

Máster en Estadística e Investigación Operativa

Título: Modelación de una Unidad de Anestesia y Pabellones quirúrgicos en un hospital Chileno usando un Lenguaje de Especificación y Descripción

Autor: Jorge Leiva Olmos

Director: Jordi Ocaña Rebull
Pau Fonseca i Casas
Jaume Barceló Bugeda

Departamento: Estadística e Investigación Operativa

Convocatoria: 10 / Junio / 2010



Facultat de Matemàtiques
i Estadística

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Facultat de Matemàtiques i Estadística
Universitat Politècnica de Catalunya

Tesis de master

**Modelación de una Unidad de Anestesia
y Pabellones quirúrgicos en un hospital
Chileno usando un Lenguaje de
Especificación y Descripción**

Jorge Leiva Olmos

Directores: Jordi Ocaña Rebull, Pau Fonseca i Casas y Jaume Barceló Bugada

Departamento de Estadística e Investigación Operativa

*El futuro pertenece a quienes creen en la belleza de sus sueños. (Anna Eleanor Roosevelt).
Con Especial agradecimiento a mi esposa, constructora silenciosa de mi presente. A mis
tutores por la paciencia, confianza, enseñanzas y horas invertidas en este proyecto. A mi
familia y a todos quienes me han apoyado durante toda la vida.*

Resumen

En este trabajo se aborda el problema de llevar a cabo una modelización formal de los procesos relacionados con la Unidad de Anestesia y Pabellones Quirúrgicos en un hospital chileno. Para realizar este modelo utilizamos el Lenguaje de Especificación y Descripción (SDL). Como resultado se obtuvo un diseño gráfico que consiste en un sistema con 10 bloques, 50 procesos, 102 canales de comunicación y 126 señales de información. También incluye el entorno del sistema que corresponde a las unidades de emergencia, servicios clínicos y unidades de apoyo. El modelo tuvo bastante éxito para documentar y comprender el conocimiento de los procesos que se desarrollan en la unidad. La metodología propuesta permite diseñar un modelo que analiza el sistema de forma modular y estándar. También ayuda en la gestión hospitalaria y facilita la simulación.

Keywords: Simulation, Modeling language, SDL, Surgical pavilions.

Abstract

In this work, addresses the problem of performing a formal modeling of the processes related to the surgical pavilions and an anesthesia unit on a Chilean hospital. To perform this modeling we used Specification and Description Language (SDL). As a result we obtained a graphic layout consisting of: 1 system, 10 blocks, 50 processes, 102 channels of communication and 126 information signals. It also includes system environment emergency units, clinical services, and support units. The model was very successful in order to document and to understand the tacit knowledge of the anesthesia and pavilions surgical units. The proposed methodology allows designing a model that permits to scan the system in a modular and standard way. It also helps in the hospital management and facilitates the simulation.

Keywords: Simulation, Modeling language, SDL, Surgical pavilions.

ÍNDICE

CAPÍTULO 1.- INTRODUCCIÓN.....	4
1.1.- Introducción	4
1.2.- Planteamiento del Problema.....	6
1.3.- Fundamentación.....	7
1.4.- Objetivos	7
1.5.- Hipótesis	8
1.6.- Metodología.....	8
1.7.- Revisión de la literatura.....	9
CAPÍTULO 2. – CONCEPTOS BASICOS DE MODELACIÓN Y SIMULACIÓN	11
2.1.- El sistema	11
2.1.- La Modelación.....	12
2.3.1.- Conceptos de modelación.....	12
2.3.1.- Modelos de simulación	15
2.3.- La Simulación.....	16
2.3.1.- Conceptos de simulación	16
2.3.2.- Modelos de simulación de eventos discretos	19
2.4.- Verificación y validación de un modelo de simulación.....	19
CAPÍTULO 3. - EL HOSPITAL Y LA UNIDAD DE PABELLONES QUIRÚRGICOS	21
3.1.- El Hospital Dr. Gustavo Fricke	21
3.2.- La demanda y la oferta	21
3.3.- La Unidad de Anestesia y Pabellones Quirúrgicos.....	24
CAPÍTULO 4.- MODELACIÓN DE LA UNIDAD PABELLÓN.....	27
4.1.- Specification and Description Language, SDL	27
4.2.- Desarrollo del Modelo con SDL	30
4.2.1.- El Sistema y su Entorno.	30
4.2.2.- El Sistema y los Bloques.....	30
4.2.3.- Los Bloques y los Procesos.....	34
4.2.4.- Los Procesos y Procedimientos.	45
4.3.- Simulación con SDLPS.....	47
4.2.5.- Validación y evaluación de la calidad de los datos.....	50
CAPÍTULO 5.- CONCLUSIONES	52
ANEXO	54
I.- Conceptos del Lenguaje de Descripción y Especificación SDL	54
A.- Introducción.....	54
B.- Algunas características.....	55
C.- La estructura del sistema.....	56
D.- Las Entidades.....	58
E.- Comportamiento del sistema.....	68

F.- La comunicación.....	69
G.- Los Símbolos	75
H.- Listas de mensajes de secuencia (MSC).....	77
I.- Metodología para el desarrollo de un modelo en SDL.....	78
J.- Las Herramientas de SDL y MSC.....	84
BIBLIOGRAFÍA.....	86

CAPÍTULO 1.- INTRODUCCIÓN

1.1.- Introducción

En la actualidad, la mayor parte de las organizaciones, empresas y procesos productivos han crecido en complejidad y han visto aumentados los requerimientos de competitividad, flexibilidad y calidad en sus actividades; por ello han debido innovar y adaptarse a los constantes cambios provocados por su pertenencia a un mercado cada vez más global. Posiblemente influenciadas por las nuevas tendencias en el sector, muchas empresas se han visto involucradas en un proceso de reingeniería de su proceso productivo (Guasch, y otros, 2002).

El panorama antes descrito, se ha experimentado en el proceso de modernización iniciado en la última década por el Estado de Chile. En el sector salud ha dado inicio a un profundo proceso de reforma a partir del año 2002, que incluye nuevos objetivos sanitarios, un plan de salud pública, un nuevo modelo de atención, la configuración de una nueva autoridad sanitaria, el aseguramiento solidario y la participación ciudadana, como ejes centrales que buscan cambios estructurales y de gestión que modifiquen sustancialmente su organización y los procesos que le dan vida.

El Gobierno ha formulado objetivos sanitarios nacionales que apuntan a establecer prioridades, definir estrategias y planificar las actividades necesarias para mejorar la salud de la población. Dentro de los objetivos se encuentra la reducción de las listas de espera para consultas de especialidades, procedimientos, exámenes, intervenciones quirúrgicas y el cumplimiento de las garantías de las patologías AUGE¹. La alta demanda, la escasez de recursos y la complejidad del sistema, dificultan el cumplimiento de estas metas sanitarias. Por ello permanentemente se impulsan estrategias a fin de satisfacer la necesidad sanitaria de la población.

El Hospital Dr. Gustavo Fricke no escapa a esta realidad. Con una población usuaria cercana al medio millón de personas, es uno de los establecimientos de alta complejidad con mayores brechas entre la demanda y oferta en Chile. Realiza más de 250.000 consultas de especialistas anuales y 13.000 intervenciones quirúrgicas (Ministerio de Salud de Chile, 2009). Las listas de espera al corte de diciembre del 2009 eran de 23.579 consultas de especialistas² y 9.820 intervenciones quirúrgicas³. Respecto a las Garantías Explícitas en Salud, durante el año 2008 se generaron en el hospital, 35.957 garantías, de las cuales 8.819 (24,5%) fueron resueltas mediante compra de servicios en el extra sistema y 2.517 (7,0%) quedaron pendientes (Ministerio de Salud de Chile, 2009) (Ministerio de Salud de Chile, 2009).

¹ AUGE: Acceso Universal de Garantías Explícitas. Corresponde a un conjunto de patologías que tienen por ley garantizado el acceso a la atención, plazos, calidad y protección financiera. Actualmente corresponden a 56 patologías y en su selección se incorporan criterios de magnitud (prevalencia e incidencia), trascendencia, efectividad, carga financiera en los hogares, área social que afecta y costo de implementación para el estado.

² 90% de los pacientes esperando más de 30 días para ser atendidos

³ 93% de los pacientes esperando más de 90 días para ser intervenidos quirúrgicamente.

La disponibilidad de horas de especialistas, camas hospitalarias, servicio de anestesia y horas de pabellón, son entre otros, los recursos más críticos que influyen en el aumento de las listas de espera y el retraso en el cumplimiento de Garantías GES⁴. Entre las distintas áreas del hospital, la Unidad de Anestesia y Pabellones Quirúrgicos (UAPQ) tiene un papel estratégico en la calidad de la atención al paciente y los objetivos sanitarios. Su alta demanda, costo y complejidad de funcionamiento, hacen que permanentemente se busquen estrategias para mejorar su rendimiento.

Para ello se deben utilizar herramientas analíticas que ayuden y faciliten la toma de decisiones, determinar cambios en los procedimientos, en los flujos de información, en la organización y en las políticas de funcionamiento (Guasch, y otros, 2002). Esto permite reaccionar y adaptarse a las contingencias y constantes cambios a los que se ven sometidos. Entre las herramientas más utilizadas como apoyo a la gestión administrativa están la estadística, la modelación, la simulación y la optimización matemática.

Un requerimiento fundamental en este sentido es que se haya especificado y documentado cada una de las actividades, tareas y responsabilidades del sistema. Lamentablemente, algunos hospitales no han descrito el modelo de operación y por otra parte hay un gran desconocimiento de herramientas que apoyen el manejo administrativo y logístico de la institución.

El presente trabajo intenta dar pasos en este sentido, describiendo la UAPQ del Hospital Dr. Gustavo Fricke mediante el lenguaje de especificación de sistema Specification and Description Language (SDL). Esto ha permitido documentar y comprender el funcionamiento de la Unidad, diseñar un modelo que permite analizar el sistema en forma modular y estándar, protocolizar tareas e información transmitida por canales informales, facilitar la simulación estadística y apoyar la gestión hospitalaria.

⁴ GES: Garantías Explícitas en Salud. Corresponden a las Garantías de las patologías AUGE.

1.2.- Planteamiento del Problema

Una condición necesaria para la mejora del uso de la Unidad de Anestesia y Pabellones Quirúrgicos es conocer los procesos que en ella se desarrollan (actividades, funciones, relaciones y comunicaciones, entre otros elementos). Algunos de los motivos que dificultan la especificación y documentación del sistema son:

- Algunos hospitales no tienen descrito el modelo de funcionamiento, no cuentan con un manual de procedimientos o el manual existente no puede ser actualizado con la frecuencia que se requiere.
- Existen funciones y tareas no protocolizadas, transmitidas muchas veces por canales informales o asumidas de forma implícita.
- Existen actividades de procesos (clínicos y administrativos) sin cuantificar (demanda, oferta, tiempo de ocupación y recursos asociados).
- Los lenguajes utilizados para la descripción de sistema no siempre permiten la participación y entendimiento de un grupo multidisciplinario.
- La complejidad del sistema hacen que esta descripción sea extensa y ardua.
- El gran volumen, la falta de disponibilidad y estructura de datos (acopio, estandarización y errores de registros), hacen difícil el acceso a la información.
- La escases de recursos disponibles (económicos, humanos, materiales y tiempo) para realizar este tipo de tareas, hacen que se privilegien frecuentemente la necesidad y actividad sanitaria.
- El desconocimiento de nuevas herramientas que permiten aportar al manejo administrativo y logístico de la institución.

En la actualidad, en la UAPQ del Hospital Dr. Gustavo Fricke, no existen estudios que describan sus procesos de modo formal, ni un manual de procesos, funciones y perfiles. Lo anterior es necesario para comprender la estructura y el comportamiento del sistema. Una vez formalizado el proceso, se puede estudiar la aplicación de técnicas que cuantifiquen los tiempos de espera, demanda, oferta y tiempos de ocupación, entre otros, en cada etapa del proceso.

Dada su complejidad operativa, este trabajo se centra en la definición del problema, desarrollar el modelo conceptual del sistema de la UAPQ y simular una parte del modelo, para evaluar su idoneidad mediante (validación conceptual y validez). Esto sentará las bases para que en un futuro se pueda seguir desarrollando el modelo, evaluar propuestas de mejoras al sistema y estudiar la optimización de su uso (por ejemplo, la programación de horas de cirugías en pabellón, la cual actualmente se realiza mediante una mecánica empírica y que carece de una metodología científica), entre otros estudios.

1.3.- Fundamentación

El trabajo se fundamenta en cuatro ejes:

1. Los requerimientos ministeriales y el nuevo modelo de autogestión para los hospitales en Chile, basado en la búsqueda de eficacia, eficiencia, calidad y estabilidad económica del establecimiento.
2. Los pabellones quirúrgicos son considerados como una de las áreas más críticas de un hospital, tanto por la gran cantidad de procesos que se desarrollan paralelamente en él, los recursos involucrados y la complejidad de su funcionamiento.
3. La disposición del hospital para desarrollar el estudio, mediante el acceso a los datos y a la información de los procesos.
4. Un modelo es un producto en sí (Brade, 2000) y muy útil para conocer el panorama completo del sistema.

1.4.- Objetivos

Objetivo General

Desarrollar un modelo teórico de la actividad de la Unidad de Anestesia y Pabellones Quirúrgicos.

Objetivos Específicos

1. Recopilar información respecto al funcionamiento, oferta, recursos, necesidades y restricciones actuales de la unidad de pabellón.
2. Describir el sistema en un lenguaje formal que permita su estudio en forma estructurada y modular.
3. Especificar una parte del sistema, para permitir una simulación.

1.5.- Hipótesis

Las hipótesis de estudio son:

- Los procesos de la Unidad de Anestesia y Pabellones Quirúrgicos se pueden representar mediante un modelo.
- Specification and Description Language (SDL) permite construir el modelo.
- Esta especificación del sistema permitirá protocolizar tareas implícitamente conocidas.
- El modelo facilitará la simulación informática del sistema.

1.6.- Metodología

El estudio abarcó los tipos cualitativos (bibliográficos y documentación de procesos) y cuantitativos (de campo y de laboratorio). Comenzó en mayo de 2010 y abarcó una duración de 12 meses de trabajo.

La información de los procesos fue entregada por el equipo responsable de la UAPQ. La recogida y validación de los datos se realizó desde las base de datos del hospital.

En el diseño del modelo se utilizó el lenguaje de especificación y descripción de sistemas (SDL), específicamente su versión gráfica SDL/GR (Telecommunication standardization sector of ITU, 1999), utilizando Microsoft Visio ® para representar los diagramas. Sus características y la justificación de su elección, se presentan el Capítulo 4.

1.7.- Revisión de la literatura

Al estudiar qué trabajos se han realizados respecto a la gestión hospitalaria relacionada con la modelación, se encontró que muchas investigaciones se han llevado a cabo sobre el funcionamiento, uso y mejora de los pabellones quirúrgicos. En su mayoría se enfocan a la revisión de literatura (Cardoen, et al., 2010), descripción del sistema (Baumgart, et al., 2007), cuantificar tiempos de ocupación (Dexter, et al., 2003), evaluar la capacidad o uso óptimo (Ballard Sarah M, et al., 2006) y (Dexter, et al., 1999), entre otros. Para ello, emplean ocupan técnicas de modelación, simulación y optimización, en forma individual o combinada. La técnica de descripción de sistemas más utilizada es el Diagrama de Flujo (Baumgart, et al., 2007). Respecto a las técnicas de simulación más utilizadas están: Teoría de Colas (Lucas, et al., 2001), Simulación Discreta (Dexter, et al., 2003) y Método de Monte Carlo (Dexter, et al., 1999). Dentro de los programas informáticos más utilizados para la simulación se encuentran (i) lenguajes de programación, tales como Matlab (Shaw, et al., 2007), C (Denton, et al., 2006), Visual Basic y Java (Persson, et al., 2009) y (ii) aplicaciones, tales como Microsoft Excel (Dexter, et al., 2003), Arena® de Rockwell (Ballard Sarah M, et al., 2006), Simio® (Zheng, et al., 2010), Extend® (Fei, et al., 2006), MedModel® (Lowery, et al., 2009), Witness® (Niu, et al., 2007) y Network® (Su, et al., 2010). Las herramientas de optimización más empleadas son la Programación Lineal (Dexter, et al., 2002), Programación Entera (Cardoen, et al., 2006), Programación no Lineal Entera (Jeang, et al., 2010), Programación Entera Mixta (Jebali, et al., 2006), Programación Estocástica (Denton, et al., 2006) y Métodos Heurísticos (Fei, et al., 2006).

Por último, otras técnicas empleadas son el Análisis Envolvente de Datos (Cerda, et al., 2001), Modelos de Ecuaciones Estructurales (Dexter, et al., 1999), Análisis de Costo Mínimo (Dexter, et al., 2000), Minería de datos (Steins, et al., 2010) y Modelos DSS (Sistemas de Soporte de Decisión) (Kusters, et al., 1996).

Muchos estudios caracterizan a la UAPQ como un recurso con oferta permanente, indiferenciación de las salas de cirugías y estableciendo que entidades se relacionan, pero no como se relacionan. Esto presenta una serie de dificultades a la hora de describir adecuadamente el sistema, estimar la demanda, los tiempos de ocupación y de ocio de los pabellones quirúrgicos, tiempos de esperas e incorporar elementos de incertidumbre y comportamiento humano (Sibbel, et al., 2001) . Nuestro enfoque parte de considerar la UAPQ como un sistema (con procesos y subprocesos clínicos y administrativos), cuyos componentes interaccionan de forma especificada explícitamente (canales de comunicación, señales, datos, direccionalidad, temporalidad, jerarquía, etc.). En concreto, para superar algunas de las dificultades antes expuestas, se incorporaron en la especificación de la UAPQ las siguientes características:

- La especificación de los procesos y subprocesos administrativos y clínicos.
- La especificación de los canales, señales, direccionalidad, temporalidad y jerarquía que relacionan o poseen los procesos.
- La categorización de los pabellones. Dada la mayor complejidad de las patologías, el avance tecnológico y las nuevas técnicas quirúrgicas, algunas de las salas de cirugías han sido diseñadas, equipadas y/o destinadas para determinadas

prestaciones (por especialidad, tales como cirugías cardiovasculares, oftalmológicas o traumatológicas, entre otras o según complejidad, tales como urgencias, cirugía menor, cirugía mayor ambulatoria o cirugía mayor no ambulatoria), mientras que otras salas siguen siendo indiferenciadas (es decir, se pueden realizar todo tipo de intervenciones quirúrgicas).

- La especificación modular del sistema, sin perder la generalidad (es decir, modelar y describir el sistema como un todo).
- La descripción mediante un lenguaje gráfico que permita la participación y entendimiento de un grupo multidisciplinario (médicos, personal clínico, administrativos, etc.).
- La elección de un lenguaje de especificación que permita la simulación modular, facilitando así, un estudio por etapas.

Estos elementos dieron lugar a un mejor conocimiento y descripción del comportamiento del sistema.

CAPÍTULO 2. – CONCEPTOS BASICOS DE MODELACIÓN Y SIMULACIÓN

2.1.- El sistema

Un sistema puede definirse como una colección de objetos o entidades que interactúan entre sí para alcanzar un cierto objetivo (Schmidt, et al., 1970). Una característica fundamental de un sistema es que se puede establecer perfectamente lo que pertenece a él y lo que no, también se puede especificar cómo interacciona con el entorno. El planteamiento es jerárquico, ya que si se separa una nueva parte se estará ante un nuevo sistema. Otra propiedad es el hecho que puede ser controlado y observado, sin que estos términos tengan el mismo significado que el que se aplica en teoría de control (Guasch, y otros, 2002).

Un sistema puede definirse también como un conjunto de variables (Asby, 1956). De esta forma, obtener un subsistema, no implica necesariamente tomar una parte del sistema inicial, sino un subconjunto de las variables del mismo. Es decir, la división tiene lugar a nivel de abstracción matemática y en el contexto del modelado. Las interacciones del mismo con el mundo natural se pueden dividir en dos categorías:

- Variables generadas por el entorno y que tienen influencia en el comportamiento del sistema. Normalmente se las referencia como entradas.
- Variables que están determinadas por el sistema y que, a su vez, pueden tener influencia en el comportamiento del entorno. Se llaman salidas.

Bajo esta forma de actuar, se pueden asignar valores a alguna de las entradas al sistema y observar el comportamiento del mismo considerando los valores de las salidas. Esto lleva a una nueva definición: Un sistema es una fuente potencial de datos (Zeigler, 1976).

El estado de un sistema puede ser definido como el conjunto de variables necesarias para caracterizar o describir todos aquellos aspectos de interés del sistema en un cierto instante de tiempo (Law, et al., 2001). A estas variables las denominamos variables de estado. Cabe reiterar que el número y tipo de éstas vienen condicionados por los objetivos del estudio (Guasch, y otros, 2002).

Sin perder generalidad y considerando como finalidad tan solo el estudio del comportamiento de un sistema en el dominio temporal, se pueden clasificar los sistemas en: continuos, discretos, orientados a eventos discretos y combinados, atendiendo tan sólo a la relación entre la evolución de las propiedades de interés y la variable independiente tiempo. Una breve descripción de esta clasificación es:

- Sistemas continuos: las variables del estado del sistema evolucionan de modo continuo a lo largo del tiempo

- **Sistemas discretos:** se caracterizan porque las propiedades de interés del sistema cambian únicamente en un cierto instante o secuencia de instantes, y permanecen constantes el resto del tiempo. La secuencia de instantes en los cuales el estado del sistema puede presentar un cambio, obedece normalmente a un patrón periódico.
- **Sistemas orientados a eventos discretos:** al igual que los sistemas discretos, se caracterizan porque las propiedades de interés del sistema cambian únicamente en una secuencia de instantes de tiempo y, podemos considerar que permanecen constantes el resto del tiempo. La secuencia de instantes en los cuales el estado del sistema puede presentar un cambio, obedece a un patrón aleatorio.
- **Sistemas combinados:** aquellos que combinan subsistemas que siguen filosofías continuas o discretas, respectivamente. Es el caso de los sistemas que poseen componentes que deben ser necesariamente modelados según alguno de dichos enfoques específicos.

2.1.- La Modelación

2.3.1.- Conceptos de modelación

La modelación es el proceso de construir un modelo. Permite describir el funcionamiento de los procesos. Es muy útil para conocer el panorama general. Muestra cómo se combinan los distintos componentes para producir algún resultado. Al especificar las relaciones que hay entre los componentes del sistema, se facilita la comprensión de los vínculos entre las diversas actividades y el impacto que tienen entre sí (Rico, et al., 2008).

Un modelo se define como una representación (normalmente simplificada) de un sistema en un instante de tiempo o espacio concreto, realizado para comprender el sistema real. Se desarrolla siempre a partir de una serie de aproximaciones e hipótesis y, consecuentemente, representa tan sólo parcialmente la realidad. Se construye para una finalidad específica y debe ser formulado para que sea útil para ese fin, por tanto, un mismo sistema real puede tener varios modelos (Guasch, y otros, 2002).

Existen muchos tipos de modelos (mentales, lingüísticos, físicos, gráficos, matemáticos, etc.). Dadas las características de la UAPQ, el objetivo de este trabajo es construir un modelo gráfico que permita la simulación digital (de aquí en adelante, llamada simplemente simulación). Para ello es necesario formalizar el conocimiento que se tiene del sistema de modo conciso, sin ambigüedades (interpretación única), y que puedan ser procesados informáticamente. El lenguaje SDL (se presentará más adelante) cumple con estas cualidades.

Algunas de las características que debe cumplir un buen modelo son (Guasch, y otros, 2002):

- Un modelo es un objeto o concepto que utilizamos para representar cualquier otra entidad (un sistema). Así pues, mediante un proceso de abstracción, se muestran en un formato adecuado las características de interés de un objeto (sistema) real o hipotético.
- Un modelo es una representación simplificada de un sistema que nos facilitará explicar, comprender, cambiar, preservar, prever y posiblemente controlar el comportamiento de un sistema.
- Un modelo es el sustituto de un sistema físico concreto.
- Un modelo debe representar el conocimiento que se tiene de un sistema de modo que facilite su interpretación, formalizando tan sólo los factores que son importantes para los objetivos de modelado.
- Un modelo debe ser tan sencillo como sea posible, porque el desarrollo de modelos universales es impracticable y poco económico, siempre y cuando represente adecuadamente los aspectos de interés.
- Un modelo tiene que ser por necesidad un compromiso entre la simplicidad y la necesidad de recoger todos los aspectos esenciales del sistema en estudio. Debe ser lo suficientemente sencillo como para facilitar su mantenimiento, adaptación y reutilización.

Tomando en cuenta el conocimiento y la formalización de los procesos, se pueden destacar dos tipos de modelos:

- Modelo Mental: Basado en el conocimiento que se tiene sobre un aspecto de la realidad adquirido a través de la experiencia e intuición, del cual se extraen aquellas características esenciales para representar el aspecto considerado. Este es el modelo que comúnmente se maneja en la UAPQ.
- Modelo Formal: Basado en las hipótesis empleadas en los modelos mentales, estableciendo a partir de ellas las relaciones formales que definen el comportamiento del aspecto de la realidad en cuestión. Utiliza la capacidad del computador, que aunque no es capaz de establecer las relaciones por sí mismo, si está capacitado para desarrollar las consecuencias dinámicas de las interacciones del sistema que representa el modelo.

Un modelo formal es más explícito que un modelo mental. Su implementación en el computador produce el modelo computarizado, tal como muestra en la figura 1:

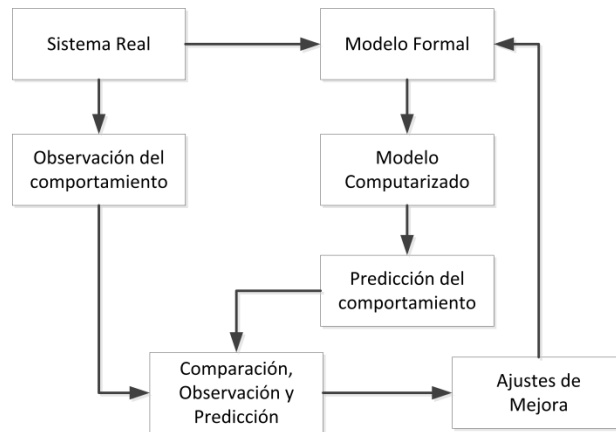


Figura 1. Modelización de un sistema (Fernandez de Cañete, 2006).

La adecuada definición del modelo formal facilita la simulación y con ella, la evaluación de la verificación y validez de los modelos construidos y experimentación, de modo de evaluar propuestas de mejoras al sistema.

