

MAESTRÍA
EN
INGENIERÍA DE SOFTWARE

TESIS

TEMA: USO DE PATRONES EN EL PROCESO DE
CONSTRUCCIÓN DE ESCENARIOS

ALUMNO: ING. MARCELA RIDAO

DIRECTOR: ING. JORGE H. DOORN

CODIRECTOR: DR. JULIO CESAR SAMPAIO DO PRADO LEITE

ÍNDICE

ÍNDICE	I
ESTADO DEL ARTE	1
1.1 INGENIERÍA DE SOFTWARE	1
1.2 INGENIERÍA DE REQUISITOS.....	3
BASELINE DE REQUISITOS	8
2.1 COMPONENTES DE LA BASELINE DE REQUISITOS.....	9
2.2 MODELO DE OBJETOS.....	9
LEL Y ESCENARIOS	12
3.1 USO DEL LENGUAJE NATURAL	12
3.2 SÍMBOLOS DEL LEL.....	13
3.3 PROCESO DE CONSTRUCCIÓN	15
3.4 ESCENARIOS	18
3.5 COMPONENTES DE UN ESCENARIO	19
3.6 CONSTRUCCIÓN DE ESCENARIOS	23
3.6.1 <i>Derivar</i>	25
Identificar actores	25
Identificar escenarios	25
Crear escenarios	26
3.6.2 <i>Describir</i>	29
Completar componentes.....	30
Crear subescenarios	30
Completar restricciones	31
Completar excepciones	31
3.6.3 <i>Organizar</i>	31
Reorganizar	32
Definir relaciones entre escenarios	34
Integrar	35
Verificar	35
Validar	37
ASPECTOS RECURRENTES EN LA CONSTRUCCIÓN DE ESCENARIOS	38
4.1 REGULARIDADES	38
4.2 CLASIFICACIÓN DE EPISODIOS.....	39
4.3 CLASIFICACIÓN DE SUBESCENARIOS	41
4.4 CLASIFICACIÓN DE ESCENARIOS	43
4.5 ESTADÍSTICAS.....	45
PATRONES DE ESCENARIOS	46
5.1 PATRONES	46
5.2 PATRONES DE ESCENARIOS	47
5.3 CATÁLOGO	48
5.4 EJEMPLOS	48
5.5 MECANISMO DE DERIVACIÓN DE ESCENARIOS USANDO PATRONES	51
5.6 EJEMPLO.....	53
INTEGRACIÓN DE LOS PATRONES AL PROCESO DE CONSTRUCCIÓN	58

6.1 DESCRIPCIÓN DEL PROCESO DE CONSTRUCCIÓN CON PATRONES	58
6.2 APLICACIÓN A UN CASO DE ESTUDIO	61
6.2.1 <i>Ejemplo de derivación sin patrones</i>	61
6.2.2 <i>Ejemplo de derivación con patrones</i>	63
6.2.3 <i>Comparación entre ambos procedimientos de derivación</i>	65
CONCLUSIONES Y TRABAJO FUTURO	68
REFERENCIAS	70
ANEXO A: CASOS DE ESTUDIO	
ANEXO B: CATALOGO DE PATRONES	

CAPÍTULO 1

ESTADO DEL ARTE

1.1 INGENIERÍA DE SOFTWARE

El mundo moderno depende cada día más de los sistemas de software. Los sistemas computacionales están en todas partes, desde los sistemas de freno de los autos, hasta el control de los hornos de microondas. Hoy en día, las grandes corporaciones conocen la importancia estratégica de sus políticas en relación con las tecnologías de información y el papel fundamental del software como parte de esas tecnologías [Leite97a].

En los 40's y 50's, ya era común el hardware de propósito general. El software, sin embargo, era diseñado a medida para cada aplicación y tenía una distribución relativamente limitada. El software como producto, es decir, programas desarrollados para ser vendidos a uno o más clientes estaba en su infancia. La mayoría del software era desarrollado y finalmente usado por la misma persona u organización. El diseño era un proceso implícito llevado a cabo en la cabeza de alguien y la documentación era en general inexistente [Pressman97a]. Un programa era considerado exitoso si: a) ejecutaba, b) ejecutaba rápidamente, c) daba una respuesta aceptable, y la calidad dependía altamente de la habilidad del programador.

Hacia fines de los 60 comenzaron a producirse problemas recurrentes en el desarrollo de sistemas que se conocen como la “crisis del software”, caracterizada por problemas en el desarrollo de software, que provocaban demoras en los plazos de entrega, altos presupuestos, falta de respuesta a los requerimientos de los clientes y/o usuarios, y dificultad para usar, mantener y mejorar los sistemas [Dorfman97a]. A medida que los sistemas se hacían más grandes, su calidad era sospechosa. La gente responsable del desarrollo de software tenía limitados datos históricos para usar como guías y menos control sobre el curso de un proyecto. En 1969, un conjunto de técnicas, llamadas Ingeniería de Software, surgió como respuesta a esa crisis. Estas técnicas ven al software como un producto de ingeniería que requiere planeamiento, análisis, diseño, implementación, testeo y mantenimiento [Pressman97a].

Según Nauer y Randall [Nauer69] el término Ingeniería de Software significa “el establecimiento y uso de principios de ingeniería para obtener en forma económica, software confiable y que trabaje eficientemente en máquinas reales”.

Luego, durante casi veinte años, hubo grandes debates sobre si la creación de software era un arte, una ciencia, o disciplina [Hoare84]. Finalmente, el término “Ingeniería de Software” ha tenido cierto nivel de aceptación dentro de la comunidad de software. Sin embargo, lo que significa exactamente, sigue algo confuso.

Guezzi et al. en [Guezzi91] definen la Ingeniería de Software como “el campo de la

ciencia de la computación que trata la construcción de sistemas de software que son tan grandes o tan complejos que tienen que ser construidos por un equipo o equipos de ingenieros”.

Según Alan Davis en [Thayer97], Ingeniería de Software es “la aplicación de principios científicos a: 1) la transformación ordenada de un problema en una solución de software, y 2) el mantenimiento subsiguiente de ese software hasta el final de su vida útil”.

Pressman establece los objetivos de la ingeniería de software: “Los objetivos claves de la Ingeniería de Software son definir, crear y aplicar 1) una metodología bien definida dirigida a un ciclo de vida de planeamiento, desarrollo, y mantenimiento; 2) un conjunto establecido de componentes de software que documenta cada paso en el ciclo de vida y muestra un seguimiento paso a paso, y 3) un conjunto de hitos predecibles que pueden ser revisados a intervalos regulares a través del ciclo de vida del software”.

Ha habido grandes avances en la tecnología de Ingeniería de Software en todas las áreas: análisis de requisitos, estrategias de implementación, modelos de costos, etc. Sin embargo, la Ingeniería de Software aún está por debajo de las necesidades de calidad demandadas por sistemas cada vez más complejos. En este contexto, existen considerables esfuerzos de investigación y desarrollo con el objetivo de perfeccionar el proceso de producción de software, tanto a través de estudios teóricos, como de estudios aplicados.

Cuando la Ingeniería de Software ingresa en su cuarta década, tiene muchas de las fuerzas y fragilidades que son experimentadas por los seres humanos de la misma edad. La inocencia y el entusiasmo de sus primeros años han sido reemplazados por expectativas más razonables (y aún por un saludable cinismo) basados en años de experiencia. La Ingeniería de Software llega a su edad adulta con muchos logros, pero con trabajo aún por hacer [Pressman97b].

El Dr. Windston Royce enfatizó esta situación en [Royce91]: “La construcción de nuevo software a gusto del usuario/cliente y sin errores es un problema inesperadamente difícil. Es tal vez el problema más difícil de la ingeniería hoy en día, y ha sido reconocido como tal durante más de 15 años. La “crisis del software”, se ha convertido en la crisis más larga en el mundo de la ingeniería, y aún continúa.

A pesar de la creciente participación del software en el mundo actual, y de los avances producidos, el software como producto aún utiliza procesos de producción bastante insatisfactorios. Son varios los casos en que errores en el software han traído problemas no sólo financieros, sino también en vidas humanas. Es función de la Ingeniería de Software evitar que estos errores ocurran, produciendo productos de software más robustos y proveyendo procesos de producción más confiables.

Durante mucho tiempo, la literatura y la investigación en Ingeniería de Software centraban sus esfuerzos en el modelaje, procurando desarrollar lenguajes y técnicas de representación. Por ello, se prestaba muy poca atención a la tarea de descubrir lo que el sistema tenía que hacer, puesto que lo más importante era la representación del sistema.

El punto de partida del proceso de producción de software es el momento en que se define lo que se quiere. Por lo tanto, el desarrollo del software sólo puede ser iniciado cuando se tiene bien establecido lo que se quiere producir. Cuando se trata de sistemas complejos, esta definición de lo que se quiere no es trivial. Por ello, muchos productos de software no se comportan como sería deseable. Hacer una definición que cubra todas las

necesidades de un sistema complejo es una tarea difícil. Y es más difícil aún si no se cuenta con métodos, técnicas y herramientas adecuadas.

En este ámbito se sitúa la Ingeniería de Requisitos, como un área de investigación que procura atacar un punto fundamental en el proceso de producción, que es la definición de lo que se quiere producir. Cabe a la Ingeniería de Requisitos, como subárea de la Ingeniería de Software, proponer métodos, técnicas y herramientas que faciliten el trabajo de definición de lo que se quiere de un software. Por lo tanto, la Ingeniería de Requisitos tiene una interacción muy fuerte con aquéllos que demandan un producto de software, sean éstos el mercado o los clientes de una aplicación que será especialmente construida.

1.2 INGENIERÍA DE REQUISITOS

Hubo un tiempo en que la frase “los requisitos dicen lo que el sistema hará y no cómo lo hará” resumía todo lo referido a Ingeniería de Requisitos. Hoy, mucho tiempo después, la investigación en Ingeniería de Requisitos ha producido un gran volumen de conocimiento que incluye terminología, métodos, lenguajes y herramientas [Zave97].

El problema de la Ingeniería de Requisitos es determinar los ‘requisitos’ para sistemas basados en computadoras. Hay actualmente un debate acerca de la definición y el ámbito de los requisitos. Por ello, no es posible dar una única definición de esta disciplina.

Jackson afirma que la Ingeniería de Requisitos se ubica en el punto de encuentro entre lo informal y lo formal del desarrollo de software. Los programas computacionales son efectivamente construcciones formales, que utilizan métodos matemáticos basados en teoría de tipos, precondiciones y postcondiciones, e invariantes. Pero el mundo de los seres humanos y objetos físicos en el cual los requisitos se ubican es informal, y no puede ser tratado adecuadamente por métodos puramente formales [Jackson01].

Davis sugiere que la Ingeniería de Requisitos consiste en el análisis, documentación y posterior evolución de las necesidades del usuario y el comportamiento externo del sistema a ser construido [Davis90]. Para Goguen, los requisitos son propiedades que un sistema debería tener para tener éxito en el ambiente en el cual será usado [Jirotko94].

En [Thayer97] se define Ingeniería de Requisitos como la ciencia y disciplina interesada en establecer y documentar los requisitos de software. Consiste en la elicitación, el análisis, la especificación, la verificación y la administración de los requisitos.

Según Kotonya et al. el proceso de Ingeniería de Requisitos involucra un claro entendimiento de los requisitos del sistema deseado. Esto incluye los servicios requeridos por el sistema, los usuarios del sistema, su entorno y las restricciones asociadas. Este proceso involucra la captura, análisis y resolución de muchas ideas, perspectivas y relaciones a distintos niveles de detalle [Kotonya96].

La falla en el desarrollo y documentación de buenas especificaciones de requisitos es una de las principales causas de errores en el desarrollo de software. Actualmente, la dificultad para escribir especificaciones de requisitos buenas, correctas, completas y medibles es uno de los problemas más grandes en la producción de software. Algunas situaciones que originan esa dificultad son:

- La inhabilidad de los ingenieros para escribir una especificación de requisitos

correcta

- El deseo de los gerentes de truncar la actividad de requisitos porque creen que el mayor esfuerzo de cualquier desarrollo de software está en la programación y el testeo
- La falta de cooperación de los clientes en la verificación de los requisitos
- Los problemas asociados con la identificación de qué herramienta y/o metodología usar
- La tendencia en grandes corporaciones de asignar personal con poca experiencia o conocimiento en software a las tareas relacionadas con la especificación de requisitos

Cuando se detectó la “crisis del Software” en los 60, se dirigieron muchos esfuerzos a encontrar las causas de los problemas que llevaron a esa crisis. Las investigaciones determinaron que las deficiencias en los requisitos constituían una de las principales fuentes de problema. En [Dorfman97b] se presentan algunos de los beneficios que se obtienen a partir de una buena definición de requisitos:

- Acuerdo entre desarrolladores, clientes y usuarios en el trabajo que tiene que hacerse y los criterios de aceptación para el sistema a producir
- Una base sólida para la estimación de recursos (costo, cantidad y habilidades del personal, equipo y tiempo)
- Mejoras en la usabilidad, mantenimiento, y otros atributos de calidad del sistema
- Logro de los objetivos con mínimos recursos

Otro de los beneficios de una buena ingeniería de requisitos es la prevención de backtracking en etapas tardías del proceso de desarrollo. Averiguar correctamente las necesidades de los clientes, disminuye costosas tareas de reingeniería después de la codificación [Kovitz01].

Es importante destacar, además, que el valor de los buenos requisitos, así como la importancia de hacerlos bien, crece dramáticamente con el tamaño y la complejidad del sistema en desarrollo.

Una vez establecido el valor de una buena especificación de requisitos, surgen otras cuestiones: ¿Cómo desarrollarlos? ¿Cómo saber si un conjunto de requisitos es bueno?. ¿Qué estándares, herramientas y métodos pueden ayudar? ¿Estas herramientas existen o deben ser desarrolladas? Estas preguntas no tienen una respuesta definitiva, puesto que continuamente surgen nuevas herramientas y métodos. Pero lo importante es entender que la Ingeniería de Requisitos debe intentar, antes que nada, elegir el o los modelos más apropiados para el caso en cuestión. Dependiendo de la elección del modelo, van a variar los métodos, las técnicas y herramientas utilizadas durante la definición de requisitos.

En resumen, la Ingeniería de Requisitos establece el proceso de Definición de Requisitos como un proceso en el cual se elicit, modela y analiza lo que debe hacerse. Este proceso debe lidiar con diferentes puntos de vista, y usar una combinación de métodos, herramientas y personal. El producto de este proceso es un modelo, que produce un documento llamado Requisitos. Este proceso sucede en un contexto previamente definido que llamamos el Universo de Discurso, en el cual el software deberá ser desarrollado y

operado. 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 [Leite97a].

Debido a que uno de los objetivos de esta etapa del proceso de desarrollo de software es aumentar el conocimiento del dominio del problema, la comunidad de Ingeniería de Software ha desarrollado diversas estrategias para elicitar y especificar los fenómenos propios de cada Universo de Discurso. Algunos autores proponen la utilización de aproximaciones basadas en el lenguaje natural; otros se inclinan por los lenguajes artificiales y las representaciones. Unos pocos recomiendan la construcción de un vocabulario que capture la jerga usada por los expertos del dominio [Arango93] [Leite90]. La mayoría de ellos ([Anton01] [Benner93], [Carroll95], [Gough95], [Jacobson92], [Potts94], [Potts95], [Rubin92] [van Lamsweerde01] y [Zorman95]) adhiere al uso de escenarios o casos de uso para describir el comportamiento del macrosistema.

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 [Allenby01].

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 requisitos. Los ingenieros de requisitos entenderán, modelarán y analizarán el dominio de la aplicación donde el software se utilizará y los clientes/usuarios validarán si la visión de los ingenieros es correcta o no [Hadad99].

Los escenarios pueden ser un medio de lograr este objetivo, puesto que proveen un atractivo medio de comunicación entre los *stakeholders* del Universo de Discurso. Y es en este punto donde los escenarios se vuelven importantes, puesto que pueden mantener mucha información en una forma que dichos *stakeholders* podrían reconocer.

Existe una variedad de interpretaciones, sintaxis y mecanismos de construcción para escenarios, tales como: narrativas, videos, guiones, prototipos escritos y otros [Carroll95] [Booch91] [Jacobson92] [Zorman95] [Potts94]. Sin embargo, no todos los enfoques coinciden. Existen contradicciones básicas entre ellos. Por ejemplo, no hay un acuerdo con respecto a si los escenarios deberían ser construidos utilizando un método top-down o bottom-up [Hadad99].

Aún cuando no lo establece claramente, Booch parece adherir a un enfoque top-down: “La aplicación más compleja puede ser caracterizada en términos de unas pocas docenas de escenarios primarios. Los escenarios primarios modelan el problema central. Un escenario primario representa alguna función fundamental del sistema. Los escenarios secundarios representan alguna variación respecto del tema de un escenario primario. El comportamiento completo de un sistema de software puede ser capturado a través de una red de escenarios, de un modo similar al uso de un guión para hacer una película” [Booch94].

Firesmith ve a los escenarios primarios de Booch como escenarios relevantes, y los vincula a través de un diagrama de ciclo de vida de escenarios: “Un diagrama de ciclo de

vida de escenarios es usado para documentar las interacciones válidas entre los escenarios del nivel más alto de un sistema.” y “Debido a que los escenarios son abstracciones funcionales, a menudo hay una fuerte tendencia a descomponerlos funcionalmente” [Firesmith94].

La propuesta de Sutcliffe también puede ser vista como un enfoque top-down teniendo en cuenta las heurísticas: “1. Los requisitos iniciales capturan y dominan la familiarización. A partir del análisis preliminar del dominio, se desarrolla un escenario básico, 2. Especificación y desarrollo del demostrador de concepto (prototipo inicial). Se utilizan diagramas de diseño racional para explicar opciones de diseño en los puntos clave, 3. Análisis de requerimientos – sesión de validación para criticar el demostrador de conceptos” [Sutcliffe97].

El proceso de formalización de escenarios en un árbol de escenarios usando gramáticas regulares, descrito en [Hsia94] es otro ejemplo de un enfoque top-down.

La propuesta de [Dano97], en cambio, es más cercana a un enfoque bottom-up, puesto que sugiere: “...coleccionar y describir casos de uso con una notación tabular”, “... crear Redes de Petri para cada caso de uso para proveer al analista con el formalismo necesario”, y “...establecer vínculos formales entre casos de uso para obtener una descripción global de la aplicación”.

Robertson indica la necesidad de vincular escenarios: “Puesto que un solo escenario brinda información parcial, el análisis basado en escenarios involucra la observación de un conjunto de escenarios.”, y “La técnica sistemática de pregunta-respuesta ayuda a crear un puente entre los eventos específicos del escenario y las situaciones generales que habilitan, causan, o explican los eventos” [Robertson95].

La propuesta de Potts et al. también puede ser incluida en el enfoque bottom-up: “Los escenarios son representados en dos niveles de detalle: 1. Episodios o fases que son secuencias de acciones de baja granularidad, 2. Familias de escenarios utilizando la relación de Jacobson” [Potts94].

Es también útil observar la construcción de casos de uso, donde puede verse el enfoque top-down: “1. Determinar el sistema de software (determinar los límites y el alcance, identificar actores, identificar interfaces del sistema, desarrollar la primera partición en subsistemas), 2. Clasificar a los actores (activos o pasivos, roles), 3. Determinar casos de uso del sistema (descomponer al sistema en trozos discretos significativos tanto para las personas del negocio como para los desarrolladores), 4. Construir conversaciones (secuencia de interacciones entre los actores y el sistema para cada caso de uso)” [Wirfs95].

La propuesta de [Ober98] también corresponde a un enfoque top-down, puesto que establece: “1. Analizar el problema”, “2. Entender las necesidades del *stakeholder*”, que involucra “Encontrar actores y casos de uso” y “Bosquejar el modelo de casos de uso”, “3. Definir el sistema”, que incluye: “Refinar el modelo de casos de uso” y “Describir los casos de uso”, “4. Administrar el alcance del proyecto”, que implica: “Priorizar los casos de uso” y “Modelar una vista de casos de uso de la arquitectura del sistema”, “5. Refinar la definición del sistema” que incluye: “Detallar los casos de uso”, y finalmente “6. Administrar cambios en los requisitos”.

La propuesta de Schneider et al. consiste en cuatro fases principales que siguen un enfoque top-down: “1. Fase inicial: se identifican actores y se desarrollan casos de uso de

alto nivel que ayudarán a determinar el alcance del proyecto; 2. Fase de elaboración: se desarrollan casos de uso más detallados, que serán usados para crear el plan para la próxima fase. El producto de esta fase son escenarios primarios detallados y escenarios secundarios; 3. Fase de construcción: se usan los casos de uso como punto de partida para diseñar y desarrollar planes de prueba; 4. Fase de transición: los casos de uso pueden ser usados para desarrollar guías para el usuario y para su entrenamiento” [Schneider98].

Un enfoque intermedio aunque cercano al top-down es el de Constantine que indica que el proceso para generar casos de uso involucra: “Identificar casos de uso candidatos basados en el modelo de roles de usuario y *brainstorming*: generar una lista de casos de uso; bosquejar un mapa de casos de uso basado en el conocimiento inicial (relaciones entre casos de uso); Desarrollar narrativas de casos de uso en una forma estructurada; Reducir casos de uso concretos a su forma esencial; Cambiar entre las narrativas y el mapa cuando sea necesario” [Constantine98].

Jacobson, en cambio, sigue un enfoque estrictamente bottom-up: “Primero describimos los casos de uso básicos, y luego describimos como mejorarlos para lograr casos de uso más avanzados, algunos de los cuales pueden depender de los básicos. Se deberían obtener casos de uso abstractos a partir de los concretos, y no a la inversa. Las asociaciones nos permiten capturar los requisitos funcionales de un sistema complejo, de la misma manera en que aprendemos acerca de cualquier nuevo tema: primero entendemos las funciones básicas, luego agregamos complejidad” [Jacobson94a] [Jacobson94b].

Las heurísticas de Gough et al. siguen un enfoque bastante cercano al que se utilizará en este trabajo: “1. Crear documentos en lenguaje natural: documentos de alcance del proyecto, documentos de necesidades del cliente, documentos de necesidades de servicio, análisis de la competencia y del mercado. Especificar funcionalmente el sistema existente, 2. Identificar los actores principales del sistema, 3. Derivar los encabezados de los escenarios, 4. Completar la descripción de los escenarios.”. En las conclusiones establecen: “la necesidad de proveer una vista de alto nivel de la interacción de los escenarios como un medio de manejar la complejidad de un número grande de escenarios” [Gough95].

En [Weidenhaupt98] pueden verse cuatro procesos de construcción de escenarios diferentes: “i) Descomposición top-down, ii) Escenarios desde caja negra a caja traslúcida, iii) Desde una definición informal de escenarios a una definición formal, y iv) Desarrollo incremental de escenarios.”

El enfoque que se utilizará en este trabajo es el de [Leite00], que propone una estrategia middle-out: “En nuestro proceso, donde los escenarios son desarrollados en primer término para entender el UdeD, comenzamos con un enfoque bottom-up, y luego organizamos los escenarios utilizando la estrategia middle-out.”

En [Leite00] se propone una estrategia para la creación y uso de escenarios que enfoca dos importantes problemas: la organización y la calidad. El problema de organización surge cuando se hace necesario manejar un gran número de escenarios. El problema de la calidad se refiere a la confiabilidad de las descripciones presentadas como escenarios. Con respecto al primer problema, se presenta una innovadora estrategia middle-out que sistematiza el proceso de construcción utilizando relaciones tipadas y heurísticas operacionales. Con respecto a la calidad, se proveen políticas y procedimientos para detectar defectos y errores en escenarios.

CAPÍTULO 2

BASELINE DE REQUISITOS

El enfoque de Leite [Leite97b] se basa en una baseline orientada al cliente. Esta baseline es un metamodelo que contiene descripciones sobre el dominio del problema y el artefacto de software que ha de ser construido dentro de ese dominio. Estas descripciones son escritas en lenguaje natural siguiendo patrones determinados y están relacionadas entre sí. El uso de lenguaje natural posibilita la intervención de los clientes a la hora de validar el conjunto de especificaciones obtenidas.

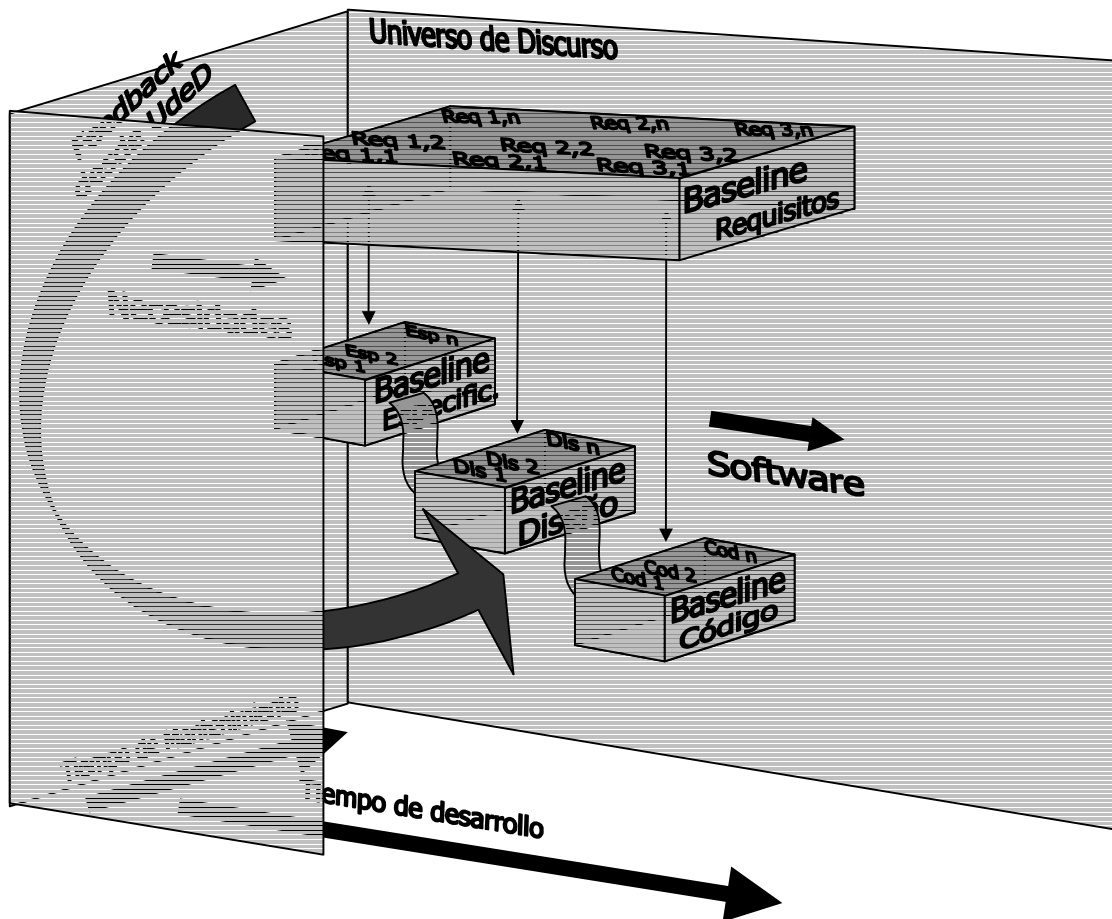


Figura 1 – Baseline de Requisitos y el proceso de desarrollo de software

La baseline de requisitos es una estructura dinámica que se genera durante el proceso de ingeniería de requisitos que evoluciona junto al proceso de desarrollo de software y que acompaña las tareas de mantenimiento. Es importante enfatizar la importancia de la capacidad de referencias cruzadas entre la Baseline de Requisitos y el proceso de desarrollo

y mantenimiento. Esto permite el seguimiento de los requisitos en cualquier punto durante el desarrollo y el mantenimiento hacia su punto de origen [Garcia99]. En la figura 1 se observa como la evolución de la baseline va aparejada con la evolución del proceso de software, tanto en lo que se refiere al desarrollo, como al mantenimiento.

2.1 COMPONENTES DE LA BASELINE DE REQUISITOS

La baseline, de acuerdo con [Leite97b], está compuesta por cinco vistas:

- LEL, vista del modelo léxico
- BMV, vista del modelo básico
- SMV, vista del modelo de escenarios
- HV, vista de hipertexto
- CV, vista de configuración

La vista léxica está basada en un metamodelo diseñado para ayudar a la elicitación del lenguaje usado en el macrosistema. Este metamodelo se denomina Léxico Extendido del Lenguaje (LEL).

La vista básica está soportada por diagramas de entidad-relación que representan los requisitos externos propuestos por le cliente del macrosistema.

La vista de escenarios describe situaciones del comportamiento de la aplicación en un momento específico.

La vista de hipertexto trabaja como un integrador de la vista léxica, la vista de escenarios y la vista básica, habilitando la definición de vínculos dentro de la misma vista y con las demás. Los escenarios, por ejemplo, tienen un fuerte vínculo con el LEL ya que hacen referencia a los términos de dicho léxico.

La vista de configuración es un sistema de versiones que lleva la historia de los cambios en la vista léxica, en la vista de escenarios y en la vista básica durante el desarrollo y el mantenimiento. Esta vista muestra la evolución del sistema de software.

Es importante destacar que las vistas hipertexto y de configuración son ortogonales a las otras tres. Constituyen un soporte de servicios indispensable para garantizar el acceso a la información almacenada (HV) y el seguimiento de los requisitos y sus revisiones (CV).

2.2 MODELO DE OBJETOS

A partir de los modelos correspondientes al LEL y a los escenarios puede encontrarse un modelo preliminar de objetos. Los escenarios son descriptos basándose en actores que colaboran entre sí para alcanzar un objetivo. Si se consideran los impactos de un término de tipo sujeto u objeto del LEL, es posible hacer el seguimiento de dicho término en los escenarios en los que participa. Esto permite analizar el comportamiento global de un actor o un recurso. De este modo, el LEL permite relacionar los escenarios y determinar el comportamiento completo de una entidad dentro del sistema. Si se analizan los escenarios en forma aislada, sobre todo en proyectos grandes, se corre el riesgo de no tener en cuenta

que un objeto puede participar en varios escenarios.

