al., **Enhancing a Requirements Baseline with Scenarios**. Proceedings of RE 97', IEEE Third International Requirements Engineering Symposium, IEEE Computer Society Press, 1997, pp 44-53
- [Leite00]** Leite, J.C.S.P., et. al., **A Scenario Construction Process**, Requirements Engineering Journal Vol.5, N° 1, 2000, pp. 38-61.
- [Liu01]** Liu L, Yu E., **From Requirements to Architectural Design – Using Goals and Scenarios**, Proceedings first STRAW workshop, 2001.

- [Loucopoulos95]** Loucopoulos P. Karakostas V., **System Requirements Engineering**, McGraw-Hill International series in Software Engineering, ISBN 0-07-707843-8, 1995.
- [Mylopoulos94]** Mylopoulos J., , Yu E., **Understanding “Why” in Software Process Modelling, Analysis, and Design**, Proceedings of the 16th International Conference Software Engineering, May 16-21, 1994, Sorrento, Italy.
- [Mylopoulos01]** Mylopoulos J., Chung L., Liao S., Wang H., Yu E., **Exploring Alternatives During Requirements Analysis**, IEEE Software, 2001.
- [Nuseibeh00]** Nuseibeh B., Easterbrook S., **Requirements Engineering: A Roadmap**, ICSE2000, 2000, Limerick, Irlanda.
- [Oshiro03]** Oshiro K., Watahiki K., Saeki M., **Goal-Oriented Idea Generation Method for Requirements Elicitation**, 11th IEEE International Requirements Engineering Conference, 2003, pp. 363-364.
- [Pfleeger02]** Pfleeger Shari Lawrence, **Ingeniería de Software, Teoría y Práctica**. Editorial Prentice Hall. Primera edición. ISBN 987-9460-71-5. Págs. 52-54.
- [Pressman98]** Pressman Roger, **Ingeniería del Software, Un Enfoque Práctico**. Editorial McGraw-Hill. Cuarta Edición. ISBN 84-481-1186-9, Págs. 40.
- [Regev01]** Regev Gil, Wegmann Alain, **Goals, Interpretations, and Policies in Information Systems Design**, Technical Report N° DSC/2001/043, 2001.

- [Ridao01]** Ridao Marcela, **Uso de Patrones en el Proceso de Construcción De Escenarios**, Trabajo de Tesis para la Maestría de Ingeniería de Software, UNLP, 2001.
- [Rolland98]** Rolland Colette, Souveyet C., Ben Achour C., **Guiding Goal Modelling Using Scenarios**, IEEE Transactions on Software Engineering, special issue on Scenario Management, 1998.
- [Rolland99]** Rolland C., Grosz G., Kla R., **Experience With Goal-Scenario Coupling In Requirements Engineering**, IEEE International Symposium on Requirements Engineering, Ireland, 1999.
- [Robertson02]** Robertson Suzanne, **Project Sociology: Identifying and involving the stakeholders**, ICFAI University Press, 2002.
- [Thayer90]** Institute of Electrical and Electronic Engineers, **IEEE Guide to Software Requirements Specifications. ANSI/IEEE Standard 830**. En Thayer, R. y Dorfman, M., *System and Software Requirements Engineering*, IEEE Computer Society Press, Los Alamitos, 1990.
- [Thomas03]** Thomas P., Oliveros A., **Elicitación de Objetivos, un estudio comparativo**, Congreso Argentino de Ciencias de la Computación CACIC2003, UNLP, 2003.
- [Thomas04]** Thomas P., Oliveros A., **Proceso de Elicitación de Objetivos**, Congreso Argentino de Ciencias de la Computación CACIC2004, Universidad Nacional de La Matanza, 2004.

[Thomas05] Thomas P., Oliveros A., **Elicitación de Objetivos a partir de Escenarios**, VIII Workshop on Requirements Engineering WER'05, Porto, Portugal, Junio 2005.

[Uchitel04] Uchitel S., et al.: **Fluent-Based Animation: Exploiting the Relation between Goals and Scenarios for Requirements Validation**, 12th International Requirements Engineering Conference, Kyoto, Japón, Septiembre 2004.