Hay dos puntos de vista a la hora de establecer un modelo de un sistema (Fernandez de Cañete, 2006):

- **Conductista:** La construcción del modelo se realiza a partir del procesamiento de datos históricos de la evolución del sistema. Se trata de ajustar un modelo previamente elaborado a los datos disponibles. No se pretende establecer la estructura interna del sistema, sino que se supone una estructura interna a priori que reproduzca el comportamiento observado del sistema.
- **Estructuralista:** La construcción del modelo se realiza siguiendo un análisis cuidadoso y detenido de los distintos elementos que intervienen en el sistema observado. De aquí se extrae la lógica interna del modelo que conduce a la obtención de la estructura, realizándose posteriormente un ajuste de los parámetros libres del modelo con los datos históricos.

Y respecto a las fases en la construcción de un modelo, estas corresponden a:

- **Conceptualización:** Consiste en la obtención de una comprensión mental de un cierto fenómeno del mundo real (información a través de la opinión de expertos y la literatura al respecto, definición de aspectos del problema a resolver, identificación de elementos del sistema para establecer sus límites, entre otros.)
- **Formulación:** Trata de representar los elementos manejados en la fase anterior mediante un lenguaje formal (gráfico, matemático, implementación computacional, etc.)

- Evaluación: consiste en el análisis del modelo y su sometimiento a criterios de aceptabilidad (ensayos mediante simulación de las hipótesis que sustenta el modelo y su consistencia, análisis de la sensibilidad ante modificaciones, etc.).

El proceso de modelación no es lineal, pasa por sucesivas etapas, desarrollando progresivamente nuevos o mejorados modelos de acuerdo con un cierto criterio de aceptabilidad. El proceso de modelado puede dividirse entonces en dos etapas: (i) etapa inicial y (ii) etapa de perfeccionamiento. Las sucesivas etapas consistirán en una eliminación progresiva de las hipótesis más simplificadoras de manera que el modelo se aproxime cada vez más a la realidad.

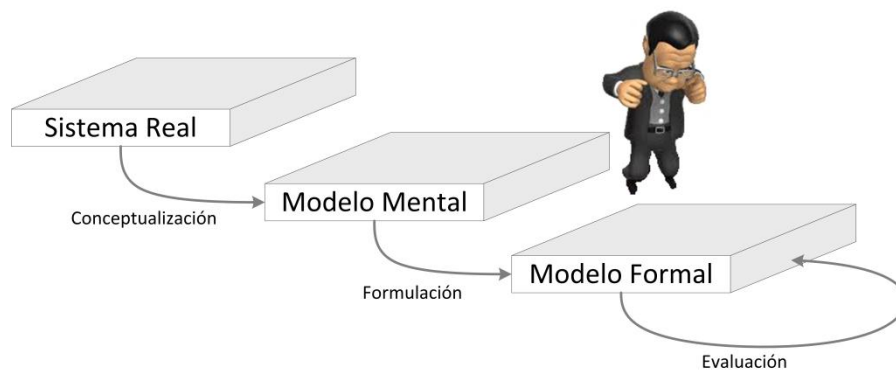


Figura 2. Fases en la construcción de un modelo.

2.3.1.- Modelos de simulación

En el entorno de la simulación y atendiendo a las características que debe poseer estos modelos y los objetivos del estudio, los modelos suelen clasificarse de las siguientes formas (Guasch, y otros, 2002):

- Modelos estáticos respecto a modelos dinámicos:
 - Los Modelos Estáticos suelen utilizarse para representar el sistema en un cierto instante de tiempo; y en su formulación no se considera el avance del tiempo. Este tipo de modelos es muy útil cuando el sistema se encuentra en equilibrio (no evoluciona respecto al tiempo). Si se cambia el punto de equilibrio alterando uno o más de los valores del sistema, el modelo permite deducir el resto de los valores, pero no muestra la manera en que cambiaron.
 - En contraposición a los modelos estáticos, los modelos dinámicos permiten deducir cómo las variables de interés del sistema en estudio evolucionan respecto al tiempo.
- Modelos deterministas respecto a modelos estocásticos
 - Un modelo se denomina determinista si su nuevo estado puede ser completamente definido a partir del estado previo y de sus entradas. Es decir, ofrece un único conjunto de valores de salida para un conjunto de entradas conocidas.

- Los modelos estocásticos requieren de una o más variables aleatorias para formalizar las dinámicas de interés. En consecuencia, el modelo no genera un único conjunto de salida cuando es utilizado para realizar un experimento, sino que los resultados generados son utilizados para estimar el comportamiento real del sistema.
- Modelos continuos respecto a modelos discretos
 - Los modelos continuos se caracterizan por representar la evolución de las variables de interés de forma continua.
 - De modo análogo a la definición de modelos continuos, los modelos discretos se caracterizan por representar la evolución de las variables de interés de forma discreta.

Es importante notar a partir de la clasificación de modelos realizada, que es posible describir un sistema continuo mediante un modelo discreto y, al revés, también es posible describir un sistema discreto mediante un modelo continuo. La decisión de utilizar un modelo continuo o un modelo discreto depende de los objetivos particulares de cada estudio y no tanto de las características del sistema.

2.3.- La Simulación

2.3.1.- Conceptos de simulación

La simulación es una técnica que permite imitar (o simular) en un computador el comportamiento de un sistema físico o teórico según ciertas condiciones particulares de operación. En el caso de algunos problemas reales es una metodología indispensable para resolverlos (por costo, problemas éticos, imposibilidad de replicación, etc.).

Es una metodología que permite apoyar la toma de decisiones, ya sea en el diseño de sistemas, antes que éstos sean construidos o probando políticas de funcionamiento, antes que éstas sean implantadas. La simulación, no resuelve los problemas por sí misma, sino que ayuda a identificar los problemas relevantes y evaluar cuantitativamente las soluciones alternativas. Una desventaja es que no entrega resultados exactos, pero permite modelar sistemas complejos (muchas veces es mejor una respuesta aproximada al problema correcto que una respuesta correcta al problema aproximado). Es la técnica de modelado estocástico más útil, de mayor reconocimiento en diversos campos de aplicación. Como regla general, la simulación es apropiada cuando se requiere desarrollar un modelo analógico es muy difícil o quizás aún imposible, el sistema tiene una o más variables aleatorias relacionadas, la dinámica del sistema es extremadamente compleja, el objetivo es observar el comportamiento del sistema sobre un período de tiempo o se requiere representación mediante animación.

Siempre es preferible estudiar y experimentar con el sistema real. Cuando esto no es posible, se puede desarrollar un modelo físico o teórico que describa las dinámicas de interés. En el caso de este último tipo de modelo, se implementa en un simulador

(programa o aplicación informática) para poder experimentar con el modelo del sistema y luego analizar los resultados, tal como se muestra en la figura 3.

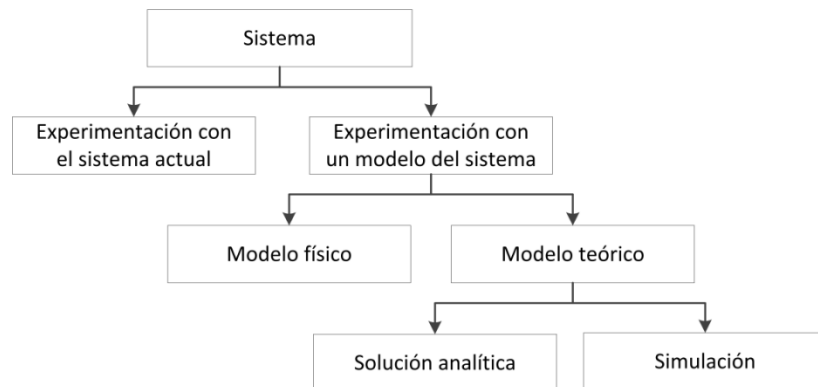


Figura 3. Aplicación de la simulación en el estudio de un sistema (Law, et al., 2001).

Para llevar a cabo un estudio de simulación, se debe prestar atención a una variedad de elementos que van desde la definición del problema, realización de diseños experimentales hasta la gestión de personal. Los pasos (y las relaciones entre ellos) para realizar este tipo de estudio se describen a continuación y se presentan en la figura 4 (Law, et al., 2001):

1. *Formular el problema y el plan de estudio.* Se deben especificar los objetivos y delimitar los aspectos del estudio.
2. *Recopilar datos y formular el modelo.* La información y datos permiten especificar los procedimientos operativos y determinar la distribución de probabilidades de las variables involucradas. Un buen punto de partida es que el modelo solo tenga los elementos esenciales del sistema y posteriormente ir incorporando reglas o detalles que lo hagan más sofisticado.
3. *Evaluar la validez del modelo teórico construido hasta el momento.* Debe incorporarse la participación de las personas que están involucradas en los procesos del sistema. La distribución de probabilidad de las variables estudiadas, se pueden evaluar con test de bondad de ajuste.
4. *Construir un programa y verificarlo.* Esto implica traducir el modelo formal o teórico en términos de un programa informático. En la elección un lenguaje de programación o programa de simulación, deben incorporarse los propósitos, el tamaño y complejidad de implementación.
5. *Hacer pruebas piloto.* Estas serán empleadas para evaluar el modelo en el siguiente paso (6).
6. *Evaluar la validez del modelo computacional construido hasta el momento.* Se debe comprobar si el modelo construido funciona correctamente (está libre de errores) y

si es válido como representación del sistema para los propósitos del estudio, es decir, comprobar que las respuestas que proporcionados por el modelo a las preguntas pueden ser aceptadas como válidas (sensibilidad del modelo).

7. *Diseños de experimentos.* Identificar los factores de diseño a simular como si se estuviera trabajando con el sistema real. Estos se traducen evaluar el funcionamiento y/o cambios en los procesos evaluado a través de experimentos informáticos.
8. *Hacer pruebas de producción.* Especificar el procedimiento de la toma de muestra experimental para recopilar los datos para el análisis estadístico que proporcionará las respuestas esperadas. Realizar estas pruebas.
9. *Analizar los datos de salida.* Aplicar técnicas estadísticas para evaluar los resultados.
10. *Documentar, presentar e implementar los resultados.* Debido a que los modelos de simulación son aplicados en más de una ocasión, es importante documentar los supuestos que se incluyeron en el modelo, la implementación informática y cualquier otro elemento que sea de interés en el estudio.

Por último, un estudio de simulación cuyos resultados nunca son implementados, es probablemente un fracaso, por ello, resultados altamente creíbles tienen más posibilidad de ser usados.

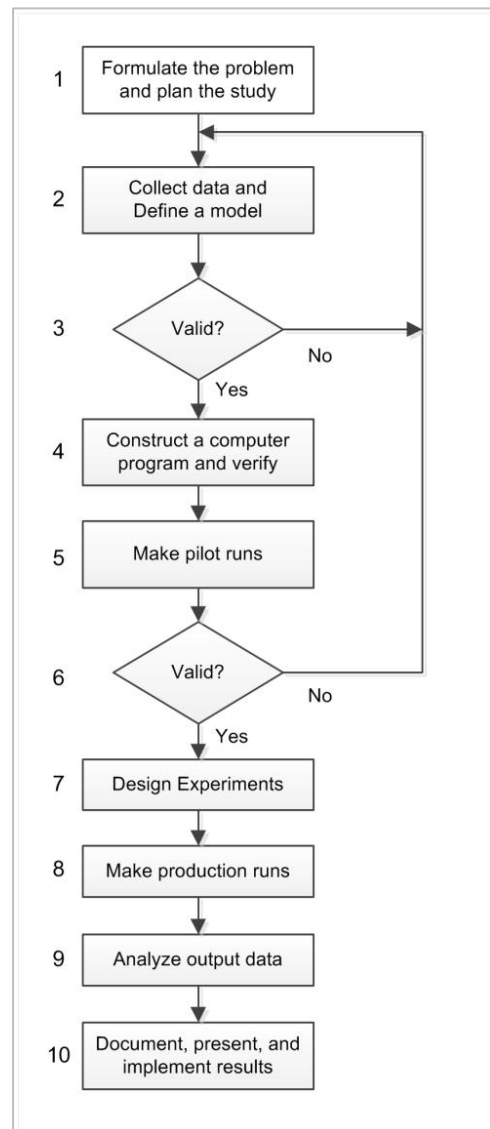


Figura 4. Pasos en un estudio de simulación (Law, et al., 2001)

2.3.2.- Modelos de simulación de eventos discretos

Los Modelos de eventos discretos son modelos en los que las variables de estado cambian de valor en instantes no periódicos del tiempo. Estos instantes de tiempo se corresponden con la ocurrencia de un evento. Por tanto, un evento se define como una acción instantánea que puede cambiar el estado del modelo (Guasch, y otros, 2002). Dentro de una taxonomía de modelos de simulación, los modelos de eventos discretos se clasificarían dentro de los modelos estocásticos, dinámicos y discretos, tal como muestra la figura 5.

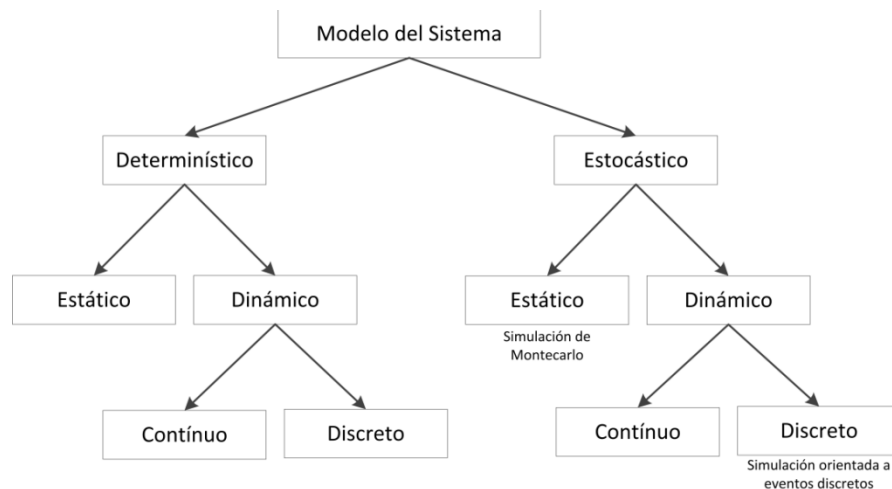


Figura 5. Taxonomía de modelos de sistemas (De la Fuente).

La orientación en este trabajo consiste en aplicar una metodología de modelado de sistemas orientada a eventos discretos.

2.4.- Verificación y validación de un modelo de simulación.

El modelo en sí, es un producto (Brade, 2000). Como se mencionó previamente, las características esperadas son la simplicidad y la capacidad de representar el comportamiento de sistema real. Esta representatividad se evalúa mediante la validación y verificación del modelo. La validación está relacionada con la correspondencia entre el modelo teórico y la realidad. La verificación evalúa la consistencia interna del modelo computacional en relación con el modelo teórico.

Una versión simplificada del proceso de desarrollo de un modelo simulación se presenta en la figura 6. La entidad problema es el sistema (real o propuesto), el modelo conceptual es la representación matemática-lógica-verbal (gráfica) de la entidad problema para un particular estudio y el modelo computarizado es el modelo conceptual desarrollado en una aplicación informática o creado con un lenguaje de programación (Sargent, 2007).

El modelo conceptual se desarrolla a través de diferentes fases de análisis y modelamiento, incluyendo el comportamiento, la estructura y los datos del sistema. El modelo computarizado se desarrolla mediante programación informática o un software de aplicación, usando el modelo conceptual. Estas fases permiten obtener información del modelo a través de la experimentación, que permite validar el modelo conceptual y verificar el modelo computarizado (Sargent, 2007). La validación del modelo conceptual se define como la determinación de que las teorías y las hipótesis que sustentan el modelo conceptual son correctas y que la representación del modelo de la entidad problema es "razonable" para la finalidad prevista del modelo. La verificación del modelo computarizado se define como el aseguramiento de que el desarrollo informático (con la programación o software) y el modelo conceptual es la correcta. La validación operacional se define como la determinación de que las salidas del modelo computacional tienen una precisión suficiente para el propósito previsto sobre el dominio de aplicabilidad del modelo. La validez de los datos se preocupa de disponer de los datos (en calidad y cantidad) necesarios para la construcción de los modelos. La evaluación de los experimentos realizados determina si son los adecuados y correctos (Sargent, 2007). La figura 5 muestra una versión simplificada de un proceso de modelación, donde se presentan las fases descritas previamente.

Unidad de Anestesia y Pabellones Quirúrgicos del Hospital Dr. Gustavo Fricke (Chile) no tiene descrito el modelo de funcionamiento, existen tareas y funciones transmitidas por canales informales y no documentadas. Este trabajo se enfoca a definir el problema, desarrollar el modelo conceptual, validar los datos y simular una parte del proceso.

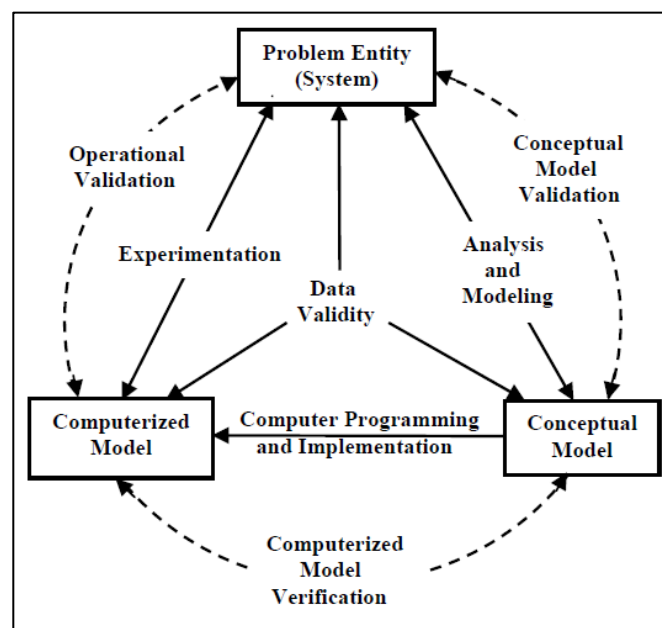


Figura 6. Versión simplificada de un proceso de modelación (Sargent, 2007).

CAPÍTULO 3. - EL HOSPITAL Y LA UNIDAD DE PABELLONES QUIRÚRGICOS

3.1.- El Hospital Dr. Gustavo Fricke

Ubicado en la región de Valparaíso, zona central de Chile, el Hospital Dr. Gustavo Fricke tuvo sus inicios el 21 de Julio de 1878 cuando se inauguró el Hospicio de Viña del Mar. En septiembre de 1879, el Hospicio adquirió el carácter de Enfermería, contando con 24 camas, lo que significó la construcción del Hospital. Sucesivas administraciones aumentaron sus dependencias, hasta que el 13 de Diciembre de 1954 se inauguró el actual edificio. El 2 de Octubre de 1969, pasó a llevar el nombre del Doctor Gustavo Fricke, gestor de su construcción.

A partir de 1980 es el Hospital Base del Servicio de Salud Viña del Mar-Quillota. Es un Hospital de alta complejidad y docente. Posee las especialidades y subespecialidades de Medicina, Cirugía, Ginecología, Obstetricia, Pediatría, Oncología, entre otras. Cuenta con tres Unidades de Emergencias (Adultos, Infantil y Gineco-Obstetra) y en la actualidad es Centro de Referencia Nacional para Trasplante Cardíaco.

Su capacidad hospitalaria es de 447 camas. Entre las estadísticas anuales se encuentran más de 24.000 egresos hospitalarios, 13.000 intervenciones quirúrgicas, 250.000 consultas médicas de especialidades, 22.000 procedimientos y 1.400.000 exámenes efectuados. Posee un Índice ocupacional cercano al 90% y con un costo de operación cercano a los 30.000.000 millones de pesos.

Su población usuaria cercana al medio millón de personas y es uno de los establecimientos a nivel nacional con mayores brechas entre la demanda y oferta. Al mes de diciembre del 2009, las listas de espera eran del orden de 9.820 intervenciones quirúrgicas y 23.579 consultas, siendo una de las más grandes de Chile.

3.2.- La demanda y la oferta

A nivel de hospital, las atenciones de salud se dividen en dos tipos:

- 1) Las atenciones de urgencias.
- 2) Las atenciones electivas.

Las atenciones de emergencia no son programables ni pueden ser postergadas. Esto hace que la oferta del hospital siempre deba ser evaluada en función de esta demanda. Las atenciones electivas (consultas, exámenes, procedimientos e intervenciones quirúrgicas) son aquellas atenciones programadas y además que conforman las listas de espera⁵.

⁵ Es el registro de todas las personas que han recibido la indicación de atención, de consulta ambulatoria de especialidad médica u odontológica, procedimiento en atención especializada e intervención quirúrgica programada, por un profesional

Las atenciones a su vez se dividen en atenciones abiertas (aquellas que no requieren hospitalización) y atención cerrada (que requieren hospitalización). Por último una vez atendidos, los pacientes pueden ser dados de alta (atención abierta) o egresados (atención cerrada) o pueden ser incorporados en las Listas de Espera, en caso de requerir una nueva prestación. La estructura de las atenciones se muestra en la siguiente figura:

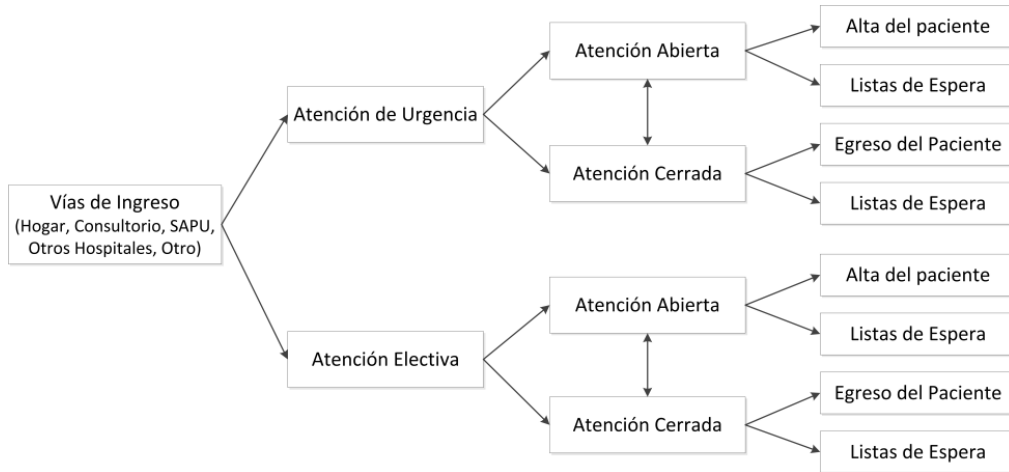


Figura 7. La estructura de las atenciones hospitalarias

Por último, los siguientes esquemas resumen el recorrido de la estancia de un paciente en el hospital, desde el momento en que requiere una necesidad, las distintas vías de ingresos, la atención abierta, la atención cerrada, listas de espera, asignación de camas para hospitalización, intervenciones quirúrgicas, compra de servicios⁶ y alta (o egreso) del paciente.

autorizado por la red de salud, teniendo documentada tal solicitud en el formulario correspondiente. La inclusión en el registro debe considerar, a todas las personas, aun cuando la atención requerida no forme parte de la cartera de servicios del establecimiento de referencia, en cuyo caso el gestor de red, debe resolver a través de la oferta de su red y la macro red.

⁶ Corresponden a las prestaciones que se compran en instituciones de salud privadas.

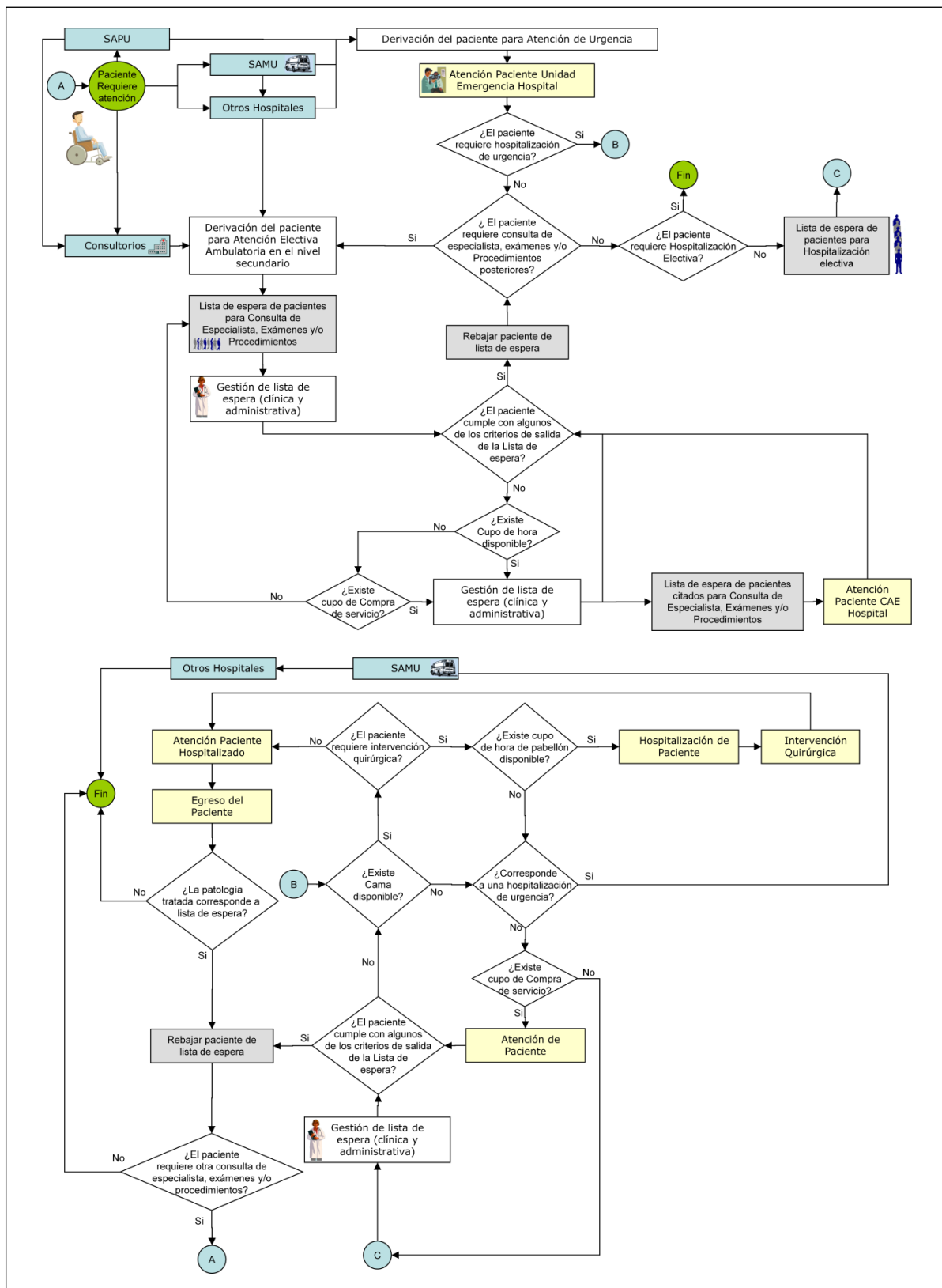


Figura 8. Recorrido de la estancia de un paciente en el hospital⁷. El proceso se inicia cuando se genera una necesidad en el paciente y finaliza con el alta. Según los requerimientos clínicos, puede haber atenciones ambulatorias, de urgencias, exámenes, procedimientos, hospitalizaciones y/o intervenciones quirúrgicas.