Analizar el modelo del LEL y de escenarios en forma conjunta, como se indicó, permite encontrar objetos, sus responsabilidades y colaboraciones con otras clases [Wirfs95]. De este modo se obtiene un modelo preliminar de objetos. Las relaciones entre los tres modelos permiten tener información completa y complementaria sobre el dominio de la aplicación y permiten la trazabilidad de la información.

El modelo preliminar de objetos definido constituye la base para la etapa de diseño, donde se definirán las jerarquías de herencia y se definirá la estructura interna de las clases. El modelo obtenido sólo es una visión del dominio del problema en términos de objetos de la aplicación. De este modelo se deberá identificar qué clases serán implementadas con tecnología y cuáles representan clases externas al sistema. Este paso se hace con la colaboración del usuario. De esta manera se obtiene un modelo de objetos que será el punto de partida para un modelo de diseño de objetos, con la mayor parte del trabajo ya hecho (encontrar clases, responsabilidades y colaboraciones).

Para describir el proceso completo de derivación de Objetos se utilizará el modelo SADT [Ross77]. Las cajas representan actividades; las flechas desde la izquierda representan las entradas requeridas por la actividad; las flechas desde abajo representan controles; las flechas desde arriba representan mecanismos y las flechas hacia la derecha representan salidas desde la actividad. Este modelo será utilizado también en los siguientes capítulos para describir el proceso de construcción del LEL y de escenarios.

En la figura 2 se presenta el modelo SADT correspondiente al proceso de Derivación de Objetos con sus entradas, salidas, controles y mecanismos.

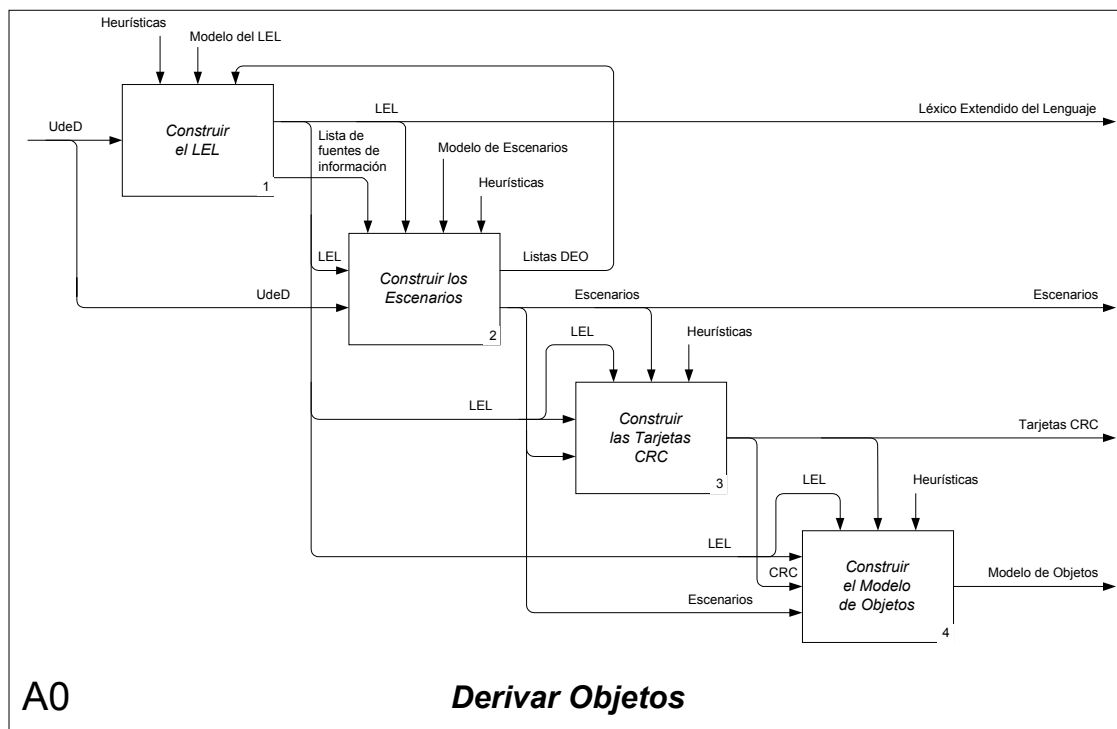


Figura 2 – SADT del proceso Derivar Objetos

En el capítulo siguiente se presentan, en detalle, las etapas correspondientes a la Construcción del LEL y la Construcción de Escenarios, puesto que el desarrollo de este trabajo se basa principalmente en estos dos modelos.

CAPÍTULO 3

LEL Y ESCENARIOS

El Léxico Extendido del Lenguaje es una representación de los símbolos del lenguaje del dominio del problema, que intenta capturar el vocabulario de una aplicación. Su objetivo principal es que el ingeniero de software entienda el lenguaje que habla el usuario, entendiendo los términos que utiliza, sin preocuparse por entender el problema [Leite93] [Leite97b].

El LEL involucra la denotación y la connotación de cada símbolo descubierto como una palabra o frase relevante al dominio de la aplicación. El propósito de construcción del léxico no sólo es habilitar una buena comunicación y acuerdo entre los clientes/usuarios y el equipo de ingeniería sino también facilitar la construcción de escenarios y ayudar a su descripción, facilitando la validación.

Un escenario es una descripción parcial y concreta del comportamiento de un sistema en una determinada situación. Es una descripción parcial, porque no necesita describir todas las características de las entidades involucradas, sólo se describe aquello que está relacionado con un comportamiento particular del sistema analizado. A pesar de estar acotados a un determinado comportamiento, describen todo el contexto que involucra a esa actividad: recursos del sistema, objetivos de los usuarios, contexto social en que se desarrolla, entidades involucradas. Proveen un “retrato” de como esa actividad se lleva a cabo.

3.1 USO DEL LENGUAJE NATURAL

Como puede verse en el Capítulo 1, existen diferentes estrategias de construcción de escenarios. La que utilizaremos en este trabajo se basa en el vocabulario del Universo de Discurso. El Léxico Extendido del Lenguaje permite registrar ese vocabulario, reflejando las palabras peculiares y más usadas en el UdeD.

Los escenarios logran diferentes objetivos dependiendo de la fase donde se usen durante el proceso de desarrollo de software. En la fase de Ingeniería de Requisitos, los objetivos del escenario son:

- capturar los requisitos,
- proveer un medio de comunicación entre los stakeholders,
- proveer un soporte para trazabilidad

El uso de lenguaje natural para describir situaciones cumple con el objetivo de mejorar la comunicación con los stakeholders. El uso del LEL y Escenarios para la elicitación de requisitos y su utilización a través de todo el proceso de desarrollo de software permite la validación con el cliente/usuario. El propósito principal del léxico es capturar el

vocabulario de la aplicación y su semántica, posponiendo la comprensión de la funcionalidad de la aplicación. Los escenarios son usados para entender la aplicación y su funcionalidad: cada escenario describe una situación específica de la aplicación centrando la atención en su comportamiento [Leite00].

El uso del LEL como soporte para la construcción de los escenarios aborda el problema de la ambigüedad en las descripciones basadas en lenguaje natural. Cada vez que se utiliza un símbolo del LEL en la descripción de un escenario, ese símbolo se convierte en hipervínculo, permitiendo así la comunicación entre ambas representaciones.

3.2 SÍMBOLOS DEL LEL

Este léxico se construye utilizando lenguaje natural y está compuesto, en primera instancia, por símbolos que pueden ser objetos activos o Sujetos (realizan acciones), objetos pasivos u Objetos (las acciones se realizan sobre ellos), Verbos (acciones del sistema) y Estados significativos del sistema [Leite97a].

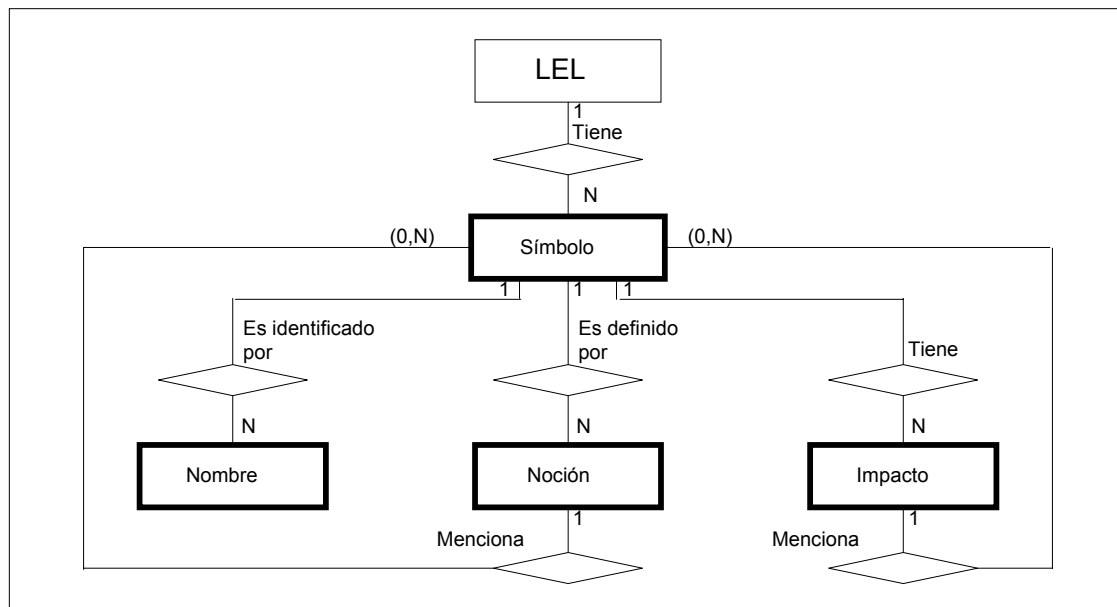


Figura 3 – Diagrama Entidad-Relación para el modelo del Léxico Extendido del Lenguaje

En la figura 3 se presenta el diagrama Entidad-Relación [Chen76] para el modelo del Léxico Extendido del Lenguaje. Cada símbolo en el léxico tiene uno o más nombres o frases que lo identifican y dos tipos de descripciones, la noción y el impacto. La noción describe la denotación de la palabra o frase. Indica quién, cuándo ocurre, qué procesos involucra, qué significado tiene el símbolo, etc. El impacto describe la connotación del símbolo, es decir, su repercusión en el sistema. Esta descripción puede contener acciones que se ejecutan sobre otros objetos o que se aplican al que se está describiendo, situaciones derivadas de la que se está definiendo, etc. Cada entrada puede contener una o más nociones y uno o más impactos [Leite96].

En la descripción de los símbolos deben cumplirse simultáneamente dos reglas básicas:

Principio de circularidad: en la descripción de la noción o impacto de los símbolos se debe maximizar el uso de otros símbolos del léxico. De esta manera, el conjunto de símbolos determina una red, que permite representar al LEL mediante un hipertexto que puede ser navegado para conocer todo el vocabulario del problema.

Principio del vocabulario mínimo: se debe minimizar el uso de símbolos externos al lenguaje de la aplicación. De este modo, se acota el lenguaje al menor conjunto de símbolos posible. Si se utilizan símbolos externos, éstos deben pertenecer al vocabulario básico del lenguaje natural que se está utilizando [Leite97b]:

A continuación se presenta el modelo utilizado para representar los símbolos del LEL:

<p>LEL: Representación de los símbolos en el lenguaje del dominio de la aplicación Sintaxis: $\{\text{Símbolo}\}_1^N$</p> <p>Símbolo: Entrada del léxico que tiene un significado especial en el dominio de la aplicación Sintaxis: $\{\text{Nombre}\}_1^N + \{\text{Noción}\}_1^N + \{\text{Impacto}\}_1^N$</p> <p>Nombre: Identificación del símbolo. Más de uno representa sinónimos Sintaxis: Palabra Frase</p> <p>Noción: Denotación del símbolo. Debe ser expresado usando referencias a otros símbolos y usando un vocabulario mínimo Sintaxis: Sentencia</p> <p>Impacto: Connotación del símbolo. Debe ser expresado usando referencias a otros símbolos y usando un vocabulario mínimo Sintaxis: Sentencia</p>
--

Figura 4 – Modelo del Léxico Extendido del Lenguaje

donde:

Sentencia está compuesto por Símbolos y No-Símbolos pertenecientes al vocabulario mínimo,
+ significa composición,
{x} significa cero o más ocurrencias de x,
() es usado para agrupamiento,
| significa or, y
[x] significa que x es opcional

3.3 PROCESO DE CONSTRUCCIÓN

El proceso de construcción consta de 6 etapas dependientes entre sí, y que, en algunos casos, se desarrollan en forma simultánea.

1. *Identificar las fuentes de información.*
2. *Identificar los símbolos.*
3. *Clasificar los símbolos.*
4. *Describir los símbolos.*
5. *Verificar el LEL.*
6. *Validar el LEL con los clientes-usuarios.*

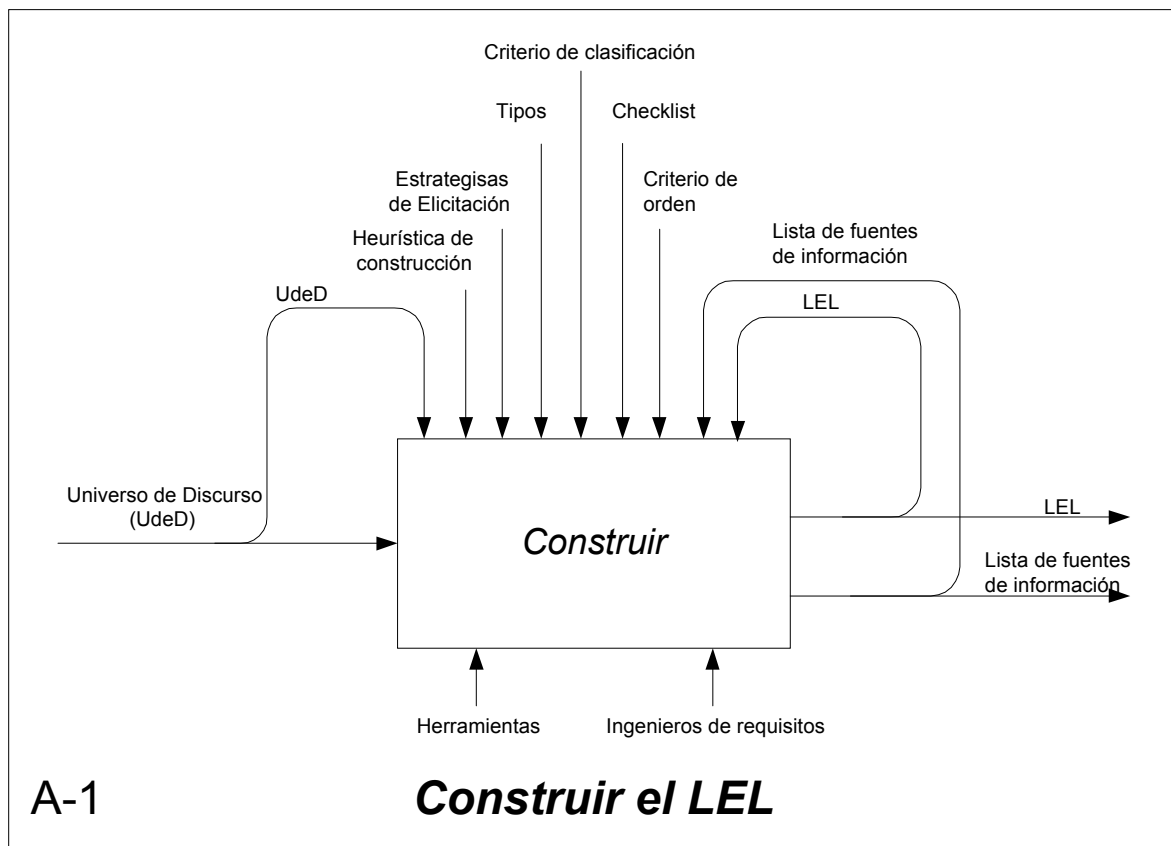


Figura 5 – SADT del proceso de construcción del LEL

En la figura 5 se presenta un modelo SADT para el proceso del Construcción del LEL con sus entradas, salidas y controles, y luego, en la figura 6 se presenta la explosión del SADT de la figura 5, considerando las etapas del proceso de construcción.

El modelo SADT del proceso de construcción del LEL muestra un flujo principal compuesto por cuatro tareas: *Identificar las fuentes de información*, *Identificar los símbolos*, *Clasificar los símbolos* y *Describir los símbolos*. Existe un ciclo entre las etapas de *Validación* y de *Verificación*. Después de verificar el LEL, el proceso vuelve a la etapa de *Descripción de los símbolos* donde las correcciones son realizadas teniendo en cuenta las

listas DEO (Discrepancias, Errores y Omisiones). Después de la tarea de *Validación*, el proceso regresa a la actividad *Identificar símbolos* y/o a la actividad *Describir símbolos*, dependiendo de la lista DEO de validación, para hacer los cambios necesarios. Por ejemplo, mientras se describe un símbolo puede descubrirse un tipo mal asignado, lo que obliga a volver atrás en el proceso y reclasificarlo [Garcia99].

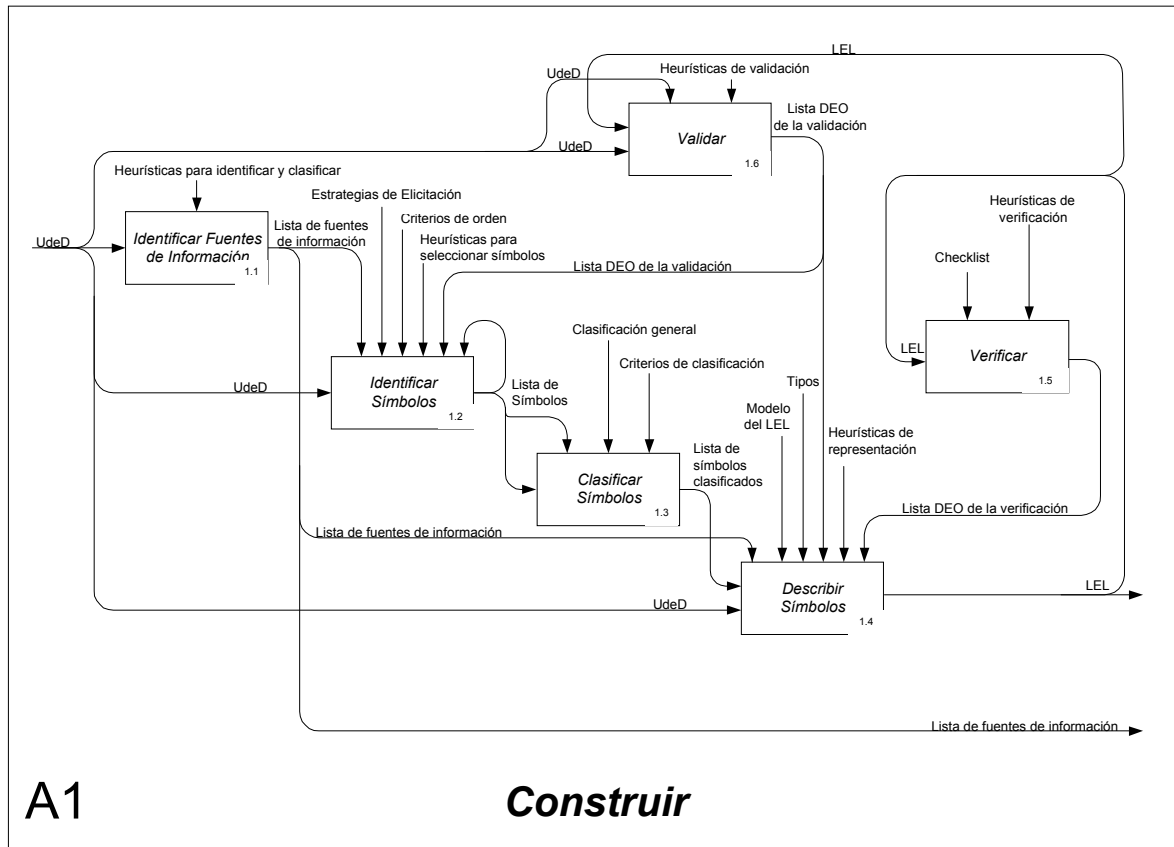


Figura 6 – SADT de las etapas del proceso de construcción del LEL

El primer paso para la construcción del LEL es la identificación de las fuentes de información. Estas fuentes están contenidas en el UdeD, cuyos límites son raramente precisos, siendo incluso redefinidos, en algunos casos, a lo largo del proceso de desarrollo.

Las principales fuentes de información son los documentos y las personas involucradas en el dominio de la aplicación, libros de temas relacionados y otros sistemas disponibles en el mercado.

Posteriormente, se procede a identificar los símbolos, para lo cual se seleccionan las estrategias de elicitación, se recolectan y organizan los símbolos. Las principales técnicas para recolectar información son la lectura de documentos, entrevistas, observaciones, aproximaciones antropológicas, reuniones, anotaciones del diseño de software y otras. La estrategia depende fundamentalmente de las fuentes identificadas en la etapa previa, aunque las entrevistas son el medio más corriente para la recolección de símbolos. En ellas, el ingeniero puede reconocer fácilmente el vocabulario que el cliente-usuario emplea en su ambiente. Esta técnica se combina normalmente con lectura de documentos principalmente formularios, manuales y políticas organizacionales. Aplicando las técnicas seleccionadas, el

ingeniero de requisitos identifica los símbolos y construye una lista de símbolos candidatos.

Después de las primeras validaciones, la lista de símbolos candidatos generada inicialmente es ampliamente modificada. Se incorpora un gran número de símbolos y sinónimos, se eliminan símbolos erróneos y se clarifican símbolos confusos.

Posteriormente, se procede a clasificar los símbolos. Esta etapa asegura la integridad y la homogeneidad de las descripciones. Existe un criterio de clasificación general, que agrupa los símbolos en cuatro grandes categorías: Sujetos, Verbos, Objetos y Estados. Al finalizar esta etapa, cada uno de los símbolos de la lista queda identificado con un tipo que guiará su posterior descripción.

La descripción de los símbolos implica la definición de sus nociones e impactos, basadas en el modelo LEL y los tipos obtenidos en la clasificación. El ingeniero de requisitos, si lo necesita, recurre al paquete de información previamente obtenido.

Para un símbolo del tipo *Sujeto*, la noción define quién es el sujeto y en el impacto deben registrarse las acciones que ejecuta o recibe.

Para un símbolo del tipo *Verbo*, la noción debe decir quién ejecuta la acción, cuándo ocurre y las actividades involucradas con dicha acción. El impacto debe identificar las situaciones que impiden la ocurrencia de la acción, qué otras acciones son disparadas en el ambiente y qué situaciones son causadas por la acción.

Para un símbolo del tipo *Objeto*, la noción debe definir al objeto e identificar otros símbolos del mismo tipo con los cuales se relaciona. El impacto describe las acciones que pueden aplicarse a este objeto.

Para un símbolo del tipo *Estado*, la noción define su significado y las acciones que llevan a ese estado. El impacto debe identificar otros estados y acciones que pueden ocurrir a partir de la situación específica.

Una vez descriptos los símbolos, se procede a la *Verificación* con el objeto de controlar que el LEL producido sea consistente y homogéneo. Se realiza utilizando heurísticas de verificación y una lista de chequeo. A partir de esta tarea se obtiene una lista DEO, que es utilizada posteriormente como control de la etapa *Descripción de Símbolos*.

Mientras se identifican y describen los símbolos tiene lugar una validación informal. Estas validaciones tratan de chequear la lista candidata y, además, permiten al ingeniero de requisitos percibir el significado de los símbolos. Las siguientes validaciones son usadas para chequear el conocimiento obtenido y para mejorar el LEL mediante el esclarecimiento de dudas, corrección de las definiciones de los símbolos y la exclusión y agregado de símbolos. La tarea de *Validación* generalmente consiste en entrevistas estructuradas o reuniones con clientes-usuarios en su lugar de trabajo. Esta tarea es llevada a cabo fácilmente con los clientes-usuarios ya que éstos no tienen dificultades en la comprensión del LEL puesto que está escrito en lenguaje natural y empleando su propio vocabulario. Como resultado, se genera la lista DEO, que es utilizada para hacer correcciones en la Identificación y Descripción de los símbolos.

En las figuras 7 y 8 se presentan ejemplos de símbolos del LEL correspondiente al caso de estudio Agenda de Reuniones [Hadad98]. El primero corresponde a un símbolo de tipo Sujeto, y el segundo a uno de tipo Verbo.

CONVOCADO
<p>Noción:</p> <ul style="list-style-type: none"> - Persona invitada a la <u>reunión</u> <p>Impactos:</p> <ul style="list-style-type: none"> - Puede informar sus <u>horarios disponibles</u> - Puede dar el <u>aviso de concurrencia</u> - Debe estar en el lugar, <u>fecha</u> y <u>horarios</u> establecidos en la <u>convocatoria</u> - Puede dar el <u>aviso de no concurrencia</u> - Asigna un <u>reemplazante</u> en caso de no poder asistir y lo informa al <u>convocante</u> o <u>secretaria</u> - Puede definir el <u>material para repartir</u> - Registra el tiempo de traslado de la <u>reunión</u>

Figura 7 – Símbolo del LEL de tipo Sujeto

DISEÑO DE LA AGENDA DE REUNIONES
<p>Noción:</p> <ul style="list-style-type: none"> - Actividad que realiza el <u>convocante</u> para determinar la <u>fecha</u>, la <u>hora</u> y el <u>lugar</u> de las <u>reuniones</u>, basándose en los <u>horarios disponibles</u> de los <u>convocados</u>, la <u>disponibilidad de espacio</u> y otras <u>reuniones ya registradas</u> en la <u>agenda</u> - Tiene la finalidad de organizar los tiempos y evitar superposiciones y omisiones de <u>reuniones</u> <p>Impactos:</p> <ul style="list-style-type: none"> - Se registra en la <u>agenda</u>: el <u>objetivo</u>, la <u>fecha</u>, la <u>hora</u> y el <u>lugar</u> establecido para llevar a cabo la <u>reunión</u> - Se confecciona el <u>listado para convocatoria</u> - Se confecciona el <u>temario</u> - Se registra en el <u>cronograma de reuniones</u> - Se reserva el <u>espacio físico</u> - Se reserva el <u>material físico</u>

Figura 8 – Símbolo del LEL de tipo Verbo

3.4 ESCENARIOS

Los escenarios describen situaciones teniendo en cuenta aspectos de uso, permitiendo: conocer el problema, unificar criterios, ganar compromiso con clientes / usuarios, organizar los detalles involucrados y entrenar a nuevos participantes. El uso de escenarios como una técnica para entender el problema a resolver usando un sistema de software ha sido recomendado por varios autores [Potts95], [Booch91], [Jacobson92], [Zorman95] y dichas propuestas se hicieron muy importantes para extender el uso de escenarios en la práctica real.

Por otra parte, 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 útil desarrollar un conjunto de escenarios con el objeto de utilizarlos para modelar el comportamiento del sistema. En este caso, los usuarios finales simulan su interacción con el sistema usando el escenario. Así, pueden explicar al equipo de ingenieros de requisitos lo que están haciendo y la información que necesitan del sistema para llevar a cabo la tarea descrita en el escenario. Además, el proceso de desarrollar un escenario, aun sin considerar la interacción del usuario, puede ayudar a entender los requisitos. Descubrir escenarios posibles expone el rango de posibles interacciones del sistema y revela facilidades del sistema que podrían ser requeridas [Kotonya98].

Por otra parte, los escenarios pueden ser utilizados como un medio para obtener métricas. En [Sutcliffe01], por ejemplo, se propone un método que utiliza escenarios para medir la carga de trabajo de los agentes en términos de las tareas que tienen que ejecutar, y lograr así estimaciones de tiempo y probabilidades de falla. En [Gans01] se presenta una utilización similar de escenarios de flujo de información para determinar los perfiles de calidad requeridos. Y en [Cleland01] se propone una métrica dinámica basada en requisitos, que se sustenta en el hecho que los escenarios representan el comportamiento anticipado del sistema, y proveen, en una etapa muy temprana del diseño, los medios para predecir cuantitativamente ese comportamiento.

Los escenarios pueden ser vistos como historias que explican como se usa el sistema. Son principalmente útiles para agregar detalle a una descripción de requisitos. Una vez que se tiene una idea básica de las facilidades que un sistema debería proveer, se pueden desarrollar escenarios alrededor de esas facilidades.

Los escenarios pueden ser escritos de diferentes maneras, pero deberían incluir al menos, la siguiente información:

- Una descripción del estado del sistema antes de entrar al escenario
- El flujo normal de eventos en el escenario
- Excepciones al flujo normal de eventos
- Información acerca de otras actividades que podrían estar sucediendo al mismo tiempo
- Una descripción del estado del sistema después de completar el escenario

3.5 COMPONENTES DE UN ESCENARIO

El modelo de escenario de la metodología de [Leite00] es una estructura compuesta por las siguientes entidades: Título, Objetivo, Contexto, Recursos, Actores, Episodios y Excepciones y el atributo restricción. Actores y Recursos son enumeraciones. Título, Objetivo, Contexto y Excepciones son sentencias declarativas, mientras Episodios es un conjunto de sentencias expresadas en un lenguaje simple que da una descripción operacional de comportamiento.

La figura 9 presenta la estructura de un escenario usando el diagrama entidad-relación, donde sólo se muestran las relaciones relevantes.