⁷ SAMU: Servicio Ambulatorio Médico Urgencias (ambulancias); CAE: Consultorio Adosado de Especialidades; SOME: Sección de Orientación Médico Estadística.

3.3.- La Unidad de Anestesia y Pabellones Quirúrgicos

La Unidad de Anestesia y Pabellones Quirúrgicos del Hospital Dr. Gustavo Fricke es una de las unidades estratégicas en el quehacer del hospital. Su actividad influye directamente en la atención quirúrgica de urgencia y en la reducción de listas de espera. Tiene dentro de la cadena de prestaciones procesos iniciales, intermedios y finales⁸. Entrega servicio a sus usuarios internos que son los servicios quirúrgicos y a sus usuarios externos que son los pacientes. Su infraestructura está dividida en dos secciones; 1) Pabellón central, ubicado en el segundo piso del edificio, cuenta con 12 pabellones quirúrgicos (9 de mediana y alta complejidad, 1 de baja complejidad y 2 de urgencias) y 2) Pabellón de Maternidad, ubicado en el cuarto piso del edificio, cuenta con 1 pabellón de mediana complejidad.



Figura 9. Salas de cirugías

Tal como ya se apuntó, la oferta de pabellón es limitada y se ve condicionada por la alta demanda, escasos recursos (económicos y humanos) y la complejidad de funcionamiento. Entre estos aspectos se encuentran:

- La demanda de urgencia (la cual no es programable).
- La disponibilidad horaria (hábil, lunes a viernes de 08:00 a 17:00)
- La disponibilidad de Anestésista.
- La disponibilidad de arsenal quirúrgico.
- El presupuesto y directrices institucionales.
- Los compromisos sanitarios.
- La oferta de camas hospitalarias.
- La oferta de horas de especialistas.
- Movilizaciones gremiales.
- Otros factores.

⁸ Una prestación de intervención quirúrgica, corresponde a la actividad terapéutica que implica la incisión de la piel u otros planos, con el fin de extirpar, drenar, liberar o efectuar un aseo quirúrgico ante un cuadro patológico.

Estos factores influyen en:

- El Aumento en el porcentaje de cirugías suspendidas.
- Disponer de una programación horaria de pabellones compleja.
- La utilización del recurso (en horario hábil, la exigencia ministerial es que sea al menos de un 75% de ocupación).
- La disminución del impacto de las estrategias de reducción de listas de espera a través de la utilización de pabellón para cirugías electivas en extensión horaria (17:00 a 21:00).
- Desencadenan el aumento de compras de servicios, lo que conlleva un mayor endeudamiento.
- El incumplimiento de metas sanitarias, compromisos de gestión y garantías GES.
- El aumento la insatisfacción usuaria.

Respecto al funcionamiento, se dispone de tres tipos de horarios, el de urgencia, el normal y el extensión horaria⁹. Los tipos de cirugías realizadas se desglosan en Maternal (Ginecológica y Obstétrica), Cirugía mayor no ambulatoria (donde el paciente requiere al menos un día de hospitalización), Cirugía mayor ambulatoria (en este caso el paciente está hospitalizado un tiempo máximo de 12 horas) y Cirugía menor (el paciente no requiere hospitalización), tal como muestra el siguiente esquema:

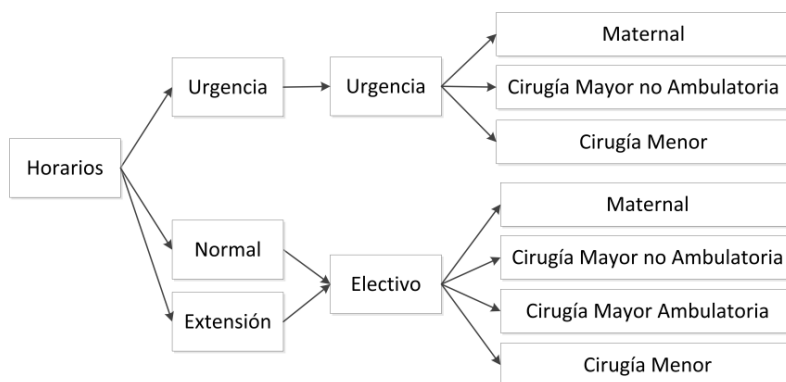


Figura 10. Oferta horaria de la Unidad de Anestesia y Pabellones Quirúrgicos

Respecto a la actividad y su entorno, la Unidad de Pabellones Quirúrgicos interactúa con los Servicios Clínicos, las Unidades de Emergencias y con las Unidades de Apoyo.

Los servicios clínicos son los encargados de gestionar las Listas de espera y por lo tanto, decidir que paciente se opera y cuando, tomando en consideración la oferta y recursos disponibles. Por otra parte, las Unidades de Emergencias destinan pacientes según la demanda que tengan.

⁹ En extensión horaria se realizan intervenciones quirúrgicas por medio de compras de servicios de horas de anestesta, cirujanos y personal asociado.

De las salas de cirugías, 10 de ellas se destinan para cirugías electivas (uno de los cuales se destinan a cirugía menor), mientras que para las Unidades de Emergencias se destinan dos salas de cirugías y dependiendo de la necesidad, podrían ampliarse a otras salas asignadas a cirugías electivas.

Por último, la función de las unidades de apoyo es colaborar en la realización del servicio, prestando para ello servicios de carácter logístico y de apoyo terapéutico y de diagnóstico. La siguiente figura resume las unidades de apoyo que interactúan con la unidad de pabellón.

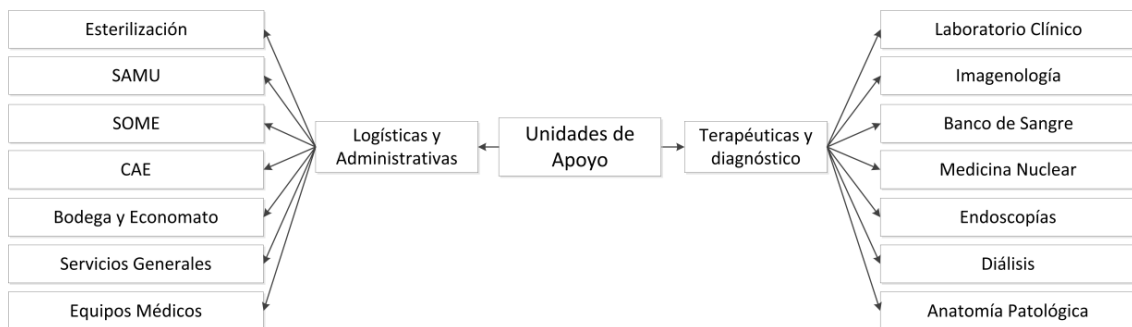


Figura 11. Unidades de apoyo que interactúan con la Unidad de Pabellón

CAPÍTULO 4.- MODELACIÓN DE LA UNIDAD PABELLÓN

Como se mencionó previamente, la modelación (de aquí en adelante, especificación) de la Unidad de Anestesia y Pabellones Quirúrgicos del Hospital Dr. Gustavo Fricke se realizará con SDL. Esto permitirá comprender y explicar el funcionamiento de la unidad, permitiendo además, generar nuevas preguntas o inquietudes acerca del comportamiento del proceso. Esta formalización del sistema puede ser una herramienta inestimable para obtener estos beneficios y puede ser considerada un producto por sí mismo.

En la Unidad de Pabellón del hospital estudiado, no existe aún un modelo formal que describa los procesos que se desarrollan. Tampoco existe un manual de procesos, instrucciones de trabajo, políticas, funciones y perfiles. A cada integrante del proceso se le instruye respecto a que actividades debe hacer, en forma empírica, transmitiendo esta información por distintos canales (principalmente verbal) y aprendiendo el proceso con las indicaciones de las personas a cargo de la unidad, compañeros de trabajo y la práctica diaria de la realización de sus tareas. Esto implica que frente a una determinada decisión o duda, sea muchas veces el criterio personal el que influya en la elección de un determinado camino. Entre los aspectos que afectan la disposición de un manual de procedimientos es la permanente contingencia que enfrenta a diario la unidad, lo que lleva a dejar los procesos administrativos en segundo plano. A pesar de ello, cuando un servicio clínico cuenta con un manual, es complicado mantenerlo actualizado, dada la complejidad y rapidez con que cambian los procesos, estrategias, datos e informaciones que se manejan.

A pesar de lo descrito anteriormente, la UAPQ es una de las unidades con mejor rendimiento en el hospital. Esto puede deberse que a pesar de que no disponen de un manual de procedimientos, su personal está entrenado para las tareas que deben realizar y manejan claramente y de forma implícita los procesos.

Especificar y formalizar las actividades de la unidad permitirá que todos sus actores conozcan e interpreten de igual manera los objetivos y tareas a realizar, disminuyendo así los errores y discrepancias en la ejecución de sus tareas.

4.1.- Specification and Description Language, SDL.

SDL es un lenguaje de descripción y especificación de sistemas¹⁰. En sus inicios fue creado para modelar sistemas de telecomunicaciones, sin embargo, se puede utilizar también en sistemas industriales, de organización y el comportamiento social, entre otros. Está incluido en la recomendación Z.100 de la recomendación de la Unión Internacional de Telecomunicaciones (UIT) respecto a Lenguajes y aspectos generales de software de Sistema de Telecomunicaciones (Telecommunication standardization sector of ITU, 1999).

¹⁰ Para obtener más información sobre SDL se puede consultar (Telecommunication standardization sector of ITU, 1999) o consultar el anexo "Conceptos del Lenguaje de Descripción y Especificación SDL".

Esta recomendación es clara, y fácil de utilizar en combinación con otros lenguajes estándar como UML.

SDL le permite elegir entre dos formas diferentes para representar los sistemas. La primera es SDL/GR (Graphical Representation), que es un lenguaje gráfico que define la estructura de control y el flujo del sistema. El segundo es SDL/PR (Phrase Representation), que es una representación textual. El éxito de SDL se puede atribuir a su forma de presentación gráfica. Esto hace que sea fácil de entender las especificaciones y expresiones diseñadas. Otro factor es la utilización del concepto de una máquina de estados finitos extendido (EFSM). SDL proporciona una manera conveniente para especificar sistemas con varias instancias de comunicación a través de EFSM. El lenguaje fue diseñado para la especificación de evento orientado, en tiempo real e interactivo de sistemas complejos.

Estos sistemas pueden incluir diferentes actividades concurrentes que utilizan señales que viajan a través de los canales, para realizar la comunicación. SDL se basa en la definición de cuatro niveles (un tipo de diagrama para cada nivel) para describir la estructura y el comportamiento del modelo: El sistema, los bloques, los procesos y procedimientos. Los bloques y los procesos se llaman agentes, el bloque exterior se llama sistema y es un agente en sí mismo. Un sistema de SDL está formado por uno o más agentes y sus vías de comunicación. Un agente se comunica con el medio ambiente exterior. Los agentes contienen otros agentes y por lo tanto es posible definir una estructura jerárquica. La figura 12 muestra esta jerarquía. La comunicación se basa en el intercambio de mensajes asíncrono.

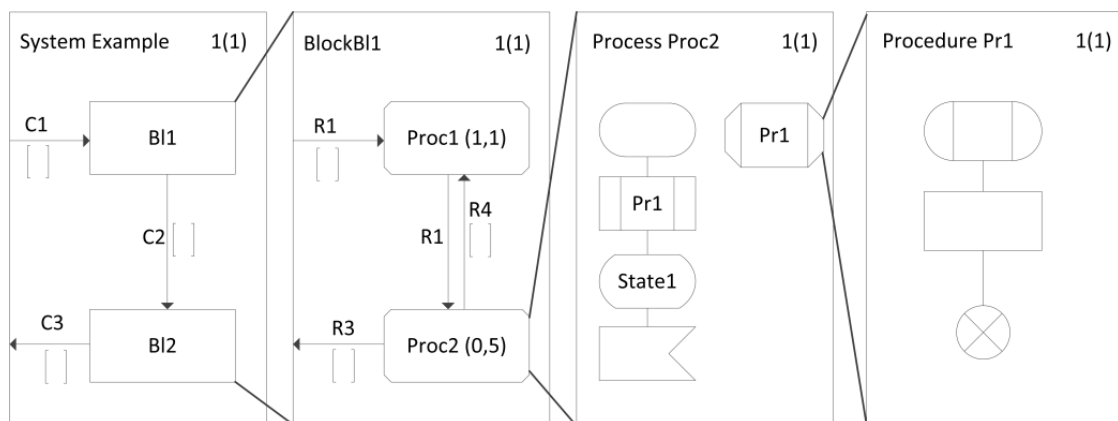


Figura 12. Estructura de un sistema en lenguaje SDL (Doldi, 2003).

SDL extiende el paradigma EFSM en dos aspectos importantes (El Barachi, et al.):

- Cada agente tiene un puerto de entrada en los cuales las colas de las señales siguen una disciplina FIFO (primero en llegar-primero en salir).
- Los datos pueden ser recibidos por las señales de entrada, almacenados en variables, manipulados, utilizando expresiones, que sirve para decidir cómo se comportará el agente y ser enviados por las señales de salida.

La arquitectura y el comportamiento son dos características independientes en el sistema. A fin de describir plenamente las propiedades estructurales y de comportamiento de un sistema, SDL se basa en tres modelos semánticos. Estos son:

1. Modelo de arquitectura, el cual consta de:
 - a. Una descomposición en términos de interconexión de las entidades funcionales, tales como: Sistema, bloques, sub-bloques, procesos, procedimientos y canales.
 - b. Una organización jerárquica siguiendo el diseño de arriba hacia abajo.
2. Modelo de comportamiento, el cual describe:
 - a. Un conjunto de procesos cooperativos, donde la comunicación se realiza a través del intercambio de señales y variables.
 - b. Cada proceso se describe por medio de EFSM.
3. Modelo de datos, el cual describe:
 - a. Las estructuras de datos manipulados por los procesos.
 - b. Los tipos abstractos de datos.

Las principales ventajas que tiene describir un sistema mediante SDL es que permite:

- Documentar, comprender y modelar sistemas simples y complejos mediante su entorno gráfico.
- Representar de manera detallada, la estructura, el comportamiento y las relaciones existentes entre los diferentes elementos del proceso.
- Simplificar la comprensión del modelo (y por tanto la comprensión del sistema).
- Trabajar en un lenguaje estándar, lo que facilita traducir la descripción del sistema en otros lenguajes.
- Ayuda a verificar y validar el modelo. Debido a que tiene una semántica completa, la descripción del sistema puede ser rápidamente chequeada y depurada.
- Separar las funcionalidades y los aspectos ambientales.
- Representar el sistema en forma modular, lo que facilita estudiar el problema descomponiéndolo en partes y en forma general.
- Formalizar los canales de comunicación y los mensajes entre las distintas entidades que interactúan en el sistema.
- Conocer el comportamiento del sistema, manteniendo la independencia entre el modelo y la herramienta seleccionada en caso de llevar a cabo una simulación.
- Contar con una serie de herramientas que permiten evaluar, editar, simular y mejorar la descripción y especificación del sistema mediante SDL. Algunas de estas herramientas se presentan en el anexo.

Estas características hacen de SDL un lenguaje de especificación completo, no ambiguo y de fácil entendimiento. Estas fueron las causas de su elección en este trabajo en relación a otros lenguajes de modelación.

4.2.- Desarrollo del Modelo con SDL.

La forma de modelar la Unidad de Anestesia y Pabellones Quirúrgicos en SDL será de modo jerárquico, es decir, se describirá primero el sistema y su entorno, luego bloques asociados, los procesos y por último, el procedimiento en cada proceso. En cada etapa se describirán además los canales y señales asociados.

4.2.1.- El Sistema y su Entorno.

Como se mencionó previamente, se puede definir un sistema como una colección de objetos o entidades que interactúan entre sí para alcanzar un cierto objetivo. En este caso, el sistema corresponde a la UAPQ y los componentes serán los procesos y las relaciones que se desarrollan en su interior. Respecto al entorno, corresponde a todas las entidades que están fuera del sistema pero que interactúan con él. Para el caso de la UAPQ, el entorno lo conforman las Unidades de Emergencias, Servicios Clínicos y Unidades de Apoyo (Diagnóstico y Terapéutico, Logísticos, Administrativos y de Gestión de pacientes). Luego, el sistema queda como se muestra en la figura 13:

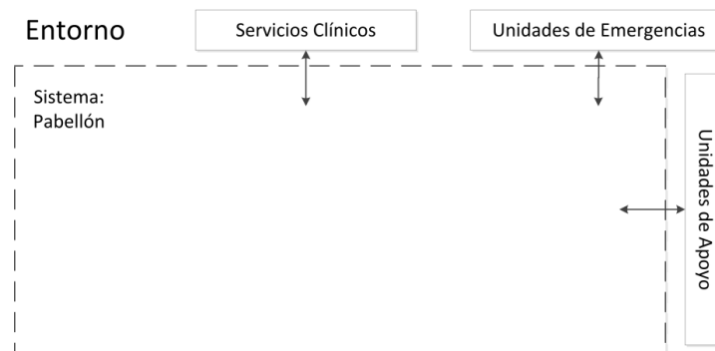


Figura 13. El sistema de la Unidad de Pabellón y su entorno

4.2.2.- El Sistema y los Bloques.

Un bloque es un conjunto de procesos que realizan una función en común. De acuerdo a esta definición, los bloques se construyeron en base a los objetivos y actividades de los distintos grupos de procesos que se desarrollan en la unidad. Los bloques resultantes son:

1. **Comité Quirúrgico:** En este bloque se evalúa y distribuye la disponibilidad semanal de horas de cirugías.
2. **Tabla Quirúrgica:** Se determinan los pacientes a ser intervenidos y se asignan las salas de cirugías y el médico anestesista.
3. **Solicitud de Insumos:** Se determina y solicita el stock y los insumos requeridos para realizar las cirugías.
4. **Recepción del paciente:** Recepción del paciente previo a la cirugía.
5. **Pre-anestesia:** Preparación del paciente antes de la cirugía.
6. **Anestesia:** Administración de la anestesia y chequeo de las condiciones de salud del paciente.
7. **Intervención Quirúrgica:** Operación del paciente.
8. **Recuperación del paciente:** Recepción del paciente post operatorio y chequeo de condiciones de salud.
9. **Requerimiento de Mantención:** Se determinan necesidades y se solicitan reparación o mejoras a la infraestructura.
10. **Requerimiento de Equipos Médicos:** Se determinan necesidades y se solicitan reparación o mejoras de equipos médicos.

Los bloques se conectan entre sí mediante canales y se comunican con señales. Un canal es una ruta para transportar señales entre dos entidades, mientras que una señal es un flujo de información entre dos entidades.

Un canal puede ser unidireccional o bidireccional, según sea el tipo de comunicación que se desarrolle. A cada canal se le asignó un nombre de acuerdo a la numeración del bloque y junto a él se detallan en paréntesis las señales que se transmiten. Estas fueron especificadas en el modelo de acuerdo al tipo de actividad que en cada bloque se desarrolla.

El esquema en SDL del Sistema y Bloques, los canales y señales asociadas se presenta en la figuras 14 y 15.

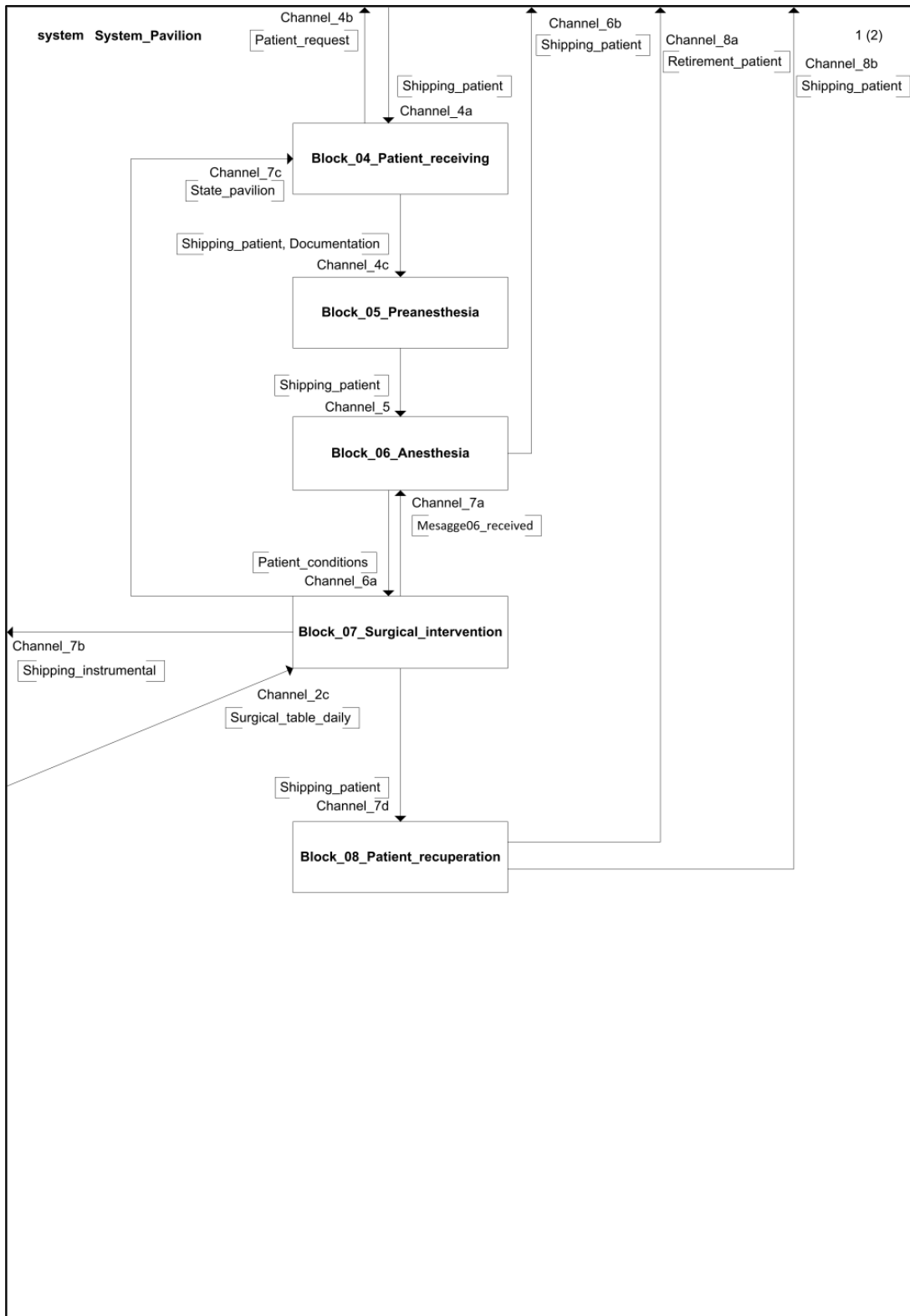


Figura 14. Sistema y Bloques de la Unidad de Anestesia y Pabellones Quirúrgicos (1 de 2).

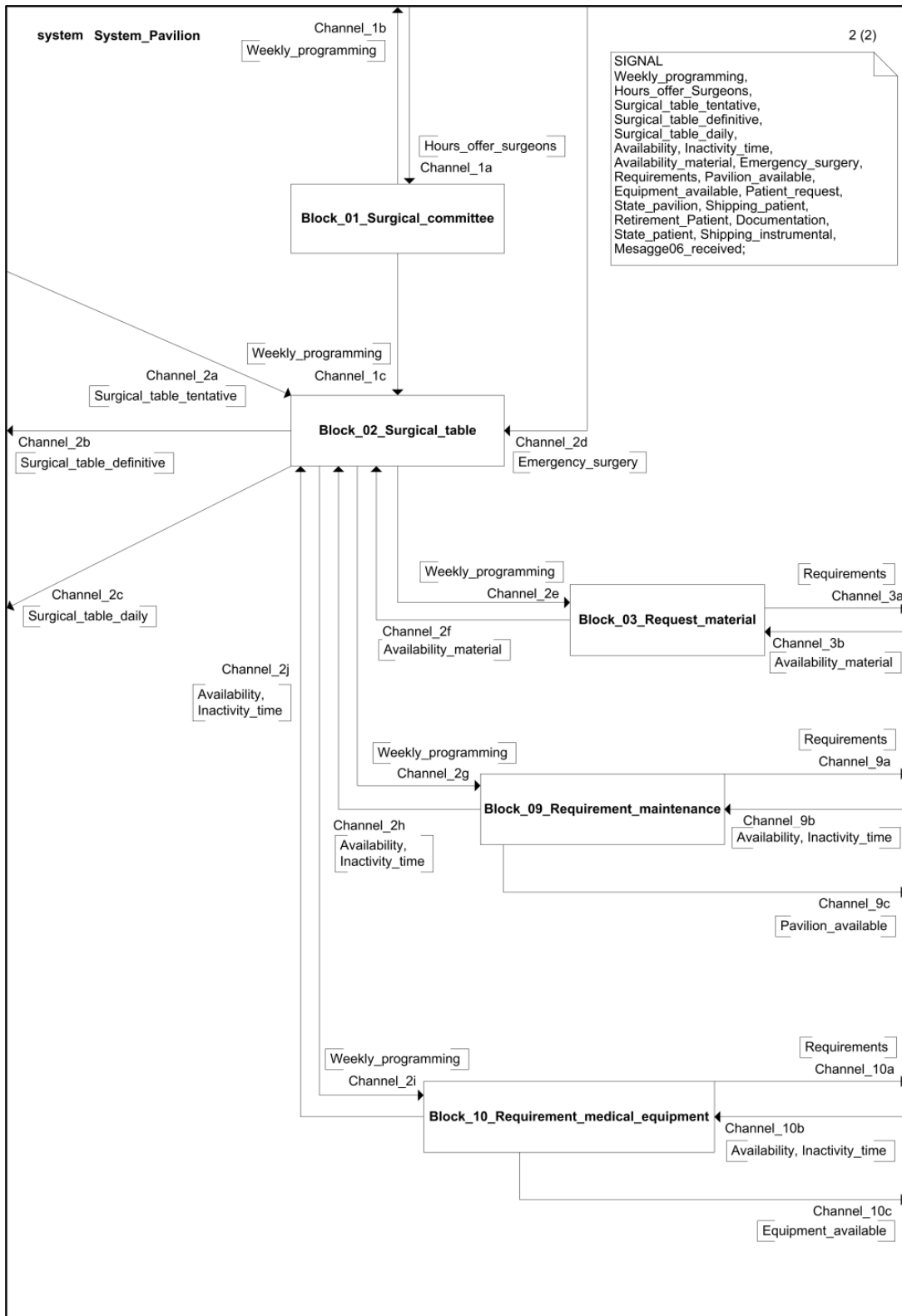


Figura 15. Sistema y Bloques de la Unidad de Anestesia y Pabellones Quirúrgicos (2 de 2).