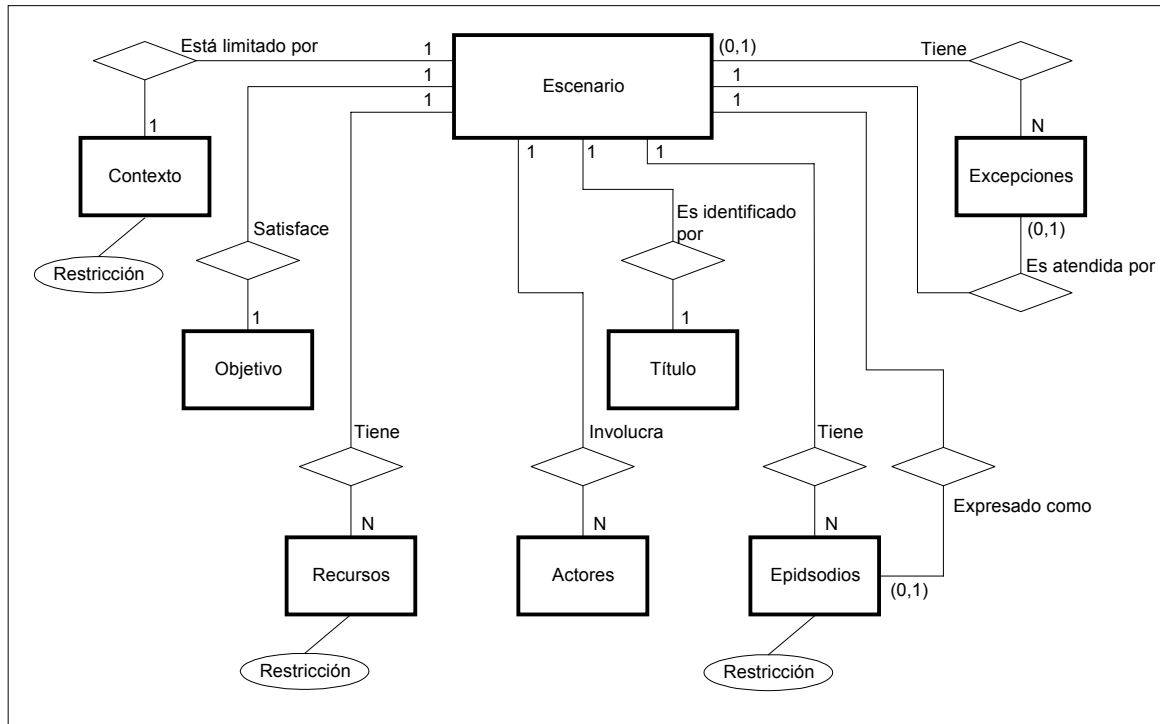


Figura 9 – Diagrama Entidad-Relación para el modelo de Escenarios

Un escenario, identificado por un **título**, debe satisfacer un **objetivo** que se alcanza mediante la ejecución de los **episodios**. Éstos representan el curso de acción principal, pero incluyen también variaciones o alternativas posibles. Cada **episodio** representa una acción realizada por un **actor**, con la participación de otros **actores** y la utilización de **recursos**.

Mientras se ejecutan los **episodios** puede surgir una **excepción**, que señala un obstáculo para lograr el objetivo. El tratamiento de la **excepción** puede satisfacer o no el **objetivo** original. La comprensión de un escenario se ve facilitada por el uso de lenguaje natural, situaciones bien limitadas y el uso de **subescenarios**. Estos permiten manejar el problema de la explosión de escenarios señalada en [Cockburn95].

Un **subescenario** es usado cuando:

- se detecta comportamiento común en varios escenarios,
- aparecen cursos de acción condicionales o alternativos complejos en un escenario, y
- se detecta en un escenario la necesidad de mejorar una situación con un objetivo concreto y preciso

El **contexto** se describe detallando una ubicación geográfica, una ubicación temporal y precondiciones. Cada uno de estos subcomponentes puede ser expresado por una o más sentencias simples vinculadas por los conectores lógicos *and*, *or*.

El atributo **restricción** es usado para caracterizar requisitos no funcionales aplicados a contexto, recursos y episodios.

Un escenario puede ser interrumpido por **excepciones**. Cada **excepción** se describe

como una sentencia simple que especifica la causa de la interrupción. Si se incluye el título de otro escenario, la **excepción** será tratada por dicho escenario.

La posibilidad de registrar **excepciones** en escenarios ha sido cuestionada, en algunos casos, argumentando que las discusiones sobre excepciones podrían distraer a los expertos del dominio, alejándolos del tratamiento de los puntos centrales del sistema, en otros, aduciendo que podría verse afectada la “vendibilidad” del sistema. Sin embargo, negar o descuidar importantes excepciones en ingeniería de requisitos, podría resultar en la disconformidad del cliente y altísimos costos a largo plazo [Rivero98].

Los **episodios** puede ser simples, condicionales y opcionales. Los *episodios simples* son aquéllos necesarios para completar el escenarios. *Episodios condicionales* son aquéllos cuya ocurrencia depende de una condición específica. La condición puede ser interna o externa al escenario. Las condiciones internas pueden ser debidas a precondiciones alternativas, restricciones de actores o recursos y episodios previos. Los *episodios opcionales* son aquéllos que pueden ocurrir o no dependiendo de condiciones pueden no estar detalladas explícitamente.

Independientemente de su tipo, un **episodio** puede ser expresado como una acción simple o puede ser concebido en si mismo como un escenario, posibilitando de esa manera la descomposición de un escenario en **subescenarios**.

En la figura 10 se presenta el modelo basado en lenguaje natural que se utilizará en adelante para describir escenarios. [Leite97b].

En la representación, se utiliza la siguiente simbología:

- + significa composición,
- {x} significa cero o más ocurrencias de x,
- () es usado para agrupamiento,
- | significa **or**, y
- [x] denota que x es opcional

En la figura 11 se presenta un ejemplo de escenario correspondiente al caso de estudio “Agenda de Reuniones”.

El modelo de escenarios debería ser visto como una guía estructural para:

- obtener un estilo homogéneo de descripción para el conjunto de escenarios:
- demostrar los diferentes aspectos que los escenarios pueden cubrir y
- facilitar la verificación de escenarios (principalmente por un proceso automatizado)

El modelo de escenarios provee la descripción de comportamientos con diferentes órdenes temporales. Una secuencia de episodios implica un *orden de precedencia*, pero un *orden no secuencial* requiere agrupar dos o más episodios según la sintaxis mostrada en la figura 10. Así, es posible expresar un orden paralelo o indistinto.

<p>Escenario: descripción de una situación en el dominio de la aplicación. Sintaxis: Título + Objetivo + Contexto + {Recursos}₁^N + {Actores}₁^N + {Episodios}₂^N + {Excepciones}</p> <p>Título: identificación del escenario. En el caso de un subescenario, el título es el mismo que la sentencia del episodio (ver en la definición de Episodios más abajo), sin las restricciones. Sintaxis: Frase ([Actor Recurso] + Verbo + Predicado)</p> <p>Objetivo: meta a ser alcanzada en el dominio de la aplicación. El escenario describe el cumplimiento del objetivo. Sintaxis: [Actor Recurso] + Verbo + Predicado</p> <p>Contexto: compuesto al menos por uno de los siguientes subcomponentes:</p> <p>Ubicación Geográfica: ubicación física del escenario. Ubicación Temporal: especificación de tiempo para el desarrollo del escenario. Precondición: estado inicial del escenario. Sintaxis: {Ubicación Geográfica} + {Ubicación Temporal} + {Precondición} donde Ubicación Geográfica es: Frase + {Restricción} donde Ubicación Temporal es: Frase + {Restricción} donde Precondición es: [Sujeto Actor Recuso] + Verbo + Predicado + {Restricción}</p> <p>Recursos: elementos físicos relevantes o información que debe estar disponible en el escenario. Sintaxis: Nombre + {Restricción}</p> <p>Actores: personas, dispositivos u organización que tienen un rol en el escenario. Sintaxis: Nombre</p> <p>Episodios: conjunto de acciones que describen al escenario y proveen su comportamiento. Un episodio también puede ser descripto como un escenario. Sintaxis (usando una representación tipo BNF): <episodios> ::= <serie de grupos> <serie de episodios> <serie de grupos> ::= <grupo> <grupo> <grupo no secuencial> <serie de grupos> <grupo> <grupo> ::= <grupo secuencial> <grupo no secuencial> <grupo secuencial> ::= <sentencia básica> <grupo secuencial> <sentencia básica > <grupo no secuencial> ::= # <serie de episodios> # <serie de episodios > ::= <sentencia básica > <sentencia básica> <serie de episodios> <sentencia básica > <sentencia básica> ::= <sentencia simple> <sentencia condicional> <sentencia opcional> <sentencia simple> ::= <sentencia episodio> CR <sentencia condicional> ::= IF <condición> THEN <sentencia episodio> CR <sentencia opcional> ::= [<sentencia episodio >] CR donde <sentencia episodio > es descripta como: (([Actor Recurso] + Verbo + Predicado) ([Actor Recurso] + [Verbo] + Título)) + {Restricción}</p> <p>Excepciones: usualmente reflejan la falta o mal funcionamiento de un recurso necesario. Una excepción impide el cumplimiento del objetivo del escenario. El tratamiento de la excepción puede ser expresado por medio de otro escenario. Sintaxis: Causa [(Solución)] donde Causa es: Frase ([Sujeto Actor Recurso] + Verbo + Predicado) donde Solución es: Título</p> <p><i>Restricción:</i> alcance o requisito de calidad referido a una entidad dada. Es un atributo de Recursos, Episodios básicos o subcomponentes de Contexto. Sintaxis: ([Sujeto Actor Recurso] + [No] Debe + Verbo + Predicado) Frase</p>

Figura 10 – Modelo de Escenario

TÍTULO:	Organizar la reunión
OBJETIVO:	Asegurar el desarrollo eficiente de la <u>reunión</u>
CONTEXTO:	Debe realizarse previamente el <u>diseño de la agenda de reuniones</u>
ACTORES:	<u>convocante</u> <u>secretaria</u> <u>convocados</u>
RECURSOS:	<u>agenda</u> <u>listado para convocatoria</u> <u>temario</u> medios de comunicación (teléfono, fax, correo, computadora, etc.) <u>material físico</u>
EPISODIOS:	El <u>convocante</u> instruye a la <u>secretaria</u> sobre la <u>convocatoria</u> a la <u>reunión</u> CONVOCAR A LA REUNIÓN # AVISAR LA CONCURRENCIA AVISAR LA NO CONCURRENCIA [SOLICITAR MATERIAL FÍSICO] SI la <u>convocatoria</u> fue hecha con anticipación ENTONCES RECORDAR LA REUNIÓN [La <u>secretaria</u> asegura el <u>material físico</u> para la <u>fecha de reunión</u>] La <u>secretaria</u> asegura el <u>espacio físico</u> para la <u>fecha de reunión</u> #
EXCEPCIONES:	El <u>convocante</u> o principales <u>convocados</u> no pueden asistir a la <u>reunión</u> (ANULAR LA REUNIÓN) La <u>fecha de reunión</u> debe ser cambiada (TRASLADAR LA FECHA DE REUNIÓN) Surge la necesidad de un cambio en los <u>requerimientos</u> de la <u>reunión</u> (CAMBIAR LOS REQUERIMIENTOS DE LA REUNIÓN)

Figura 11 – Ejemplo de escenario

3.6 CONSTRUCCIÓN DE ESCENARIOS

El proceso de construcción de escenarios comienza en el léxico del dominio de la aplicación, produciendo una primera versión de los escenarios derivada exclusivamente desde el LEL. Estos escenarios son mejorados utilizando otras fuentes de información y organizados para obtener un conjunto consistente de escenarios que representen al dominio de la aplicación. Durante o después de estas actividades, los escenarios son verificados y validados con los clientes/usuarios para detectar Discrepancias, Errores y Omisiones (DEO).

En la figura 12 se presenta un modelo SADT para el proceso del Construcción de Escenarios con sus entradas, mecanismos, salidas y controles.

En la figura 13 se presenta la explosión del SADT de la figura 12, donde pueden verse las siguientes actividades:

1. *Derivar*
2. *Describir*
3. *Organizar*
4. *Verificar*
5. *Validar*

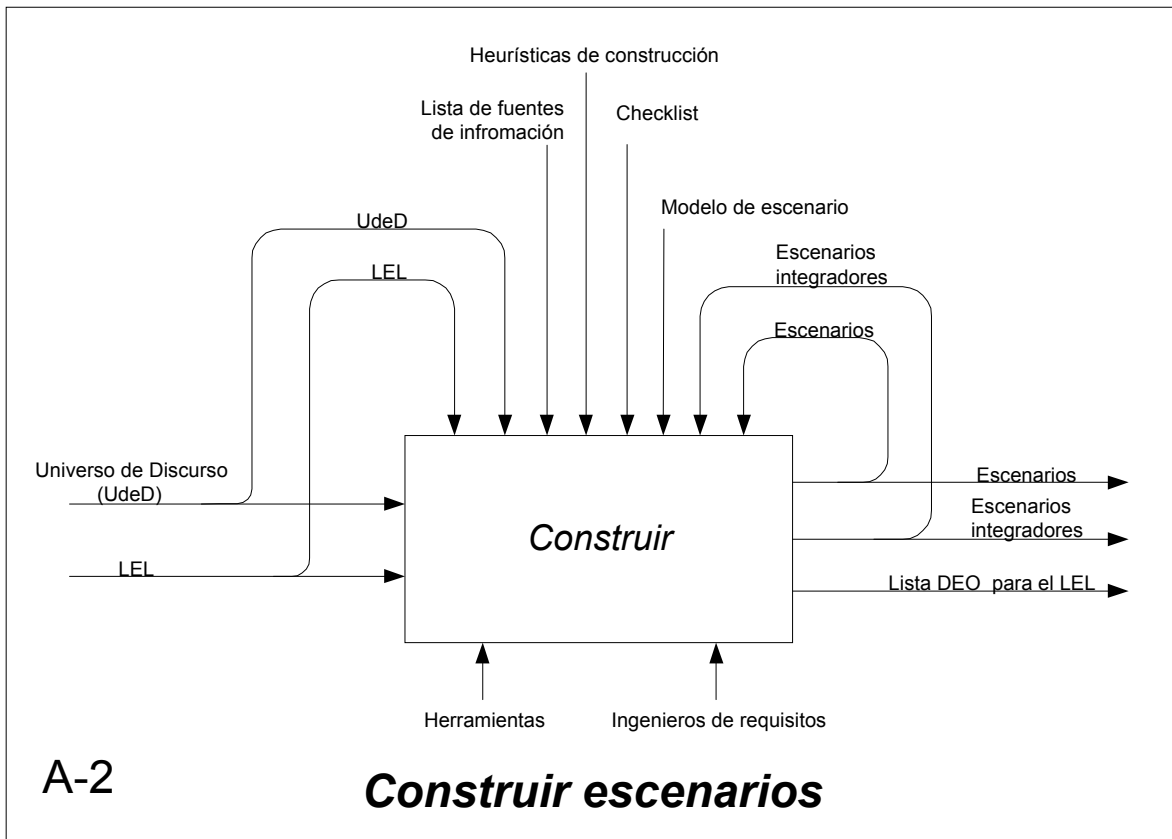


Figura 12 – SADT del proceso de construcción de escenarios

Aunque el proceso de Construcción de Escenarios de la figura 13 muestra un flujo principal compuesto por tres tareas: *Derivar*, *Describir* y *Organizar*, estas actividades no son estrictamente secuenciales. Algunas actividades pueden ser concurrentes debido al mecanismo de feedback, siempre presente en estas situaciones [Goguen92]. Hay un feedback cuando se llevan a cabo las actividades *Verificar* y *Validar*, retornando a la tarea *Describir*, donde se hacen correcciones en base a las listas DEO¹. La actividad *Verificar* ocurre después de describir los escenarios, y después de la organización cuando aparecen nuevos escenarios o se generan los escenarios de integración. Los escenarios son validados con los clientes/usuarios después de la verificación.

El proceso de construcción de escenarios podría producir también tres listas DEO que actuarán como feedback para el proceso de construcción del LEL para mantener la consistencia entre el vocabulario de los escenarios y el LEL propiamente dicho. Estas listas DEO son creadas durante las etapas *Describir*, *Verificar* y *Validar*, y pueden ser descubiertas en este punto porque el conocimiento sobre el dominio de la aplicación comienza a incrementarse y mejorarse fuertemente. Es posible detectar principalmente impactos omitidos, algunas discrepancias entre símbolos y raramente nuevos símbolos del UdeD.

¹ Una lista DEO contiene las discrepancias, errores y omisiones descubiertas durante las actividades de Validación o Verificación, donde se incluyen sugerencias para correcciones.

En las próximas secciones se describe cada una de las actividades del proceso de construcción de escenarios que aparecen en la figura 13.

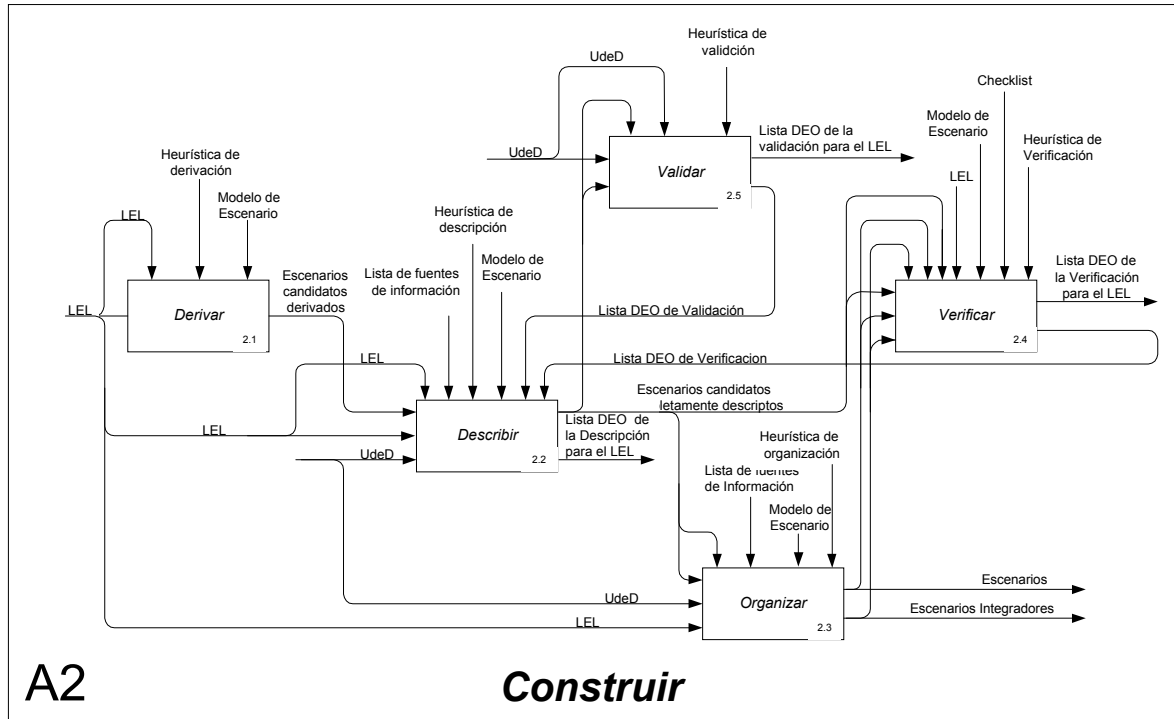


Figura 13 – SADT de las etapas del proceso de Construcción de los escenarios

3.6.1 DERIVAR

Esta tarea apunta a generar los escenarios candidatos a partir de la información del LEL usando el modelo de escenario y aplicando las heurísticas de derivación. La etapa de *Derivación de escenarios* consiste de tres pasos: identificar los actores del UdeD, identificar los escenarios candidatos y crearlos usando el LEL. La figura 14 muestra el modelo SADT que descompone la actividad *Derivar*.

A continuación, se describe brevemente cada una de las etapas correspondientes a la actividad *Derivar*.

Identificar actores

Se identifican dentro del LEL los símbolos que representan actores del UdeD. Deben pertenecer al tipo sujeto. Los actores son clasificados en actores principales y secundarios. Los primeros son aquellos que ejecutan acciones directamente en el dominio de la aplicación, y los segundos son los que reciben y/o dan información, pero no comparten responsabilidades en la acción.

Identificar escenarios

Se extraen del LEL los impactos de los símbolos elegidos como actores principales y

secundarios. Cada impacto representa un posible escenario, y es incorporado a la lista de escenarios candidatos. El título del escenario se construye con la acción (verbo) incluido en el impacto, pero expresado en infinitivo más un predicado también tomado del impacto.

Cuando diferentes actores ejecutan la misma acción, es muy probable que dos o más escenarios de la lista compartan el título. En ese caso todos ellos excepto uno deberían ser eliminados de la lista.

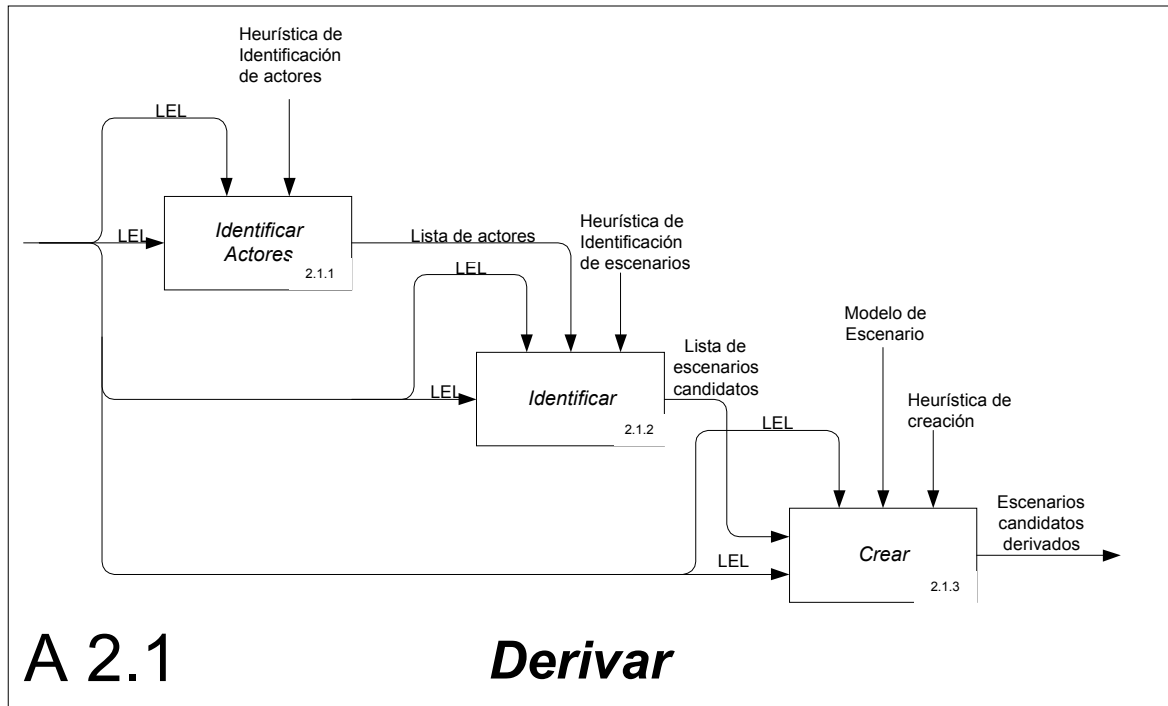


Figura 14 – SADT de la etapa Derivar Escenarios

Crear escenarios

La intención de esta etapa es construir el escenario aprovechando la información del LEL en la mayor medida posible, mediante la aplicación de las heurísticas de creación. El producto de esta etapa lo constituyen los escenarios candidatos derivados.

El contenido de cada impacto del símbolo de tipo Sujeto que llevó a un escenario candidato es analizado para encontrar símbolos del léxico del tipo Verbo.

- ☞ Si el impacto contiene un símbolo Verbo:
 - El objetivo se define según el título y la noción del símbolo Verbo, y el punto de vista de la aplicación.
 - Los actores y recursos del escenario son identificados a partir de la información contenida en el símbolo Verbo y deberían ser símbolos de tipo Sujeto y Objeto respectivamente.
 - Los episodios se derivan a partir de cada uno de los impactos del símbolo Verbo.

- ➔ Si el impacto no contiene un símbolo Verbo:
 - Los símbolos del léxico contenidos en el impacto son identificados y considerados como posibles fuentes de información.
 - El objetivo se define de acuerdo al título del escenario y el punto de vista de la aplicación.
 - Leyendo la definición completa de los símbolos mencionados, se seleccionan posibles actores y recursos. Los primeros se derivan de los símbolos de tipo Sujeto y los segundos de los de tipo Objeto.
 - Los episodios no se derivan del LEL. Su definición se posterga hasta una etapa posterior.

En ambos casos, la ubicación geográfica y temporal del contexto puede ser extraída del impacto del símbolo del LEL que originó el escenario (símbolo Sujeto). La información relevante que debería ser registrada en las precondiciones del contexto podría estar disponible no sólo en ese impacto, sino también en la relación de secuencia con otros impactos del mismo símbolo. Si se encuentra una secuencia, cualquier impacto anterior a la ocurrencia del escenario deberá ser leído cuidadosamente para obtener precondiciones.

Mientras se definen los episodios pueden aparecer excepciones cuyas causas deber ser identificadas.

En la figura 15 se presenta un ejemplo de la actividad *Derivar* para el caso de estudio “Agenda de Reuniones” y en la figura 16 puede verse la versión final del escenario. Las palabras o frases subrayadas son símbolos del léxico correspondiente al caso de estudio. Los números sobre las flechas representan los diferentes pasos en la aplicación de la heurística:

1. A partir de las entradas del LEL se genera la lista de actores,
2. Se consulta la entrada del LEL correspondiente a cada uno de los actores de la lista,
3. A partir de los impactos de esta entrada del LEL, se produce la lista de escenarios candidatos,
4. Se consulta el símbolo de tipo Verbo del LEL correspondiente a uno de los escenarios candidatos,
5. Se usa la noción del símbolo Verbo del LEL para definir el objetivo, los actores y los recursos,
6. Se utilizan los impactos del símbolo Verbo del LEL como base para describir los episodios, y
7. A partir de los impactos correspondientes al símbolo Sujeto del LEL, considerados en el paso 3, se deriva la información de contexto.

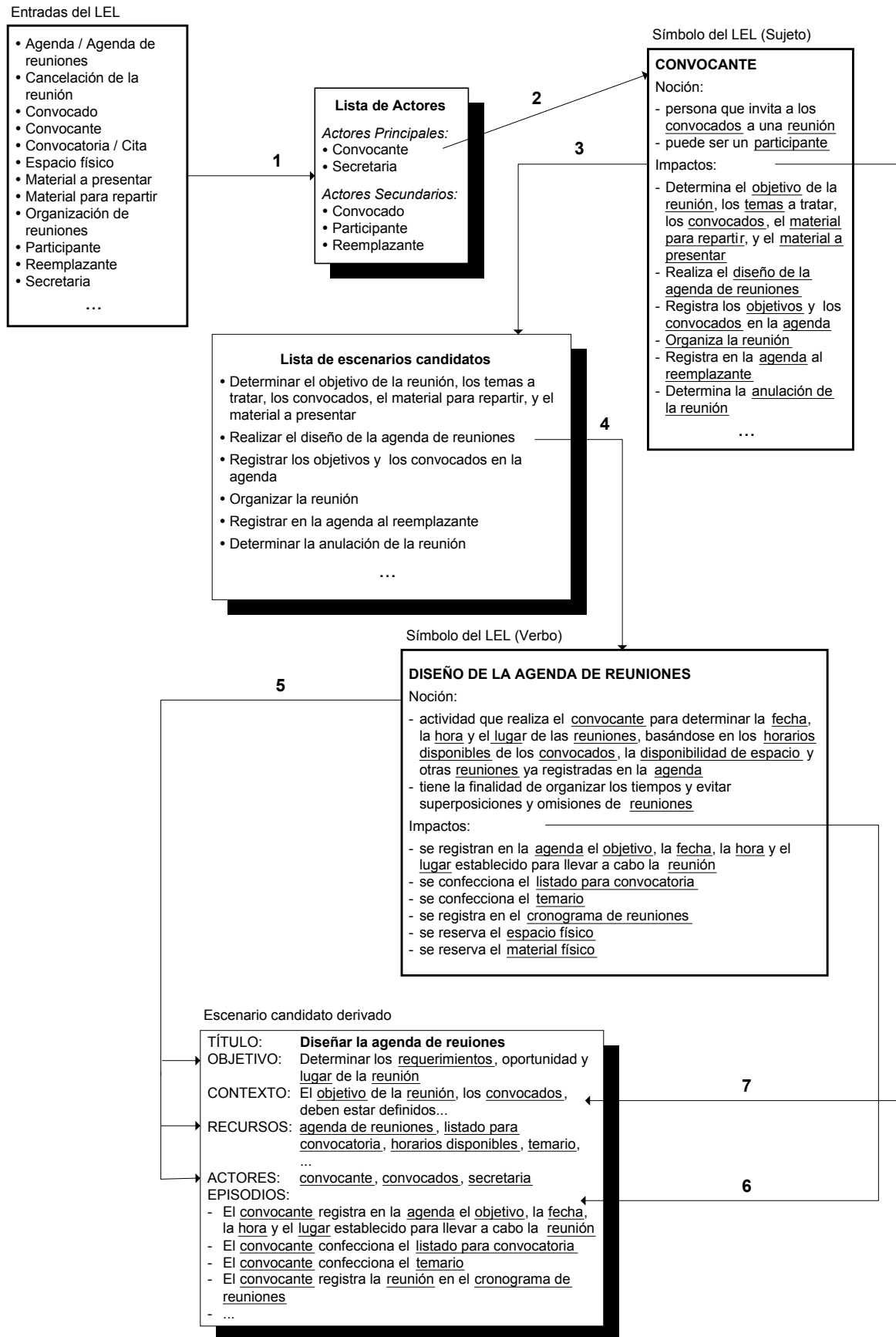


Figura 15 – Ejemplo de derivación de escenarios

TÍTULO:	Diseñar la agenda de reuniones
OBJETIVO:	Determinar los <u>requerimientos</u> , <u>oportunidad</u> y <u>lugar</u> de la <u>reunión</u>
CONTEXTO:	Debe presentarse previamente la necesidad de una <u>reunión</u>
ACTORES:	<u>convocante</u> <u>secretaria</u>
RECURSOS:	<u>agenda de reuniones</u> <u>listado para convocatoria</u> <u>horarios disponibles</u> <u>temario</u> ...
EPISODIOS:	[El <u>convocante</u> obtiene los datos de la <u>reunión</u> a diseñar del <u>esquema de base</u>] #SI los <u>horarios disponibles</u> de los <u>convocados</u> no están registrados ENTONCES SOLICITAR HORARIOS DISPONIBLES El <u>convocante</u> consulta en su <u>agenda de reuniones</u> sus <u>horarios disponibles</u> # ESTABLECER LA FECHA DE REUNIÓN [El <u>convocante</u> determina el <u>material para repartir</u>] #El <u>convocante</u> o la <u>secretaria</u> registran en la <u>agenda</u> el <u>objetivo</u> , la <u>fecha</u> , la <u>hora</u> , el <u>lugar</u> , los <u>temas</u> a discutir, los <u>convocados</u> , el <u>material a presentar</u> y el <u>material para repartir</u> [El <u>convocante</u> confecciona el <u>temario</u>] GENERAR EL LISTADO PARA CONVOCATORIA El <u>convocante</u> o la <u>secretaria</u> registran la <u>reunión</u> en el <u>cronograma de reuniones</u> El <u>convocante</u> o la <u>secretaria</u> reservan el <u>espacio físico</u> [El <u>convocante</u> o la <u>secretaria</u> reservan el <u>material físico</u>] [El <u>convocante</u> o la <u>secretaria</u> registran el <u>material físico</u> en el <u>cronograma de reuniones</u> #
EXCEPCIONES:	Conflictos en los <u>horarios disponibles</u> de los <u>convocados</u> Conflictos en la <u>disponibilidad de espacio</u> Conflictos en la <u>disponibilidad de material físico</u>

Figura 16 – Versión final del escenario DISEÑAR LA AGENDA DE REUNIONES

3.6.2 DESCRIBIR

En esta actividad se completan los escenarios candidatos agregando información del UdeD utilizando las heurísticas de descripción y tomando como base el modelo de escenario y los símbolos del LEL. El resultado es un conjunto de escenarios candidatos completamente descriptos. Esta etapa consta de cuatro actividades, como puede verse en la figura 17: *Completar componentes*, *Crear subescenarios*, *Completar restricciones* y *Completar Excepciones*.