4.2.3.- Los Bloques y los Procesos.

Los procesos son las actividades que ocurren en un bloque. Para conocerlos, se realizó un estudio amplio y detallado de las distintas actividades que se realizan en la UAPQ, lo que conllevó a revisar en varias oportunidades las estructuras y relaciones existentes. El resultado obtenido fue un sistema con 50 procesos. El esquema jerárquico del sistema de UAPQ se presenta en la figura 16 y los esquemas en SDL de cada bloque, incorporando sus procesos, canales y señales asociadas, se presentan desde la figura 17 a la figura 26:

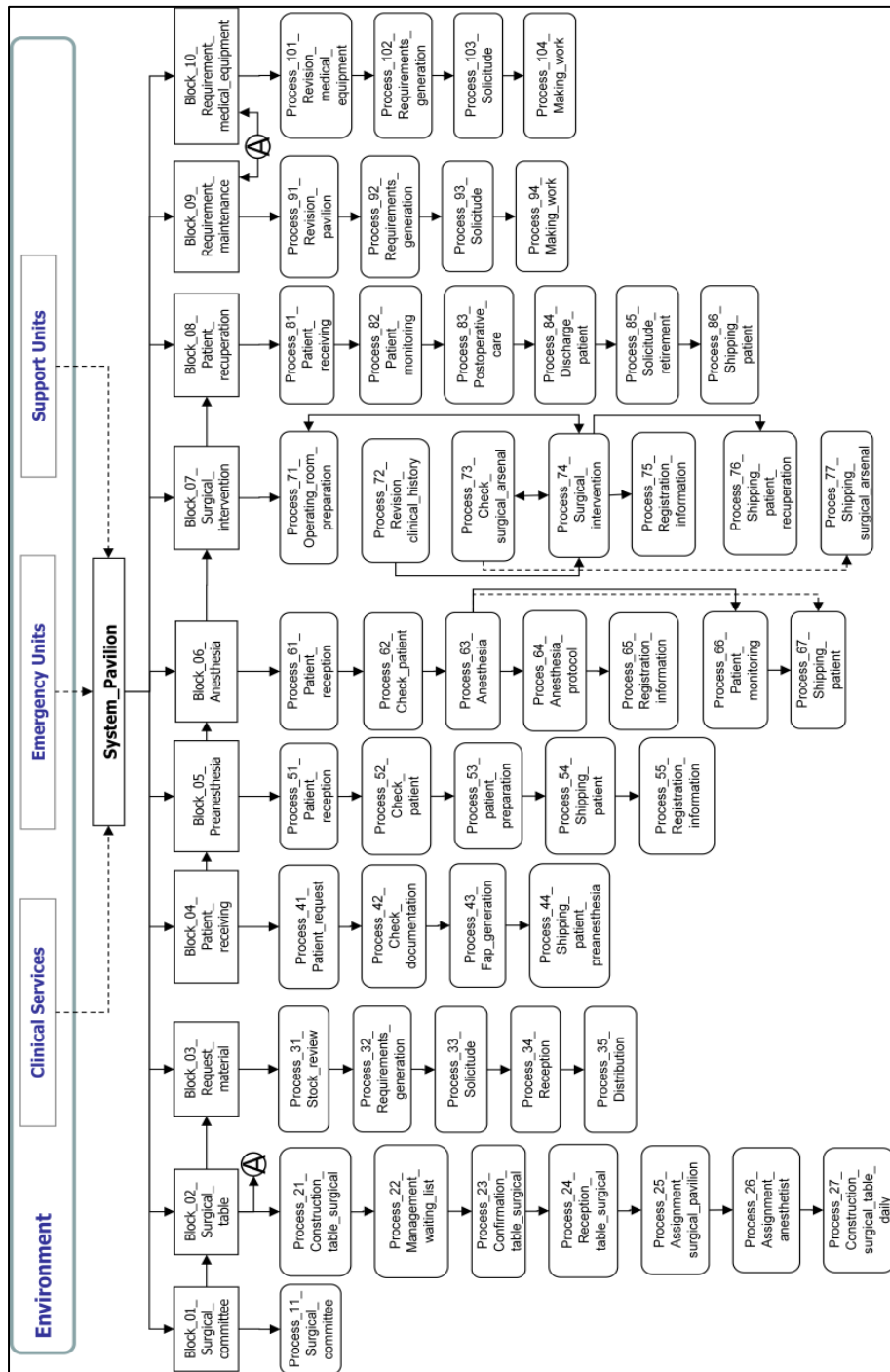


Figura 16. Esquema jerárquico del sistema de UAPQ.

Bloque 1: Comité Quirúrgico (Block_01_Surgical_committee)¹¹

Procesos:

- Comité Quirúrgico (Process_11_Surgical_committee).

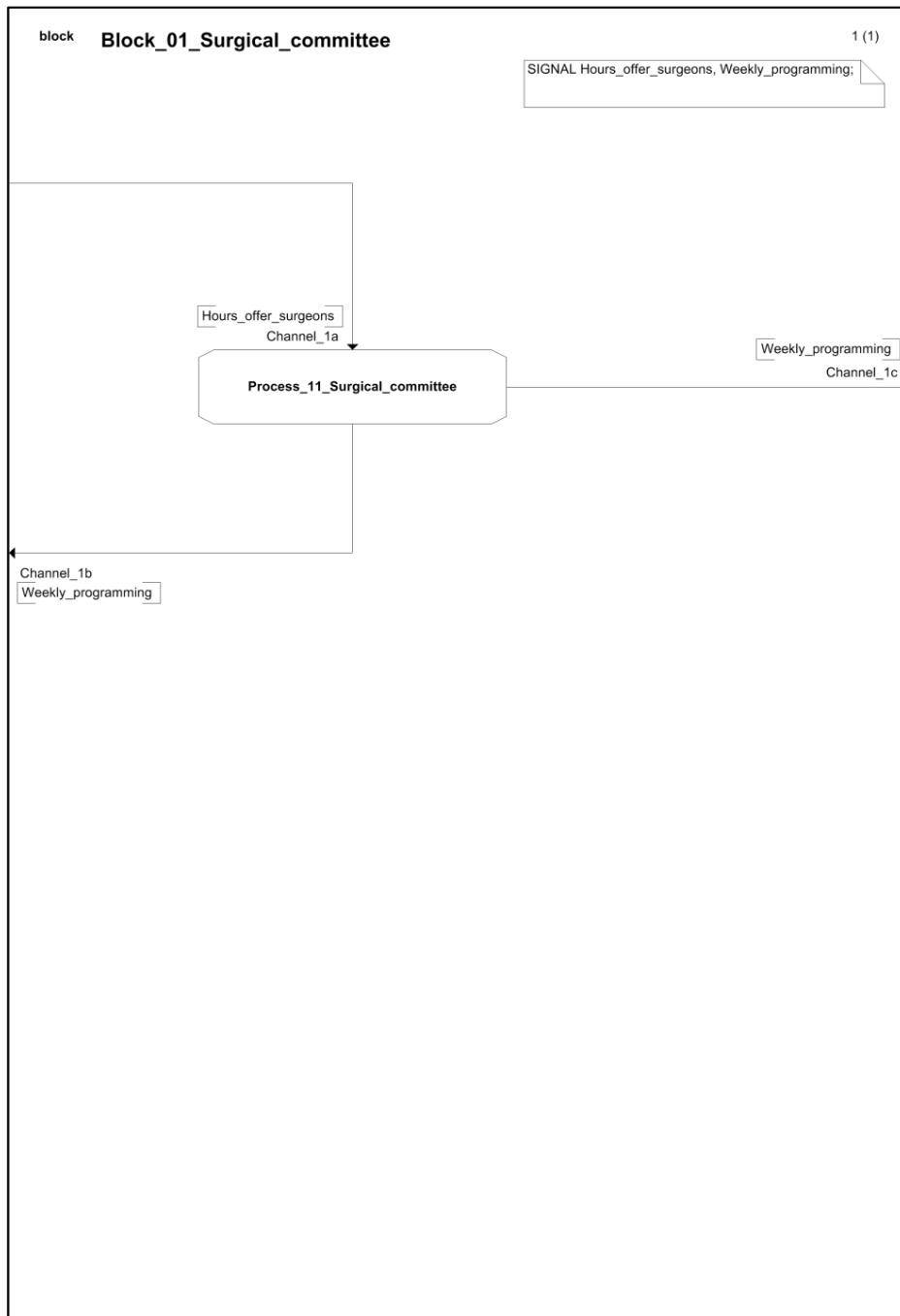


Figura 17. Bloque 1 - Comité Quirúrgico y Procesos asociados.

¹¹ En paréntesis los nombres utilizados en los esquemas SDL.

Bloque 2: Tabla Quirúrgica (Block_02_Surgical_table)

Procesos:

- Construcción de Tabla Quirúrgica por servicio clínico (Process_21_Construction_table_surgical).
- Gestión administrativa de Lista de Espera (Process_22_Management_waiting_list).
- Confirmación de Tabla Quirúrgica (Process_23_Confirmation_table_surgical).
- Chequeo y recepción de tablas quirúrgicas (Process_24_Reception_table_surgical).
- Asignación de pabellón quirúrgico (Process_25_Assignment_surgical_pavilion).
- Asignación de médico anestesista (Process_26_Assignment_anesthetist).
- Construcción de tabla quirúrgica diaria (Process_27_Construction_surgical_table_daily).

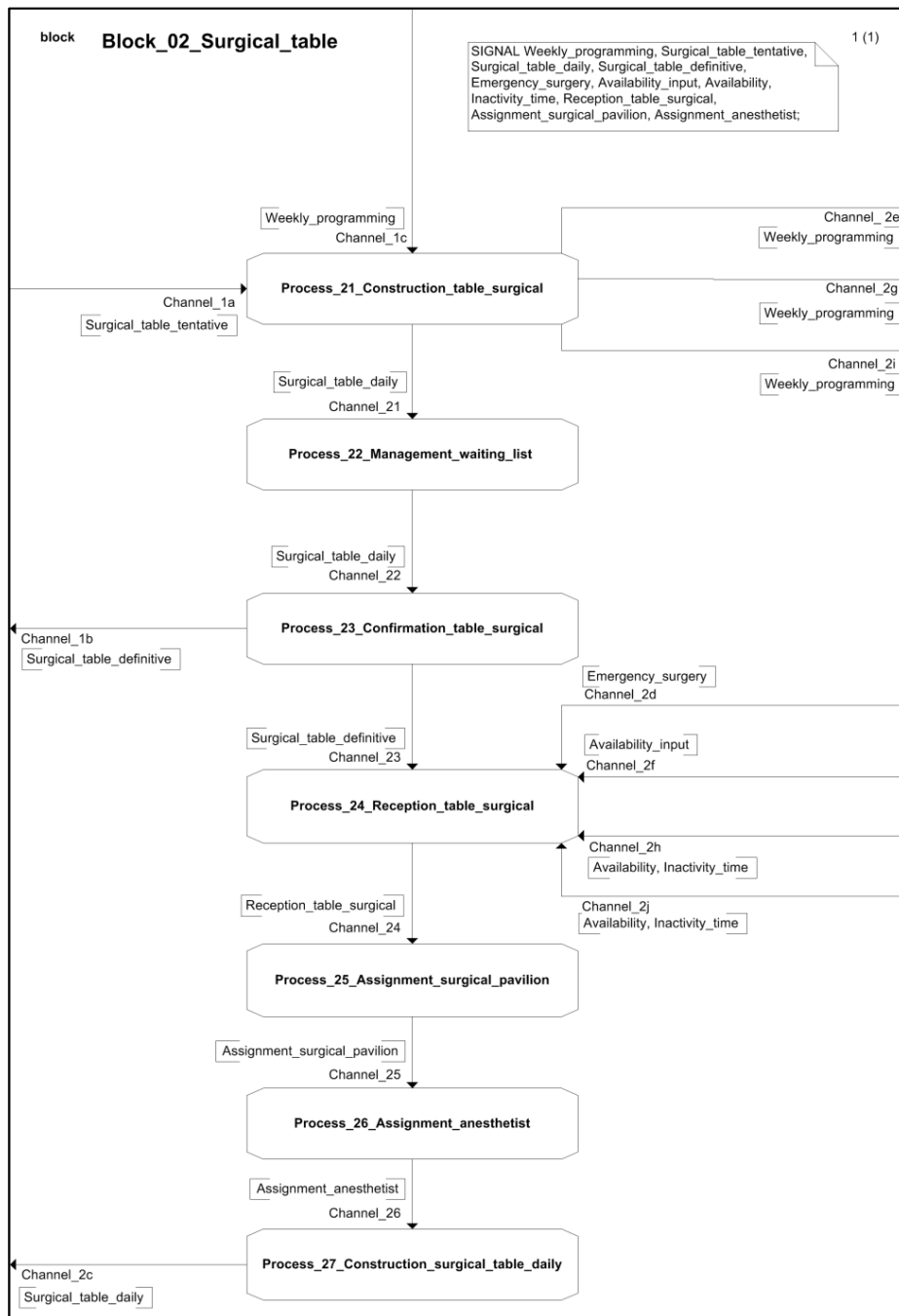


Figura 18. Bloque 2 - Tabla Quirúrgica y Procesos asociados.

Bloque 3: Solicitud de Insumos (Block_03_Request_material)

Procesos:

- Chequeo de stock de insumos (Process_31_Stock_review).
- Generación de requerimientos (Process_32_Requirements_generation).
- Solicitud de requerimientos (Process_33_Solicitud).
- Recepción de insumos (Process_34_Reception).
- Distribución de insumos (Process_35_Distribution).

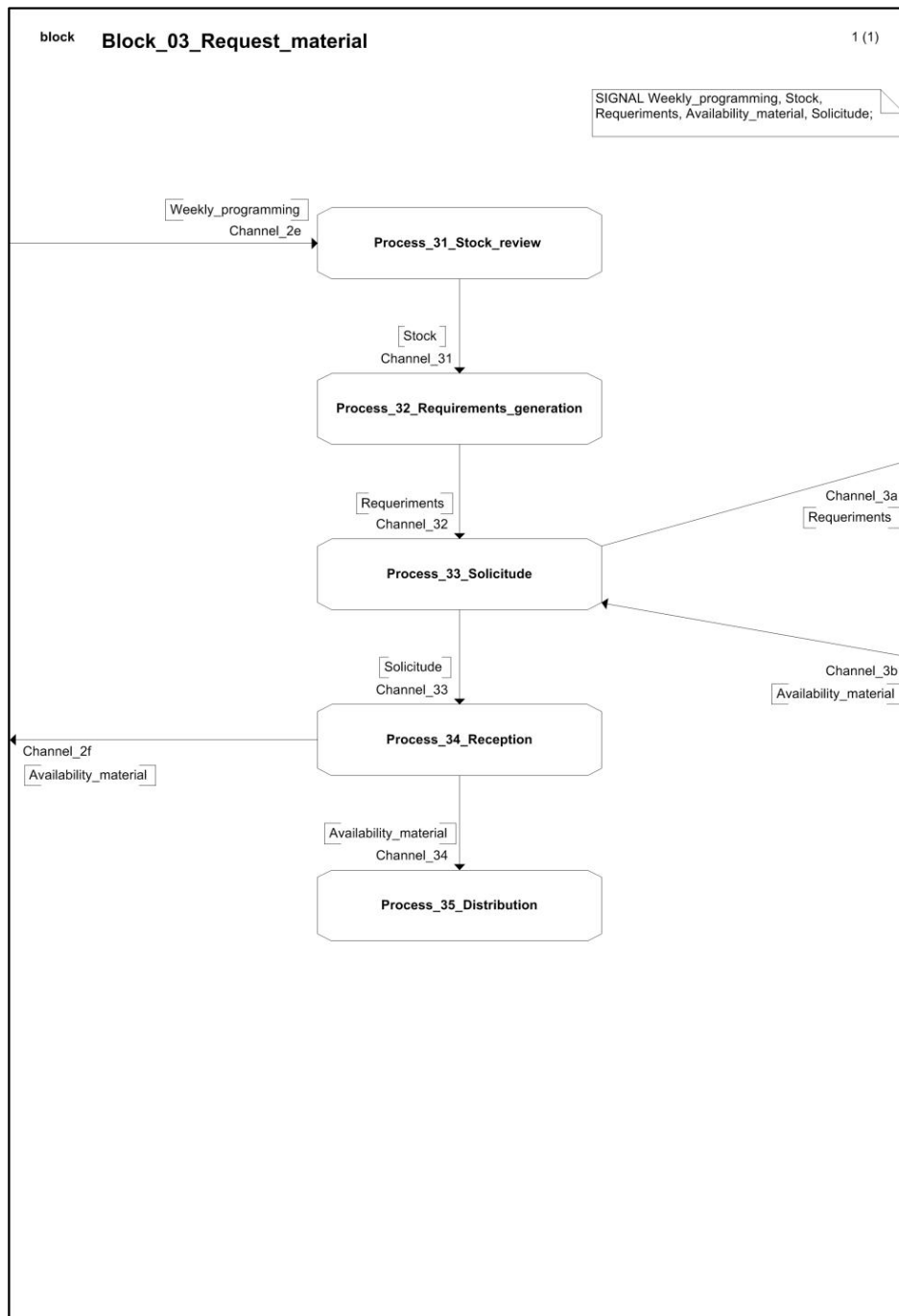


Figura 19. Bloque 3 - Solicitud de Insumos y Procesos asociados.

Bloque 4: Recepción del paciente (Block_04_Patient_receiving)

Procesos:

- Solicitud del paciente a servicios clínicos (Process_41_Patient_request).
- Recepción del paciente y chequeo de documentación (Process_42_Check_documentation).
- Generación de FAP¹² de pabellón (Process_43_Fap_generation).
- Envío de paciente a Pre-anestesia (Process_44_Shipping_patient_preanesthesia).

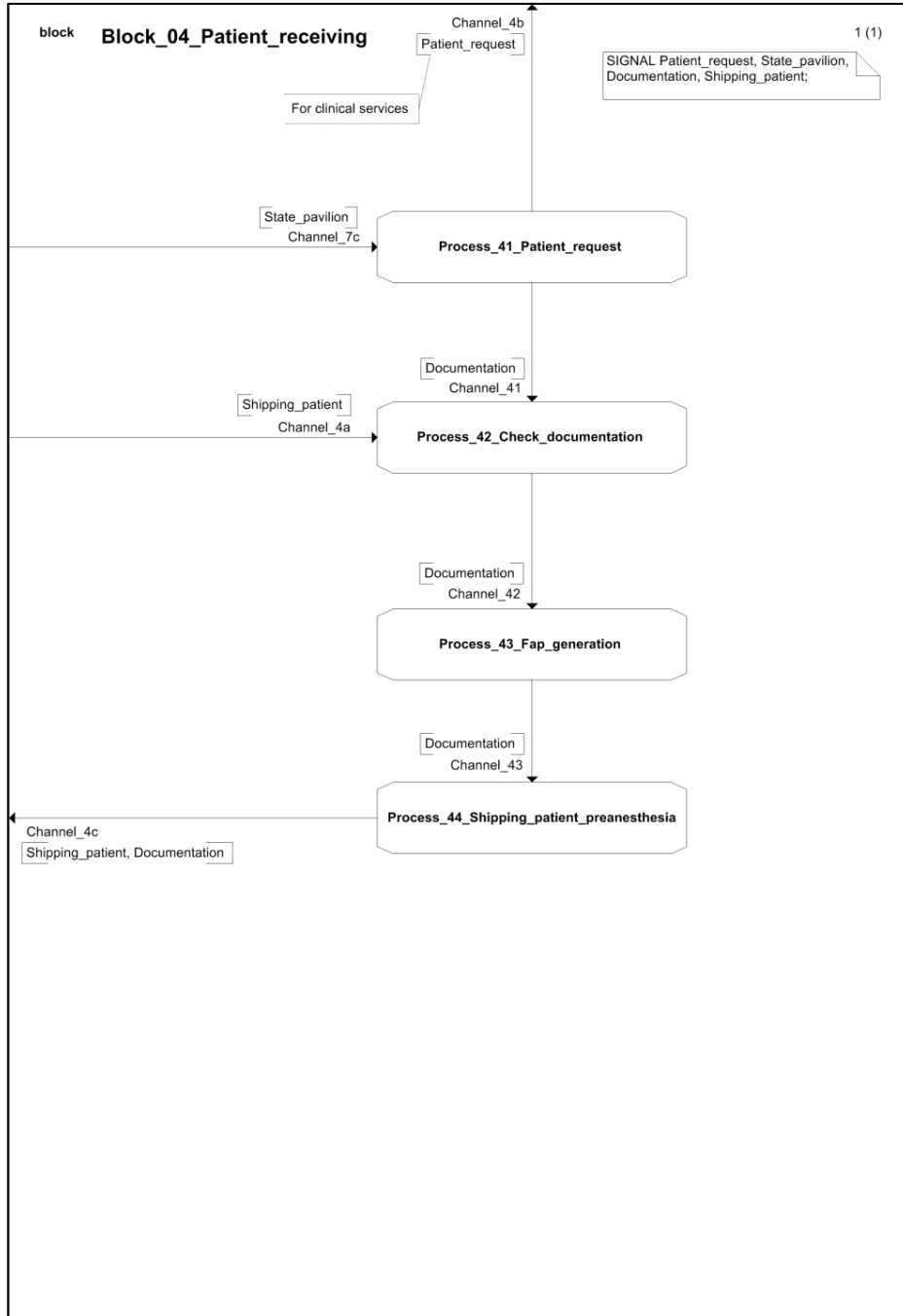


Figura 20. Bloque 4 - Recepción del paciente y Procesos asociados.

¹² FAP: Formulario de Atención de Prestaciones de Pabellón.

Bloque 5: Pre anestesia (Block_05_Preanesthesia)

Procesos:

- Recepción de paciente (Process_51_Patient_reception).
- Chequeo de condiciones de salud del paciente (Process_52_Check_patient).
- Preparación del paciente para cirugía (Process_53_patient_preparation).
- Envío de paciente a sala de cirugía (Process_54_Shipping_patient).
- Registro de información en FAP de Pabellón (Process_55_Registration_information).

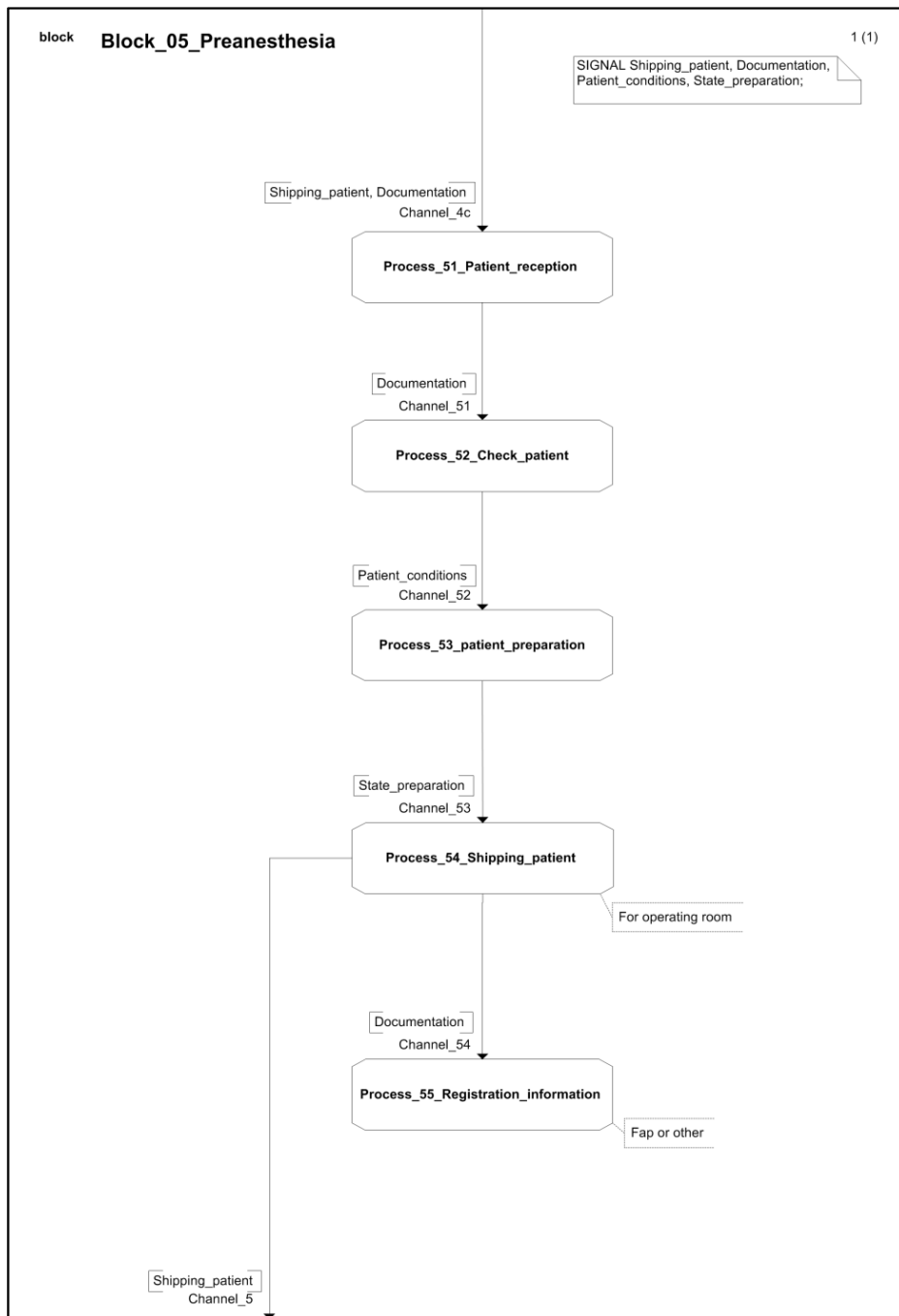


Figura 21. Bloque 5 - Pre anestesia y Procesos asociados.