A partir de la *Derivación*, se obtiene un conjunto incompleto de escenarios que debe ser completado en dos sentidos: se debe mejorar el contenido de los escenarios candidatos y se pueden agregar escenarios nuevos.

Para ello puede utilizarse, si está disponible, la información obtenida durante el proceso de construcción del LEL pero no incluida en él, y si es necesario, el ingeniero de requisitos puede retornar a las fuentes de información para obtener más datos y confirmar los existentes.

Como puede verse en la figura 13, existe un feedback entre las etapas de *Verificación* y *Validación*, representado por las listas DEO. Estas listas pueden originar cambios en la descripción del escenario. En la actividad *Describir* también puede generarse una lista DEO para el LEL.

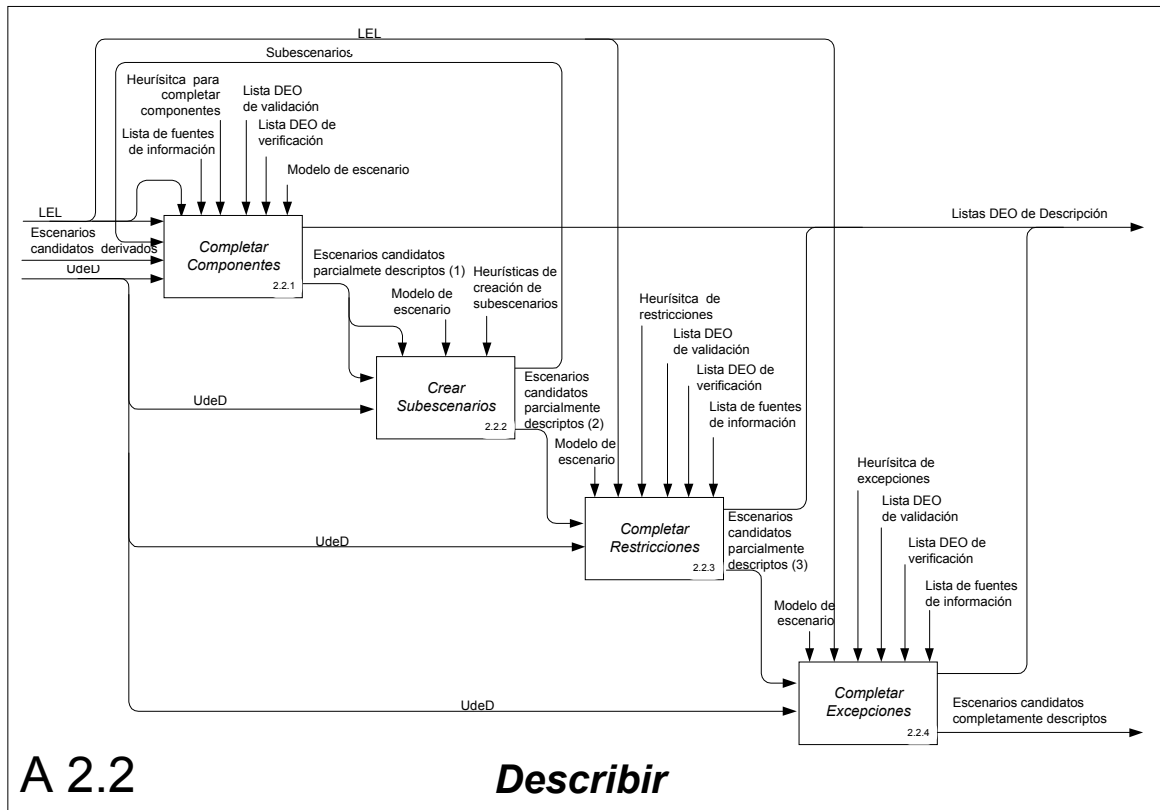


Figura 17 – SADT de la etapa Describir Escenarios

Completar componentes

Esta actividad generalmente se basa en entrevistas estructuradas, observaciones y lectura de documentos. La información obtenida permite confirmar y mejorar el curso normal de eventos del escenario. Puesto que algunos escenarios pueden estar parcialmente descriptos en este punto, en esta etapa deberían revisarse las descripciones iniciales.

Después de terminar de describir el conjunto de escenarios, los episodios deben ser revisados cuidadosamente para confirmar el orden secuencial o detectar paralelismos, y para encontrar episodios opcionales.

Crear subescenarios

Los subescenarios son una posible solución al problema de la explosión de escenarios señalada en [Cockburn95]. Se utilizan cuando:

- se detecta comportamiento común en varios escenarios,
- aparecen cursos de acción condicionales o alternativos complejos en un escenario, o
- se detecta en un escenario la necesidad de mejorar una situación con un objetivo concreto y preciso

Completar restricciones

Las restricciones se utilizan para caracterizar requisitos no funcionales aplicados a Contexto, Recursos y Episodios. Algunas pueden ser elicitadas desde el UdeD y otras pueden surgir examinando los episodios.

Completar excepciones

Finalmente, se deben detectar los casos alternativos y excepciones. Algunas causas de excepción son elicitadas desde las fuentes de información mientras que otras pueden deducirse analizando los episodios y la no disponibilidad o malfuncionamiento de los recursos. Cuando se descubren las causas de una excepción, los ingenieros de requisitos deberían investigar cómo es tratada la excepción en el UdeD, surgiendo así una nueva situación que podría necesitar ser descrita a través de un escenario separado.

3.6.3 ORGANIZAR

Esta actividad es la más compleja y sistematizada del proceso de construcción de escenarios. Se basa en la idea de *Escenarios Integradores*, descripciones “artificiales” cuyo único objetivo es lograr que el conjunto de escenarios sea más manejable y comprensible. Los escenarios integradores dan una visión global de la aplicación.

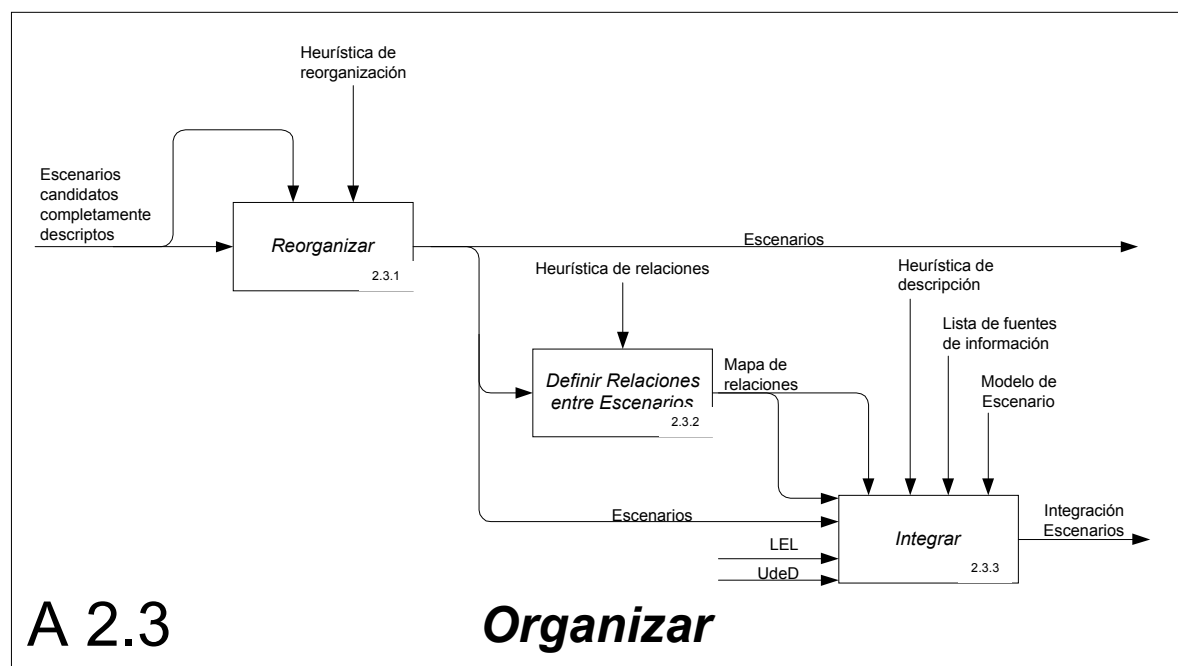


Figura 18 – SADT de la etapa Organizar escenarios

Cuando las aplicaciones son grandes, el número de escenarios puede ser inmanejable y el ingeniero de requisitos puede verse inmerso en detalles, perdiendo la visión global de la aplicación. Para enfrentar este problema, se propone la construcción de escenarios

integradores, que dan una visión global de la relación existente entre varias situaciones, puesto que cada episodio de un escenario integrador corresponde a un escenario. Por lo tanto, el propósito principal de los escenarios integradores es vincular escenarios dispersos, proveyendo una visión global de la aplicación, preservando siempre el formato de lenguaje natural utilizado para la descripción de los escenarios.

Así que, esta etapa comienza con los escenarios completamente descriptos, y los organiza mediante la reorganización y la integración, como puede verse en la figura 18.

Si bien la actividad *Organizar* está ubicada en tercer lugar en la figura 13, las actividades *Verificar* y *Validar* deben ocurrir antes de iniciar la organización de los escenarios completamente descriptos. En otras palabras, esta es una tarea que se lleva a cabo sólo cuando se confía en tener un conjunto de escenarios bien definidos. Por ello, lo ideal es comenzar con las listas DEO vacías. No obstante, los escenarios completamente descriptos pueden tener aún algunas debilidades importantes, como:

- falta de homogeneidad,
- problemas semánticos menores, y
- falta de perspectiva global

Estos problemas pueden ser minimizados al aplicar las heurísticas de organización que se detallan más adelante. En esta etapa se utilizan, además, el LEL, el modelo de escenarios y es posible que sea necesario volver al UdeD.

Las heurísticas que se aplican en esta etapa contienen taxonomías de operaciones y relaciones, que se describen a continuación en cada una de las etapas de la actividad *Organizar*.

Reorganizar

Los escenarios pueden tener diferente grado de detalle o solapamiento de información, especialmente cuando son construidos por más de un ingeniero de requisitos. Esta situación se complica más a medida que se manejan más escenarios y el equipo de trabajo es más numeroso. Entonces, las heurísticas de reorganización, aplicadas a escenarios completamente descriptos, tienen por objeto obtener escenarios consistentes y homogéneos.

Básicamente la reorganización consiste en juntar dos o más escenarios en uno, o dividir un escenario en uno o más. La primera situación se da cuando un único escenario ha sido artificialmente dividido durante las etapas *Derivar* o *Describir*. La necesidad de partir el escenario surge cuando éste contiene más de una situación.

Para establecer la necesidad de operaciones de composición/descomposición, algunas propiedades y relaciones entre escenarios deberían ser previamente identificadas. Cuando un escenario presenta propiedades especiales, principalmente en sus episodios, se puede aplicar la descomposición. La composición, mientras tanto, puede ser usada cuando se descubren relaciones específicas entre escenarios: superposiciones, orden de precedencia o relaciones de jerarquía.

Debe notarse que tanto la operación de composición como la de descomposición involucra el riesgo de degradación semántica. La descomposición crea dos o más escenarios a partir de uno. Este procedimiento no debe crear situaciones “artificiales”; en otras

palabras, la descomposición debe ser aplicada sólo cuando los escenarios resultantes van a describir mejor al UdeD. La composición, por el contrario, reemplaza dos o más escenarios por uno. Esto parece implicar que una situación del UdeD es eliminada. Sin embargo, esto sólo puede ocurrir cuando uno de los escenarios describe algo que no es una situación real, o cuando una única situación real del UdeD fue descripta por dos escenarios.

Las operaciones de reorganización son sugeridas como guías para mejorar la comprensión y administración de los escenarios. Por lo tanto, esta etapa no debería verse como una actividad obligatoria, sino como una buena práctica.

A continuación se describen las diferentes operaciones de reorganización:

- **Empotrar subescenarios:** Puede ser aplicada cuando se detectan subescenarios no relevantes con pocas ocurrencias en otros escenarios. Esta operación incorpora los episodios del subescenario en cada escenario que lo menciona. El subescenario original se borra cuando se han empotrado todas sus ocurrencias.
- **Agregar subescenarios:** Puede ser aplicada cuando se detecta un conjunto de episodios muy relevantes o un conjunto de episodios con diferente nivel de detalle en relación con el resto. Se crea un subescenario que contiene episodios de uno o más escenarios, y el grupo de episodios es reemplazado por el título del subescenario que los contiene.
- **Intercalar escenarios:** Dos o más escenarios están fuertemente superpuestos si sus objetivos y contextos son similares y tienen muchas coincidencias en los episodios. Esta operación copia los episodios comunes de los escenarios originales a un nuevo escenario y crea nuevos episodios condicionales usando los episodios no compartidos, siendo la condición la parte correspondiente de la precondición del escenario original. Los escenarios originales son eliminados.
- **Dividir escenarios:** Puede aplicarse cuando se detecta la presencia de varios episodios condicionales con la misma condición. Dividir produce dos nuevos escenarios que tendrán a la condición disparadora como precondición. Los episodios disparados son movidos al escenario correspondiente sin la condición, en tanto que los episodios no disparados por la condición son copiados a ambos escenarios. El escenario original es eliminado.
- **Fusionar escenarios:** Puede ser aplicada a escenarios que presentan un orden de precedencia contigua o secuencial, objetivos complementarios y contextos acoplados. No puede existir una brecha temporal entre los escenarios. Esta operación copia los episodios de cada uno de los escenarios originales al nuevo escenario en el orden correspondiente. Los escenarios originales son eliminados.
- **Partir escenarios:** Puede ser aplicada a un escenario si hay una brecha temporal entre los episodios o cuando se detecta un contexto temporal muy largo. Esta operación copia todos los episodios que preceden a la brecha temporal a un nuevo escenario, y aquéllos que le siguen a otro nuevo escenario. Luego, se elimina el escenario original. Las precondiciones del segundo escenario pueden reflejar el orden de precedencia secuencial respecto del primero.

A continuación se presentan heurísticas que ayudan en la etapa *Reorganizar Escenarios*:

- Después de aplicar operaciones de composición o descomposición, cada nuevo escenario debe representar una situación del UdeD
- La descomposición no debe ser aplicada cuando el objetivo no puede ser desacoplado
- Cuando los objetivos son diferentes y sólo pueden ser expresados juntos usando conjunciones, las operaciones de *Intercalar* y *Fusionar* no son recomendadas.
- Cuando las ubicaciones geográficas y temporales podrían ser fragmentadas, se recomienda la operación *Partir Escenarios*

Puede verse que para aplicar las operaciones de composición se requiere la existencia de relaciones específicas que luego desaparecerán entre los escenarios. Las operaciones de descomposición, mientras tanto, generan relaciones entre los escenarios resultantes del mismo tipo que las eliminadas por la composición. El conjunto de relaciones resultante se usará en la siguiente etapa.

Definir relaciones entre escenarios

Varios autores han propuesto relaciones entre escenarios, tales como las asociaciones y asociaciones extendidas de [Jacobson94b], conexiones semánticas de [Booch94], vínculos temporales de [Dano97], y otros. Estas relaciones fueron identificadas y utilizadas con otros propósitos. En este caso, el objetivo de las relaciones a definir es la construcción de escenarios integradores, así que se necesita un nuevo conjunto de relaciones. En esta etapa se identifican diferentes relaciones entre escenarios con el fin de integrarlos.

A continuación se presentan los tipos de relaciones identificados en [Leite00]:

- **Relación jerárquica:** El comportamiento de un escenario se describe a través de un conjunto de episodios, que podrían ser acciones simples o subescenarios. Un escenario puede contener más de un subescenario o ninguno. Un subescenario puede estar incluido en uno o más escenarios y puede a su vez, contener subescenarios. Así, una jerarquía puede definirse como el conjunto compuesto por un escenario y sus subescenarios.
- **Relación de superposición:** Se establece entre escenarios con porciones comunes. Esta relación se observa principalmente cuando varios episodios comunes se presentan en diferentes escenarios.
- **Relación de orden:** Se establece entre dos jerarquías de escenarios cuando una precede a otra. Una jerarquía puede preceder a otras jerarquías estableciendo un orden temporal parcial con ellas. Se establece una secuencia cuando una jerarquía es precedida inmediatamente por otra. Puesto que la segunda puede a su vez estar precedida por una tercera, un gran número de jerarquías puede estar involucrado en una secuencia comenzando al menos con una jerarquía inicial.
- **Relación a través de excepciones:** Se establece entre un escenario y aquellos escenarios que tratan sus excepciones. Cuando un escenario tiene una excepción, se describe su causa y si se especifica un tratamiento de la excepción, se menciona un escenario para tratarla. Un escenario puede estar relacionado con uno o más escenarios que resuelven excepciones. Un escenario que resuelve excepciones puede tratar excepciones que ocurren en diferentes escenarios.

Integrar

Esta actividad implementa el comportamiento “middle out” del proceso de construcción de escenarios [Hadad99], siendo los tres primeros pasos bottom-up y los últimos dos top-down. La integración de escenarios como tal podría ser vista como un procedimiento de cinco pasos:

- Construir jerarquías de escenarios
- Detectar orden parcial entre jerarquías
- Construir secuencia de jerarquías
- Construir el esqueleto de la integración
- Proponer Título, Objetivo y Contexto para los escenarios integradores

. Los cuatro pasos iniciales se basan principalmente en información de sintaxis, mientras el último requiere más conocimiento semántico. En esta actividad los escenarios son agrupados en jerarquías que luego son agrupadas en secuencias. Finalmente, estas secuencias se usan para construir escenarios integradores. La figura 19 muestra un ejemplo de un escenario integrador producido para el caso de estudio “Agenda de Reuniones”. Nótese que puesto que un escenario integrador es sólo producido para organizar un conjunto de escenarios, no usa las entidades recursos y actores.

TÍTULO:	Administrar la <u>agenda de reuniones</u>
OBJETIVO:	Administrar la agenda de reuniones eficiente y efectivamente
CONTEXTO:	Debe existir un asunto a tratar o resolver por dos o más personas
ACTORES:	-
RECURSOS:	-
EPISODIOS:	<p>REQUERIR UNA REUNIÓN</p> <p>DISEÑAR LA AGENDA DE REUNIONES</p> <p># ORGANIZAR LA REUNIÓN</p> <p>SI el <u>convocante</u> o los principales <u>convocados</u> no pueden asistir a la <u>reunión</u> ENTONCES ANULAR LA <u>REUNIÓN</u></p> <p>SI la <u>fecha de la reunión</u> debe ser cambiada ENTONCES TRASLADAR LA <u>FECHA DE LA REUNIÓN</u></p> <p>SI surge la necesidad de cambios en los requerimientos de la <u>reunión</u> ENTONCES CAMBIAR LOS REQUERIMIENTOS DE LA REUNIÓN #</p> <p>SI la <u>reunión</u> no fue cancelada ENTONCES ASISTIR A LA <u>REUNIÓN</u></p>

Figura 19 – Ejemplo de un Escenario Integrador

Verificar

Esta actividad es llevada a cabo al menos dos veces durante el proceso de construcción de escenarios; la primera sobre el conjunto de escenarios completamente descriptos y la segunda luego de la actividad *Organizar*. Esta etapa se realiza siguiendo un checklist con heurísticas de verificación. Como resultado, se producen dos listas DEO, una utilizada en la actividad *Describir*, y la otra en el proceso de construcción del LEL. La verificación se realiza en tres niveles: intraescenarios, interesenarios y contra el LEL.

La verificación intraescenarios incluye:

➤ ***Verificar sintaxis***

- chequear la completitud de cada componente
- chequear la existencia de más de un episodio por escenario
- chequear la sintaxis de cada componente según lo establecido en el modelo de escenario

➤ ***Verificar relaciones entre componentes:***

- chequear que cada actor participe en al menos un episodio
- chequear que cada actor mencionado en los episodios esté incluido en el componente Actores
- chequear que cada recurso sea usado en al menos un episodio
- chequear que cada recurso mencionado en los episodios esté incluido en el componente Recursos

➤ ***Verificar semántica:***

- chequear coherencia entre el Título y el Objetivo
- asegurar que el conjunto de episodios satisface el objetivo y está dentro del contexto
- asegurar que las acciones presentes en las precondiciones ya han sido realizadas
- asegurar que los episodios sólo contienen acciones a ser realizadas
- asegurar el mismo nivel de detalle dentro del escenario

La verificación interesenarios incluye:

➤ ***Verificar existencia de subescenarios:***

- chequear que cada episodio identificado como un subescenario existe dentro del conjunto de escenarios
- chequear que el conjunto de episodios de cada subescenario no haya sido incluido en otro escenarios
- chequear que cada excepción sea tratada por un escenario

➤ ***Verificar contexto:***

- chequear que cada precondición es un hecho incontrolable o es satisfecha por otro escenarios
- chequear la coherencia entre las precondiciones de los subescenarios y las precondiciones de los escenarios
- chequear que las ubicaciones geográfica y temporal de los subescenarios son iguales o más restringidas que las de los escenarios

➤ ***Verificar equivalencia en el conjunto de escenarios:***

- chequear que las coincidencias de objetivo sólo se den en situaciones diferentes
- chequear que las coincidencias de episodios sólo se den en situaciones diferentes

- chequear que las coincidencias de contexto sólo se den en situaciones diferentes

La verificación contra el LEL incluye:

- chequear que cada símbolo del léxico esté identificado
- chequear el uso correcto de los símbolos del léxico
- chequear que los actores sean preferentemente símbolos del tipo Sujeto
- chequear que los recursos sean preferentemente símbolos del tipo Objeto
- chequear que los impactos de los símbolos del tipo Sujeto estén cubiertos por los escenarios

Al usar las listas DEO de verificación, se modifican los escenarios y el LEL. Si se necesitan correcciones importantes, podría requerirse una nueva verificación. Cuando un ciclo *Describir-Verificar* ha finalizado, la actividad *Organizar* debería proveer un conjunto más grande de escenarios, que a su tiempo serán verificados iniciando un posible ciclo *Describir-Organizar-Verificar*. Esta actividad es efectuada actualmente por medio de inspecciones [Doorn98].

Es importante resaltar que la tarea de *Verificación* es manejada por chequeos de sintaxis, referencias cruzadas entre el LEL y los escenarios y por un checklist. Algunas de las heurísticas de este checklist ya fueron presentadas. En el checklist se va registrando lo que se aprende en términos de verificación al utilizar el modelo de escenarios.

Validar

Los escenarios son validados con los clientes/usuarios efectuando entrevistas estructuradas o reuniones. Durante la validación, debe prestarse especial atención al componente Dudas para aquellos escenarios que lo incluyen.

Es importante destacar que aunque se use una descripción estructurada, los escenarios son escritos en lenguaje natural, empleando el propio vocabulario de los clientes/usuarios y describiendo una situación específica bien delimitada.

La tarea de *Validación* permite la detección de DEO en los escenarios. Los errores son principalmente detectados al leer cada escenario al cliente/usuario. Algunas omisiones pueden aparecer también durante esa lectura, y otras al cuestionar la falta de información o detalles en los escenarios. Las discrepancias pueden aparecer durante las entrevistas pero surgen generalmente más adelante, cuando se analiza la información recolectada.

Debe notarse que el objetivo principal de la *Validación* es confirmar la información elicitada y detectar DEO. Sin embargo, un efecto lateral de esta etapa es la elicitación de nueva información. La validación de un conjunto de escenarios, después de un proceso de verificación, debería confirmar que las situaciones del UdeD han sido reportadas de acuerdo con la percepción de los actores del UdeD (clientes/usuarios) a cargo de la lectura y discusión de los escenarios.

CAPÍTULO 4

ASPECTOS RECURRENTE EN LA CONSTRUCCIÓN DE ESCENARIOS

En el capítulo anterior se describieron los escenarios como un medio de asegurar la comunicación entre los ingenieros de requisitos y los clientes/usuarios, puesto que permiten mantener mucha información en una forma que los involucrados pueden reconocer, proveyendo así un atractivo medio de comunicación entre ellos.

La falta de precisión acerca de cuándo y cómo los escenarios deberían ser usados ha sido extendida a los ingenieros que están usando estas técnicas en la práctica. La mayoría de los desarrolladores ven la creación de escenarios más como una tarea artesanal que como un trabajo de ingeniería. Estudios recientes respecto al uso real de escenarios en la Ingeniería de Requisitos [Weidenhaupt98] han probado claramente este hecho, apuntando a la necesidad de definiciones más detalladas acerca de la construcción de escenarios como una contribución inevitable para incrementar su uso en situaciones reales.

En [Cockburn95] se presentan guías para estructurar casos de uso y escenarios: “la gente en un proyecto está confundida acerca de cómo escribirlos”. Por ello [Schneider98] también propone un modelo para describir casos de uso y escenarios. [Jarke98] también señala la necesidad de guías para escribir escenarios estructurados.

Como una posible respuesta a estas inquietudes, en este trabajo se propone una modificación a la heurística de construcción de escenarios incorporando patrones que brindarán orientación en el proceso de construcción y además ofrecerán una fuente de confirmación del escenario en desarrollo [Ridao01a] [Ridao01b].

La idea central del uso de patrones en la construcción de escenarios consiste en aportar la experiencia acerca de situaciones similares acumulada en los patrones a las técnicas existentes. Mediante un análisis más estructurado de las situaciones, se propone que se detecten características recurrentes que permitan identificar a la situación, completa o parcialmente, con uno o más patrones de un catálogo. En este capítulo se estudian los aspectos básicos del proceso de construcción de escenarios que son susceptibles de ser considerados ladrillos básicos para la definición de patrones [Ridao00a].

4.1 REGULARIDADES

Se pretende representar mediante patrones, técnicas de creación de escenarios que han llevado a conjuntos de escenarios de alta calidad, y así, poder reusar sus estructuras en la construcción de nuevos escenarios que representen en forma consistente el dominio de una aplicación. Para ello, se han analizado varios conjuntos de escenarios con el objeto de

detectar características comunes, y así encontrar candidatos a patrones que luego ayudaran en el proceso de creación de nuevos escenarios. Se pretende tener una estructura básica, que pueda ser reusada, de modo de facilitar la construcción.

Se han analizado los siguientes casos de estudio, resultado de proyectos de investigación realizados en diferentes Universidades e Institutos de Investigación:

Sistema Nacional para la Emisión de Pasaportes [Leite96] [Leite97a]: desarrollado en la Universidad de Belgrano.

Sistema de Agenda de Reuniones [Hadad98]: desarrollado en la Universidad de Belgrano, basándose en el caso de estudio propuesto por van Lamsweerde et al. [van Lamsweerde93].

Sistema de Plan de Ahorro para la Adquisición de Automóviles [Mauco97] [Rivero98]: desarrollado en ISISTAN, Universidad Nacional del Centro de la Pcia. de Buenos Aires.

Sistema para el Control de Pos Graduación [Breitman98]: desarrollado en la Pontificia Universidade Católica do Rio de Janeiro para el Dto. de Informática de PUC-Río.

Sistema de Biblioteca: desarrollado en la Pontificia Universidade Católica do Rio de Janeiro.

Sistema Editor de Texto: desarrollado en la Pontificia Universidade Católica do Rio de Janeiro.

Uso recursivo de las heurísticas para la construcción de LEL y Escenarios del proceso de construcción de LEL y Escenarios: llevado a cabo en el ISISTAN, Universidad Nacional del Centro de la Provincia de Buenos Aires.

A partir del análisis de los episodios de los casos de estudio se pudo determinar la existencia de diferentes tipos de situaciones con características bien definidas. Estas características no sólo se refieren a los episodios sino también a los otros elementos del escenario. Esto se debe a que los episodios, por ser los que describen casi completamente la situación, permiten derivar la naturaleza del resto de los elementos, como por ejemplo el tipo y número de actores, el número de recursos, etc.

4.2 CLASIFICACIÓN DE EPISODIOS

Debido a que el componente Episodios es el lugar en que se encuentran más aspectos recurrentes en un escenario, es apropiado comenzar el análisis en este componente. Los primeros aspectos a considerar son el número de actores que intervienen en el episodio, si éste requiere o no respuesta, si dicha respuesta es inmediata o diferida, y la participación o rol de los actores en el episodio.

Según el número de actores involucrados, los episodios pueden ser:

- **x** (no-arios): Son aquéllos en los que no participa ningún actor
- **u** (unarios): Son aquéllos que sólo involucran a un actor

- **n** (n-arios)²: Son aquéllos que involucran a más de un actor, en forma explícita o implícita

Los episodios tipo **x** aparecen muy raramente y representan sucesos asociados con aspectos impersonales tales como fenómenos climáticos, transcurso del tiempo, etc.

Por otra parte, en muchos casos los episodios se vinculan entre sí a través de una relación del tipo solicitud-respuesta. En algunas circunstancias la respuesta es inmediata, y aparece en el mismo escenario, mientras que en otras, esa respuesta se encuentra en un escenario diferente. Esta observación lleva a una nueva clasificación de los episodios.

Según la respuesta, los episodios pueden ser:

- **sr** (sin respuesta): Son aquéllos en los que participa más de un actor y que no requieren otro episodio como respuesta
- **ri** (con respuesta inmediata): Son aquéllos en los que la acción de uno de los actores requiere una respuesta inmediata de otro
- **rd** (con respuesta diferida): Son aquéllos en los que la acción de uno de los actores requiere una respuesta de otro que puede o debe demorarse

Los actores interactúan de diferentes maneras en los episodios. En algunos casos, un solo actor lleva a cabo una actividad; en otros, los actores colaboran entre sí para cumplir con un objetivo común; y en otros se produce una interacción que hace que los episodios se vinculen entre sí a través de solicitudes y respuestas.

Según la participación de los actores, los episodios pueden ser:

- **p** (producción): Son aquéllos en los que un único actor, en forma autónoma, realiza un cambio en el macrosistema
- **s** (servicio): Son aquéllos en los que uno de los actores adquiere el rol de actor activo y realiza una acción en beneficio de uno o más actores pasivos
- **c** (colaboración): Son aquéllos en los que dos o más actores realizan una acción que requiere de la participación cooperativa de todos ellos, produciendo un efecto global sobre el macrosistema
- **d** (demanda): Son aquéllos en los que uno de los actores desempeña un rol activo y uno o más cumplen roles pasivos. El actor activo realiza una acción que tiene implícita la exigencia de una respuesta por parte de el o los actores pasivos
- **r** (respuesta): Son aquéllos en los que un actor, que fue pasivo en un episodio de tipo **d** asume el rol activo y satisface el mencionado pedido
- **i** (interacción): Son aquéllos que reúnen las propiedades de respuesta y demanda. Es decir que satisfacen un pedido previo y generan uno nuevo

Los episodios **d**, **r** e **i** suelen agruparse en secuencias **d i ... i r** que constituyen una suerte de “negociación” en el sentido amplio de la palabra. Un pedido inicial **d** de un actor provoca una respuesta **i** de otro que incluye una nueva demanda. Esto se repite varias veces hasta la aparición de un episodio tipo **r**. Estas secuencias pueden quedar inconclusas cuando falta el episodio **r**.

Desde el punto de vista del tipo de la respuesta es posible ver las mismas secuencias como: **ri . . . ri sr**, o **ri . . . ri rd**. En la primera el episodio **sr** corresponde al tipo **r**

² La gran mayoría de los episodios n-arios son en realidad binarios, por lo que de aquí en adelante se los referirá como binarios (**b**), excepto cuando se deba destacar que puede haber más de dos actores

(respuesta) de la última clasificación y en la segunda el episodio **rd** deja pendiente la negociación.

En la figura 20 se presenta un fragmento de los episodios del Escenario SACAR FOTOGRAFÍA del caso de estudio Sistema Nacional para la Emisión de Pasaportes, donde pueden verse ejemplos de alguno de los tipos de episodios.

El primer episodio corresponde al tipo d. Los episodios 2 al 4 son de tipo i, el 5 es una p y el 6, una r. Si se analiza la respuesta de los episodios del ejemplo, los 4 primeros corresponden al tipo ri, y el último a sr.

EPISODIOS:	
...	
	El <u>solicitante</u> llega con su <u>formulario de solicitud</u> lleno a la <u>Cabina de Fotografía</u>
	El <u>solicitante</u> entrega el <u>formulario</u> al empleado de <u>caja</u>
	El empleado de <u>caja</u> informa el importe
	El <u>solicitante</u> paga
	El empleado de <u>caja</u> sella el <u>formulario</u> con el número de fotografía policial
	El empleado devuelve el <u>formulario</u> al <u>solicitante</u>
...	

Figura 20 – Ejemplo de Episodios de un Escenario

Las clasificaciones presentadas no son completamente ortogonales ya que existe un cierto grado de acoplamiento entre los criterios utilizados. En la Tabla 1 se presentan las combinaciones posibles entre el tipo de respuesta y la participación de los actores, y en la Tabla 2, la relación entre el tipo de respuesta y la cardinalidad.

	p	s	c	d	r	i
sr	Si	Si	Si	-	Si	-
ri	-	-	-	Si	-	Si
rd	-	-	-	Si	-	Si

Tabla 1: Combinaciones posibles entre tipo de respuesta y la participación de los actores

	sr	ri	rd
u	Si	-	-
b	Si	Si	Si

Tabla 2: Combinaciones posibles entre tipo de respuesta y cardinalidad

La tabla 1 pone de manifiesto que la necesidad de una respuesta, inmediata o diferida, existe si y sólo si se trata de episodios **d** o **i**. En la tabla 2 se indica que la necesidad de una respuesta implica la existencia de dos actores, pero no a la inversa.

4.3 CLASIFICACIÓN DE SUBESCENARIOS

Un subescenario, al ser mencionado como episodio de otro escenario, se comporta de manera similar a un episodio simple en cuanto a las clasificaciones de la sección anterior.

Por ello, en esta sección se utilizan, para los subescenarios, los mismos criterios de clasificación utilizados para episodios, y los mismos códigos, excepto que se escriben con mayúsculas para indicar su condición de subescenarios.

A continuación, se indica como puede determinarse la clase de un subescenario según las características de sus propios componentes.

Los subescenarios son generalmente n-arios, excepcionalmente pueden ser unarios, y no se han encontrado ejemplos de subescenarios sin actores.

Según la respuesta, los subescenarios unarios (**U**) siempre se verán como episodios sin respuesta (**SR**) tal como se puede observar en la Tabla 2. Los casos de subescenarios Binarios o N-arios, en cambio, pueden tener varias conductas con respecto a la respuesta. A continuación, se presenta un análisis más detallado, utilizando una extensión de las expresiones regulares, donde * representa 0 o más ocurrencias del elemento precedente, + representa 1 o más ocurrencias del elemento precedente, | representa un or entre el elemento que lo precede y el que le sigue, # ... # representa un bloque no secuencial, y <nombre>: es la definición de un símbolo no terminal:

BSR (Binario Sin Respuesta): Puede tratarse de un subescenario que contiene sólo episodios binarios sin respuesta o bien un subescenario que contiene una negociación terminada por un episodio binario sin respuesta.

<ubs>: (bsr* u+ bsr+ u* | u* bsr+ u+ | bsr+ bsr u* | (u* bri+ u*)+ bsr u*)+

BSR: <ubs> | (<ubs>* # <ubs> # <ubs>*)+

BRD (Binario con respuesta diferida): Un subescenario que contiene una negociación que queda inconclusa debido a la presencia de un episodio binario con respuesta diferida, que sigue siendo evaluada como diferida en el contexto del escenario que lo menciona.

<sec>: (u | bsr | (bri+ u*)+ bsr*)+

<uno>: (u+ | bsr+ | bri+)

<pari>: # <uno>+ <uno> #

<pard>: # (<uno>+ brd <uno>* | <uno>* brd <uno>+)+ # |

(brd* <uno>* brd+ <uno>* brd+ <uno>* brd*)+

BRD: (<sec> | <pari>)+ brd | (<sec> | <pari>)* <pard>

BRI (Binario con respuesta inmediata): Un subescenario que contiene una negociación que queda inconclusa debido a la presencia de un episodio binario con respuesta diferida, que es satisfecha en el contexto del escenario que lo menciona.

BRI: idéntica composición que **BRD**

En los tres casos, es posible que aparezcan episodios **u** (unarios) en los que alguno de los actores realiza alguna actividad típicamente de producción que contribuye a la negociación. Los episodios pueden presentarse en forma secuencial, o en bloques no secuenciales.

En algunos casos, el tipo de subescenarios desde el punto de vista de la participación se puede derivar de la clasificación de sus episodios según el mismo criterio. En otros, en cambio, es necesario recurrir al objetivo del subescenario para detectar el “espíritu” del

mismo. A continuación se detallan los casos más evidentes en los que no es necesario estudiar el objetivo:

P (Producción): Un subescenario que contiene sólo episodios de producción que pueden estar en secuencia y/o en bloques no secuenciales

P: $p+ p | (p^* \# p+ p \# p^*)+$

S (Servicio): Un subescenario que contiene sólo episodios de servicio, que pueden estar en secuencia y/o en bloques no secuenciales

S: $s+ s | (s^* \# s+ s \# s^*)+$

C (Colaboración): Un subescenario que contiene sólo episodios de colaboración, que pueden estar en secuencia y/o en bloques no secuenciales

C: $c+ c | (c^* \# c+ c \# c^*)+$

D (Demanda): Un subescenario que contiene un episodio **d** seguido por uno o más episodios **i** que pueden estar en secuencia y/o en bloques no secuenciales

D: $d i+ | d (i^* \# i+ i \# i^*)+$

R (Respuesta): Un subescenario que comienza con uno o más episodios **i** y termina con uno de tipo **r**. Los episodios **i** pueden estar en secuencia y/o en bloques no secuenciales

R: $i+ r | (i^* \# i+ i \# i^*)+ r$

I (Interacción): Un subescenario que comienza con un episodio **i** y termina con uno de tipo **i**

I: $i+ i | (i^* \# i+ i \# i^*)+$

4.4 CLASIFICACIÓN DE ESCENARIOS

A los efectos de visualizar la existencia de posibles patrones de escenarios es importante enfatizar el hecho que un escenario describe una situación y por lo tanto un patrón de escenarios debe poder ser visto como un patrón de situaciones. Es así que resulta mucho más atractiva la creación de patrones basándose en criterios de clasificación con mayor contenido semántico que faciliten la comprensión de la estructura interna del escenario.

Los términos Producción, Servicio y Colaboración describen apropiadamente tanto el impacto de un subescenario sobre los episodios siguientes, como la estructura interna de un escenario. Esto no ocurre en el caso de la Demanda, Respuesta e Interacción, porque estos términos enmascaran la existencia de varios actores realizando acciones coordinadas cada uno con su propio objetivo y la cantidad de situaciones que se generan como consecuencia del escenario bajo análisis.

Así, se definen los siguientes patrones:

Producción: Describe la realización de una actividad productiva que provocará un efecto sobre el macrosistema.

Servicio: Describe la prestación de un servicio que es necesario para uno de los actores.

Colaboración: Describe la asociación de varios actores para realizar una actividad cooperativa con el fin de lograr un objetivo común.

Negociación Inconclusa: Describe la iniciación de una actividad que requiere una secuencia coordinada de acciones por parte de los actores. Este tipo de escenario requiere que otra situación termine con la negociación.

Negociación Inconclusa con Disparador de Escenarios: Describe la iniciación de una actividad que requiere una secuencia coordinada de acciones por parte de los actores y que crea la necesidad de varias situaciones futuras.

Fin de Negociación: Describe una secuencia coordinada de acciones por parte de los actores que da por finalizada una actividad iniciada en otro escenario.

Etapas de Negociación: Describe una secuencia coordinada de acciones por parte de los actores que continúa con una actividad iniciada en otra situación y cuya finalización queda inconclusa.

Etapas de Negociación con Disparador de Escenarios: Describe una secuencia coordinada de acciones por parte de los actores que continúa con una actividad iniciada en otra situación y cuya finalización deberá ser resuelta en varias situaciones futuras.

Negociación Terminada: Describe la realización completa de una actividad que requiere una secuencia coordinada de acciones por parte de los actores.

Se observa que existen situaciones donde se presentan combinaciones de diferentes tipos de episodios, dando lugar a otros tipos de escenarios. De esas combinaciones, surgen los siguientes patrones:

Producción y/o Servicio y/o Colaboración: Realización de una actividad productiva, combinada con servicios y actividades cooperativas.

Negociación inconclusa con producción y/o servicio y/o colaboración: Realización de una actividad centrada en transacciones que se combina con actividades de producción, servicio y/o colaboración. Este tipo de escenario requiere que otra situación finalice la negociación.

Fin de Negociación con producción y/o servicio y/o colaboración: Finalización de una actividad centrada en transacciones que se combina con actividades de producción, servicio, y/o colaboración.

Etapas de Negociación con producción y/o servicio y/o colaboración: Continuación de una actividad centrada en transacciones que se combina con actividades de producción, servicio y/o colaboración. La finalización de la actividad queda inconclusa.

Negociación terminada con producción y/o servicio y/o colaboración: Realización de una actividad centrada en transacciones que se combina con actividades de producción, servicio o colaboración.

Negociación inconclusa con Disparador de Escenarios y producción y/o servicio y/o colaboración: Ejecución inconclusa de una actividad basada en transacciones, combinada con producción, servicio y/o actividades cooperativas, que dispara situaciones futuras.

Etapas de Negociación con Disparador de Escenarios y producción y/o servicio y/o colaboración: Continuación de una actividad centrada en transacciones que se combina con

actividades de producción, servicio y/o colaboración. La finalización de la actividad deberá ser resuelta en varias situaciones futuras.

4.5 ESTADÍSTICAS

Como se indicó en la sección 4.1, los escenarios de diferentes casos de estudio fueron analizados en busca de recurrencias que permitieran encontrar patrones. Luego, los escenarios de algunos de los casos de estudio fueron estudiados buscando correspondencia con los patrones definidos. En el anexo A se presenta la lista de escenarios de cada uno de los casos analizados y el patrón correspondiente a cada escenario. Para cada caso, se indica el número de escenarios que pudo ser identificado con un patrón y el porcentaje de cubrimiento correspondiente.

En la tabla 3 se presenta un resumen de los datos obtenidos a partir de dicho análisis. En el análisis inicial, se obtuvieron porcentajes muy altos para conjuntos de escenarios muy buenos, pero esto no ocurrió con juegos de escenarios de baja calidad. Después de analizar los escenarios para encontrar patrones, se descubrieron muchos errores. Por ejemplo, en muchos casos, podía verse que un solo escenario abarcaba dos o más situaciones, dando lugar a, por lo menos, dos escenarios diferentes. Así es que, muchos de los escenarios fueron vueltos a escribir, obteniendo así un considerable crecimiento en el grado de cubrimiento. El promedio final de cubrimiento entre todos los casos de estudio es de aproximadamente 93%.

<i>Caso de estudio</i>	<i>(a)</i>	<i>(b)</i>	<i>(c)</i>	<i>(d)</i>	<i>(e)</i>
<i>Sistema Nacional para la Emisión de Pasaportes</i>	24	5	25	21	84%
<i>Sistema de Agenda de Reuniones</i>	14	0	14	13	93%
<i>Sistema de Plan de Ahorro</i>	22	14	38	38	100%
<i>Sistema para el Control de Pos Graduación</i>	8	4	12	11	91.7%
<i>Autoaplicación de la Metodología</i>	64	0	64	61	95.5%

- (a) Número inicial de escenarios
- (b) Número de escenarios reescritos
- (c) Número final de escenarios
- (d) Escenarios que se corresponden con un patrón
- (e) Cubrimiento

Tabla 3. Resultados estadísticos del análisis de los casos de estudio al aplicar patrones

Como puede observarse en la tabla, cuatro escenarios del primer caso, uno del segundo, uno del tercero y tres del quinto no pudieron ser cubiertos completamente por un patrón. Sin embargo, no se trataba de situaciones completamente ajenas a los patrones del catálogo. Pudo observarse que existía una relación de subconjunto entre cada escenario y un patrón [Breitman98]. Si se consideraran secuencias de patrones, la columna (e) de la tabla contendría un 100% de cubrimiento para todos los casos.

CAPÍTULO 5

PATRONES DE ESCENARIOS

La idea de consolidar la experticia de profesionales en un ámbito del conocimiento humano mediante patrones que registren problemas y soluciones típicas de su dominio, ha sido explotada ampliamente en muchas disciplinas, en particular, en el proceso de desarrollo de software, y ha producido importantes innovaciones. La explotación de este enfoque está lejos de ser completa o siquiera razonablemente difundida. La Ingeniería de Requisitos es una de las áreas donde se han realizado muy pocos intentos de patronización.

Los patrones permiten y han permitido en diferentes áreas del conocimiento humano reusar la esencia de la solución de un problema al enfrentar nuevos problemas similares. Es así que los patrones constituyen una especie de mecanismo de registro y concentración de experiencia. Uno de los problemas que trae aparejado el uso de patrones en la práctica, es la identificación del patrón más apropiado para el problema en cuestión.

En este trabajo se pretende analizar desde un punto de vista estructural las situaciones, con el fin de asociar la situación, completa o parcialmente, con uno o más patrones de un catálogo. Una vez determinado el patrón que más se aproxime a la situación real, se reusa su estructura con el fin de derivar el escenario más fácil y directamente. En las secciones siguientes se describen técnicas que permiten seleccionar el patrón más adecuado para describir una situación [Ridao00b].

5.1 PATRONES

Cuando los expertos trabajan con un problema en particular, es inusual que conciban una nueva solución completamente distinta de las existentes. A menudo, recurren a problemas similares que ya han resuelto, y reusan la esencia de su solución para resolver el nuevo problema [Buschmann96]. Tal experiencia es parte de lo que los hace expertos. Si fuera posible recordar los detalles de un problema previo y cómo se solucionó, se podría reusar esa experiencia cuando se presente un problema similar en lugar de redescubrir la solución.

En la vida diaria, permanentemente se reconocen situaciones recurrentes, que se utilizan una y otra vez. Esto sucede porque las acciones, características y relaciones que son notables se repiten. Sin embargo, cada una de esas situaciones no sólo contiene esas partes repetidas, sino también porciones que varían. Las partes recurrentes se destacan porque nuestras mentes están construidas de esa forma [Gabriel96].

Un patrón es una pieza de literatura que describe un problema de diseño y una solución general para el problema en un contexto particular. La importancia de utilizar patrones en la creación de sistemas complejos ha sido largamente reconocida en muchas disciplinas. En particular, Christopher Alexander y sus colegas fueron los primeros en proponer la idea de

usar un lenguaje de patrones en la construcción de edificios y ciudades.

Según Alexander, “Cada patrón describe un problema que ocurre una y otra vez, y describe la solución a ese problema, de tal manera que dicha solución pueda ser usada un millón de veces más, sin hacerlo necesariamente dos veces del mismo modo” [Alexander77].

Se pueden encontrar patrones en edificios, en el tráfico vehicular, en organizaciones de gente, y en el software. Alexander extrajo los patrones de las estructuras de edificios y ciudades de numerosas culturas. El sostenía que documentando esos patrones podría ayudar a la gente en la construcción de los edificios de sus comunidades. Los patrones son estructuras que han evolucionado a través de los años para satisfacer necesidades culturales. Son reglas, manejadas por principios, que sirven a necesidades humanas y sociales. Son también la documentación de esas estructuras. Sin embargo, la mayoría de la comunidad de patrones se ha alejado de la interpretación literal de Alexander. Su mayor legado a la comunidad es su visión y su valioso sistema, pero su visión se aleja bastante de la práctica de software [Coplien96].

En [Gamma95] y [Buschmann96] se enumeran las propiedades fundamentales de los patrones de diseño. A continuación se resumen las más significativas. Cada patrón:

- Está dirigido a un problema recurrente que surge en situaciones específicas, y presenta una solución para dicho problema
- Documenta experiencia existente y bien probada
- Provee un vocabulario común
- Es un medio de documentación de arquitecturas de software
- Soporta la construcción de software con propiedades definidas
- Ayuda a manejar la complejidad del software
- Puede ser independiente del dominio, o puede estar dirigido a aspectos de un dominio específico

Un patrón es “una descripción de una solución a un problema que ocurre en un contexto específico” [Meszaros99], pero esta definición es muy amplia. Lo que hace a los patrones diferentes es su habilidad para explicar el motivo para usar la solución (el “por qué”) además de describir la solución (el “cómo”).

5.2 PATRONES DE ESCENARIOS

La arquitectura de software es expresada en términos de lenguajes, librerías, y frameworks. Los patrones expresan cómo otros arquitectos han combinado exitosamente esos elementos. Prestando atención a los problemas recurrentes, los patrones pueden manejar la evolución de herramientas, procesos, lenguajes y frameworks, proveyendo artefactos reusables para atacar esos problemas. Así, es posible usar esos artefactos directamente, y enfocar la energía en aspectos más elevados [Foote98].

En el capítulo anterior se presentó el análisis de conjuntos de escenarios que permitieron detectar situaciones recurrentes, y así encontrar patrones que ayudarán en el

proceso de creación de nuevos escenarios.

La idea central de la construcción de escenarios utilizando patrones consiste en estudiar las situaciones desde un punto de vista diferente al utilizado por otras técnicas de elaboración de escenarios. Se pretende tener una visión más conceptual y estructural de las situaciones, con el fin de identificar la naturaleza intrínseca de las mismas. Con esa visión, será posible determinar el tipo de escenario correspondiente a cada situación y así, elegir un patrón de un catálogo, reusando su estructura con el fin de derivar el escenario más fácil y directamente.

Si bien existen heurísticas que permiten construir escenarios a partir de los términos del LEL, es posible que una definición incompleta o incorrecta de algún término lleve a la construcción de escenarios erróneos. Contar con patrones de escenarios, permitirá reconocer un patrón para cada situación y así se podrá refinar y corregir la definición inicial con el fin de describir más precisamente la realidad, y poder derivar un producto de software de mejor calidad.

Los patrones consisten en un texto guía, que para cada componente del escenario incluye pautas acerca del contenido que deberá tener dicho componente. Por ejemplo, para la Episodios, se da una descripción general del tipo de episodios, indicando cuántos episodios de cada tipo deben aparecer en el escenario y el orden en que deben escribirse.

5.3 CATÁLOGO

Los patrones son una forma literaria, tal como los sonetos o novelas o historias cortas. El formato del patrón cumple un propósito: introducir al lector en un problema, describir el contexto donde el problema podría surgir, analizar el problema, y presentar una solución. Existe una gran variedad de formatos de patrones. Si bien todos los formatos tienen su base en el de Alexander, algunos, como el GOF [Gamma95] y el de Coplien presentan una especie de plantilla que incluye diferentes secciones con sus títulos correspondientes, mientras que otros, como el de Portland son descriptos como narrativas [Coplien96].

En el Anexo B se presenta un catálogo con los patrones obtenidos a partir del análisis de recurrencias presentado en el capítulo 4. Los patrones son presentados como escenarios completamente descriptos a los que se ha agregado un reducido número de reglas de conformación como una suerte de metacomponentes del escenario. Cada componente del escenario, según la estructura definida en [Leite97a], es decir, Título, Objetivo, Contexto, Actores, Recursos, Episodios y Excepción, ha sido completado con un texto nominal que se aspira sea reemplazado en el escenario real generado al usar el patrón pero que a su vez guíe en la redacción del componente.

5.4 EJEMPLOS

En esta sección se presentan ejemplos de patrones del catálogo y escenarios que se identifican con cada uno de ellos.

En la figura 21 se presenta el patrón de Producción y en la figura 22 el escenario DISEÑAR LA AGENDA DE REUNIONES [Hadad98] correspondiente al caso de estudio

Sistema de Agenda de Reuniones.

PATRÓN: **PRODUCCIÓN**

TÍTULO: **Realización de una actividad productiva**

OBJETIVO: Producir un efecto sobre el macrosistema

CONTEXTO:
Ubicación geográfica: generalmente el lugar de trabajo del actor principal
Precondiciones: puede tener precondiciones
Ubicación temporal: generalmente determinado por el actor principal y posiblemente prolongado

ACTORES: Varios, al menos uno

RECURSOS: Al menos uno, generalmente muchos

EPISODIOS:
POR LO MENOS DOS COMO EL SIGUIENTE
Un actor realiza alguna actividad que produce algún efecto sobre el macrosistema.
PUEDEN ESTAR EN SECUENCIA O CONSTITUIR GRUPOS NO SECUENCIALES

EXCEPCIÓN: Circunstancia que obstaculiza el cumplimiento del objetivo

Figura 21 – Descripción del Patrón Producción

TÍTULO: **Diseñar la agenda de reuniones**

OBJETIVO: Determinar los requerimientos, oportunidad y lugar de la reunión

CONTEXTO: Debe presentarse previamente la necesidad de una reunión

ACTORES: convocante
secretaria

RECURSOS: agenda de reuniones
listado para convocatoria
horarios disponibles
temario

EPISODIOS:

1. [El convocante obtiene los datos de la reunión a diseñar del esquema de base]
2. **#SI** los horarios disponibles de los convocados no están registrados **ENTONCES SOLICITAR HORARIOS DISPONIBLES**
3. El convocante consulta en su agenda de reuniones sus horarios disponibles **#**
4. ESTABLECER LA FECHA DE REUNIÓN
5. [El convocante determina el material para repartir]
6. **#El convocante o la secretaria registran en la agenda el objetivo, la fecha, la hora, el lugar, los temas a discutir, los convocados, el material a presentary el material para repartir**
7. [El convocante confecciona el temario]
8. GENERAR EL LISTADO PARA CONVOCATORIA
9. El convocante o la secretaria registran la reunión en el cronograma de reuniones
10. El convocante o la secretaria reservan el espacio físico
11. [El convocante o la secretaria reservan el material físico]
12. [El convocante o la secretaria registran el material físico en el cronograma de reuniones **#**

EXCEPCIONES:
Conflictos en los horarios disponibles de los convocados
Conflictos en la disponibilidad de espacio
Conflictos en la disponibilidad de material físico

Figura 22 – Ejemplo de un escenario de tipo Producción

PATRÓN: NEGOCIACIÓN INCONCLUSA

TÍTULO:	Ejecución inconclusa de una actividad basada en transacciones
OBJETIVO:	Iniciar una actividad que requiere una secuencia coordinada de acciones por parte de los actores
CONTEXTO:	Ubicación geográfica: generalmente el lugar de trabajo del actor principal Precondiciones: puede tener precondiciones Ubicación temporal: generalmente determinado por el actor principal y posiblemente breve
ACTORES:	Varios, al menos dos
RECURSOS:	Pocos o ninguno
EPISODIOS:	<p>UNO COMO EL SIGUIENTE Un actor realiza una acción que requiere respuesta inmediata de otro actor.</p> <p>VARIOS O NINGUNO COMO EL SIGUIENTE Un actor realiza una acción que responde a una acción anterior y que a su vez requiere respuesta inmediata de otro actor.</p> <p>(DEBE ESTAR PRECEDIDO POR UNA ACCIÓN QUE REQUIERA RESPUESTA INMEDIATA)</p> <p>UNO COMO EL SIGUIENTE Un actor realiza una acción que responde a una acción anterior y que requiere una respuesta que puede o debe demorarse.</p> <p>(DEBE ESTAR PRECEDIDO POR UNA ACCIÓN QUE REQUIERA RESPUESTA INMEDIATA)</p> <p>PUEDEN ESTAR EN SECUENCIA O CONSTITUIR GRUPOS NO SECUENCIALES. SÓLO ES NECESARIO RESPETAR ORDEN DONDE ESTÁ EXPLÍCITAMENTE INDICADO.</p>
EXCEPCIÓN:	Circunstancia que obstaculiza el cumplimiento del objetivo

Figura 23 – Descripción del Patrón Negociación Inconclusa

TÍTULO:	Inicio del trámite de reválida telefónica
OBJETIVO:	Solicitar el trámite de reválida por teléfono
CONTEXTO:	Se efectúa en el Box de Reválidas
ACTORES:	<u>Solicitante</u> <u>Empleado de Box de Reválidas telefónicas</u>
RECURSOS:	Teléfono <u>Pasaporte viejo</u>
EPISODIOS:	<ol style="list-style-type: none"> 1. El <u>solicitante</u> llama al <u>Box de Reválidas</u> 2. El empleado pide número de <u>pasaporte</u> 3. El <u>solicitante</u> informa el número de <u>pasaporte</u> 4. El empleado toma nota de <u>Número de identificación</u> y le informa un código a <u>solicitante</u> que éste deberá indicar cuando se presente ante <u>Box de Reválidas telefónicas</u> 5. El empleado le informa a <u>solicitante</u> la fecha y hora en la que debe presentarse en <u>Box de Reválidas telefónicas</u>

Figura 24 – Ejemplo de un escenario de tipo Negociación Inconclusa

En cada uno de los ejemplos se puede observar que todos los componentes, además de los episodios, se ajustan a la descripción de los mismos en el patrón correspondiente.

Analizando los episodios del escenario de la figura 22 puede verse que se trata de

producciones. Los episodios 2, 4 y 8 corresponden a subescenarios, que fueron analizados en forma independiente y resultaron ser también casos de producción. Por ello es posible determinar que se trata de un ejemplo puro de Producción.

En la figura 23 se presenta el patrón de Negociación Inconclusa y en la figura 24 el escenario INICIO DEL TRÁMITE DE REVÁLIDA TELEFÓNICA correspondiente al Sistema Nacional de Emisión de Pasaportes.

En el ejemplo de la figura 24 se detecta claramente una negociación que comienza con un episodio de tipo demanda (1), que es seguido por una secuencia de episodios de tipo interacción (2 al 5). Es importante destacar que el episodio 2 no estaba en el escenario original. El patrón permitió detectar su ausencia y solucionar el inconveniente, que fue confirmado en el UdeD.

5.5 MECANISMO DE DERIVACIÓN DE ESCENARIOS USANDO PATRONES

Antes de ampliar y precisar el catálogo de patrones de escenarios es necesario corroborar que es viable su utilización en una situación real. Para ello se creó una heurística que permite asociar un patrón con una situación. Esta heurística ha sido probada en algunos de los casos de estudio disponibles [Leite00].