Bloque 6: Anestesia (Block_06_Anesthesia)

Procesos:

- Recepción del paciente (Process_61_Patient_reception).
- Evaluación del paciente (Process_62_Check_patient).
- Administración de anestesia (Process_63_Anesthesia).
- Generación y registro de formulario protocolo de anestesia (Proces_64_Anesthesia_protocol).
- Registro de información en FAP de Pabellón (Process_65_Registration_information).
- Chequeo y monitoreo de condiciones del paciente (Process_66_Patient_monitoring).
- Devolución de paciente (Process_67_Shipping_patient).

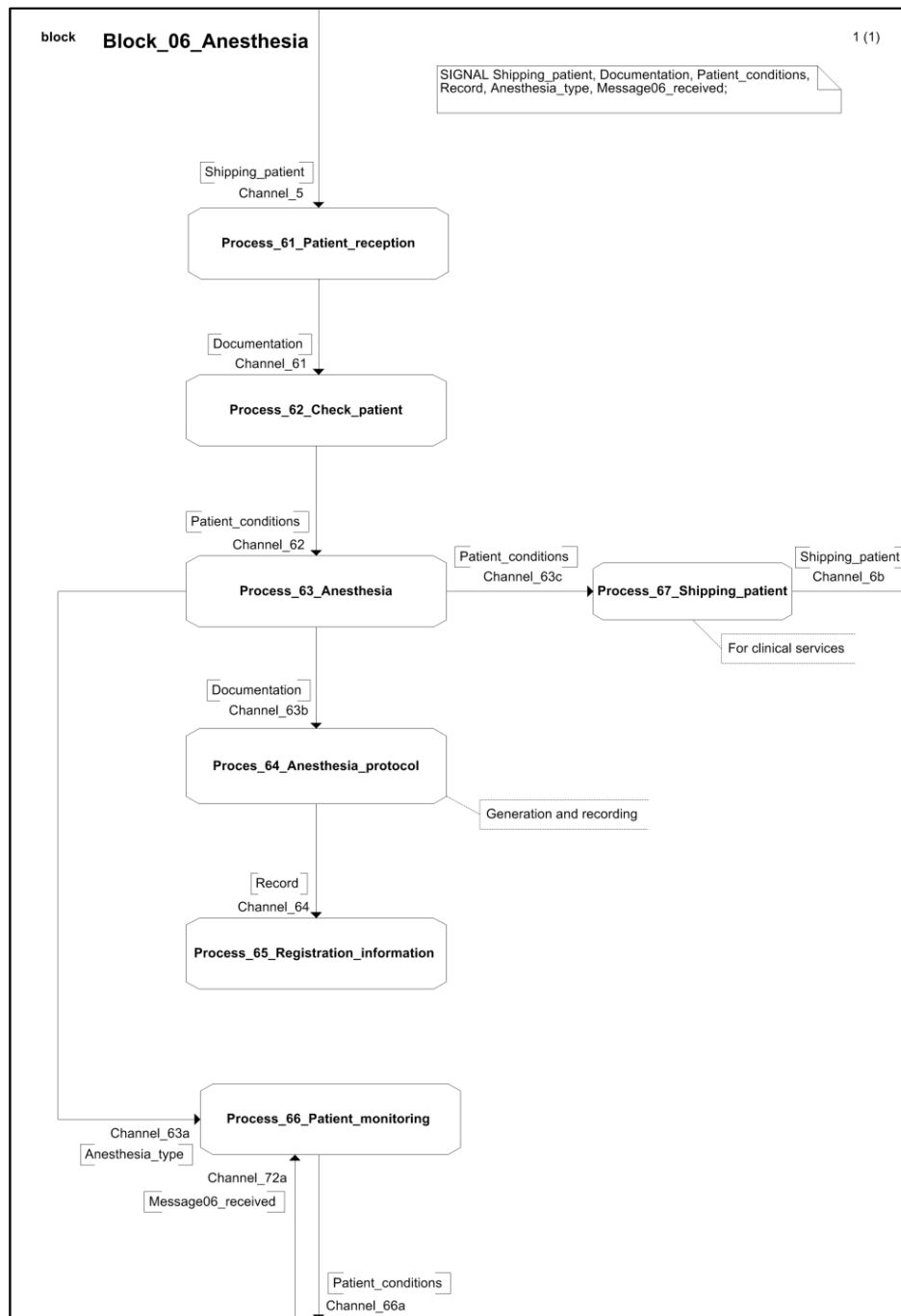


Figura 22. Bloque 6 - Anestesia y Procesos asociados.

Bloque 7: Intervención Quirúrgica (Block_07_Surgical_intervention)

Procesos:

- Preparación de la sala de cirugía (Process_71_Operating_room_preparation).
- Chequeo de antecedentes clínicos (Process_72_Revision_clinical_history).
- Chequeo de arsenal quirúrgico (Process_73_Check_surgical_arsenal).
- Intervención quirúrgica (Process_74_Surgical_intervention).
- Registro de información en FAP de Pabellón (Process_75_Registration_information).
- Envío de paciente a sala de Recuperación (Process_76_Shipping_patient_recuperation).



Figura 23. Bloque 7 - Intervención Quirúrgica y Procesos asociados.

Bloque 8: Recuperación del paciente (Block_08_Patient_recuperation)

Procesos:

- Recepción del paciente (Process_81_Patient_receiving).
- Chequeo y monitoreo del paciente (Process_82_Patient_monitoring).
- Atenciones post operatorias (Process_83_Postoperative_care).
- Chequeo de criterios de alta del paciente (Process_84_Discharge_patient).
- Solicitud de retiro del paciente (Process_85_Solicitud_retirement).
- Entrega del paciente (Process_86_Shipping_patient).

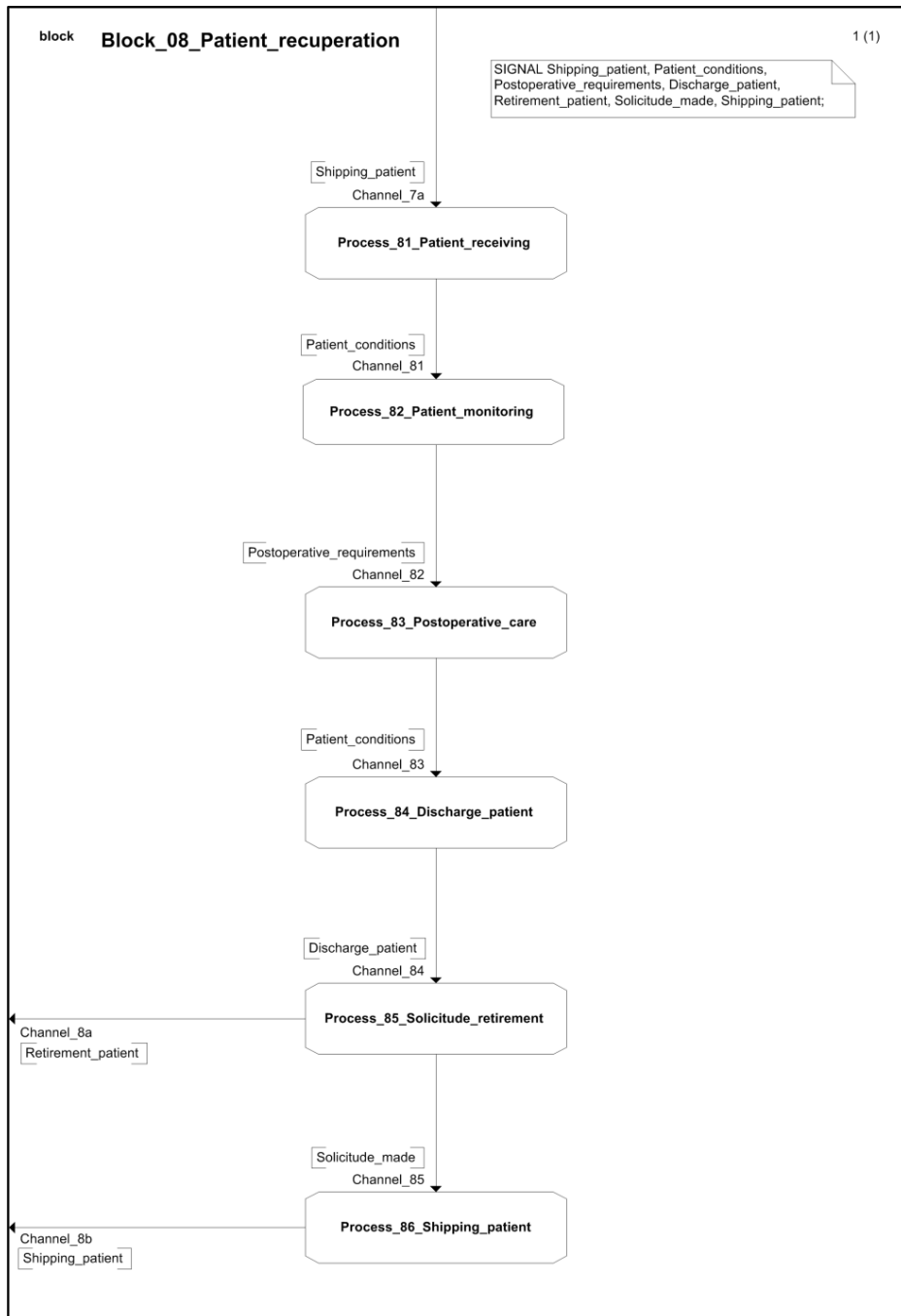


Figura 24. Bloque 8 - Recuperación del paciente y Procesos asociados.

Bloque 9: Requerimiento de Mantenimiento¹³ (Block_09_Requirement_maintenance)

Procesos:

- Chequeo de estado del pabellón (Process_91_Revision_pavillon).
- Generación de requerimientos (Process_92_Requirements_generation).
- Solicitud de requerimientos (Process_93_Solicitud).
- Realización de trabajos (Process_94_Making_work).

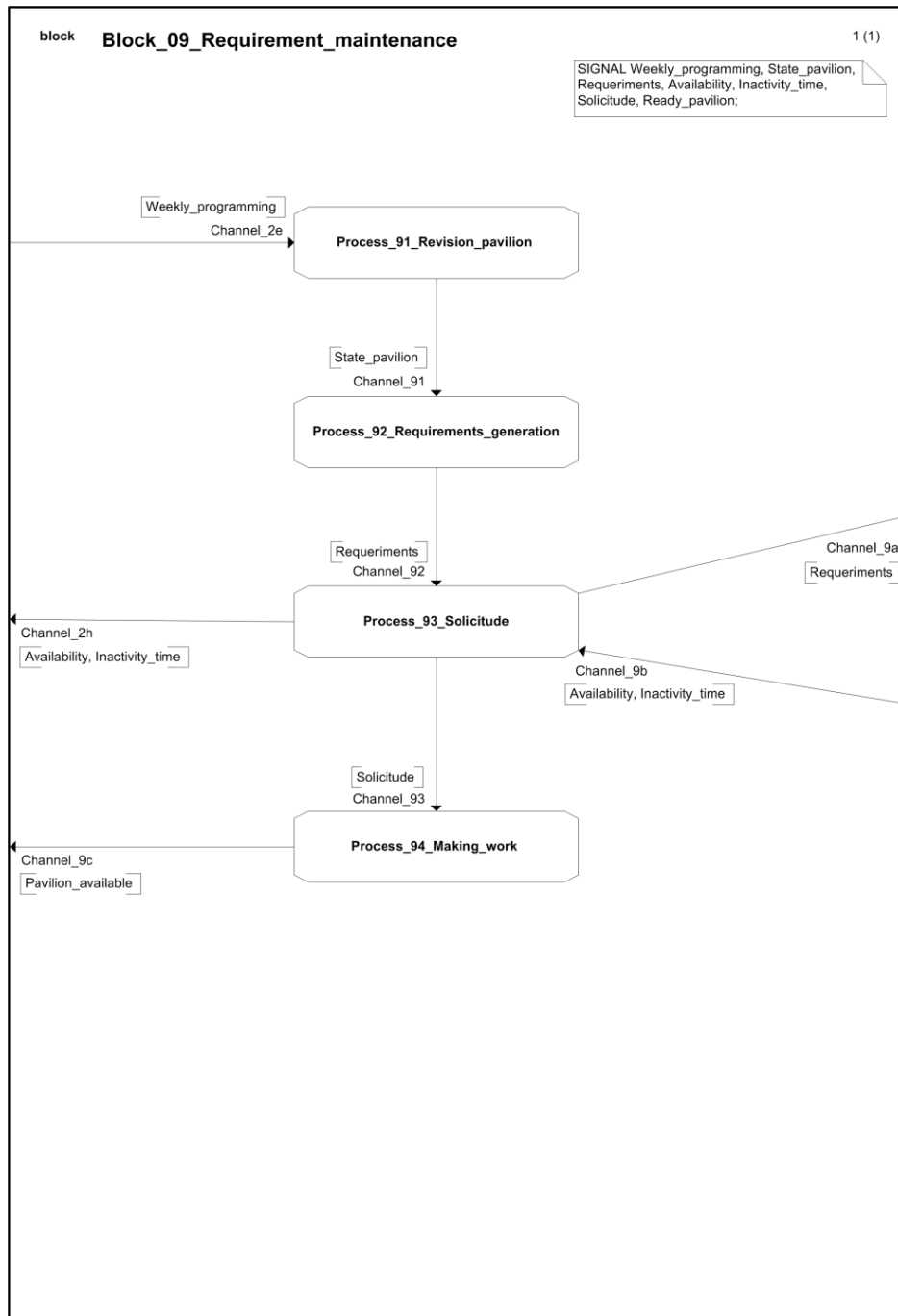


Figura 25. Bloque 9 - Requerimientos de Mantenimiento y Procesos asociados.

¹³ Infraestructura y/o servicios generales.

Bloque 10: Requerimiento de Equipos Médicos (Block_10_Requirement_medical_equipment)

Procesos:

- Chequeo del estado del equipo médico (Process_101_Revision_medical_equipment).
- Generación de requerimientos¹⁴ (Process_102_Requirements_generation).
- Solicitud de requerimientos (Process_103_Solicitud).
- Realización de trabajos (Process_104_Making_work).

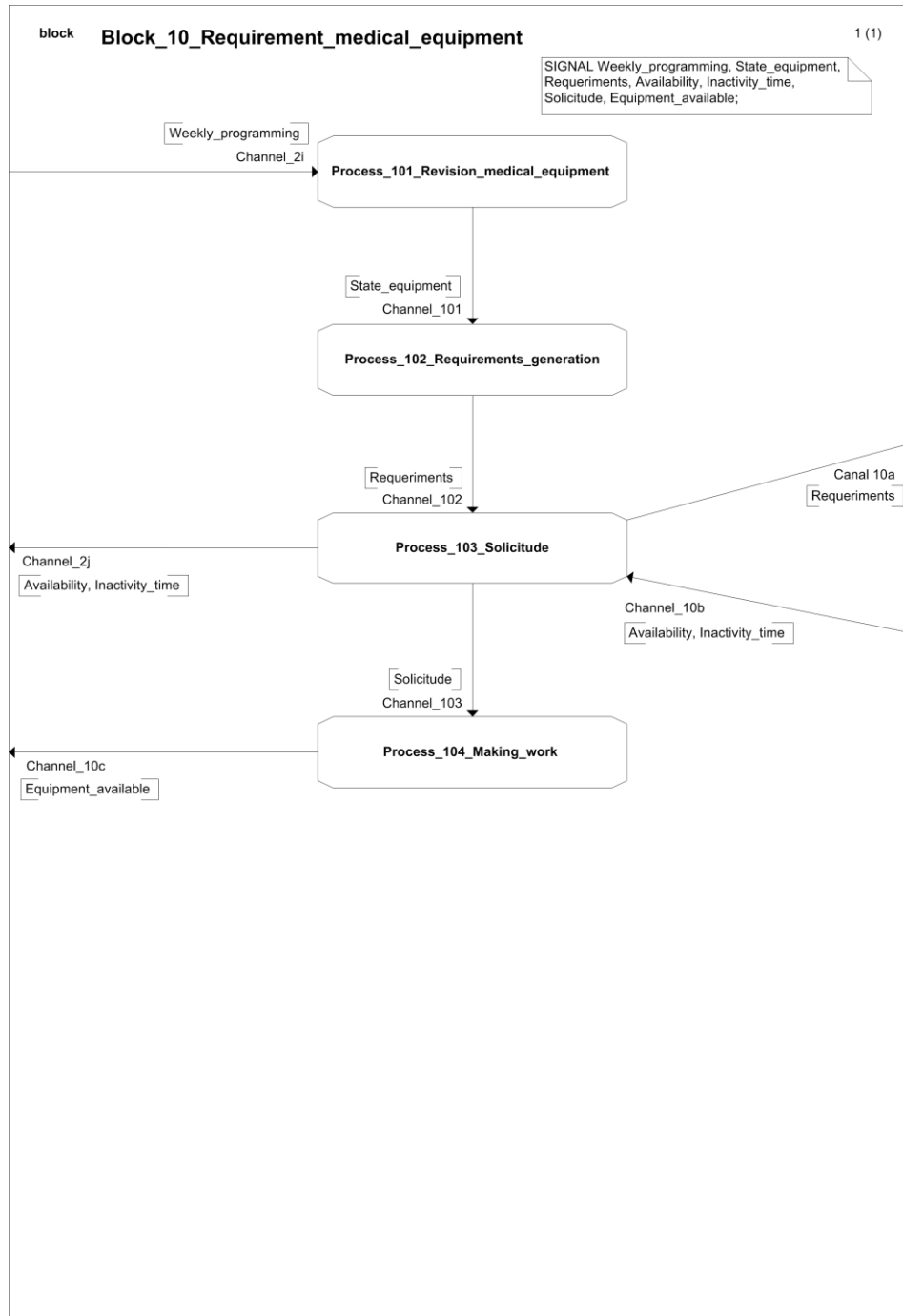


Figura 26. Bloque 10 - Requerimientos de Equipos Médicos y Procesos asociados.

¹⁴ Mantención, reparación o cambio del equipo.

4.2.4.- Los Procesos y Procedimientos.

Un proceso contiene estados y transiciones entre estados, cuya función es describir las tareas o el comportamiento que ocurre en el bloque. Este conjunto de estados se describe por medio de EFSM (definido previamente). Para que un cambio de estado ocurra, el proceso debe recibir una señal válida de otro proceso o del entorno (una señal puede considerarse como un estímulo). Al realizar una transición (de un estado a otro), se pueden realizar operaciones, tales como: manipular los datos locales del proceso, enviar señales a otros procesos o al entorno, llamar a un procedimiento o incluso crear instancias de otros procesos (llamar a otros procesos en un determinado estado). Pueden existir varias instancias de un proceso ejecutándose concurrentemente y además simultáneamente con instancias de otros procesos. Con respecto a la recepción de señales de un proceso, cada instancia tiene una única cola de señales de entrada que no comparte con otros.

Para ejemplificar, presentaremos el Proceso Intervención quirúrgica (Process_74_Surgical_intervention) perteneciente al del Bloque 7. Este proceso inicia su actividad en espera de tres señales: La preparación de la sala de operación (Preparing_operating_room) desde Process_71_Operating_room_preparation, la información clínica del paciente (Clinical_information_1) desde Process_72_Revision_clinical_history y el Chequeo del arsenal quirúrgico (Check_surgical_arsenal) desde (Process_73_Check_surgical_arsenal).

Solo cuando se reciben las tres señales, llama al procedimiento Procedure_Rand que permite generar números aleatorios para determinar si el paciente requiere una nueva operación. Si este es el caso, se generan nuevamente las tres señales (Preparing_operating_room, Clinical_information_1 y Check_surgical_arsenal) con un tiempo de retraso (delay) menor al asociado cuando las señales provienen desde los otros procesos y comienza nuevamente el ciclo del proceso 74 quedando en el estado inicial (Wait74). Si por el contrario, el paciente no requiere más operaciones, se genera una serie de señales que representan el tiempo de la operación (End_surgery), constancia de haber recibido la información clínica (Message72_received), solicitud de chequeo del arsenal quirúrgico utilizado (Surgical_arsenal) y el estado del pabellón (State_pavilion), quedando nuevamente en el estado inicial (Wait74). El esquema de este proceso se presenta en la figura 27:

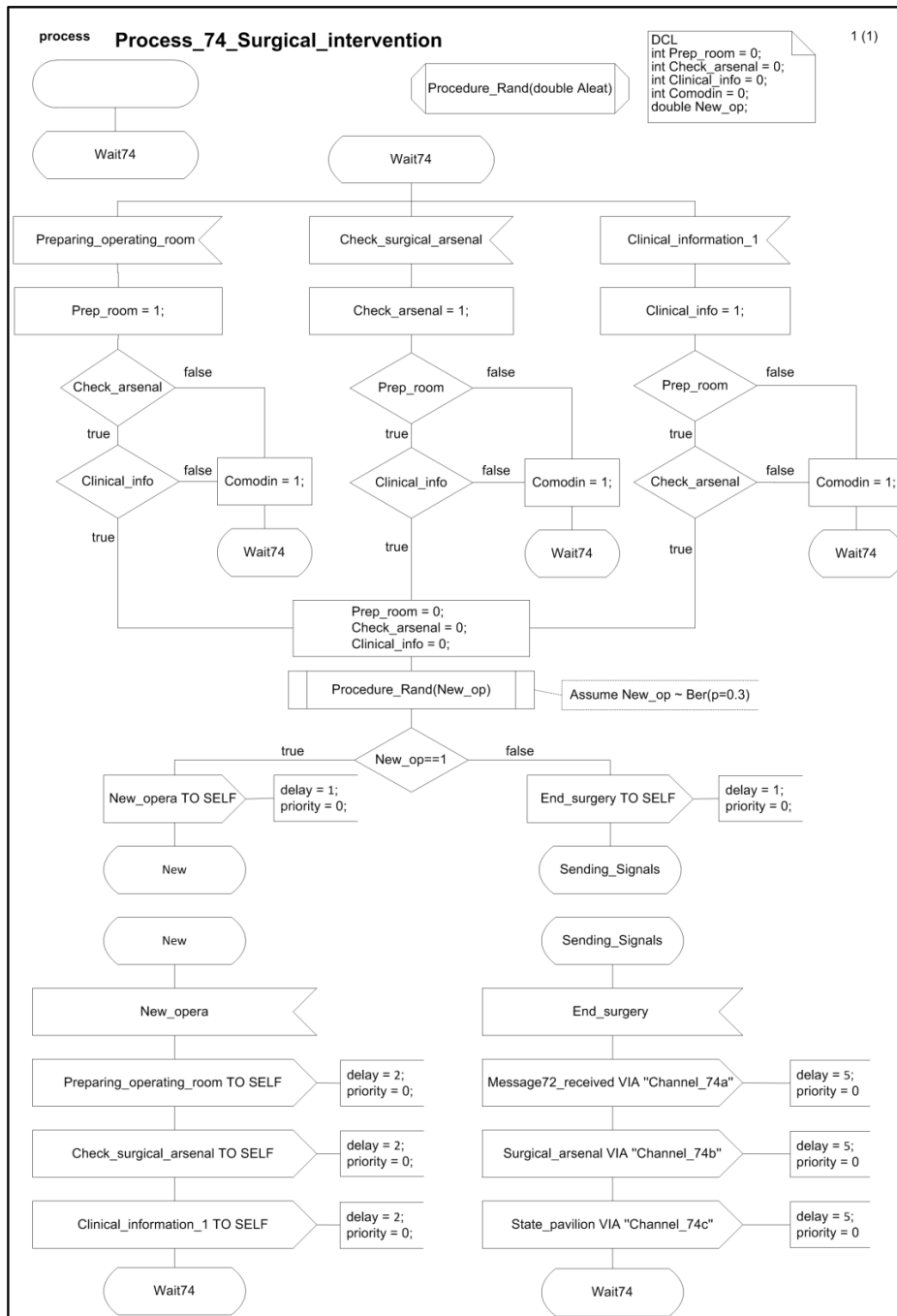


Figura 27. Process_74_Surgical_intervention.

Respecto a los procedimientos, están en el último nivel en la estructura del lenguaje SDL. Su función es especificar acciones, estados, variables, parámetros y temporizadores que se pueden dan en un determinado proceso. Pueden ser usados para factorizar partes comunes del comportamiento o clarificar la descripción. La figura 28 muestra el procedimiento Procedure_Rand, mencionado anteriormente.

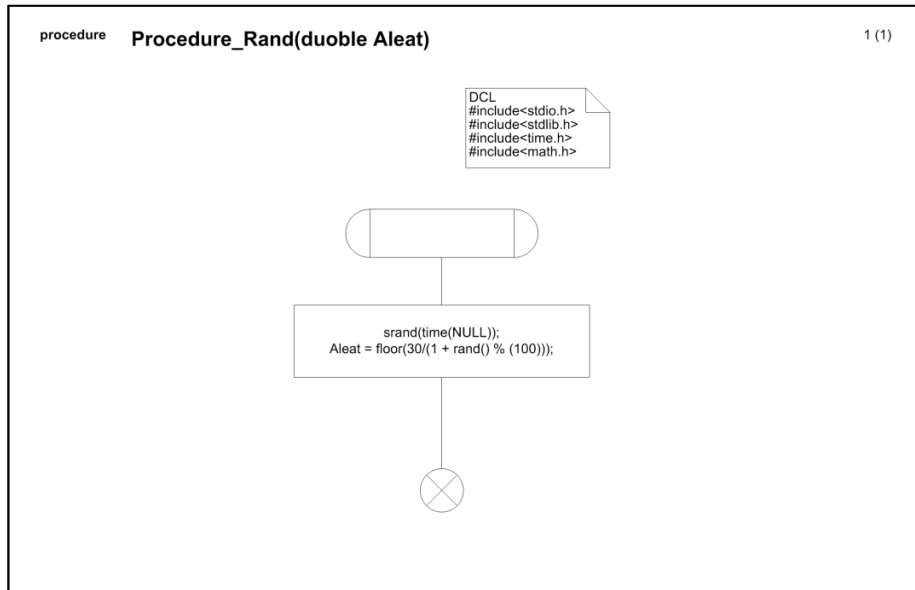


Figura 28. Procedure_Rand.

4.3.- Simulación con SDLPS.

Como se mencionó en los capítulos anteriores, SDL permite trabajar en forma modular y dada la complejidad del modelo y los objetivos planteados, se realizó la especificación, validación, verificación y simulación de una parte del sistema, al cual denominaremos “Modelo reducido”¹⁵. Por su importancia clínica, los bloques y procesos seleccionados para incluir en el modelo reducido son:

Bloque 6: Anestesia (Block_06_Anesthesia)

Procesos:

- Chequeo y monitoreo de condiciones del paciente (Process_66_Patient_monitoring).