El proceso de construcción de escenarios sin el uso de patrones [Leite00] comienza partiendo desde el léxico del dominio de la aplicación, produciendo una primera versión de los escenarios derivados exclusivamente desde el LEL.

El primer paso en la etapa de *Derivación de escenarios candidatos* consiste en la identificación de los actores del UdeD. Posteriormente, se extraen del LEL los impactos de los símbolos elegidos como actores principales y secundarios. Cada impacto da lugar a un escenario que se incorpora a la lista de escenarios candidatos. El título del escenario está compuesto por el verbo (acción) incluido en el impacto expresado en infinitivo más un predicado tomado del impacto [Hadad97].

Posteriormente se procede a crear los escenarios candidatos, extrayendo tanta información del LEL como sea posible y aplicando las heurísticas de creación.

En esta sección se presenta una serie de preguntas que se aplicarán al impacto en cuestión, con el objeto de poder determinar el patrón correspondiente al escenario candidato y así poder reusar la estructura de ese patrón para la creación del mismo. En la figura 25 se presenta el árbol de decisión para la selección del patrón correspondiente.

Al aplicar el árbol de decisión de la figura 25 a cada uno de los impactos de los símbolos del LEL correspondientes a los actores, se determina un patrón de escenario cuya estructura es conocida. Esto permite describir cada uno de los componentes del escenario con mayor precisión.

Utilizando las heurísticas del proceso de construcción de escenarios sin el uso de patrones, el título se obtiene directamente a partir del impacto del símbolo del LEL. Al haber seleccionado un patrón es ahora posible contrastar el título generado con el esperable para ese patrón, permitiendo la realización de correcciones cuando sea necesario, con una guía sólida. Luego, para cada impacto que representa un posible escenario, se verifica si contiene un símbolo del LEL que pertenezca a la clasificación Verbo. En caso afirmativo,

se define el objetivo del escenario en base al nombre del escenario, el punto de vista de la aplicación y la noción de este símbolo que representa una actividad. Nuevamente está disponible el patrón para dar marco al objetivo obtenido y eventualmente evitar desvíos. Se identifican los actores y los recursos que surgen de la información de este símbolo, contando siempre con el patrón como fuente de verificación, y se definen los episodios del escenario a partir de cada impacto del símbolo.

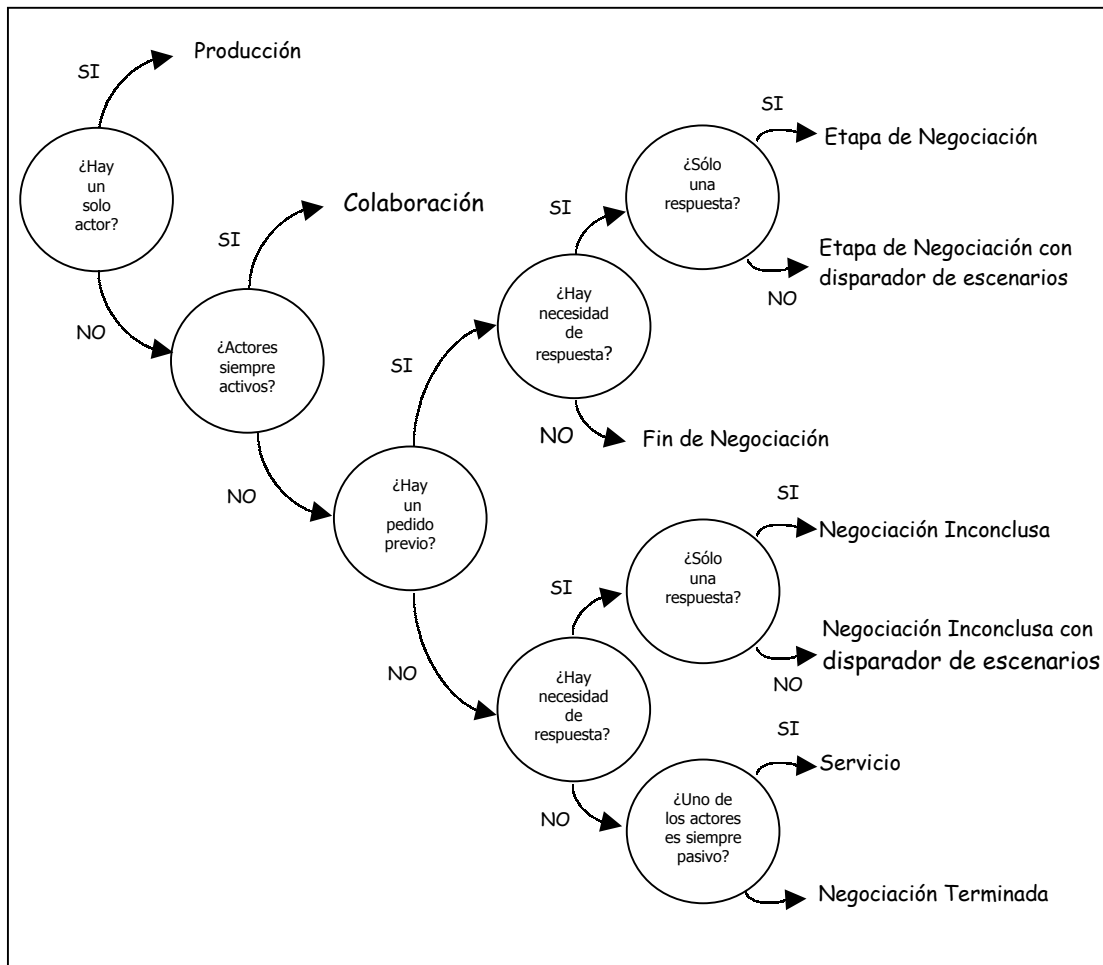


Figura 25 – Árbol de decisión para la selección de patrones

Resulta entonces evidente que al haber identificado un patrón para el escenario que se está construyendo, se dispone de una gran ayuda en la escritura de cada uno de los componentes nombrados, ya que el patrón contiene pautas orientadoras. Por lo tanto, el patrón constituye una fuente de información adicional, no sólo en lo que se refiere a contenido, sino también en cuanto a la estructura de cada componente del escenario.

Con respecto a los episodios, por ejemplo, cada patrón incluye pautas acerca del tipo, número y orden de los episodios que deberán incluirse. De ese modo, la aplicación de la heurística según la cual los episodios se definen a partir de los impactos del símbolo Verbo, puede convertirse en una tarea más sencilla y precisa.

La contrastación del componente episodios del escenario obtenido con el patrón previamente seleccionado no es directa. Es necesario haber determinado el tipo de cada uno

de los episodios involucrados según la clasificación mencionada en la sección 4.2. La determinación del tipo que corresponde a cada episodio puede también lograrse mediante un árbol de decisión, en este caso presentado en la figura 26.

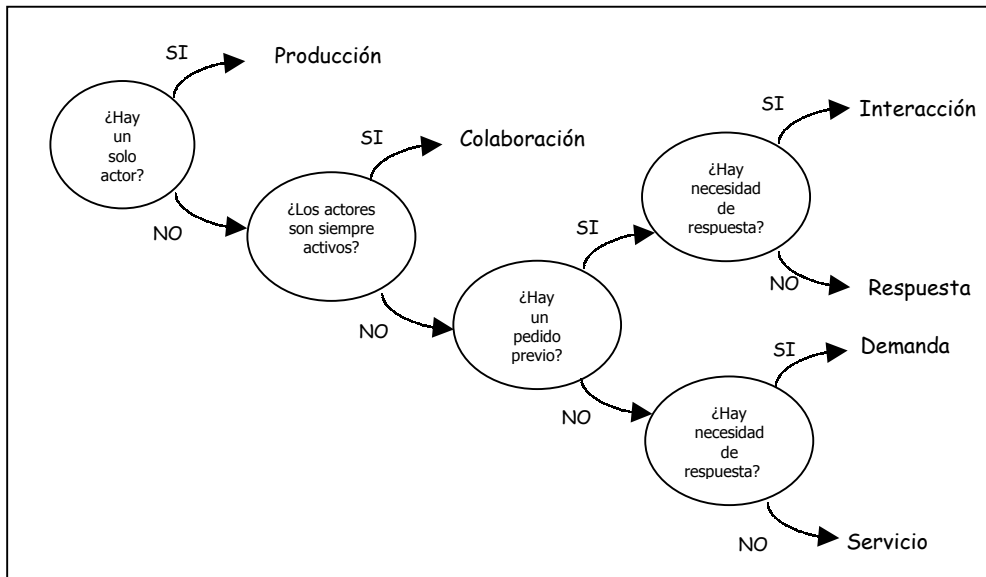


Figura 26 – Árbol de decisión para episodios de escenarios candidatos

5.6 EJEMPLO

En esta sección se presenta un ejemplo extraído del caso de estudio correspondiente a Agenda de Reuniones [van Lamsweerde93] [Hadad98].

CONVOCADO
<p>Noción:</p> <ul style="list-style-type: none"> - Persona invitada a la <u>reunión</u> <p>Impactos:</p> <ul style="list-style-type: none"> - Puede informar sus <u>horarios disponibles</u> - Puede dar el <u>aviso de concurrencia</u> - Debe estar en el lugar, <u>fecha</u> y <u>horarios</u> establecidos en la <u>convocatoria</u> ➤ <i>Puede dar el <u>aviso de no concurrencia</u></i> - Asigna un <u>reemplazante</u> en caso de no poder asistir y lo informa al <u>convocante</u> o <u>secretaría</u> - Puede definir el <u>material para repartir</u> - Registra el tiempo de traslado de la <u>reunión</u>

Figura 27 – Descripción del símbolo Convocado en el LEL

En la figura 27 se presenta la descripción del LEL del símbolo Convocado, uno de los actores secundarios de este caso de estudio. Siguiendo las heurísticas, por cada uno de los impactos se generará un escenario candidato. Si se considera el impacto: “puede dar el

aviso de no concurrencia”, se generará un escenario candidato denominado AVISAR LA NO CONCURRENCIA. En la figura 28 se presenta dicho escenario.

TÍTULO: **Avisar la no concurrencia**

OBJETIVO: Registrar que el convocado no asistirá a la reunión.

CONTEXTO: Debe realizarse previamente una convocatoria, en la cual se solicita a los convocados que confirmen o no su asistencia
El convocado pudo ya haber informado su concurrencia

ACTORES: convocado
secretaría
convocante

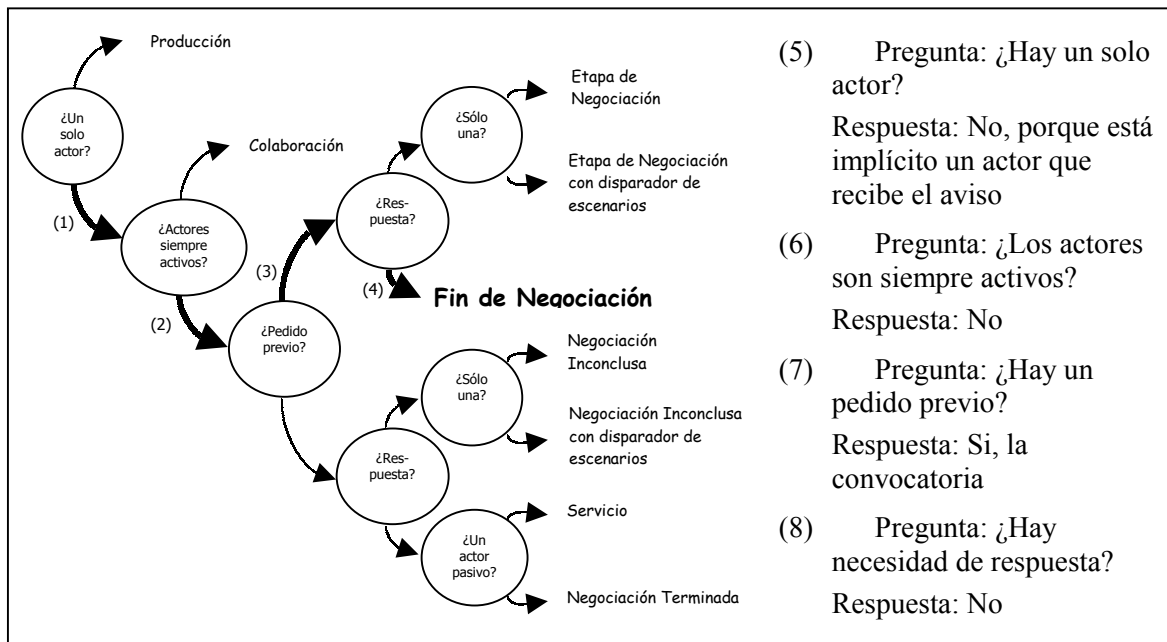
RECURSOS: agenda
listado para convocatoria
medios de comunicación (teléfono, fax, correo, computadora, etc.)

EPISODIOS:
El convocado se comunica por teléfono, fax o personalmente al sitio establecido para la confirmación de asistencia, informando al convocante o secretaría que no asistirá a la reunión. *Restricción*: debe efectuarse con anticipación a la fecha de la reunión
La secretaría o convocante registra en la agenda que no asistirá
La secretaría o convocante registra en el listado para convocatoria que no asistirá

Figura 28 – Escenario AVISAR LA NO CONCURRENCIA

Si se aplica el árbol de decisión para escenarios al impacto analizado, se obtendrá el resultado presentado en la figura 29. Este proceso permite determinar que el patrón de esta situación corresponde a un Fin de Negociación.

Escenario: AVISAR LA NO CONCURRENCIA



Patrón: Fin de Negociación

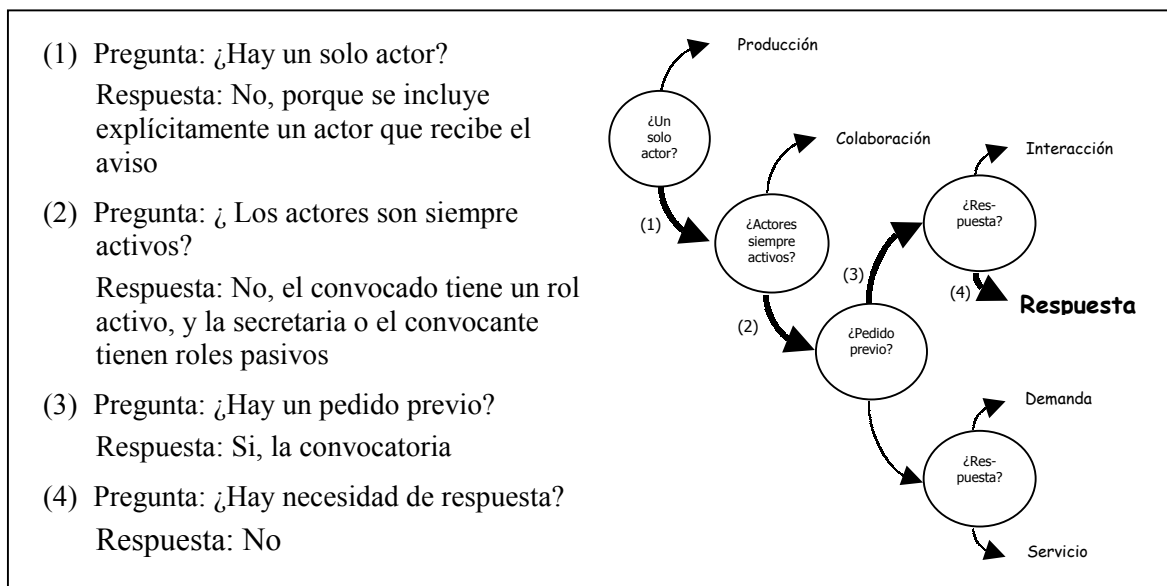
Figura 29 – Ejemplo de aplicación del árbol de decisión para escenarios

Debe notarse que el análisis efectuado en la figura 29 permite establecer que el

escenario podrá asociarse con un patrón de una familia de patrones cuya característica fundamental corresponde a Fin de Negociación. A este grupo pertenecen también patrones mixtos como Fin de Negociación con Producción y/o Servicio y/o Colaboración.

Para poder contrastar el componente Episodios del escenario con el patrón previamente seleccionado, se utiliza el segundo árbol de decisión presentado en la sección anterior.

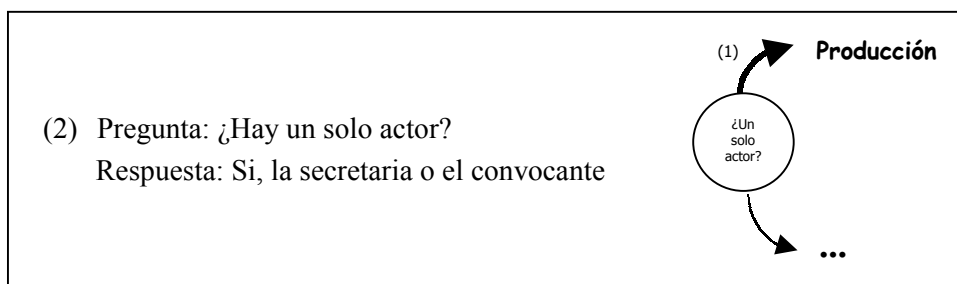
Episodio: El convocado se comunica ... informando al convocante o secretaria que no asistirá a la reunión.



Tipo de episodio: Respuesta

Figura 30 – Primer ejemplo de aplicación del árbol de decisión para episodios

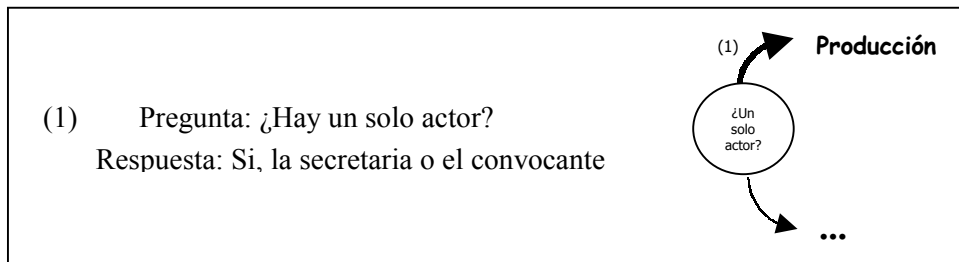
Episodio: La secretaria o convocante registra en la agenda que el convocado no asistirá



Tipo de episodio: Producción

Figura 31 – Segundo ejemplo de aplicación del árbol de decisión para episodios

Episodio: La secretaria o convocante registra en el listado para convocatoria que el convocado no asistirá



Tipo de episodio: Producción

Figura 32 – Tercer ejemplo de aplicación del árbol de decisión para episodios

El análisis de los episodios permite determinar que el primer episodio es del tipo Respuesta, característica que puede verificarse en el Patrón Fin de Negociación. Sin embargo, los otros dos episodios corresponden al tipo Producción, lo que lleva a establecer que el escenario tendrá la estructura de un Fin de Negociación con Producción, cuyo patrón puede observarse en la figura 33.

PATRÓN: FIN DE NEGOCIACIÓN CON PRODUCCIÓN

TÍTULO: Finalización de una actividad basada en transacciones

OBJETIVO: Dar por finalizada una actividad que requiere una secuencia coordinada de acciones por parte de los actores, junto con actividades de producción intercaladas

CONTEXTO:
 Ubicación geográfica: generalmente el lugar de trabajo del actor principal
 Precondiciones: otra situación debe haber dado inicio a la negociación
 Ubicación temporal: generalmente determinado por el actor principal y posiblemente prolongado

ACTORES: Varios, al menos dos

RECURSOS: Al menos uno, generalmente muchos

EPISODIOS:
VARIOS O NINGUNO COMO EL SIGUIENTE
 Un actor realiza una acción que responde a una acción anterior y que a su vez, requiere respuesta inmediata de otro actor.
POR LO MENOS UNO COMO EL SIGUIENTE
 Uno de los actores realiza alguna actividad que produce algún efecto sobre el macrosistema.
UNO COMO EL SIGUIENTE
 Un actor realiza una acción que responde a una acción anterior y que no requiere respuesta.
(DEBE ESTAR PRECEDIDO POR UNA ACCIÓN QUE REQUIERA RESPUESTA INMEDIATA)
PUEDEN ESTAR EN SECUENCIA O CONSTITUIR GRUPOS NO SECUENCIALES. SÓLO ES NECESARIO RESPETAR ORDEN DONDE ESTÁ EXPLÍCITAMENTE INDICADO.

EXCEPCIÓN: Circunstancia que obstaculiza el cumplimiento del objetivo

Figura 33 – Patrón Fin de Negociación con Producción

Cada uno de los componentes del Escenario de la figura 28 puede ser contrastado exitosamente con este patrón. Puede observarse, por ejemplo, en el componente Objetivo, la existencia de una precondition en el escenario que determina la necesidad de una situación donde se inició esta negociación. En particular, en este caso se trata de la convocatoria, que estará modelada en otro escenario.

CAPÍTULO 6

INTEGRACIÓN DE LOS PATRONES AL PROCESO DE CONSTRUCCIÓN

El proceso de construcción de escenarios, como ya se vio en detalle en el capítulo 3, comienza en el léxico del dominio de la aplicación, produciendo una primera versión de los escenarios derivada exclusivamente desde el LEL. Estos escenarios son mejorados utilizando otras fuentes de información y organizados para obtener un conjunto consistente de escenarios que representen al dominio de la aplicación.

En este capítulo se presentan la integración del Catálogo de Patrones y los Árboles de Decisión al proceso de construcción de escenarios [Ridao01b].

6.1 DESCRIPCIÓN DEL PROCESO DE CONSTRUCCIÓN CON PATRONES

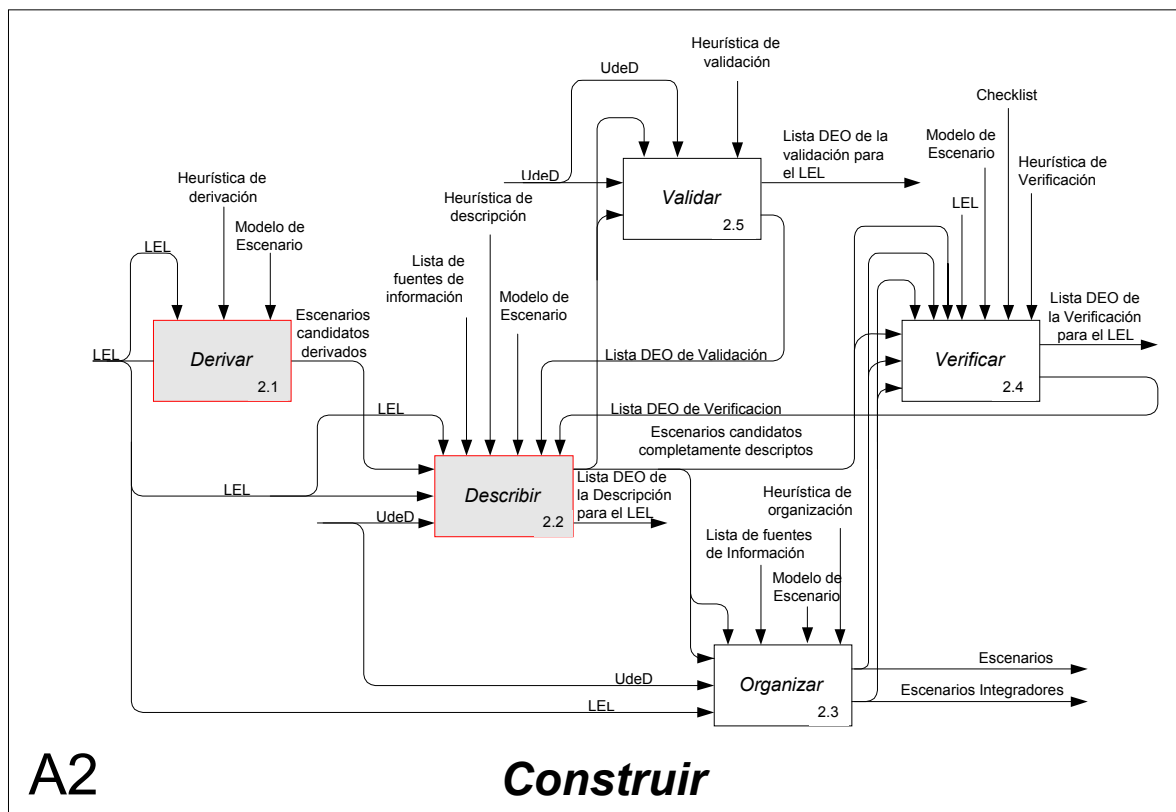


Figura 34 – SADT del Proceso de Construcción de Escenarios

El proceso completo comprende las actividades *Derivar*, *Describir*, *Organizar*, *Validar* y *Verificar*, como puede verse en el diagrama SADT de la figura 34.

El Catálogo de Patrones se incorpora en las etapas *Derivar* y *Describir*, en tanto que los Árboles de Decisión se integran a la actividad *Describir*. En esta sección se describe cómo se modifican ambas etapas.

La actividad *Derivar* está compuesta por cuatro actividades: *Identificar Actores*, *Identificar Escenarios* y *Crear Escenarios*. En la segunda, al momento de generar la lista de escenarios candidatos, es posible obtener alguna guía acerca del patrón a utilizar. En esta etapa, si varios actores comparten la misma situación, se procede a eliminar escenarios candidatos de la lista, dejando sólo uno que finalmente describirá la situación. Con la incorporación de patrones es posible mejorar este paso, ya que, si los títulos que describen la situación coinciden totalmente, es probable que el escenario a derivar corresponda a una **Colaboración**, mientras que si los títulos son similares y de alguna manera complementarios, es probable que se trate de **Negociación** o **Servicio**.

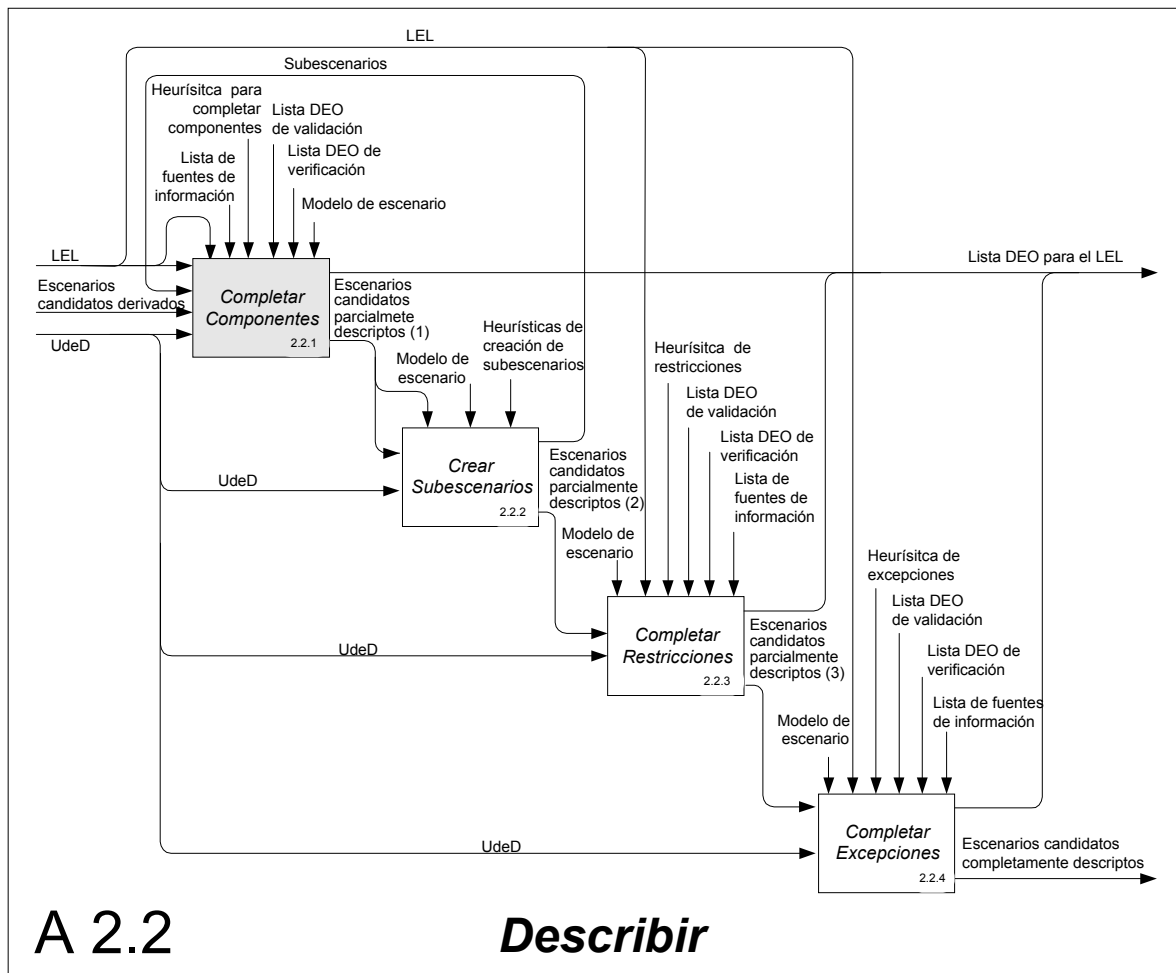


Figura 35 – SADT de la Actividad Describir

La actividad *Describir* consta de cuatro actividades: *Completar Componentes*, *Crear*

Subescenarios, *Completar Restricciones* y *Completar Excepciones*, que pueden verse en la figura 35. Los patrones se utilizan en la actividad *Completar Componentes*, cuyo diagrama SADT puede verse en la figura 36.