Bloque 7: Intervención Quirúrgica (Block_07_Surgical_intervention)

Procesos:

- Preparación de la sala de cirugía (Process_71_Operating_room_preparation).
- Chequeo de antecedentes clínicos (Process_72_Revision_clinical_history).
- Chequeo de arsenal quirúrgico (Process_73_Check_surgical_arsenal).
- Intervención quirúrgica (Process_74_Surgical_intervention).

Solo se conservaron los canales y señales involucradas con los bloques y procesos seleccionados. En esta aplicación, se trabajaron con tiempos de retrasos fijos (en una versión futura se considerarán tiempos aleatorios de acuerdo con distribuciones adecuadas. Esto se detalla aún más en el punto 4.2.5). La figura 29 muestra un resumen del modelo reducido.

¹⁵ El desarrollo completo del modelo de la UAPQ se realizará en un estudio posterior.

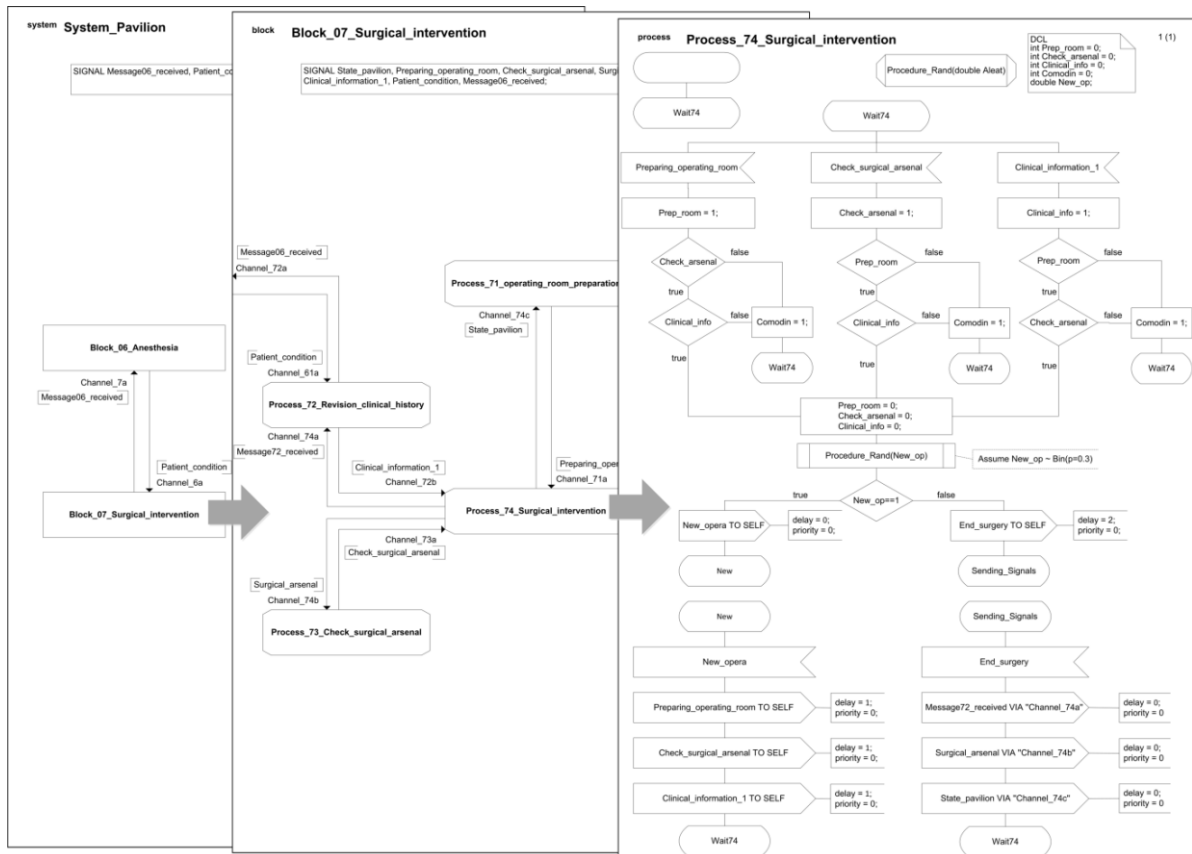


Figura 29. Sistema, Bloque 07 y Proceso 74 del modelo reducido.

SDLPS permite validar el modelo y simular eventos. Después de hacer los cambios apropiados, se ha logrado completar las etapas de validación y verificación, ilustradas en la Figura 30. Esto permite realizar simulaciones del modelo reducido. Los resultados obtenidos permiten entender el comportamiento del sistema y obtener el tiempo de ocupación y de ocio de cada proceso, entre otras estadísticas.

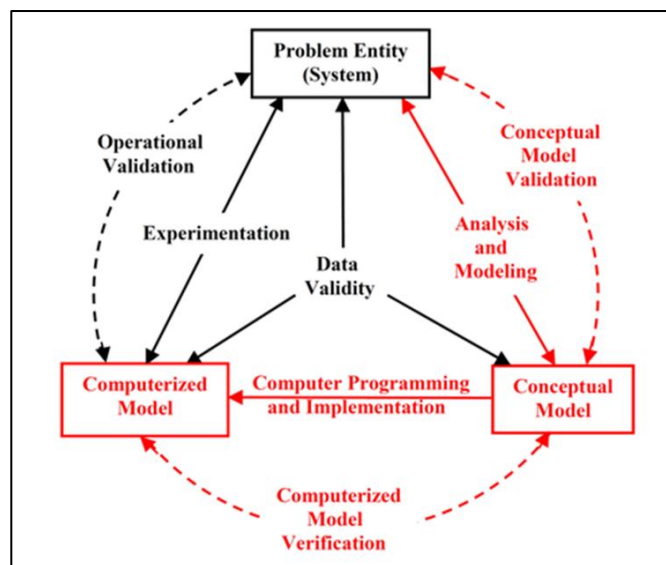


Figura 30. Versión simplificada de un proceso de modelación (Sargent, 2007).

En relación a los resultados de la simulación, la figura 31 muestra el diagrama de estado y los resultados de simulación del proceso Chequeo y monitoreo de condiciones del paciente (Process_66_Patient_monitoring). Del total de tiempo simulado, este proceso estuvo en funcionamiento 212 unidades de tiempo (t.u.). En este tiempo se generaron 446 eventos y un total de 10 ciclos del proceso. El 80% del tiempo, el proceso estuvo en estado "Wait66" (es decir, a la espera de señales) y el 20% restante en el estado "Sending_condition" (es decir, ocupado y enviando señales de la condición del paciente). El tiempo medio de pasar de un estado a otro son: 16 (t.u.) para la transición a Wait66 a Sending_condition y de 4 (t.u.) para la transición Sending_Condition a Wait66.

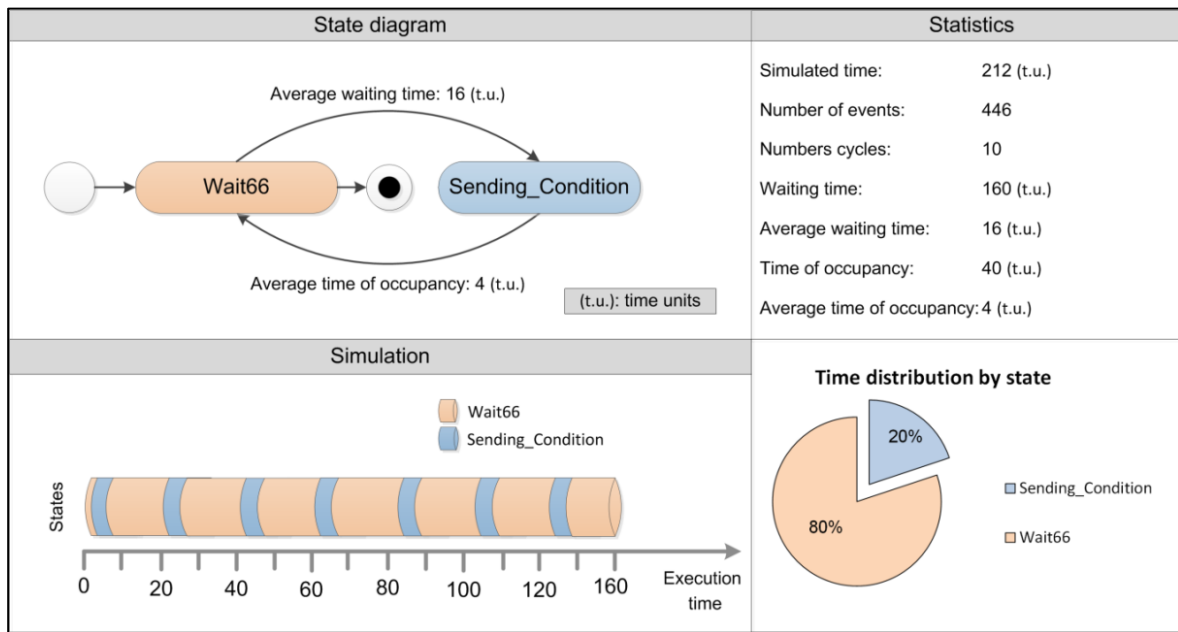


Figura 31. Resultados de la simulación para Process_66_Patient_monitoring, en el modelo reducido.

La figura 32 muestra el diagrama de estado y los resultados de simulación del proceso Intervención quirúrgica (Process_74_Surgical_intervention). Del total del tiempo simulado, este proceso estuvo en funcionamiento 476 (t.u.). En este tiempo se generaron 1.010 eventos y un total de 30 ciclos del proceso. El 73.9% de tiempo, el proceso estuvo en estado "Wait74" (es decir, a la espera de señales), el 23,1% en estado "Sending_signal" (es decir, se realizó una sola operación y luego se envió señales a los procesos de 71, 72 y 73) y el 2,9% restante en estado "New" (es decir, se realizó más de una operación y se enviaron señales internas). El tiempo medio para pasar de un estado a otro son: 13 (t.u.) para la transición Wait74 a Sending_signals y 5 (t.u.) para Sending_signals a Wait74, mientras que 9.6 (t.u.) para la transición Wait74 a New y 2 (t.u.) desde Wait74 a New.

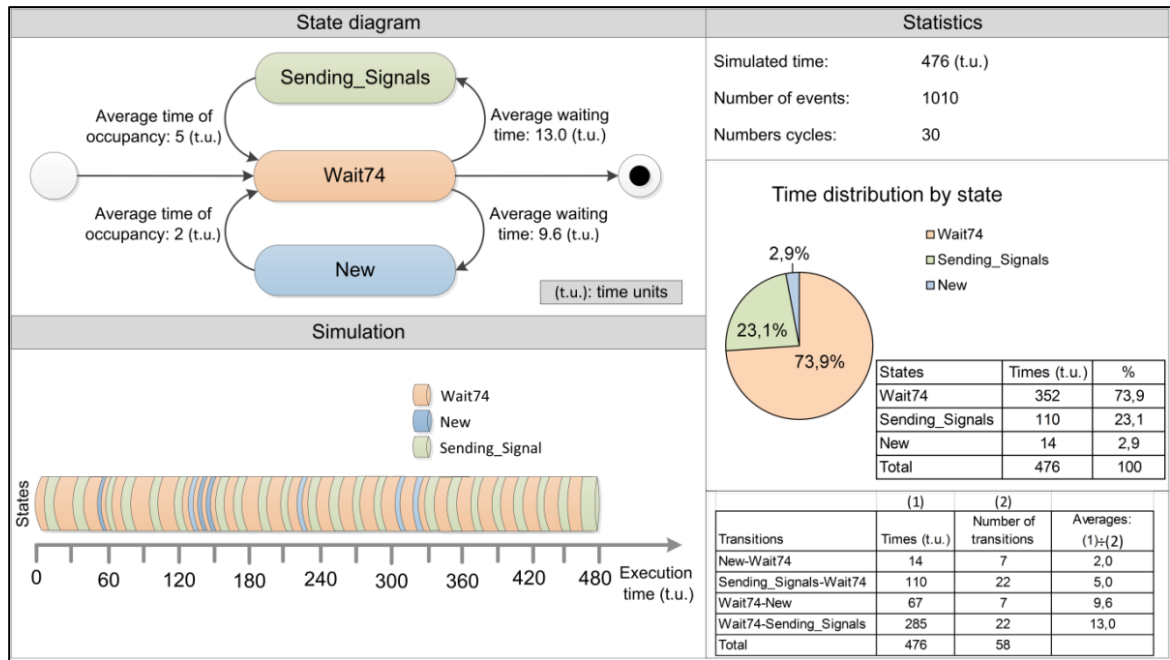


Figura 32. Resultados de la simulación para Process_74_Surgical_intervention, en el modelo reducido.

4.2.5.- Validación y evaluación de la calidad de los datos

Este trabajo ha sido la primera etapa de un estudio mayor, cuya finalidad es la modelación y simulación completa de la UAPQ. Por ello, en la actualidad ya se trabaja en el acopio, validación de los registros y evaluación de la calidad de los datos. Para ello se ha recogido la información de los últimos 5 años a partir de la base de datos UAPQ.

Debido a la complejidad del problema, el gran número de procesos y la importancia de cada actividad, se decidió trabajar con las variables asociadas a los procesos del modelo reducido. Se llevó a cabo un proceso de validación de los registros existentes (detección y corrección de errores de registros, cruces con otras bases de datos del hospital, la normalización de los códigos, etc.). Las variables que ya han sido evaluadas son:

- Tiempo de preparación de la sala de cirugía
- Tiempo de ocupación del pabellón quirúrgico
- Tiempo de operación
- Tiempo de chequeo de arsenal quirúrgico
- Tiempo de revisión de antecedentes clínicos

Para comenzar, se seleccionó los registros del pabellón quirúrgico n°1 (donde se realizan prestaciones de traumatología y cirugía cardiovascular). Se han realizado las pruebas estadísticas para evaluar la independencia de los datos (de tendencia, la aleatoriedad y autocorrelación) y las pruebas de bondad de ajuste para seleccionar las distribuciones de probabilidad asociadas a las variables estudiadas. La figura 33 muestra el

estudio de la calidad de los datos para la variable "Tiempo de ocupación del pabellón quirúrgico n°1".

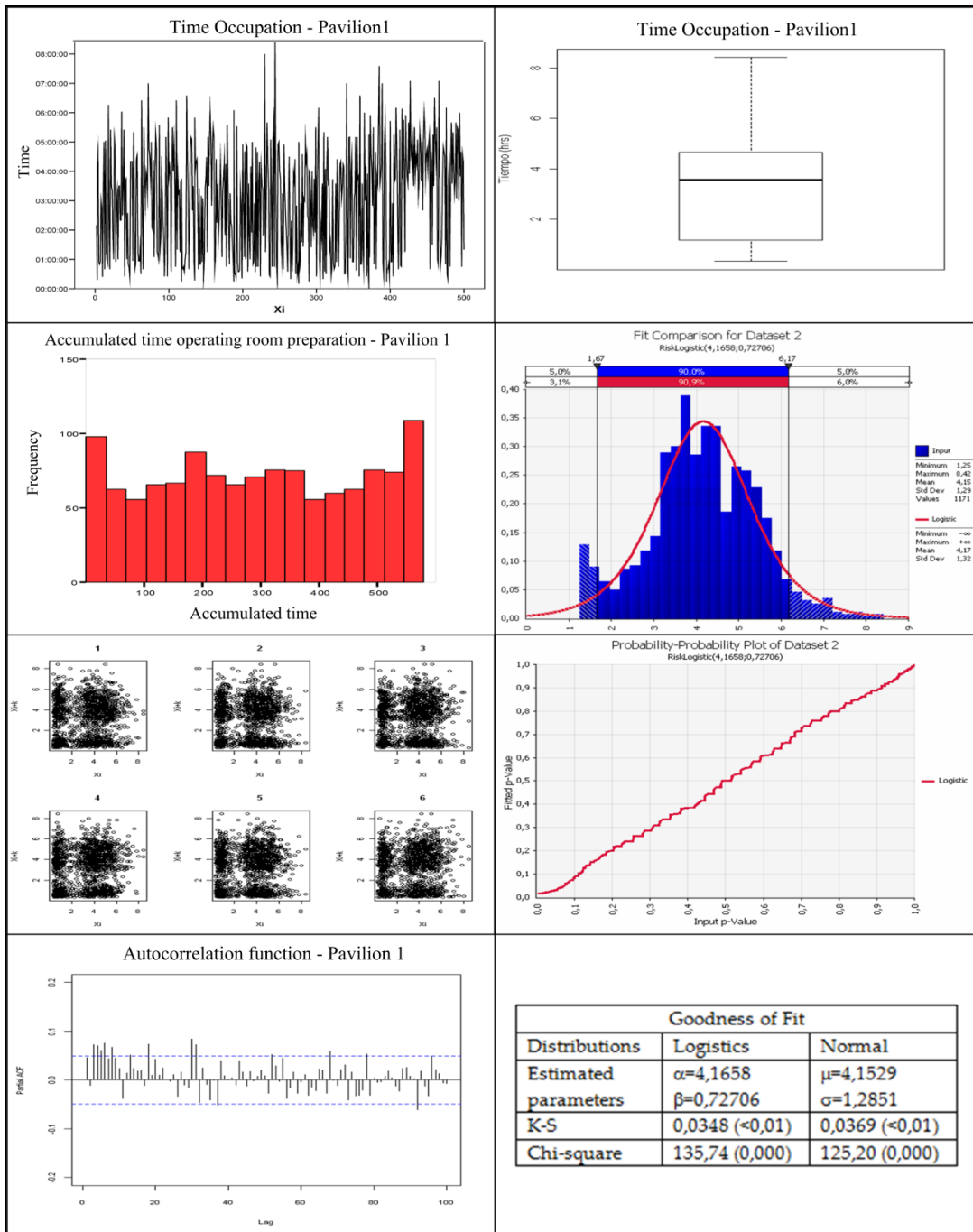


Figure 33. Estudio de la calidad de los datos para la variable: Tiempo de ocupación del pabellón quirúrgico n°1.

A futuro y una vez completada la validación y evaluación de la calidad de los datos con todas las variables del modelo reducido, se ejecutará nuevamente la simulación realizada previamente.

CAPÍTULO 5.- CONCLUSIONES

Las conclusiones más importantes del estudio son:

- El estudio permitió la creación de una documentación sin ambigüedades y una comprensión completa de los procesos de la UAPQ por parte del personal clínico y administrativo. Esto condujo a la colaboración en la gestión hospitalaria a través de la evaluación de los diferentes procesos.
- Los canales de información (para permitir la transmisión de la señal), las señales, la direccionalidad y la jerarquía entre los diferentes procesos han sido definidos. Esta definición completa de las características del sistema y la estructura modular del lenguaje permite una identificación clara de las relaciones entre los diferentes elementos del sistema.
- El diseño permite establecer pautas de actuación y mejora para las actividades administrativas, tales como los relacionados con las unidades de apoyo, incluidas las tareas y los conocimientos (a menudo tácitos) transmitidos a través de canales informales.
- Esta representación del modelo facilita la simulación desvinculándola de la herramienta utilizada para realizarla.
- La definición modular del modelo permite seguir diferentes líneas de investigación, como detalle de los procesos UAPQ o estudiar la optimización de algunos de ellos.
- El estudio permitió evaluar la validación y verificación de una parte del sistema, obteniendo a través de una simulación estadísticas respecto a su funcionamiento.

Algunas de las dificultades encontradas fueron:

- La calidad de los registros UAPQ. Esto implicó una ardua tarea para validar los datos y una dificultad para determinar la distribución de probabilidades a través de las pruebas de bondad de ajuste.
- La complejidad inherente del sistema consiste en especificar un gran número de procesos.
- Los cambios en los equipos de trabajo por parte del hospital ha producido retrasos en el trabajo.

Para concluir, la metodología propuesta, basada SDL entrega grandes ventajas para representar un sistema hospitalario utilizando un lenguaje gráfico y sin ambigüedades. SDL está normalizado por la ITU y cuenta con varias herramientas que permiten realizar la simulación, lo que implica una clara diferenciación del modelo de simulación y su aplicación, y simplificar el proceso de verificación.

ANEXO

I.- Conceptos del Lenguaje de Descripción y Especificación SDL

A.- Introducción

SDL (Specification and Description Language) es un lenguaje orientado a la especificación y descripción de sistemas complejos, interactivos, orientados a eventos, de tiempo real o que presenten un comportamiento paralelo, y donde módulos o entidades independientes se comunican por medio de señales para efectuar su función (The International Engineering Consortium, 2010).

Se desarrolló en 1972, cuando un grupo de estudio de 15 miembros de la CCITT¹⁶, representando a varios sectores de la ingeniería eléctrica y de la industria, comenzaron a definir un lenguaje para especificar sistemas de telecomunicaciones. SDL está incluido en la recomendación Z.100 de la ITU-T, que hace referencia a los Lenguajes y aspectos generales de software de Sistema de Telecomunicaciones. La primera versión del lenguaje fue lanzada en 1976. Desde entonces ha sido actualizada varias veces. La última versión corresponde al año 2010. Hoy en día SDL se utiliza en distintos tipos de sistemas, incluidos los sistemas que operan a tiempo real o sistemas sociales que incluyen la descripción del comportamiento de distintas entidades y las relaciones entre éstas (The International Engineering Consortium, 2010).

El éxito de SDL se puede atribuir a su forma de presentación gráfica. Esto hace que sea fácil de entender las especificaciones y diseños creados para representar al sistema, incluso para personas con poco conocimiento en modelación y sistemas. Otro factor es la adecuación conceptual del lenguaje: la noción de una máquina de estados finitos extendido (EFSM). SDL ofrece una forma práctica de especificar sistemas con varias instancias de comunicación mediante EFSM (Reed).

Un sistema de SDL está formado por la comunicación de uno o más agentes. La conducta de estos agentes se realiza mediante EFSM, mediante una estructura jerárquica con agentes que contienen otros agentes, variables (propiedad de los agentes), datos, o referencias de los tipos de datos, y la comunicación se basa en el intercambio de mensajes asíncronos.

¹⁶ CCITT: Comité Consultivo Internacional Telegráfico y Telefónico, actualmente, Unión Internacional de Telecomunicaciones (ITU-T).

B.- Algunas características

Entre los problemas de descripción de sistemas que intenta enfrentar SDL se encuentran:

Problema de dominio, tales como:

Concurrencia

- Responder ante la demanda de usuarios internos, externos y otros sistemas que pueden operar al mismo tiempo en el sistema.
- El sistema y el medio ambiente al mismo tiempo interactúan entre sí.

Comunicación

- El ambiente interactúa con el sistema a través de una serie de interfaces simultáneas.
- Las distancias físicas varían.
- El sistema y el medio ambiente interactúan mediante la señalización, no por manipulación directa.

Comportamiento secuencial

- En cada interfaz individual, el sistema se espera que se comporten de manera secuencial estímulo-respuesta, impulsado por eventos externos.
- Las respuestas del sistema dependerá de las condiciones en tiempo real y en los datos almacenados en el sistema.

SDL permite elegir entre dos formas diferentes para la representación de sistemas. La primera es SDL/GR (Graphical Representation) es un lenguaje gráfico que define la estructura y flujos de control del sistema. La segunda es SDL/PR (Phrase Representation), que es un lenguaje en formato de programación.

SDL se utiliza para describir dos tipos de propiedades del sistema:

1. Las propiedades estructurales, tales como arquitectura del sistema y su descomposición en forma de bloques funcionales interconectados.
2. Las propiedades del comportamiento, es decir, la reacción del sistema ante estímulos provenientes del medio ambiente y de las interacciones entre procesos que componen el sistema.

La arquitectura y el comportamiento son dos características independientes en el sistema. A fin de describir plenamente las propiedades estructurales y de comportamiento de un sistema, SDL se basa en tres modelos semánticos. Estos son:

1. Modelo de arquitectura, el cual consta de:
 - a. Una descomposición en términos de interconexión de las entidades funcionales, tales como: sistema, bloques, sub-bloques, procesos, procedimientos y canales.
 - b. Una organización jerárquica siguiendo el diseño de arriba hacia abajo.

2. Modelo de comportamiento, el cual describe:
 - a. Un conjunto de procesos cooperativos, donde la comunicación se realiza a través del intercambio de señales y el acceso a variables compartidas.
 - b. Cada proceso se describe por un conjunto finito de estados (como una máquina de estados finitos extendido (EFSM)).
3. Modelo de datos, el cual describe:
 - a. Las estructuras de datos manipulados por los procesos.
 - b. Los tipos abstractos de datos.

Las principales ventajas que tiene describir un sistema mediante SDL es que permite:

- Simplificar, documentar, comprender y modelar sistemas simples y complejos mediante su entorno gráfico.
- Representar de manera detallada, la estructura, el comportamiento y las relaciones existentes entre los diferentes elementos del proceso.
- Trabajar en un lenguaje estándar, lo que facilita traducir la descripción del sistema en otros lenguajes.
- Verificar y validar el modelo. Debido a que tiene una semántica completa, la descripción del sistema puede ser rápidamente chequeada y depurada.
- Separar las funcionalidades y los aspectos ambientales.
- Representar el sistema en forma modular, lo que facilita estudiar el problema descomponiéndolo en partes y en forma general.
- Formalizar los canales de comunicación y los mensajes entre las distintas entidades que interactúan en el sistema.
- Conocer el comportamiento del sistema, manteniendo la independencia entre el modelo y la herramienta seleccionada en caso de llevar a cabo una simulación.
- Contar con una serie de herramientas que permiten evaluar, editar, simular y mejorar la descripción y especificación del sistema mediante SDL. Algunas de estas herramientas se presentan más adelante.

C.- La estructura del sistema

Para describir un sistema, el lenguaje SDL se compone de entidades. La entidad principal es el *sistema*, el cual está compuesto por *bloques (y/o subbloques)*. Los bloques están conectados entre sí y con el entorno mediante canales, los cuales transportan señales. Dentro de los bloques están los *procesos*, los cuales se definen por estados, mediante variables, parámetros, acciones y temporizadores. Los procesos también se comunican entre sí por medio de señales. Por último, existen los *procedimientos*, los que permiten detallar algunas tareas, acciones o estados específicos que se llevan a cabo en un proceso. Estas entidades se organizan en forma jerárquica, la cual se puede resumir de la siguiente manera:

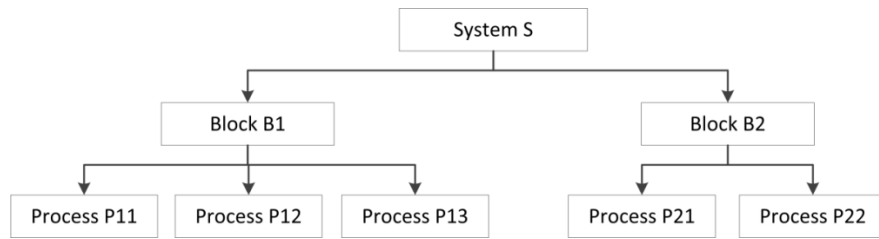


Figura 1. Esquema de jerarquía en SDL (Hogrefe).