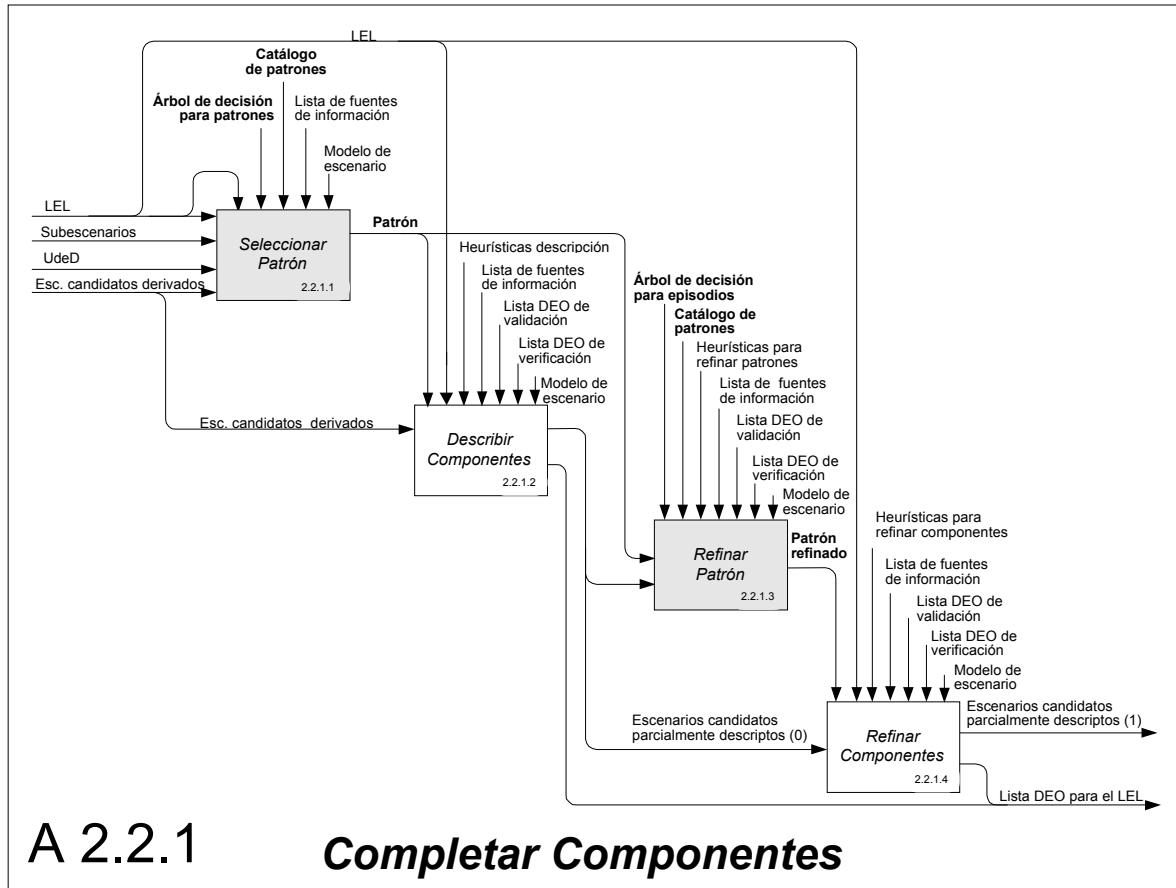


Figura 36 – SADT de la Actividad Completar Componentes

En la actividad *Seleccionar Patrón* se aplica el primer árbol de decisión a los impactos de los símbolos correspondientes a los actores del UdeD, determinando así el patrón correspondiente a la situación indicada por el mencionado impacto. Luego, en la etapa *Describir Componentes* se describe el escenario candidato utilizando las heurísticas de descripción y reusando la estructura del patrón seleccionado. Con respecto al componente Episodios, por ejemplo, cada patrón incluye reglas acerca del tipo, número y orden de los episodios.

A continuación, en la etapa *Refinar Patrón*, se aplica a cada episodio el segundo árbol de decisión, lo que permitirá refinar el patrón elegido, y mejorar su descripción. Con el patrón refinado, en la última etapa, se procede a reescribir cada componente según la información contenida en el nuevo patrón. En general, el patrón refinado es una variación del patrón seleccionado mediante el primer árbol. En la mayoría de los casos se trata de un patrón que combina a ese patrón con otro/s.

Si se introducen herramientas en el proceso de construcción de escenarios, sería útil disponer de un editor de escenarios, que permitiría que la *Descripción de Componentes* se hiciera contando con la estructura del patrón seleccionado como base, y con las pautas del

patrón como guía para la escritura de cada uno de los componentes.

En cuanto a los episodios, a medida que se fueran escribiendo, se los sometería al segundo árbol de decisión para clasificarlos y poder así contrastar el escenario que se está escribiendo con el patrón seleccionado, refinando esta selección si fuera necesario, como se indicó anteriormente.

6.2 APLICACIÓN A UN CASO DE ESTUDIO

En esta sección se presenta un ejemplo de derivación de escenarios utilizando la heurística del proceso de construcción original, en primera instancia, y luego incorporando los patrones al proceso. A continuación se describe el caso de estudio utilizado para el ejemplo.

Plan de Ahorro y Préstamo:

Un concesionario de vehículos 0 km. ofrece planes de pago en cuotas a largo plazo. Se constituye un grupo de personas físicas o legales cuyo objetivo individual es la adquisición de un vehículo mediante el pago de cuotas mensuales. Cada mes se efectúa un acto de adjudicación con el fin de entregar dos vehículos a los participantes, uno por sorteo y el otro por licitación. El sorteo da a todos los participantes la posibilidad de obtener en forma anticipada un vehículo mediante una asignación al azar. La licitación permite a aquellos participantes que desean obtener el vehículo inmediatamente, adelantar el pago de varias cuotas. El sistema administra la asignación mensual de vehículos, el cobro de las cuotas, la constitución de grupos, la sustitución, renuncia o expulsión de participantes, el seguimiento de los deudores y la actualización del valor de las cuotas en caso que el fabricante cambie el modelo y/o el precio de los vehículos. Por lo tanto, este complejo sistema requiere muchos controles legales y económicos y flexibilidad suficiente para soportar cambios constantes en el mercado y en las políticas de la compañía [Mauco97] [Rivero98] [del Fresno 98].

6.2.1 EJEMPLO DE DERIVACIÓN SIN PATRONES

En la figura 37 se presenta un ejemplo de derivación de un escenario utilizando las heurísticas del proceso de Construcción de Escenarios. Se utilizó para la representación del proceso, el esquema descrito en [Leite00]. Las palabras o frases subrayadas son símbolos del léxico correspondiente al caso de estudio. Los números sobre las flechas representan los diferentes pasos en la aplicación de la heurística:

1. A partir de las entradas del LEL se genera la lista de actores
2. Se consulta la entrada del LEL correspondiente a cada uno de los actores de la lista
3. A partir de los impactos de estas entradas del LEL, se produce la lista de escenarios candidatos.

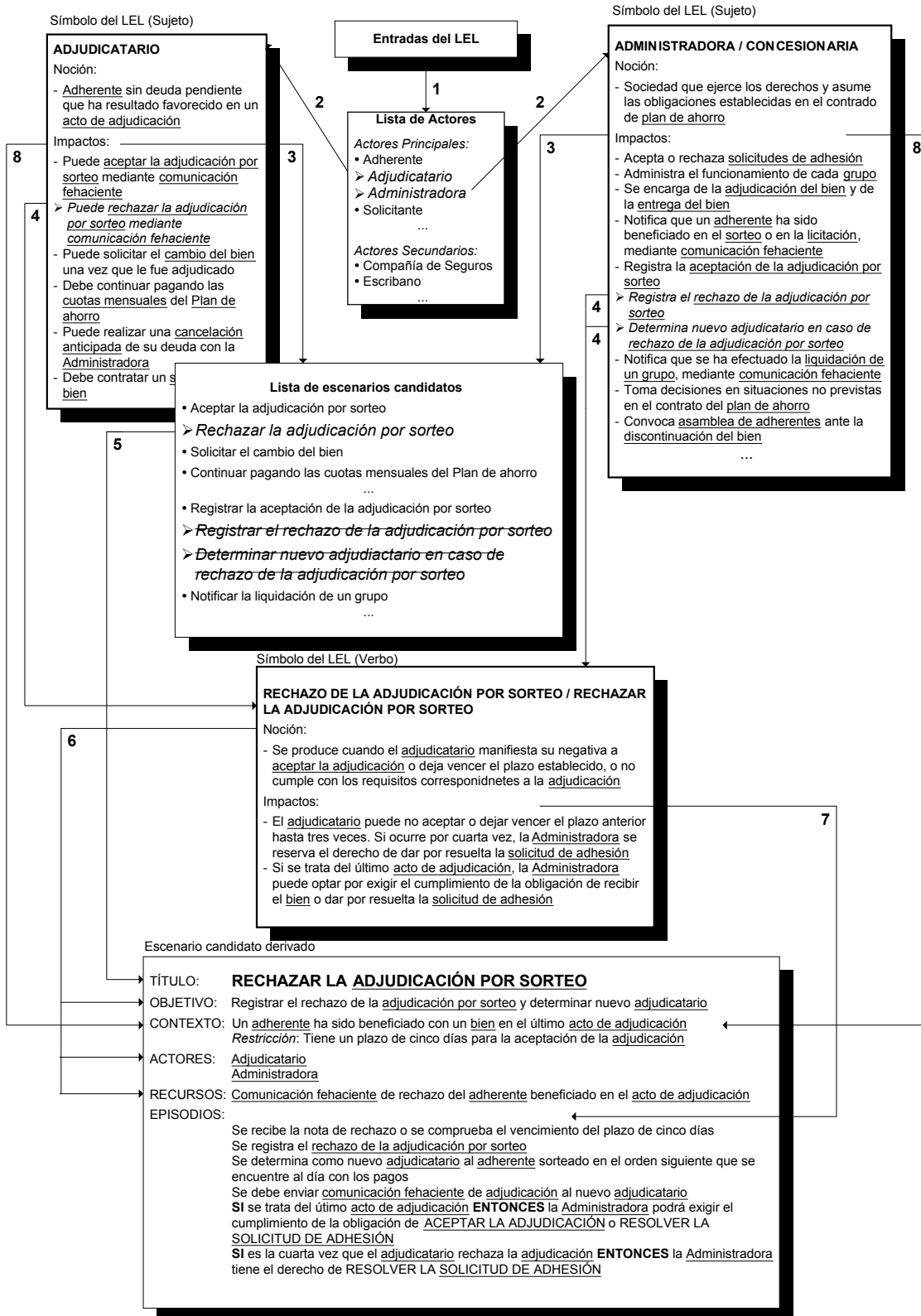


Figura 37 – Derivación de un escenario utilizando la heurística de construcción de escenarios

4. Como los impactos seleccionados contienen un símbolo de tipo Verbo, se consulta dicho símbolo en el LEL
5. Debido a que hay dos actores que participan en la misma actividad, y por lo tanto más de un impacto que la describe, se deriva un solo escenario a partir de ellos, eliminando el resto de escenarios candidatos de la lista
6. Se usa la noción del símbolo Verbo del LEL para definir el objetivo, los actores y los recursos
7. Se utilizan los impactos del símbolo Verbo del LEL como base para describir los episodios
8. A partir de los impactos correspondientes al símbolo Sujeto del LEL, considerados en el paso 3, se deriva la información de contexto

6.2.2 EJEMPLO DE DERIVACIÓN CON PATRONES

Si ahora se repite el proceso de derivación de este escenario, pero aplicando la heurística ampliada con los patrones, el proceso se modificará, como puede verse en la figura 38.

Se procede del mismo modo hasta el paso 4 inclusive, pero a partir de allí el proceso comienza a diferenciarse. A continuación se describen las etapas que aparecen en la figura. Los números entre paréntesis corresponden a los pasos de la aplicación de la heurística sin patrones.

- I. (1) A partir de las entradas del LEL se genera la lista de actores
- II. (2) Se consulta la entrada del LEL para cada uno de los actores de la lista
- III. (3) A partir de los impactos de estas entradas del LEL, se produce la lista de escenarios candidatos.
- IV. (4) Como los impactos seleccionados contienen un símbolo de tipo Verbo, se consulta dicho símbolo en el LEL
- V. Se aplica el árbol de decisión para escenarios a los impactos elegidos, seleccionado así un patrón para cada uno de ellos. (A continuación, se presenta la secuencia de preguntas que se aplica a cada uno de los episodios, y el patrón seleccionado en cada caso)

IMPACTO: Puede <u>rechazar la adjudicación por sorteo</u> mediante <u>comunicación fehaciente</u>	
Pregunta: ¿Hay un solo actor?	Respuesta: No
Pregunta: ¿Actores siempre activos?	Respuesta: No
Pregunta: ¿Hay un pedido previo?	Respuesta: Si
Pregunta: ¿Hay necesidad de respuesta?	Respuesta: Si
Pregunta: ¿Sólo una respuesta?	Respuesta: Si
PATRÓN: Etapa de Negociación	

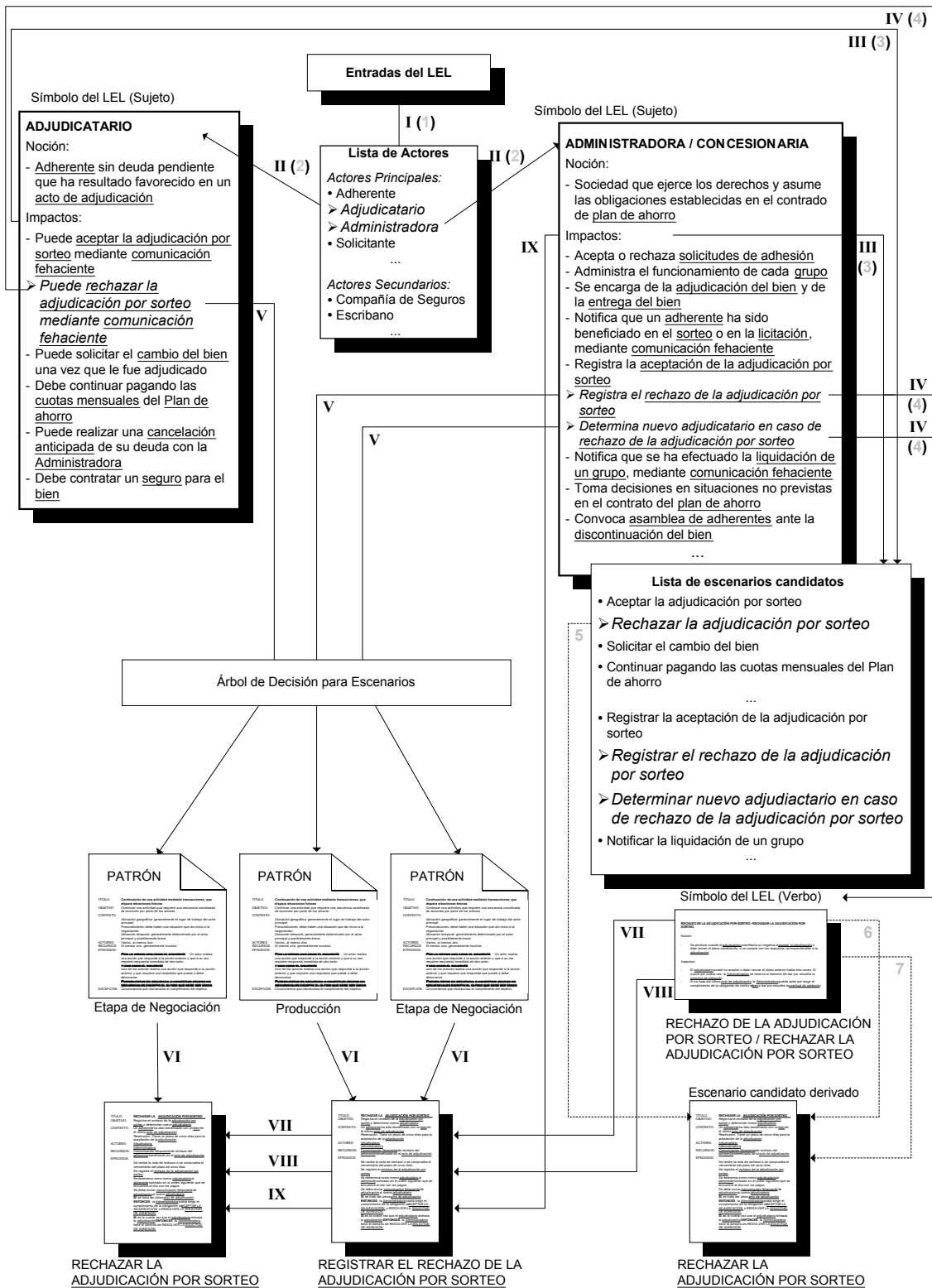


Figura 38 – Derivación de un escenario utilizando la heurística de construcción de escenarios con patrones

IMPACTO: Registra el <u>rechazo de la adjudicación por sorteo</u>	
<i>Pregunta:</i> ¿Hay un solo actor?	<i>Respuesta:</i> Si
PATRÓN: Producción	

IMPACTO: Determina nuevo <u>adjudicatario</u> en caso de <u>rechazo de la adjudicación por sorteo</u>	
<i>Pregunta:</i> ¿Hay un solo actor?	<i>Respuesta:</i> No
<i>Pregunta:</i> ¿Actores siempre activos?	<i>Respuesta:</i> No
<i>Pregunta:</i> ¿Hay un pedido previo?	<i>Respuesta:</i> Si
<i>Pregunta:</i> ¿Hay necesidad de respuesta?	<i>Respuesta:</i> Si
<i>Pregunta:</i> ¿Sólo una respuesta?	<i>Respuesta:</i> Si
PATRÓN: Etapa de Negociación	

- VI.** A partir de los patrones seleccionados y la lista de escenarios candidatos, se derivan los escenarios. Si bien los tres impactos analizados hacen referencia al mismo Verbo del LEL, al analizar el impacto correspondiente al Adjudicatario se observa que existe una brecha temporal entre el momento en que se envía la comunicación de rechazo y el momento en que la Administradora la recibe y la procesa. Por lo tanto, se deriva un escenario para el impacto del Adjudicatario, y otro para los de la Administradora.
- VII.** (6) Se usa la noción del símbolo Verbo del LEL para definir el objetivo, los actores y los recursos
- VIII.** (7) Se utilizan los impactos del símbolo Verbo del LEL como base para describir los episodios
- IX.** (8) A partir de los impactos correspondientes al símbolo Sujeto del LEL, considerados en el paso III, se deriva la información de contexto

6.2.3 COMPARACIÓN ENTRE AMBOS PROCEDIMIENTOS DE DERIVACIÓN

Comparando los procedimientos de derivación de escenarios presentados en las secciones 6.2.1 y 6.2.2, la primera diferencia notable es el número de escenarios derivados. En el primer caso, se deriva un solo escenario, cuyo título es: **Rechazar la Adjudicación por Sorteo**, mientras que, al usar patrones en el proceso de derivación, los escenarios son dos: **Rechazar la Adjudicación por Sorteo** y **Registrar el Rechazo de la Adjudicación por Sorteo**.

El escenario derivado primero, como puede verse en la figura 37, contiene a las situaciones descritas por los otros dos. Si se analiza la ubicación temporal del escenario derivado sin patrones, puede observarse que existe una brecha de duración desconocida entre el momento en que el Adjudicatario envía la comunicación fehaciente de rechazo y el momento en que la Administradora procesa dicho rechazo. Por lo tanto, el escenario está describiendo dos situaciones en lugar de una. Con el segundo método, la selección de los patrones lleva a detectar claramente ambas situaciones, derivando dos escenarios en lugar

de uno.

La heurística sin patrones sólo exige que se describa al menos uno de los componentes del Contexto (Precondiciones, Ubicación geográfica y Ubicación temporal). El contexto del escenario de la figura 37 contiene Precondiciones, pero no dice nada acerca de la Ubicación geográfica ni temporal. Por ello, es necesario analizar todo el escenario para detectar la brecha mencionada antes. Al utilizar los patrones seleccionados como guía para la descripción, se cuenta con pautas para los tres elementos, por lo cual se hace más fácil completar los tres, y se detectan más claramente los límites temporales de las situaciones.

Además, debido a que la heurística sin patrones hizo que se unificaran los tres impactos analizados derivando un solo escenario, al momento de describir los episodios fue necesario adoptar el punto de vista de uno de los actores, en este caso la Administradora. Esto hizo que se incluyeran sólo las actividades llevadas a cabo por dicho actor, omitiendo el o los episodios correspondientes al Adjudicatario, que fueron sugeridos claramente por el patrón correspondiente.

Debe notarse que ambas versiones fueron realizadas con una diferencia temporal significativa (más de un año). Este aspecto es relevante ya que el error detectado al utilizar patrones permaneció inadvertido durante un largo período incluyendo un proceso de inspección realizado sobre los escenarios de este caso de estudio [Doorn98].

Los problemas surgidos al aplicar la heurística original se solucionarían, sin embargo, al efectuar Inspecciones que contengan un checklist o un formulario orientado especialmente al problema.

El uso de patrones maneja este problema en forma más apropiada, ya que permite anticipar los posibles errores y evitarlos. Las inspecciones, en cambio, descubren los errores y exigen la reescritura de los escenarios. Detectar los problemas en el momento mismo de la derivación, permite escribir escenarios de mejor calidad, ahorrando mucho tiempo de reescritura.

Analizando todos los escenarios del caso de estudio pudo observarse que existen brechas temporales y episodios omitidos en los siguientes escenarios:

- Cesión de Derechos y Obligaciones
- Liquidar intereses punitivos
- Sustituir adherentes
- Situación de incumplimiento imputable al grupo
- Entregar el bien
- Solicitar cambio de bien
- Aceptar la adjudicación

Por lo tanto, sobre un total de 22 escenarios, un 33 % presenta problemas similares al del ejemplo analizado.

Se describieron también los siguientes casos de estudio:

Sistema Nacional para la Emisión de Pasaportes [Leite96] [Leite97b].

Este caso, desarrollado en la Universidad de Belgrano, se basó en un sistema centralizado en el Departamento de la Policía Federal, única organización autorizada para la emisión de pasaportes en Argentina.

Sistema para el Control de Pos Graduación [Breitman98]: desarrollado en la Pontificia Universidade Católica do Rio de Janeiro para el Dto. de Informática de PUC-Río.

En el primero, sobre un total de 24 escenarios del caso original, en 5 escenarios se detectaron brechas temporales, o episodios omitidos, lo que representa un 21 % del total. En el segundo caso, sobre un total de 8 escenarios, en 5 se detectó el mismo tipo de problemas, representando un 62 % del número original de escenarios.

CAPÍTULO 7

CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se ha propuesto una heurística que incluye el uso de patrones en el proceso de construcción de escenarios. La formulación y aplicación de esta heurística permitió arribar a las siguientes conclusiones:

Las regularidades existentes en los escenarios son independientes del problema, ya que se basan en la relación de los componentes del escenario y no en particularidades del dominio. Por otra parte, las regularidades son suficientemente nítidas como para ser utilizadas en la creación de patrones.

El uso de un enfoque basado en patrones fabricó un punto de vista suficientemente preciso como para hacer visibles ambigüedades e imprecisiones en escenarios de muy buena calidad que habían sido objeto de procesos de certificación previos.

La aplicación de la nueva heurística a casos de estudio preexistentes permitió obtener escenarios de mejor calidad que los obtenidos con anterioridad, especialmente en lo que se refiere al contexto temporal y a los episodios.

Se observó que un porcentaje importante de escenarios de los casos de estudio adolecían del defecto de una mala definición del contexto temporal con sus consecuencias sobre los episodios. Una brecha temporal no es una cuestión menor, ya que significa la existencia de escenarios no descubiertos o parcialmente empotrados en otros, lo que naturalmente provoca una solución de software diferente de la que realmente se necesita.

Puede afirmarse que los patrones obtenidos son verosímiles ya que aproximadamente el 93 % de los escenarios de los casos de estudio analizados pudieron ser reflejados en los mismos.

Los patrones de escenarios permiten reusar información con un alto grado de abstracción, ya que constituyen una guía en un proceso bien determinado y claro de reuso.

El uso de patrones en el proceso de construcción de escenarios enriquece las heurísticas existentes, ya que brinda orientación en el proceso de construcción y además ofrece una fuente de confirmación del escenario en desarrollo. De este modo, mejora la calidad de los escenarios producidos y se reduce el tiempo utilizado en escribir los escenarios.

La selección del patrón no es un proceso intuitivo, sino que se cuenta con un árbol de decisión que permite elegir el patrón, o la familia de patrones, con un alto grado de certeza. El segundo árbol de decisión permite seleccionar más específicamente el patrón dentro de la familia de patrones determinada por medio del primer árbol, y confirmar el escenario en desarrollo, contrastándolo con la estructura del patrón seleccionado.

En el futuro, se pretende aplicar la heurística incluyendo patrones a casos de estudio completamente nuevos, con el fin de analizar la posibilidad de reuso de los patrones

encontrados. Esta actividad se encuentra en plena realización.

Se prevé transferir los métodos desarrollados a usuarios, con el fin de utilizar la metodología en proyectos reales de desarrollo de software.

Es posible que existan otros aspectos que los patrones pueden mejorar y que no han sido detectados aún. Probablemente, a partir de las pruebas mencionadas surjan otras ventajas o aspectos no considerados. Este estudio tal vez permita, además, ampliar la base de patrones existente.

Por otra parte, es posible que el uso de patrones brinde un mayor aporte como crítica de escenarios ya construidos que como base para la construcción reusando su estructura. Para contar con una opinión más acertada acerca de este tema, será necesario finalizar las pruebas con nuevos casos de estudio y comparar los resultados de ambos enfoques.

Se estudiarán nuevas visiones de patronización, así como el uso de patrones en el proceso de construcción de escenarios integradores propuesto en [HADAD99].

Se desarrollará una herramienta para soportar el proceso propuesto en este trabajo. En principio, esta herramienta contará con un editor de escenarios, que permitirá efectuar la *Descripción de Componentes* contando con la estructura del patrón seleccionado como base, y con las pautas del patrón como guía para la escritura de cada uno de los componentes.

Se encuentra en desarrollo un nuevo caso de estudio en otra universidad. En esta nueva aplicación, se está experimentando con la noción de escenarios futuros como complemento de los escenarios actuales. Se han realizado los acuerdos para la verificación mediante los patrones a los efectos de comprobar la utilidad de los mismos en un contexto diferente de aquél a partir del cual fueron deducidos (escenarios actuales).

REFERENCIAS

- [Alexander77] Alexander, Ch., Ishikawa, S., Silverstein, M., Jacobson, M., Fidsdahl-King, I., Angel, Sh.. *"A Pattern Language"*. Oxford University Press, New York, 1977.
- [Allenby01] Allenby, K., Kelly, T., "Deriving Safety Requirements Using Scenarios", Proceedings of Fifth IEEE International Symposium on Requirements Engineering, Toronto, Canada, 2001, pp. 228-235.
- [Anton01] Anton, A., Earp, J., Potts, C., Alspaugh, T., "The Role of Policy an Stakeholder Privacy Values in Requirements Engineering", Proceedings of Fifth IEEE International Symposium on Requirements Engineering, Toronto, Canada, 2001, pp. 138-145.
- [Arango93] Arango, G., Schafer, W., Prieto, R. *"Domain Analysis Methods – Software Reusability"*. Ellis Horwood Ltd., 1993.
- [Benner93] Benner, K., Feather, M., Johnson, L., Zorman, L., "Utilizing Scenarios in the Software Development Process", Proceedings of the 8th. Knowledge-Based Software Engineering Conference (KBSE 93), IEEE, 1993.
- [Booch91] Booch, G., *"Object Oriented Design with Applications"*. The Benjamin Cumming Publishing Company, Inc., Redwood City, 1991.
- [Booch94] Booch, G., "Scenarios", *Report on Object Analysis and Design*, Vol.1, N° 3, 1994, pp. 3-6.
- [Breitman98] Breitman, K.K., Leite, J.C.S.P., "A Framework for Scenario Evolution", Proceedings of the IEEE Int. Conf. on Requirements Engineering, IEEE Computer Society Press, 1998, pp. 214-221.
- [Buschmann96] Buschmann, F., Feunier, R., Rohnert, H., Sommerlad, P., Stal, M., *"Pattern-Oriented Software Architecture: A system of Patterns"*. John Wiley & Sons, 1996.
- [Carroll95] Carroll, J., *"Scenario-Based Design: Envisioning Work and Technology in System Development"*. Introduction: The Scenario Perspective on System Development , J. Carroll, ed., John Wiley & Sons, New York, 1995.
- [Cockburn95] Cockburn, A., "Structuring Use Cases with Goals", Technical Report, Humans and Technology, 7691 Dell Rd, Salt Lake City, UT 84121, HaT.tr.95.01, <http://members.aol.com/acockburn/papers/usecases.htm>, 1995.
- [Constantine98] Constantine, L., "Joint Essential Modeling, User Requirements Modeling for Usability", International Conference on Requirements Engineering, Colorado Springs, Constantine & Lockwood, Ltd., 1998.

- [Coplien96] Coplien, J.O., *“Software Patterns”*. SIGS Books & Multimedia, USA, 1996.
- [Chen76] Chen, P., “The entity-relationship model: Towards a unified view of data”, *ACM Transactions on Database Systems*, 1(1), 1976.
- [Cleland01] Cleland-Huang, J., Chang, C., Kim, H., Balakrishnan, A., “Requirements-Based Dynamic Metrics in Object-Oriented Systems”, *Proceedings of Fifth IEEE International Symposium on Requirements Engineering*, Toronto, Canada, 2001, pp. 212-219.
- [Dano97] Dano, B., Briand, H., Barbier, F., "An Approach Based on the Concept of Use Case to Produce Dynamic Object-Oriented Specification", *Proceedings of the Third IEEE International Symposium on Requirements Engineering*, IEEE Computer Society Press, Los Alamitos, California, 1997, pp.54-64.
- [Davis90] Davis, A. *“Software Requirements: Analysis and Specification”*. Prentice Hall, 1990.
- [del Fresno98] del Fresno, M., Mauco, V.; Ridao, M.; Doorn, J.; Rivero, L. “Derivación de Objetos Utilizando LEL y Escenarios en un Caso Real”, *Anales WER’98, Workshop en Engenharia do Requisitos*, Maringá, Brazil, 1998, pp. 89-98.
- [Doorn98] Doorn, J., Kaplan, G., Hadad, G., Leite, J.C.S.P., “Inspección de Escenarios”, *Proceedings of WER’98, Workshop en Engenharia do Requisitos*, Maringá, Brazil, 1998, pp. 57-69.
- [Dorfman97a] Dorfman, M., Thayer, R. *“Software Engineering”*. IEEE Computer Society Press, 1997.
- [Dorfman97b] Dorfman, M. *“Requirements Engineering”*, in *“Software Requirements Engineering”*, edited by Thayer, R., Dorfman, M. IEEE Computer Society Press, 1997.
- [Firesmith94] Firesmith, D. G, “Modeling the Dynamic Behavior of Systems, Mechanisms, and Classes with Scenarios”, *Report on Object Analysis and Design*, Vol.1, N° 2, 1994, pp. 32-36.
- [Foote98] Foote, B. Introduction to *“Patterns Languages of Program Design 3”*. Edited by Martin, C., Riehle, D., Buschmann, F. Addison-Wesley, 1998.
- [Gabriel96] Gabriel, R. Introduction-2 to *“Patterns Languages of Program Design 2”*. Addison-Wesley, 1996.
- [Gamma95] Gamma, Helm, Johnson, Vlissides. *“Design Patterns”*. Addison-Wesley, 1995.
- [Gans01] Gans, G., Jarke, M., Kethers, S., Lakemeyer, G., Ellrich, L., Funken, C., Meister, M., “Requirements Modeling for Organization Networks: A (Dis)Trust-Based Approach”, *Proceedings of Fifth IEEE International Symposium on Requirements Engineering*, Toronto, Canada, 2001, pp.154-163.