Un ejemplo de aplicación de esta jerarquía en una estructura de un sistema es:

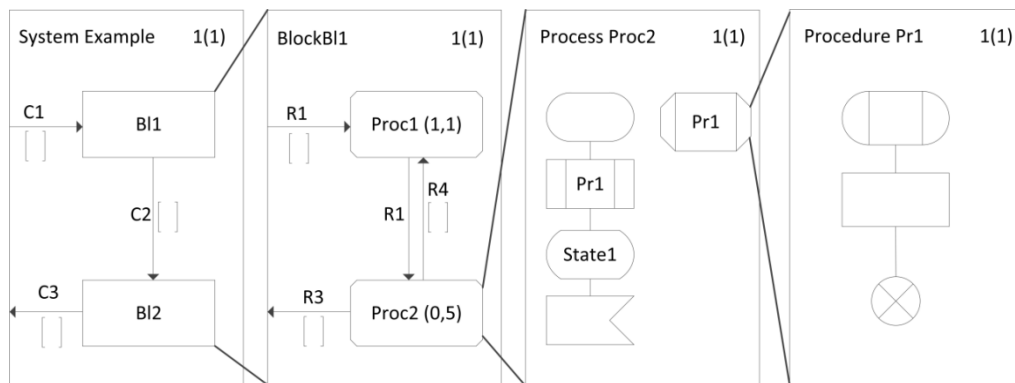


Figura 2. Estructura de un sistema en lenguaje SDL (Doldi, 2003).

Algunos conceptos y/o principios que abarca el lenguaje SDL y que se deben cumplir son:

1. Una descripción SDL debe contener exactamente un sistema.
2. El sistema debe contener al menos un bloque.
3. Un bloque puede contener procesos o bloques, pero no ambos.
4. Un proceso no puede contener otros procesos o bloques.
5. Cada proceso contiene un conjunto de estados, los que describen su comportamiento.
6. La estructura se compone del sistema, los bloques y los procesos y están relacionados jerárquicamente.
7. La comunicación se realiza a través de las señales, las vías de comunicación (los canales) y los parámetros (que pueden ser transportados por las señales).
8. El comportamiento se define con los diferentes procesos.
9. Los datos son basados en ADT (Tipos de datos abstractos).
10. La herencia es útil para describir las relaciones entre los objetos y su especialización.
11. Los bloques y los procesos son denominados agentes. Estos agentes utilizan los EFMS para describir el comportamiento del sistema.
12. La diferencia esencial entre un agente de bloque (o sistema) y agente de procesos es que las instancias de los agentes dentro de un agente de bloque se comportan al mismo tiempo y de forma asincrónica con los demás, mientras que las instancias dentro de un proceso está prevista para ejecutarse una a la vez (es decir, concurrentemente).

13. Además de contener otros agentes, los agentes pueden contener un conjunto de estados, variables de datos y procedimientos.
14. Un diagrama de agente de SDL es la definición de un conjunto de instancias del agente. Cada instancia de agente de este conjunto se crea ya sea cuando la instancia que contiene el conjunto se ha creado o por crear una acción en otra instancia del agente. El agente del sistema se crea cuando se inicializa el sistema.

D.- Las Entidades

Como se mencionó previamente, para describir un sistema, el lenguaje SDL utiliza entidades. Estas permiten estructurar y especificar el comportamiento del sistema, tanto en forma interna como con su entorno (el entorno de un sistema corresponde a todos los demás sistemas que interactúan con él). Las entidades son: el sistema, los bloques, los procesos y los procedimientos.

Una entidad queda especificada con el contexto de su definición y la naturaleza de su tipo. La identidad completa debe ser única, pero no necesariamente los nombres (los cuales sólo pueden contener letras, números y guión bajos). El detalle de cada entidad se presenta a continuación.

D.1.- El Sistema

Un sistema es la entidad de más alto nivel jerárquico y representa a todo el conjunto a modelar. Puede considerarse como el agente de bloque más externo. Está separado de su entorno¹⁷ por la frontera del sistema, con el que se comunica por medio de señales que son transportadas por los canales. En el sistema se definen los bloques que lo componen. Estos bloques están conectados por los canales, que también transmiten señales, permitiendo la comunicación entre ellos o con el entorno. En el sistema también se deben especificar las señales y los tipos de datos con los que se van a trabajar. Esta especificación debe ser visible en todos los componentes. Por último, un sistema siempre debe tener un nombre que lo identifique.

La comunicación entre el sistema y el entorno se puede resumir de la siguiente forma:

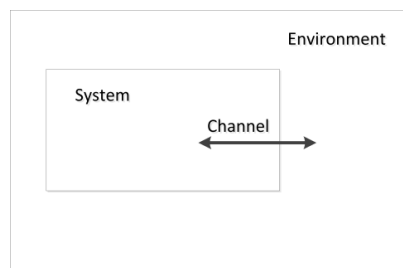


Figura 3. Comunicación de un sistema con el entorno (Hogrefe).

¹⁷ Entorno corresponde a todas las entidades que está afuera del sistema.

La especificación de un sistema es un conjunto de diagramas. Cada diagrama tiene una o más presentaciones "páginas", y cada página tiene:

- Un marco (a menudo con cierta información adjunta en el exterior).
- El esquema de partida dando la clase y la identidad del ítem, descrito en el diagrama en la esquina superior izquierda
- El nombre de la página y el número de páginas en la esquina superior derecha.

Un ejemplo de un sistema en SLD es:

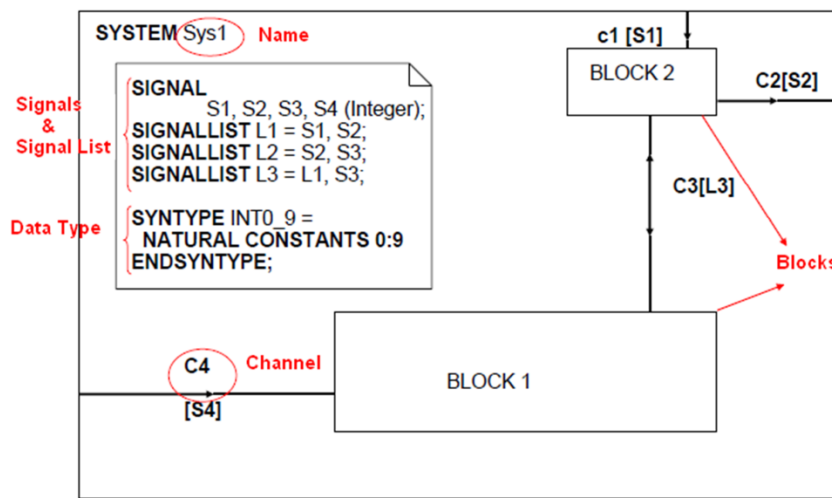


Figura 4. Ejemplo de un sistema en SLD (El Barachi, et al.).

D.2.- Los Bloques

La misión de un bloque es agrupar los procesos que realizan una función en común. Un bloque contiene uno o más procesos (al menos uno). Los bloques se conectan mediante canales, por tanto, la definición de un bloque proporciona una interfaz de comunicación con el entorno, otros bloques y los procesos. Permite establecer un ámbito para las definiciones de procesos. La siguiente figura muestra la estructura y relaciones de bloques:

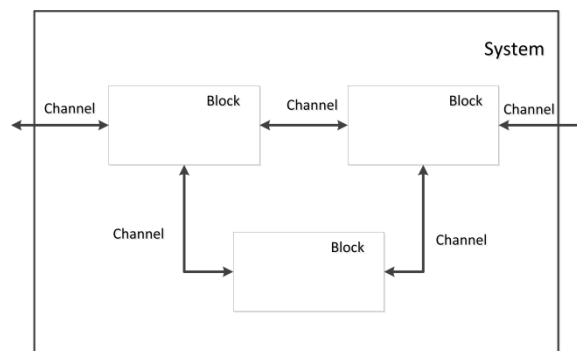


Figura 5. Estructura de bloques en un sistema (Esteban Sastre).

Existen dos tipos de bloques: 1) Bloque terminal (Terminal block), compuesto exclusivamente de procesos y 2) Bloque no terminal (Non-terminal block), compuesto exclusivamente de bloques. La figura n° 6 muestra ambos tipos:

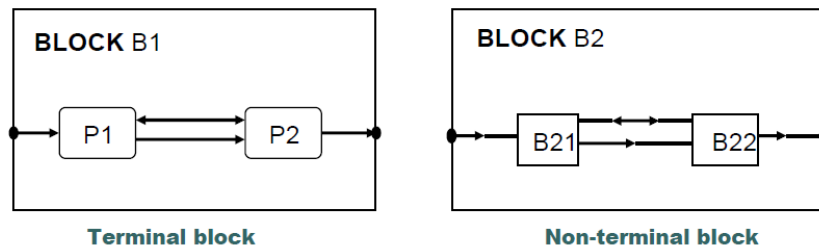


Figura 6. Tipos de bloques. El Bloque B1 está compuesto por los procesos P1 y P2. El bloque B2 está compuesto por los bloques (o sub-bloques) B21 y B22 (El Barachi, et al.).

Las características principales de cada uno son:

1. Bloque terminal (Terminal block): Su especificación incluye un nombre y la definición de:
 - a. Los procesos que describen el comportamiento del bloque.
 - b. Los medios de comunicación, es decir, las señales que conectan los procesos entre sí y con su entorno.
 - c. Las conexiones entre los canales exteriores al bloque y las rutas de señales del interior del bloque.
 - d. Las señales y lista de señales (signal and signal list) que serán usadas en el block.
 - e. Los tipos de datos, los que deben ser visibles en todos los procesos del bloque.

Un ejemplo de este tipo de bloque es:

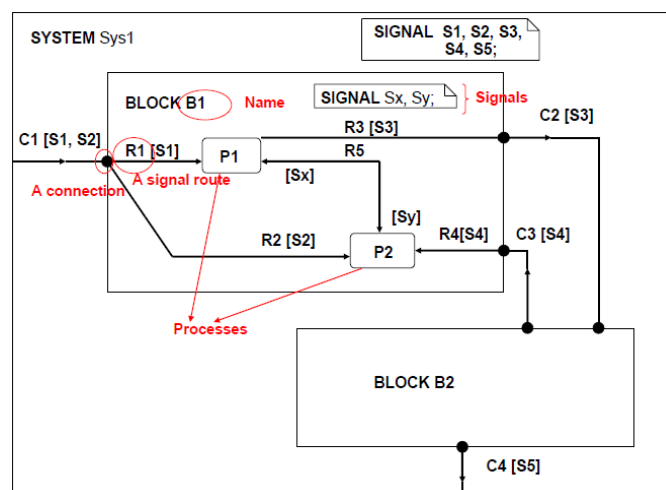


Figura 7. Estructura de un bloque terminal. El sistema Sys1 contiene los bloques B1 y B2. El bloque B1 contiene los procesos P1 y P2. El proceso P1 comunica a su entorno la señal S1 a través del canal R1 y al proceso P2 la señal Sy, a través del canal R5. El proceso P2 comunica a su entorno la señal S2 a través del canal R2 y al proceso P1 la señal Sx, a través del canal R5 (El Barachi, et al.).

2. Bloque no terminal (Non-terminal block): Su especificación incluye un nombre y la definición de la subestructura del bloque:
 - a. Similar a la definición de un sistema.
 - b. La definición de las conexiones entre los canales exteriores al bloque y canales al interior del bloque.

Un ejemplo de este tipo de bloque es:

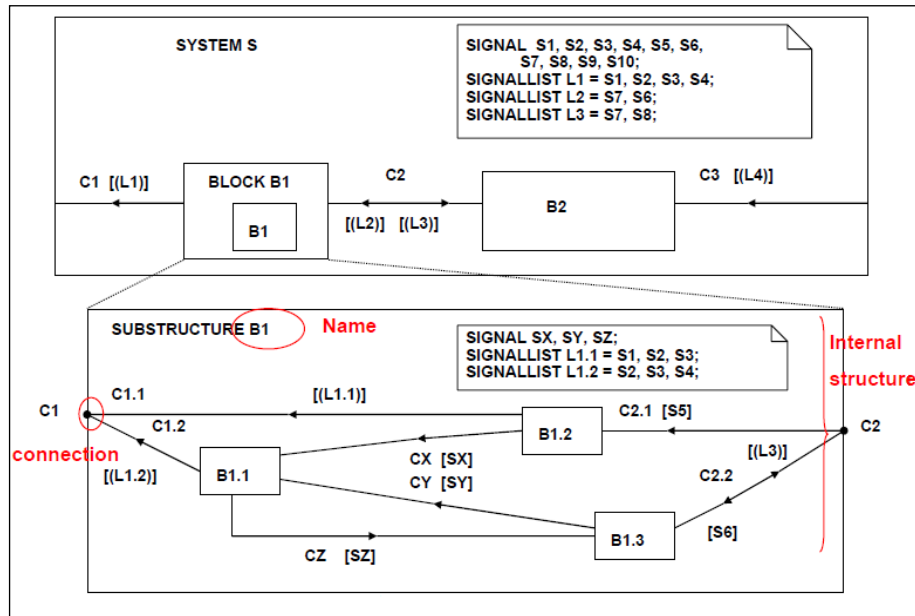


Figura 8. Estructura de un bloque no terminal. El sistema S contiene los bloques B1 y B2. El bloque B1 contiene los bloques B1.1, B1.2 y B1.3 (El Barachi, et al.).

D.3.- Los Procesos

Los procesos son las actividades que ocurren en un bloque. En su especificación se debe tener en cuenta que:

- Cada proceso debe tener un nombre que lo identifique.
- Un proceso contiene estados y transiciones entre estados, cuya función es describir las tareas o el comportamiento que ocurre en el bloque.
- Al conjunto de estados se les denomina EFSM (extended finite state machine). Corresponden al cuerpo del proceso y también deben ser definidos en la etapa especificación.
- Para que un cambio de estado ocurra, el proceso debe recibir una señal válida de otro proceso o del entorno.
- Al realizar esa transición (de un estado a otro), se pueden realizar operaciones, tales como: manipular los datos locales del proceso (por medio de variables, permitiendo cualquier tipo de datos), enviar señales a otros procesos o al entorno, llamar a un procedimiento o incluso crear instancias de otros procesos (llamar a otros procesos en un determinado estado).

- Las variables utilizadas en el proceso (variables locales), deben ser declaradas.
- Se deben definir los tipos de datos, procedimientos, macros y tiempos (timers¹⁸) que se van a utilizar.
- Las señales recibidas por el proceso se denominan señales de entrada y las señales enviadas se denominan señales de salida, aunque en el conjunto de señales de entrada válidas se debe añadir las señales de temporizador, cuya función es operar con tiempos.
- Una señal solo puede ser consumida por un proceso cuando éste se encuentra en un determinado estado.
- Un proceso puede contener parámetros, posterior a la creación de una instancia (estado del proceso). Estos deben ser definidos de manera formal en el diseño del proceso¹⁹.
- Pueden existir varias instancias de un proceso ejecutándose concurrentemente y además simultáneamente con instancias de otros procesos.
- Las instancias de un proceso pueden existir en el inicio del sistema, o pueden ser creadas después de una solicitud de otro proceso²⁰.
- Dos números se utilizan para especificar el número de instancias de un proceso: El Número de instancias creadas al iniciar el sistema y El número máximo de instancias en ejecución de forma simultánea²¹. Estas deben ser definidas en el diseño del proceso.
- Los procesos también tienen acceso al tiempo absoluto (NOW) y pueden realizar cálculos con los tiempos.
- Con respecto a la recepción de señales de un proceso, cada instancia tiene una única cola de señales de entrada que no comparte con otros.
- Para cada estado hay un conjunto de señales de conservación (saved signals). Éstas son señales que se pueden evitar o conservar en un estado concreto para posponer su procesamiento, ya que debido a que las señales se almacenan en una cola de disciplina FIFO²², puede ser de interés evitar ciertas señales para consumirlas posteriormente y no interrumpir el funcionamiento del sistema. Éstas señales serán las primeras en ser consumidas una vez se haya realizado la transición de estado en el mismo orden en el que llegaron.

¹⁸ Un Timers es un meta-proceso capaz de emitir señales al proceso en ejecución. Se definirá más adelante..

¹⁹ A esta formalización se le denomina FPAR.

²⁰ Concepto de padres e hijo (descendiente).

²¹ Por ejemplo, Process1(0, 5) indica que hay cero instancias al iniciar el proceso y que cinco es el número máximo de instancias simultáneas que pueden ser creadas por otros procesos.

²² Primero en llegar, primero en atenderse.

Un proceso dentro de un bloque se puede representar de la siguiente forma:

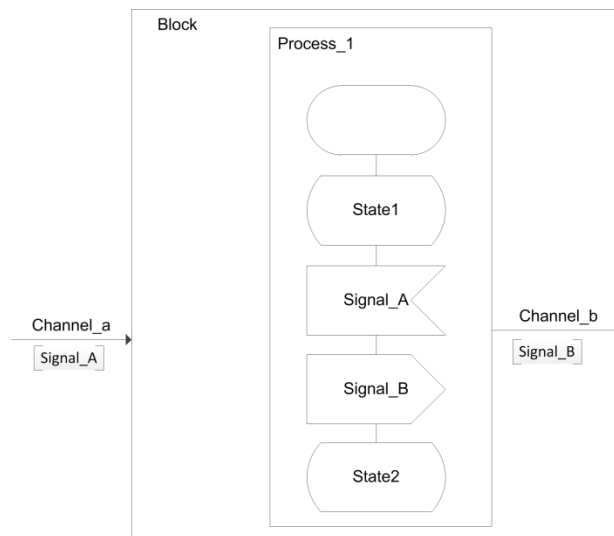


Figura 9. Proceso dentro de un bloque.

Un proceso dinámico es un proceso donde ocurre un llamado a la instancia de otro proceso. Esto se muestra en la figura nº 10:

- La creación de una instancia de proceso sólo puede hacerse mediante un proceso del mismo bloque.
- La adopción de los valores en los parámetros es opcional.
- Identificación de instancias de Procesos. Para identificar cada instancia en un determinado proceso, existen cuatro variables implícitas, que entregan distintos valores del PID²³ :
 - Sí mismo (Self): El PID de la instancia actual.
 - Remitente (Sender): El PID de la instancia que envió la señal de entrada anterior.
 - Padres (Parent): El PID de la instancia que se creó la instancia actual (Null si instancia fue creada en el inicio del sistema).
 - El hijo o descendiente (The Offspring): El PID de la última instancia creada.

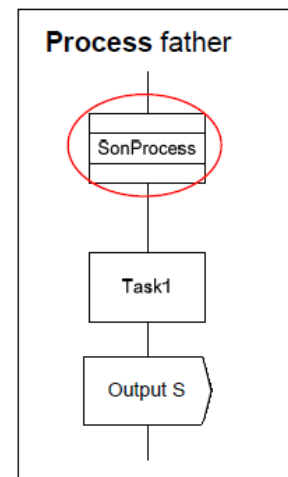


Figura 10. El proceso llamado

father llama a instancia de proceso llamado SonProcess (este último proceso sería el proceso hijo) (El Barachi, et al.).

²³ PID es un tipo de datos que permite definir las distintas características de una interfaz de un proceso.

Estos cuatro valores identifican de forma única la instancia de cada proceso. Son necesarios cuando queremos comunicarnos con una instancia de otro proceso en particular.

Un ejemplo de distintas instancias de proceso es:

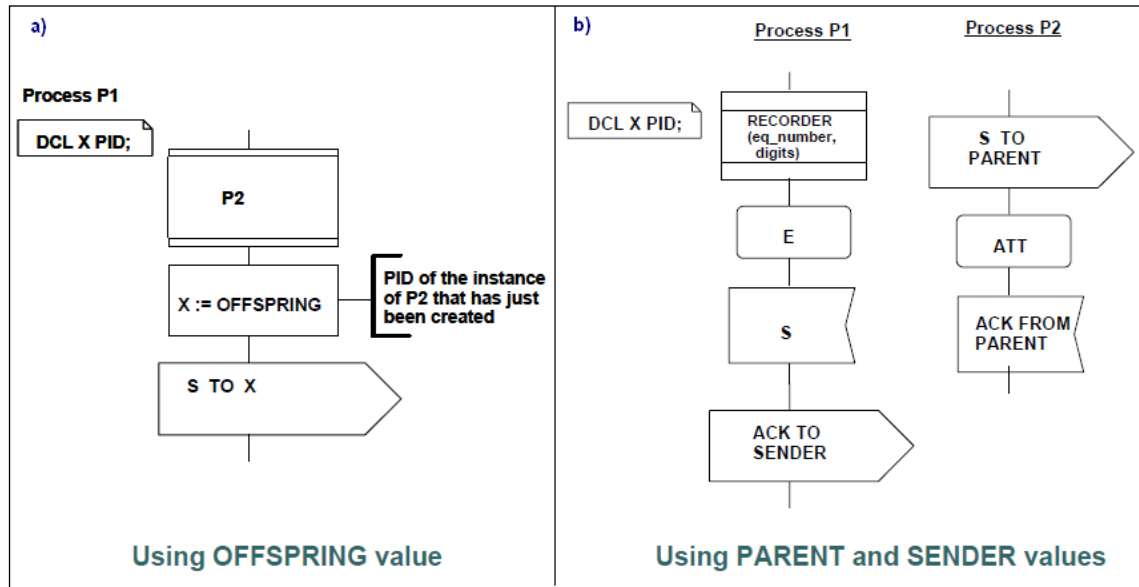


Figura 11. Aquí se presentan dos instancias distintas. En el esquema a) correspondiente al Proceso P1, el PID de la instancia de P2 se acaba de crear. Para esto se asignó el valor Offspring a la variable X. En el esquema b) se utilizan los valores Parent y Sender en los procesos P2 y P1 respectivamente (El Barachi, et al.).

- Cada interfaz añade una operación de comprobación de compatibilidad. Esta consiste en que, dada una señal, determinará si bien:
 - La señal es definida o utilizada en la interfaz ó
 - Chequea si la compatibilidad se cumple en el tipo de PID que se definió.

D.3.1- Cuerpo de un proceso.

El cuerpo de una entidad se refiere a los elementos básicos que debe poseer para una especificación correcta del comportamiento que se desea describir. En el caso de un proceso, el cuerpo tiene los siguientes elementos:

- Inicialización de transición (START), que lleva el proceso a un estado inicial.
- Conjunto de estados. , en los que se incluyen:
 - Un nombre del estado
 - Un conjunto de estímulos posible en este estado.
 - La transición, que será el efecto a la recepción de un estímulo en el proceso. Una transición se compone de:
 - Un conjunto de acciones (TAREA, OUTPUT, CREATE REQUEST, DECISION, SET, RESET).

- Un terminador de la transición, que es:
 - La llegada a un nuevo estado (NEXTSTATE)
 - Una combinación (JOIN)
 - La terminación del proceso (STOP).

El cuerpo de un proceso se puede representar como:

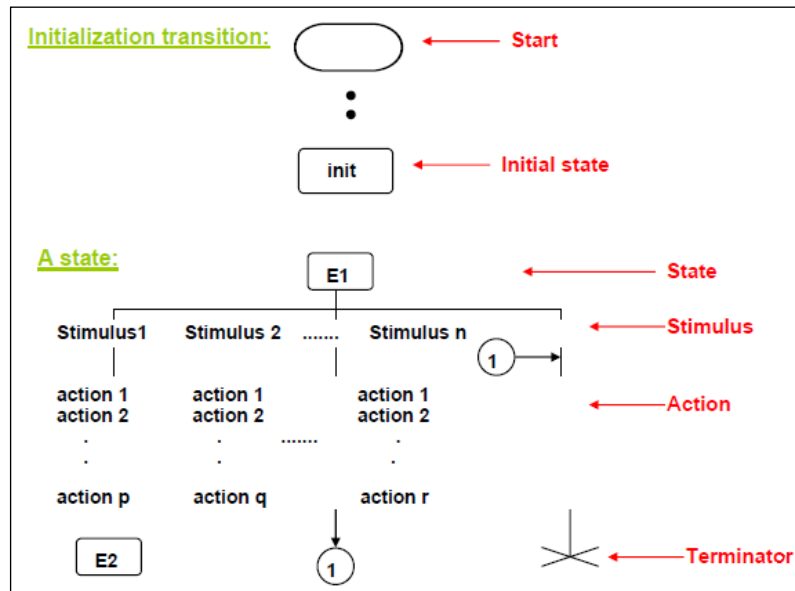


Figura 12. Cuerpo de un proceso, con sus distintos componentes (El Barachi, et al.).

D.4.- Los Procedimientos.

Un procedimiento es el último nivel en la estructura del lenguaje SDL. Su función es especificar acciones, estados, variables, parámetros y temporizadores que se pueden dar en un determinado proceso. Puede ser usado para factorizar partes comunes del comportamiento o clarificar la descripción.

Los procedimientos tienen las siguientes características:

- Pueden ser declarados en cualquier lugar.
- Pueden o no devolver un valor.
- Pueden contener declaraciones de procedimientos anidados.
- Pueden contener estados (y luego las entradas de señales).
- Pueden ser recursivos (llamarse a sí mismo o auto invocarse).
- Llamadas a procedimientos remotos: un proceso puede llamar a un procedimiento de otro proceso (como si fuese definido a nivel local).
- La combinación de valores de retorno al procedimiento y llamado a procedimiento remoto resulta en un modelo más elegante que a un intercambio de señales en los casos de dos vías de comunicación.
- No esperan la recepción de señal.

La especificación de un procedimiento en SDL incluye un nombre que lo identifique y la definición de:

- Los parámetros de manera formal²⁴.
- Los tipos de datos, otros procedimientos, macros y tiempos (timers) que se van a utilizar en el procedimiento.
- Las variables, que se deben declarar en un cuadro llamado TEXT.
- El diseño del cuerpo del procedimiento (similar al cuerpo de un proceso).