- [Garcia99] Garcia, O., Gentile, C., "*Diseño de una herramienta para construcción de LEL y Escenarios*", Graduation dissertation, Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina, 1999.
- [Goguen92] Goguen, J.A., Linde, Ch., "Techniques for Requirements Elicitation", Proceedings of the International Symposium on Requirements Engineering, IEEE Computer Society, 1992, pp. 152-164.
- [Gough95] Gough, P. A., Fodemski, F. T., Higgins, S. A., Ray, S. J., "Scenarios - An Industrial Case Study and Hypermedia Enhancements", RE95: Proceedings of the International Symposium on Requirements Engineering, IEEE Computer Society Press, Los Alamitos, California, 1995, pp. 10-17.
- [Guezzi91] Guezzi, C., Jazayeri, M., Mandrioli, D. "*Fundamentals of Software Engineering*". Prentice Hall, 1991.
- [Hadad97] Hadad, G., Kaplan, G., Oliveros, A., Leite, J.C.S.P., "Construcción de Escenarios a partir del Léxico Extendido del Lenguaje". SoST'97 (Simposio de Tecnología de Software) XXVI JAIIO, Buenos Aires, Argentina, 1997, pp. 65-77.
- [Hadad98] Hadad, G., Kaplan, G., Leite, J.C.S.P., "Léxico extendido del lenguaje y escenarios del meeting scheduler". Technical Report # 13, Dto. Investigación, U. Belgrano, Bs. As., 1998.
- [Hadad99] Hadad, G., Doorn, J., Kaplan, G., Leite, J.C.S.P., "*Enfoque Middle-Out en la Construcción e Integración de Escenarios*". Proceedings de WER'99, Buenos Aires, Argentina, 1999, pp. 79-94.
- [Hoare84] Hoare, C., "Programming: Sorcery or Science?", *IEEE Software*, Vol. 1, N° 2, April 1984.
- [Hsia94] Hsia, P., Samuel, J., Gao, J., Kung, D., Toyosima, I. and Chen, C., "Formal Approach to Scenario Analysis", *IEEE Software*, 1994, V.11, N° 2, pp. 33-41.
- [Jackson01] Jackson, M., "Formalism and Informality in RE", Proceedings of Fifth IEEE International Symposium on Requirements Engineering, Toronto, Canada, 2001, pp. 269.
- [Jacobson92] Jacobson, I., Christerson, M., Jonsson, P., Overgaard, G., "*Object-Oriented Software Engineering - A Use Case Driven Approach*", Reading, MA: Addison Wesley, New York: ACM Press, 1992.
- [Jacobson94a] Jacobson, I., "Basic Use-Case Modeling", *Report on Object Analysis and Design*, Vol.1, N° 2, 1994, pp. 15-19.
- [Jacobson94b] Jacobson, I., "Basic Use-Case Modeling (Continued)", *Report on Object Analysis and Design*, Vol.1, N° 3, 1994, pp. 7-9.
- [Jarke98] Jarke, M., Bui T.X., Carroll, J.M., "Scenario Management: An Interdisciplinary Approach", *Requirements Engineering Journal*, Vol. 3, N° 4, 1998, pp. 155-173.

- [Jirotk94] Jirotk, M., Goguen, A., “*Requirements Engineering. Social and Technical Issues*”. Academic Press, 1994.
- [Kotonya96] Kotonya, G., Sommerville, I., “*Requirements Engineering with Viewpoints*” in “*Software Requirements Engineering*”. edited by Thayer, R., Dorfman, M. IEEE Computer Society Press, 1997.
- [Kotonya98] Kotonya, G., Sommerville, I., “*Requirements Engineering. Processes and Techniquess*” . John Wiley & Sons, 1998.
- [Kovitz01] Kovitz, B., “Is backtracking so bad? The role of learning in software development”, Proceedings of Fifth IEEE International Symposium on Requirements Engineering, Toronto, Canada, 2001, pp. 272.
- [Leite90] Leite, J.C.S.P., Franco, A., “O Uso de Hipertexto na Elicitação de Linguagens da Aplicação”. Anais de IV Simpósio Brasileiro de Engenharia de Software, SBC, Brazil, 1990, pp. 134-149.
- [Leite93] Leite, J.C.S.P., Franco, A.P.M., "A Strategy for Conceptual Model Acquisition", IEEE International Symposium on Requirements Engineering, IEEE Computer Society Press, 1993, pp. 243-246.
- [Leite96] Leite, J.C.S.P., Oliveros, A., Rossi, G., Balaguer, F., Hadad, G., Kaplan, G., Maiorana, V., “Léxico extendido del lenguaje y escenarios del sistema nacional para la obtención de pasaportes”, Technical Report #7, Departamento de Investigación, Universidad de Belgrano, Buenos Aires, 1996.
- [Leite97a] Leite, J.C.S.P., “*Ingeniería de Requisitos*”. Notas de Cátedra , 1997.
- [Leite97b] Leite, J.C.S.P., Rossi, G., Balaguer, F., Maiorana, V., Kaplan, G., Hadad, G., Oliveros, A., “Enhancing a Requirements Baseline with Scenarios”, *Requirements Engineering Journal*, Vol.2, No. 4, 1997, pp. 184-198.
- [Leite00] Leite, J.C.S.P., Hadad, G.D.S., Doorn, J.H., Kaplan, G.N., “A Scenario Construction Process”, *Requirements Engineering Journal*, Vol.5, N° 1, 2000, pp. 38-61.
- [Mauco97] Mauco, V., Ridao M., del Fresno, M., Rivero, L., Doorn, J., “Ingeniería de Requisitos, Proyecto: Sistema de Planes de Ahorro”, Technical Report, ISISTAN, UNCPBA, Tandil, Argentine, 1997.
- [Meszaros99] Meszaros, G., Doble Allen, J . “A pattern Language for Pattern Writing”. Object Systems Group –Telecom Systems.
- [Nauer69] Nauer, P., Randall, B., eds. “*Software Engineering*”. NATO Scientific Affairs Division, Brussels, Belgium, 1969.
- [Ober98] Ober, R., Probasco, L, Ericsson, M., “*Applying Requirements Management with Use Cases*”. Rational Software Corporation, 1998.
- [Potts94] Potts, C., Takahashi, K., Antón, A. I., “Inquiry-Based Requirements Analysis”, *IEEE Software*, Vol. 11, N° 2, 1994, pp. 21-32.
- [Potts95] Potts, C., “Using Schematic Scenarios to Understand User Needs”, Proceedings of DIS’95 - Symposium on Designing Interactive

- Systems: Processes, Practices and Techniques, ACM Press, University of Michigan, 1995, pp. 247-256.
- [Pressman97a] Pressman, R., *“Software Engineering: A Practitioner’s Approach”*. Fourth Edition. McGraw-Hill, Inc., 1997.
- [Pressman97b] Pressman, R., “Software Engineering” in *“Software Engineering”*, edited by Dorfman, M., Thayer, R. IEEE Computer Society Press, 1997.
- [Ridao00a] Ridao, M., Doorn, J., Leite, J.C.S.P., “Aspectos Recurrentes en la Construcción de Escenarios”, *Memorias de IDEAS’00 – Jornadas Iberoamericanas de Ingeniería de Requisitos y Ambientes de Software*, Cancún, México 2000, pp. 349-360.
- [Ridao00b] Ridao, M., Doorn, J., Leite, J.C.S.P., “Uso de Patrones en la Construcción de Escenarios”, *Anais de WER 2000*, Río de Janeiro, Brazil, 2000, pp. 140-157.
- [Ridao01a] Ridao, M., Doorn, J., Leite, J.C.S.P., “Domain Independent Regularities in Scenarios”, *Proceedings of Fifth IEEE International Symposium on Requirements Engineering*, Toronto, Canada, 2001, pp. 120-127.
- [Ridao01b] Ridao, M., Doorn, J., Leite, J.C.S.P., “Incorporación de patrones al proceso de construcción de escenarios”, aceptado para publicación en *WER 2001*, Buenos Aires, Argentina, 2001.
- [Rivero98] Rivero, L., Doorn, J., del Fresno, M., Mauco, V., Ridao, M., Leonardi, M.C., “Una Estrategia de Análisis Orientada a Objetos basada en Escenarios: Aplicación en un Caso Real”, *Proceedings of WER’98, Workshop en Engenharia do Requisitos*, Maringá, Brazil, 1998, pp. 79-90.
- [Robertson95] Robertson, S.P., *“Generating Object-Oriented Design Representations via Scenario Queries, Scenario-Based Design: Envisioning Work and Technology in System Development”*. J. Carroll, ed., John Wiley & Sons, New York, 1995.
- [Ross77] Ross, D., Schoman, A., “Structured analysis for requirements definition”, *IEEE Transactions on Software Engineering*, Special Issue on Requirements Analysis, 1977, 3 (1), pp. 6-15.
- [Royce91] Royce, W., Current Problems in *“Aerospace Software Engineering: A Collection of Concepts”*. edited by Anderson, Ch., Dorfman, M., American Institute of Aeronautics, Inc., Washington DC, 1991.
- [Rubin92] Rubin, K.S., Goldberg, J., “Object Behavior Analysis”, *Communications of the ACM*, Vol. 35, N° 9, Sep. 1992.
- [Schneider98] Schneider, G., Winters, J., *“Applying Use Cases, A Practical Guide”*, Addison-Wesley, 1998.
- [Sutcliffe97] Sutcliffe, A., “A Technique Combination Approach to Requirements Engineering”, *Proceedings of the Third IEEE International*

- Symposium on Requirements Engineering, IEEE Computer Society Press, Los Alamitos, Colorado, 1997, pp. 65-74.
- [Sutcliffe01] Sutcliffe, A., “Requirements Engineering for Complex Collaborative Systems”, Proceedings of Fifth IEEE International Symposium on Requirements Engineering, Toronto, Canada, 2001, pp. 110-117.
- [Thayer97] Thayer, R., Dorfman, M., Foreword by Davis, A. “*Software Requirements Engineering*”. IEEE Computer Society Press, 1997.
- [van Lamsweerde93] van Lamsweerde, K., Darimont, R., Massonet, Ph., “The Meeting Scheduler System – Preliminary Definition”. Internal Report, Université Catholique de Louvain, 1993.
- [van Lamsweerde01] van Lamsweerde, A., “Goal-Oriented Requirements Engineering: A Guided Tour”, Proceedings of Fifth IEEE International Symposium on Requirements Engineering, Toronto, Canada, 2001, pp. 249-261.
- [Weidenhaupt98] Weidenhaupt, K., Pohl, K., Jarke, M., Haumer, P., “Scenarios in System Development: Current Practice”, *IEEE Software*, 1998, pp. 34-45.
- [Wirfs95] Wirfs-Brock, R., “Designing Objects and Their Interactions: A Brief Look at Responsibility-Driven Design”, *Scenario-Based Design: Envisioning Work and Technology in System Development*, J. Carroll, ed., John Wiley & Sons, New York, 1995, pp. 337-359.
- [Zave97] Zave, P., Jackson, M., “Four Dark Corners of Requirements Engineering”. *ACM Transactions on Software Engineering and Methodology*. Vol. 6, N°1, Jan. 1997.
- [Zorman95] Zorman, L., “*Requirements Envisaging by Utilizing Scenarios (Rebus)*”, Ph.D. Dissertation, University of Southern California, 1995.

ANEXO A

CASOS DE ESTUDIO

SISTEMA NACIONAL PARA LA EMISIÓN DE PASAPORTES

Escenario	Patrón	Estado del escenario al finalizar el análisis
ENTREGA DE FORMULARIOS	Negociación Inconclusa con Disparador de Escenarios	Reescrito
COBRAR TRÁMITE	Etapa de Negociación con Producción	Sin cambios
OBTENCIÓN DE HUELLAS DIGITALES PARA CONTROL	Etapa de Negociación con Servicio	Sin cambios
OBTENCIÓN DE HUELLAS DIGITALES PARA EL LEGAJO	Etapa de Negociación con Servicio	Sin cambios
COBRAR FOTOGRAFÍA	Etapa de Negociación	Nuevo
SACAR FOTOGRAFÍA	Servicio	Reescrito
DERIVACIÓN A CABINA DE RECEPCIÓN	Etapa de Negociación con Producción	Sin cambios
RECEPCIÓN Y ARMADO DE PASAPORTE ORIGINAL	Sucesión de patrones	Reescrito
CONTROL DE PRONTUARIO	Producción	Reescrito
CONTROL DACTILOSCÓPICO	Producción	Sin cambios
ENTREGA DE PASAPORTE	Fin de Negociación con Producción	Sin cambios
ENTREGA DE PASAPORTE EN EL BOX DE REVÁLIDAS TELEFÓNICAS	Fin de Negociación con Producción	Sin cambios
DERIVACIÓN DE PASAPORTES OBSERVADOS	Etapa de Negociación con Disparador de Escenarios y Producción	Sin cambios

Escenario	Patrón	Estado del escenario al finalizar el análisis
GESTIÓN DE PASAPORTE OBSERVADO POR MAL TRÁMITE	Negociación Terminada con Producción	Sin cambios
GESTIÓN DE PASAPORTE POR PEDIDO DE CAPTURA U ORDEN JUDICIAL	Negociación Terminada con Producción	Sin cambios
ENVÍO A INCINERACIÓN DE PASAPORTES NO RETIRADOS	Producción	Sin cambios
RECEPCIÓN PARA REVÁLIDA DE PASAPORTE SIN RENOVACIÓN TOTAL	Sucesión de patrones	Sin cambios
RECEPCIÓN Y ARMADO PARA REVÁLIDA DE PASAPORTE CON RENOVACIÓN TOTAL	Sucesión de patrones	Sin cambios
INICIO DEL TRÁMITE DE REVÁLIDA TELEFÓNICA	Negociación inconclusa	Sin cambios
PRESENTACIÓN DEL SOLICITANTE ANTE BOX REVÁLIDAS TELEFÓNICAS	Sucesión de patrones	Sin cambios
CONTROL DE DOCUMENTACIÓN	Negociación Terminada con Producción	Reescrito
TRÁMITE DE PASAPORTE ORIGINAL	Negociación Terminada con Producción y Servicio	Sin cambios
REVÁLIDA DE PASAPORTE	Negociación Terminada con Producción y Servicio	Sin cambios
INCLUSIÓN DE HIJO MENOR DE TRES AÑOS	Negociación Terminada con Producción y Servicio	Sin cambios
TRÁMITE REVÁLIDA TELEFÓNICA	Negociación Terminada con Producción y Servicio	Sin cambios

Total escenarios originales:	24
Total escenarios reescritos:	5 (21%)
Total escenarios nuevos:	1
Total escenarios eliminados:	0
Total escenarios definitivos:	25
Total escenarios con patrón:	21 (84%)

SISTEMA DE AGENDA DE REUNIONES

Escenario	Patrón	Estado del escenario al finalizar el análisis
REQUERIR UNA REUNIÓN	Producción	Sin cambios
ORGANIZAR LA REUNIÓN	Sucesión de patrones	Sin cambios
RECORDAR LA REUNIÓN	Negociación Inconclusa con Producción y Servicio	Sin cambios
ANULAR LA REUNIÓN	Producción y Servicio	Sin cambios
TRASLADAR LA FECHA DE LA REUNIÓN	Producción y Servicio	Sin cambios
CAMBIAR LOS REQUERIMIENTOS DE LA REUNIÓN	Producción y Servicio	Sin cambios
SOLICITAR HORARIOS DISPONIBLES	Negociación Terminada con Producción	Sin cambios
AVISAR LA CONCURRENCIA	Fin de Negociación con Producción y Servicio	Sin cambios
ASISTIR A LA REUNIÓN	Producción	Sin cambios
AVISAR LA NO CONCURRENCIA	Fin de Negociación con Producción	Sin cambios
SOLICITAR MATERIAL FÍSICO	Negociación Inconclusa con Producción	Sin cambios
ESTABLECER LA FECHA DE LA REUNIÓN	Producción	Sin cambios
ESTABLECER NUEVA FECHA DE REUNIÓN	Producción	Sin cambios
GENERAR EL LISTADO PARA CONVOCATORIA	Producción	Sin cambios

Total escenarios originales:	14
Total escenarios reescritos:	0
Total escenarios nuevos:	0
Total escenarios definitivos:	14
Total escenarios con patrón:	13 (93%)

SISTEMA DE PLAN DE AHORRO

Escenario	Patrón	Estado del escenario al finalizar el análisis
PEDIR SOLICITUD	Negociación Inconclusa	Nuevo
PRESENTAR SOLICITUD DE ADHESIÓN	Etapas de Negociación con Producción	Reescrito
ANALIZAR SOLICITUD DE ADHESIÓN	Producción y Servicio	Sin cambios
CONSTITUIR UN GRUPO	Producción y Servicio	Sin cambios
CESIÓN DE DERECHOS Y OBLIGACIONES	Negociación Inconclusa	Reescrito
EFFECTUAR UNA ASAMBLEA DE ADHERENTES	Colaboración y Servicio	Reescrito
CANCELAR EL PLAN DE AHORRO	_____	Eliminado
LIQUIDAR EL GRUPO	Producción y Servicio	Reescrito
LIQUIDAR INTERESES PUNTORIOS	Negociación Inconclusa	Reescrito
PAGAR INTERESES PUNTORIOS	Fin de Negociación	Nuevo
SUSTITUIR ADHERENTES	Producción y Servicio	Sin cambios
SITUACIÓN DE INCUMPLIM. IMPUTABLE AL GRUPO	Servicio	Reescrito
REALIZAR ACTO DE ADJUDICACIÓN	Negociación Inconclusa con Disparador de Escenarios y Producción y Servicio	Sin cambios
ADJUDICAR UN BIEN POR LICITACIÓN	Negociación Inconclusa con Producción y Servicio	Reescrito
RESOLVER SOLIC. DE ADHESIÓN	Producción y Servicio	Reescrito

Escenario	Patrón	Estado del escenario al finalizar el análisis
CANCELACIÓN ANTICIPADA DE DEUDA	Negociación Terminada con Producción	Reescrito
SOLICITAR CAMBIO DEL BIEN	Negociación Inconclusa	Nuevo
ADJUDICAR UN BIEN POR SORTEO	Producción	Sin cambios
ACEPTAR LA ADJUDICACIÓN	Fin de Negociación	Reescrito
PROCESAR RECHAZO ADJUDIC. POR SORTEO	Etapas de Negociación con Disparador de Escenarios, Producción y Servicio	Reescrito
ENTREGA DEL BIEN	Negociación Terminada con Producción y Servicio	Reescrito
SUSCRIBIR CONTRATO DE PRENDA	Servicio	Reescrito
ASEGURAR EL BIEN PRENDADO	Producción	Sin cambios
RECLAMAR PENALIDAD	Negociación Terminada	Reescrito
ELABORAR CONTRATO	Producción	Nuevo
REALIZAR TRÁMITE SOLICITUD DE ADHESIÓN	Negociación Inconclusa con Producción	Nuevo
REALIZAR TRÁMITE CESIÓN DE DERECHOS Y OBLIGACIONES	Negociación Terminada	Nuevo
FALTA DE PAGO DEL ADHERENTE	Negociación Terminada	Nuevo
SOLICITAR ENTREGA DEL BIEN	Negociación Inconclusa con Servicio	Nuevo
ANALIZAR SOLICITUD ENTREGA DEL BIEN	Etapas de Negociación con Producción	Nuevo
PRESENTAR CODEUDOR SOLIDARIO	Fin de Negociación	Nuevo

Escenario	Patrón	Estado del escenario al finalizar el análisis
ENTREGAR EFECTIVAMENTE EL BIEN	Fin de Negociación	Nuevo
ANALIZAR SOLICITUD CAMBIO DEL BIEN	Producción	Nuevo
PROCESAR ACEPTACIÓN DE LA ADJUDICACIÓN	Fin de Negociación con Producción	Nuevo
ESTUDIO DE LOS DATOS DEL SOLICITANTE	Producción	Nuevo
CÁLCULO DEL HABER DEL ADHERENTE	Producción	Nuevo
REGULARIZAR SITUACIÓN ADHERENTE SUSTITUIDO	Producción	Nuevo
REGULARIZAR SALDO ADJUDICATARIOS	Producción	Nuevo

Total escenarios originales:	22
Total escenarios reescritos:	14 (63%)
Total escenarios nuevos:	17
Total escenarios eliminados:	2
Total escenarios definitivos:	38
Total escenarios con patrón:	38 (100%)

SISTEMA PARA EL CONTROL DE POS GRADUACIÓN

Escenario	Patrón	Estado del escenario al finalizar el análisis
CURSAR MAESTRÍA	Negociación Terminada con Producción y Servicio	Reescrito
CURSAR DOCTORADO	Negociación Terminada con Producción y Servicio	Reescrito
CURSAR DISCIPLINA	Negociación Terminada con Servicio	Reescrito
ORIENTAR ALUMNO	El escenario abarca más de una situación	Sin cambios
DEFENDER TESIS	Negociación Terminada con Producción	Sin cambios
DEFENDER DISERTACIÓN	Negociación Terminada con Producción	Sin cambios
DICTAR CLASE	Servicio	Nuevo
INSCRIPCIÓN EN DISCIPLINA	Negociación Inconclusa	Nuevo
INSCRIPCIÓN EN EXAMEN DE LENGUA	Negociación Inconclusa	Nuevo
RENDIR EXAMEN DE LENGUA	Fin de Negociación con Producción	Reescrito
INSCRIPCIÓN EN EXAMEN DE CALIFICACIÓN	Negociación Inconclusa	Nuevo
RENDIR EXAMEN DE CALIFICACIÓN	Fin de Negociación con Producción	Reescrito

Total escenarios originales:	8
Total escenarios reescritos:	5
Total escenarios nuevos:	4
Total escenarios definitivos:	12
Total escenarios con patrón:	11 (91,7%)

**AUTOAPLICACIÓN DE LA METODOLOGÍA
(USO RECURSIVO DE LAS HEURÍSTICAS PARA LA CONSTRUCCIÓN DE LEL Y ESCENARIOS
DEL PROCESO DE CONSTRUCCIÓN DE LEL Y ESCENARIOS)**

Escenario	Patrón	Estado del escenario al finalizar el análisis
CLASIFICAR LOS SÍMBOLOS DEL LEL	Producción	Sin cambios
CONSTRUIR EL LEL	Sucesión de Patrones	Sin cambios
CONSULTAR LA LISTA DE DUDAS	Negociación Terminada con Producción	Sin cambios
DEFINIR UNA CLASIFICACIÓN PROPIA	Producción	Sin cambios
DESCRIBIR LOS SÍMBOLOS DEL LEL	Producción	Sin cambios
GENERAR LA LISTA DE SÍMBOLOS DEL LEL	Producción	Sin cambios
IDENTIFICAR FUENTES DE INFORMACIÓN	Producción con Colaboración	Sin cambios
OBTENER INFORMACIÓN	Sucesión de Patrones	Sin cambios
OBTENER LA DOCUMENTACIÓN	Colaboración	Sin cambios
ORDENAR LA DOCUMENTACIÓN DISPONIBLE	Producción	Sin cambios
REALIZAR LAS ENTREVISTAS PRELIMINARES	Negociación Terminada con Producción	Sin cambios
REDACTAR LA NOCIÓN Y EL IMPACTO	Producción	Sin cambios
REFINAR LA CLASIFICACIÓN GENERAL	Producción	Sin cambios

Escenario	Patrón	Estado del escenario al finalizar el análisis
SELECCIONAR LA CLASIFICACIÓN APROPIADA	Producción	Sin cambios
VALIDAR EL LEL	Producción con Colaboración	Sin cambios
VERIFICAR EL LEL	Producción	Sin cambios
AGREGAR JERARQUÍAS DE ESCENARIOS AL COMP. EPISODIOS	Producción	Sin cambios
AMPLIAR LA LISTA DE ESCENARIOS CANDIDATOS	Producción	Sin cambios
APLANAR EL CONTEXTO DE LAS JERARQUÍAS DE ESCENARIOS	Producción	Sin cambios
APLANAR LOS RECURSOS DE LAS JERARQUÍAS DE ESCENARIOS	Producción	Sin cambios
BUSCAR RESTRICCIONES OCULTAS	Producción	Sin cambios
COMPLETAR LOS ESCENARIOS INTEGRADORES	Producción	Sin cambios
CONSOLIDAR ESCENARIOS	Producción	Sin cambios
CONSTRUIR EL ESQUELETO DEL ESCENARIO INTEGRADOR RAÍZ	Producción	Sin cambios
CONSTRUIR LAS JERARQUÍAS DE ESCENARIOS	Producción	Sin cambios
CONSTRUIR LOS ESCENARIOS	Sucesión de Patrones	Sin cambios

Escenario	Patrón	Estado del escenario al finalizar el análisis
CONSTRUIR LOS ESQUELETOS DE LOS ESCENARIOS INTEGRADORES	Producción	Sin cambios
CONSTRUIR LOS GRUPOS DE JERARQUÍAS DE ESCENARIOS	Producción	Sin cambios
CONSULTAR EL COMPONENTE DUDAS	Negociación Terminada con Producción	Sin cambios
CREAR SUBESCENARIOS	Producción	Sin cambios
CREAR SUBESCENARIOS DE ESCENARIOS INTEGRADORES DE NIVEL INTERMEDIO	Producción	Sin cambios
CREAR SUBESCENARIOS DEL ESCENARIO INTEGRADOR RAÍZ	Producción	Sin cambios
DEFINIR LAS CAUSAS DE EXCEPCIÓN	Producción	Sin cambios
DEFINIR LAS RELACIONES DE INTEGRACIÓN	Producción	Sin cambios
DEFINIR LAS RESTRICCIONES	Producción	Sin cambios
DESCRIBIR EL COMPONENTE EPISODIOS DEL ESCENARIO INTEGRADOR	Producción	Sin cambios
DESCRIBIR LOS ESCENARIOS CANDIDATOS	Producción	Sin cambios
DETECTAR LAS CAUSAS DE EXCEPCIÓN	Producción	Sin cambios
DETECTAR ÓRDENES PARCIALES ENTRE JERARQUÍAS DE ESCENARIOS	Producción	Sin cambios

Escenario	Patrón	Estado del escenario al finalizar el análisis
DETECTAR SUBESCENARIOS	Producción	Sin cambios
DIVIDIR ESCENARIOS	Producción	Sin cambios
ELIMINAR TRANSITIVIDADES ENTRE ÓRDENES PARCIALES	Producción	Sin cambios
EMPOTRAR SUBESCENARIOS	Producción	Sin cambios
FACTORIZAR SUBESCENARIOS	Producción	Sin cambios
FUSIONAR ESCENARIOS	Producción	Sin cambios
GENERAR LA LISTA DE ESCENARIOS CANDIDATOS	Producción	Sin cambios
IDENTIFICAR ACTORES	Producción	Sin cambios
INTEGRAR ESCENARIOS	Producción	Sin cambios
ORGANIZAR LOS ESCENARIOS	Producción	Sin cambios
PARTIR ESCENARIOS	Producción	Sin cambios
REORGANIZAR LOS ESCENARIOS POR GENERALIZACIÓN / ESPECIALIZACIÓN	Producción	Sin cambios
REORGANIZAR LOS SUBESCENARIOS	Producción	Sin cambios
REORGANIZAR TEMPORALMENTE LOS ESCENARIOS	Producción	Sin cambios
VALIDAR LOS ESCENARIOS	Colaboración con Producción	Sin cambios
VERIFICACIÓN ENTRE ESCENARIOS	Producción	Sin cambios

Escenario	Patrón	Estado del escenario al finalizar el análisis
VERIFICACIÓN ENTRE LOS ESCENARIOS Y EL LEL	Producción	Sin cambios
VERIFICACIÓN INTERNA DE LOS ESCENARIOS	Producción	Sin cambios
VERIFICAR EL CONTEXTO DE LOS ESCENARIOS	Producción	Sin cambios
VERIFICAR ESCENARIOS EQUIVALENTES	Producción	Sin cambios
VERIFICAR LA EXISTENCIA DE SUBESCENARIOS	Producción	Sin cambios
VERIFICAR LA RELACIÓN ENTRE COMPONENTES DE LOS ESCENARIOS	Producción	Sin cambios
VERIFICAR LA SEMÁNTICA DE LOS COMPONENTES DE LOS ESCENARIOS	Producción	Sin cambios
VERIFICAR LA SINTAXIS DE LOS COMPONENTES DE LOS ESCENARIOS	Producción	Sin cambios
VERIFICAR LOS ESCENARIOS	Producción	Sin cambios

Total escenarios originales:	64
Total escenarios reescritos:	0
Total escenarios nuevos:	0
Total escenarios eliminados:	0
Total escenarios definitivos:	64
Total escenarios con patrón:	61 (95,5%)