Un ejemplo de la estructura de un procedimiento se muestra en la Figura 13:

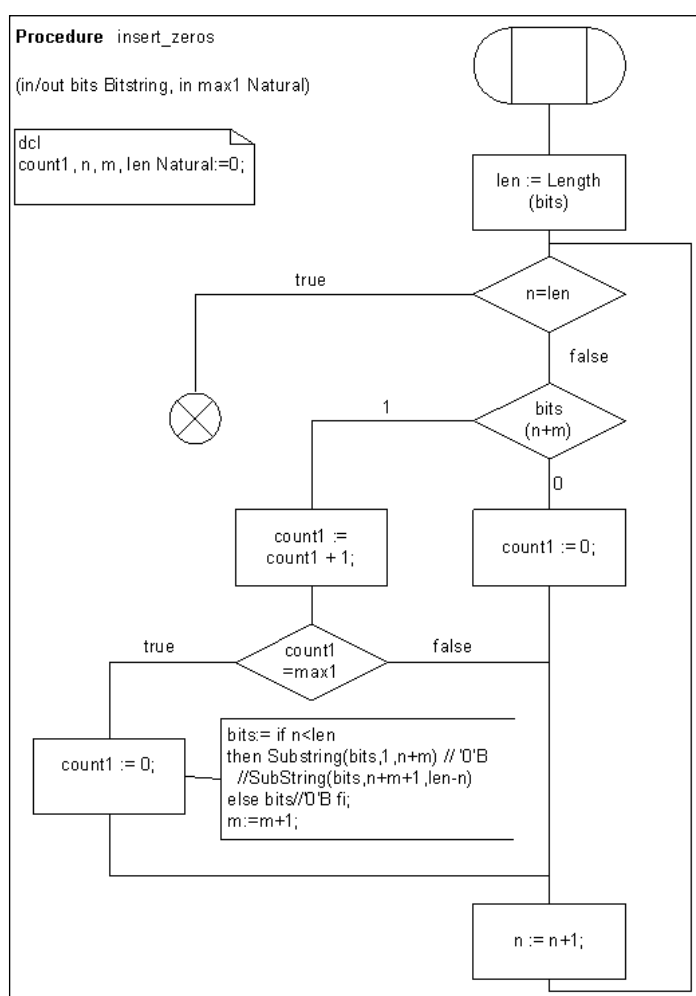


Figura 13. Estructura de un procedimiento (Sandrila Ltd).

²⁴ Si el parámetro se define como valor (Attribute IN), entonces actúa como una variable local dentro el procedimiento. Si se define como referencia (Attribute IN / OUT), entonces actúa como sinónimo para una variable de transmisión.

Un ejemplo de invocación a un procedimiento es:

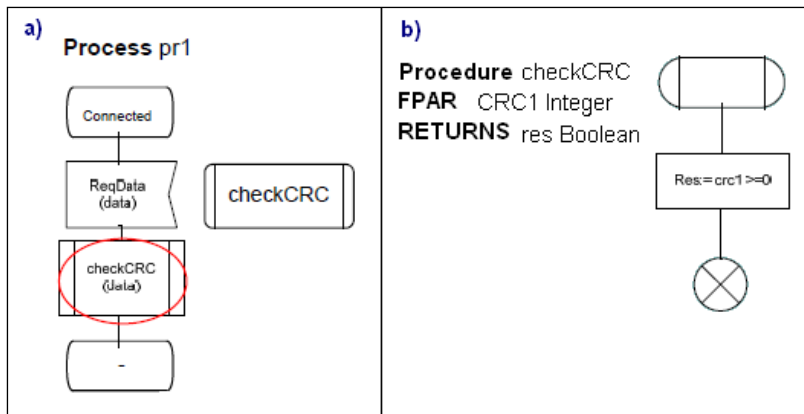


Figura 14. En el esquema a), el proceso pr1 invoca al procedimiento checkCRC. En el esquema b) se presenta el procedimiento, la formalización del parámetro CRC1 y solicita como retorno el valor res como dato tipo boolean (El Barachi, et al.).

D.5.- Las Macros.

De manera similar a los procedimientos, las macros se utilizan en la descripción del comportamiento, preferentemente para realizar acciones donde se requiere repetición. A diferencia de un procedimiento, el llamado a una macro se sustituye por la macro correspondiente en el momento de la ejecución del proceso. Pueden tener parámetros y sólo podrán ser utilizados dentro de los procesos o procedimientos.

Una macro es un conjunto de instrucciones que tienen un punto de entrada y un punto de salida. Todas las derivaciones realizadas deben permanecer dentro de la macro. La especificación de una macro incluye un nombre que la identifique y la definición de:

- Los parámetros de manera formal (FPAR)²⁵.
- El diseño del cuerpo de la macro.

Un ejemplo de invocación a una macro es:

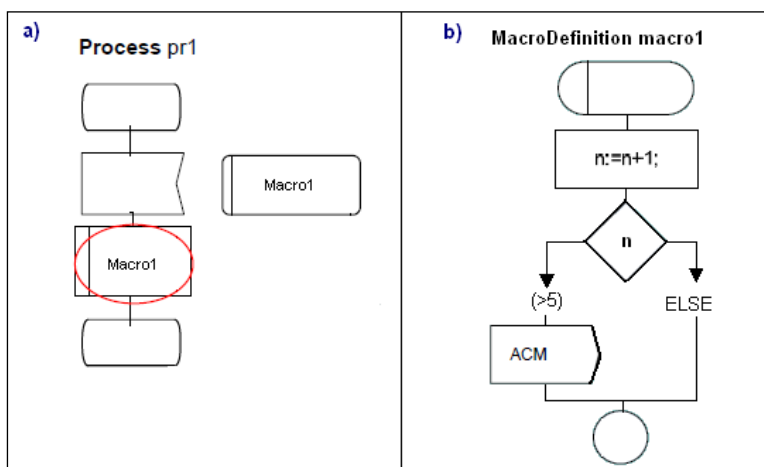


Figura 15. En el esquema a), el proceso pr1 invoca a la macro Macro1. En el esquema b) se presenta la macro invocada (El Barachi, et al.).

²⁵ Representadas por un identificador que contiene: la señal, la variable y una etiqueta.

E.- Comportamiento del sistema

Dado que se han revisado las entidades asociadas a la especificación de un sistema, se presentarán a continuación algunos aspectos importantes que permiten entender el concepto de comportamiento del sistema basados en estados. Estos son:

- Un sistema en SDL, es un conjunto de máquinas de estados finitos extendido (EFMs), comunicados por medio de señales.
- Una máquina de estados finitos (FSMs) posee las siguientes características:
 - Una cola con disciplina FIFO.
 - A la recepción de una señal *S*, una transición se realiza, una o más señales se emiten y la máquina se mueve a un nuevo estado.
 - 1 es el estado inicial de la máquina, F es el estado final, 0 es un estado reservado para el caso en que el arribo de la señal no sea esperado.

Un ejemplo de un FSMs es:

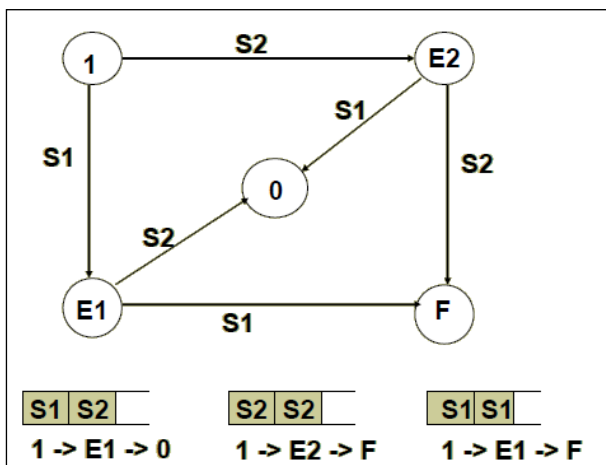


Figura 16. "1" es el estado inicial de la máquina. Al recibir la señal S1, cambia (o se mueve) a E1. Si recibe nuevamente la señal S1, la máquina cambia al estado F. En caso de recibir la señal S2, cambia (o se mueve) a E2. Si recibe nuevamente la señal S2, la máquina cambia al estado F. Tanto en el estado E1 o E2, en caso de recibir una señal inesperada se mueven al estado 0 (El Barachi, et al.).

- Esta definición de FSMs se puede aplicar a cualquier tipo de sistema, donde las máquinas pueden ser departamentos en una empresa, servicios clínicos en un hospital, cajas de atención de público, etc. Es decir, cualquier entidad que implique realización de distintas actividades y por lo tanto, cambios de estados.
- Un EFSMs tiene las siguientes características:
 - Memoria local, con la posibilidad de hacer operaciones en esta memoria.
 - Las pruebas sobre las variables locales pueden condicionar el estado que continua.
 - Una reducción considerable en el número de estados.
 - La posibilidad de trabajar en espacios de estados infinitos.

Un ejemplo de estas características es:

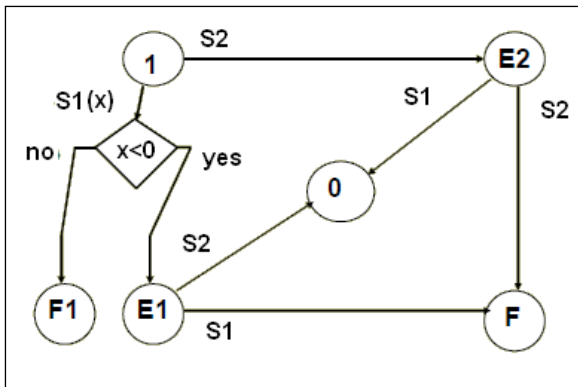


Figura 17. En este caso, el estado 1 al recibir la señal S1, conduce a la pregunta si $x < 0$. Si es correcto, el proceso llega al estado E1, en caso contrario llega al estado F1 y termina el proceso (El Barachi, et al.).

- El proceso SDL extiende el modelo de FSM a otros procesos con características distintas, tales como:
 - Disciplina de la cola es distinta de FIFO.
 - Acciones en una transición que no se limitan sólo al envío de una señal y al examen de las variables locales para determinar la siguiente transición, sino que también puede incluir:
 - La ejecución de tareas (asignación o texto estructurado).
 - Llamadas a procedimientos.
 - Creación de procesos dinámicos.
 - Armado y desarmado de temporizadores.
 - Duración de una transición que no necesariamente tiene que ser cero.
- Un proceso SDL lleva a cabo una transición antes de la inicialización situándose en el estado inicial.
- La ejecución de un temporizador es modelada por la llegada de una señal en la cola del proceso
- Las variables pueden tener varios grados de visibilidad:
 - Las variables locales, visibles sólo dentro del proceso.
 - Las variables reveladas, visibles por todos los procesos en el mismo bloque.
 - Las variables exportables, visibles por procesos en otros bloques.

F.- La comunicación

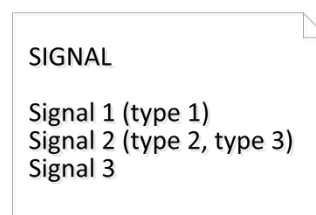
En SDL, las máquinas de estado que figuran en los procesos se comunican entre sí o con el medio ambiente mediante:

- La transmisión y recepción de señales (o variables remotas o llamadas a procedimientos).
- A través de los canales y rutas de señal.

F.1.- Las Señales

Una señal es un flujo de información entre entidades. Puede definirse como un evento transitorio asincrónico, transmitido por una instancia de proceso a otra instancia de proceso. Las características de las señales son:

- Antes de ser utilizadas, deben ser declaradas.
- La declaración de las señales en SDL/GR se realizan mediante la inclusión de un cuadro llamado TEXT, cuyo contenido debe comenzar por la etiqueta SIGNAL y posteriormente los nombres de las señales seguidas de los tipos de los parámetros entre paréntesis (siendo esto opcional), como por ejemplo:



```
SIGNAL
Signal 1 (type 1)
Signal 2 (type 2, type 3)
Signal 3
```

Figura 18. Ejemplo de señales (Esteban Sastre).

- Las señales pueden ser declaradas de dos formas:
 - Individualmente (por ejemplo, sig1, sig2, sig3, etc.) ó
 - Agrupadas en listas (por ejemplo, sigList1 = sig3, sig4).
- Una vez declaradas las señales pueden ser transmitidas a través de los canales o las rutas de señales.
- Una declaración de señal incluye:
 - El nombre de la señal.
 - Los tipos de parámetros que serán transmitidos por la señal (lista de clases).
- Una declaración de una lista de señal incluye:
 - El nombre de la lista de señal.
 - El listado de las señales y/o lista de señales que serán transmitidas.
- Las entidades que pueden recibir señales son: los bloques, los procesos, y los procedimientos. Pueden hacerlo con otra entidad de manera local o con su entorno.
- El orden en el que llegarán las señales será el mismo que el orden en el que se envían desde el otro extremo.

F.2.- Los Canales

Un canal es una ruta para transportar señales entre dos entidades. Las características de los canales son:

- En cada canal debe haber una lista de señales que transporta. Por ejemplo

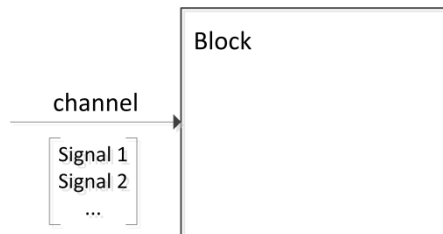


Figura 19. Ejemplo de un canal y las señales que transporta (Esteban Sastre).

- Pueden existir varios canales entre los extremos de las entidades.
- Existen dos tipos de canales de comunicación usados en SDL.
 - Los canales de no retraso: Permiten la transmisión automática o inmediata de la información (señales). Un ejemplo es:

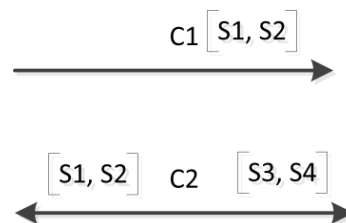


Figura 20. Representación de SDL para canales de no retraso. El nombre del canal es C1 y C2, las señales que llevan son S1, S2, S3 y S4 (Fonseca i Casas, 2008).

- Los canales de retraso: Demoran la transmisión de la información (señales). Un ejemplo es:

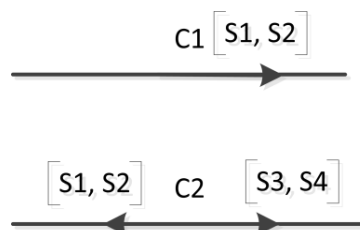


Figura 21. Representación de SDL para canales de retraso (Fonseca i Casas, 2008).

- Los canales de retraso pueden asociarse a un temporizador para indicar el tiempo de retraso de la señal.

La especificación de un canal/ruta de señal incluye:

- Un nombre que lo identifique.
- Los posibles caminos del canal (unidireccional o bidireccional).
- Para cada camino, la especificación de los extremos.
- El listado de señales que transporta (señales o lista de señales)

Algunos ejemplos de canales y señales de ruta son:

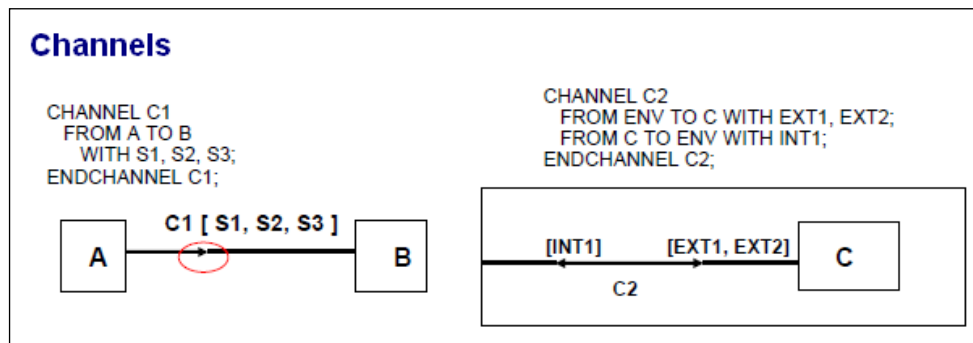


Figura 22. Ejemplos de canales y señales de ruta (El Barachi, et al.).

F.3.- Los Temporizadores

Uno de los recursos que pueden utilizar los procesos son los temporizadores. Un temporizador es un objeto que puede operar con la variable "tiempo". La forma de diseñar un temporizador y sus características son:

- Se inicializa con la sentencia SET, en cuyo caso se le asocia un valor de tiempo.
- Cuando el tiempo del sistema llega al tiempo establecido para el temporizador, se añade a la cola de señales de entrada una señal con el mismo nombre que el temporizador.
- La definición de un temporizador no se permite en los diagramas de estado o en un procedimiento.
- Otra de las operaciones que se pueden aplicar a un temporizador es la operación de RESET, la cual desactiva la operación anterior (además, si hay una señal en la cola de entrada correspondiente al temporizador también se suprime).
- Se pueden distinguir dos estados del temporizador:
 - Activo: Estará en este estado desde la inicialización hasta el consumo de la señal del temporizador por parte del proceso.
 - Inactivo: Corresponde a cualquier otro caso en que el temporizador no este activo.

- Cuando se realiza una operación de RESET, el temporizador pasará de activo a inactivo.
- Para referirse a un tiempo concreto normalmente se utiliza el tiempo absoluto del sistema (NOW), que es visible por todos los procesos y se comparte.
- Para manejar estas operaciones con SDL/GR, se debe hacer primero la declaración de un temporizador, que se realiza mediante la inclusión de un cuadro TEXT con la etiqueta TIMER seguido del nombre del temporizador. Un ejemplo sería:

Un cuadro rectangular con una esquina superior derecha doblada, que contiene el texto "TIMER T;"

Figura 23. Declaración de un Temporizador, en este caso llamado T (Esteban Sastre).

- La sentencia SET recibe dos argumentos. El primero de ellos es el tiempo en el cual vence el temporizador, como ya se mencionó previamente, se emplea la sentencia NOW para utilizar el tiempo actual del sistema sumándole los segundos que se quiere que pasen hasta la activación del temporizador. Esta sentencia se debe añadir en un cuadro llamado TASK, dentro de la definición de un procedimiento. Un ejemplo sería:

Un cuadro rectangular que contiene el texto "SET (NOW + 13, T)"

Figura 24. Activación de un Temporizador. En este caso el temporizador T se activa tras 13 segundos (Esteban Sastre).

- Por otro lado, la reinicialización de un temporizador se realiza mediante la sentencia RESET, la cual recibe un único argumento correspondiente al nombre del temporizador. También debe ir dentro de un cuadro TASK. Para reiniciar el temporizador anterior se añade:

Un cuadro rectangular que contiene el texto "RESET (T)"

Figura 25. Reinicialización de un Temporizador (Esteban Sastre).

- El comportamiento de un temporizador se puede resumir mediante la siguiente figura:

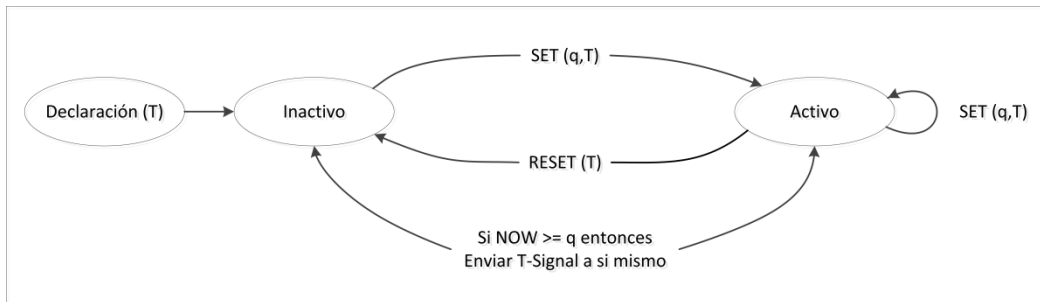


Figura 26. Comportamiento de un Temporizador. El primer estado es inactivo. Con la sentencia SET pasa a estado Activo. Con la sentencia RESET vuelve al estado inicial. Si el tiempo actual NOW es mayor o igual a q (es decir, ya se consumió el temporizador) vuelve a estado Inactivo (Esteban Sastre).

F.4.- Protocolo de parada y espera

El protocolo de comunicación de parada y espera es el protocolo más sencillo entre dos puntos lógicamente correctos. El funcionamiento consiste en:

- El emisor envía un mensaje o trama de datos al receptor.
- El receptor después de la llegada de la trama, envía una señal para indicar que puede continuar.
- El emisor debe esperar esta señal para mandar la siguiente trama.

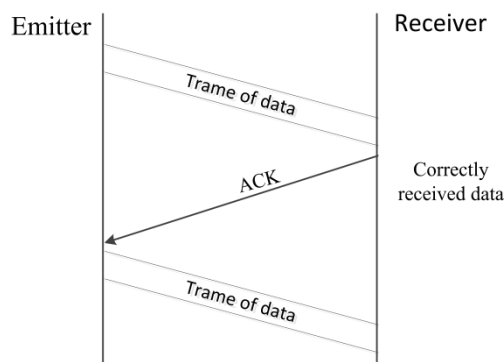


Figura 27. Ejemplo de un Protocolo de parada y espera (Esteban Sastre).

En este tipo de transmisión se pueden perder tramas, por lo que se debe tener una forma de recuperación de errores. Para ello utiliza el mecanismo de ARQ (automatic repeat request), que consiste en retransmisiones por parte del emisor si se detecta algún tipo de error. En el caso anterior, si se pierde una trama de datos, se habrá perdido información. Para solucionar esto, el emisor incorpora un temporizador de retransmisión, y en el caso de no recibir la trama (o señal) de confirmación en un intervalo determinado de tiempo, realiza una retransmisión de la misma trama.

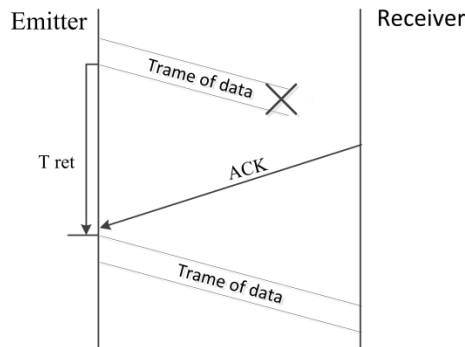


Figura 28. Retransmisión de una trama (Esteban Sastre).

Con esto se ha solucionado el problema de pérdida de tramas pero no el de pérdida de confirmaciones. Si se produce una de ellas, el receptor recibirá dos tramas duplicadas porque vencerá el temporizador y dicha trama se desechará. Para solucionar esto, se introduce numeraciones en las tramas y en las confirmaciones, de este modo el receptor cada vez que confirme la recepción de una trama, enviará además el número de trama por la que espera. La siguiente figura muestra esta situación:

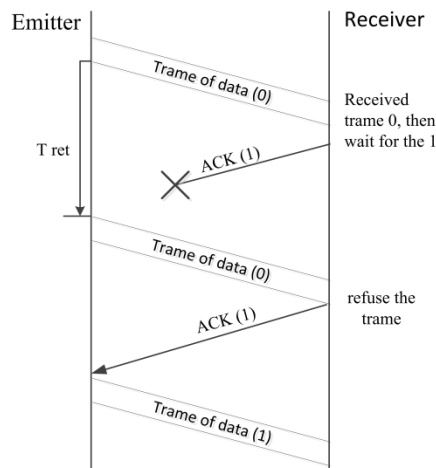










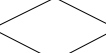







Figura 29. Numeración de tramas (Esteban Sastre).

En el ejemplo, se puede observar que una vez que llega la trama de confirmación ACK con parámetro 1 el emisor ya sabe que el receptor espera por esa trama, enviándola posteriormente. Por último, cabe destacar que sólo son necesarias dos numeraciones de tramas (en este caso, los valores 0 y 1), ya que una vez enviada la trama 1, se volverá a la trama 0.

G.- Los Símbolos

Una vez entendidas las entidades, señales y canales de las que se compone un sistema especificado en SDL, es importante revisar los principales símbolos utilizados para representar el sistema, principalmente el comportamiento de un proceso, procedimiento y una macro. Los principales símbolos son:

Tabla 1. Principales símbolos utilizados en el lenguaje SDL (Esteban Sastre).

Símbolo	Nombre	Uso
	Include	Se usa para incluir librerías SDL en el sistema. Se utiliza la directiva # include seguida de la librería que se quiere añadir.
	Text	Contiene la declaración de las estructuras, variables, temporizadores y señales. Para la declaración de variables se debe utilizar la palabra reservada DCL seguido del nombre y el tipo de la variable que se quiere declarar. Para la declaración de constantes se utiliza la palabra reservada SIGNAL seguida del nombre de la señal y los parámetros que contendrá entre paréntesis.
	Star	Indica el comienzo de un proceso. Debe existir una vez por cada proceso.
	State	Simboliza un estado del sistema.
	Input	Indica la espera de una señal de entrada por lo que se trata de un símbolo bloqueante. Debe encontrarse inmediatamente después de un símbolo de estado. Si la señal es recibida, se consume y se produce una transición de estado poniendo a disposición del proceso la información transportada por la señal. Si la señal lleva información (parámetros), debe ser copiada en variables locales del proceso, por lo tanto, debe haber una variable local en el proceso con el mismo nombre que indique el parámetro de la señal. En caso contrario se descartará el valor y no se almacenará en ningún sitio.
	Output	Envía una señal, usualmente se realiza al final de una transición.
	Task	Se usa para realizar tareas generales, como asignación de variables u operaciones sobre temporizadores. Puede contener una lista de asignaciones separadas por comas.
	Save	Añade una señal al conjunto de señales de conservación de un estado. Se puede añadir una señal concreta o usar el símbolo "*" para referirse a todas las señales.
	Decision	Es usada para escoger entre dos rutas alternativas según el resultado de una condición.
	Connector	Un conector posee una etiqueta que simboliza la continuación desde otro conector que posea la misma etiqueta, es decir, se utilizan para realizar saltos incondicionales.
	Stop	Indica la terminación de un proceso.
	Procedure call	Ejecuta la llamada a un procedimiento previamente declarado
	Procedure reference	Declaración de un procedimiento el cual es llamado en el proceso actual.
	Procedure start	Comienza la definición de un procedimiento.
	Procedure return	Retorno de procedimiento a donde fue llamado.
	Macro call	Ejecuta la llamada a una macro.

H.- Listas de mensajes de secuencia (MSC)

Es un lenguaje que puede ser usado en combinación con SDL. Permite describir las interacciones entre los distintos componentes del sistema (Sistema, bloques y procesos). Fue estandarizado por la ITU en la norma Z.120. Su última versión corresponde al año 2000. Sus principales características son:

- La comunicación entre los componentes es asincrónica.
- Un MSC no es una descripción completa del sistema, más bien se limita a expresar una ejecución parcial.
- Al igual que SDL, MSC tiene una versión gráfica (MSC / GR) y una versión de texto (MSC / PR) para formar la sintaxis.
- Un MSC es una descripción parcial del comportamiento de la comunicación de un número de instancias.
- Una instancia es una entidad abstracta de la cual se puede observar la interacción con otras entidades o con el medio ambiente (por ejemplo, en SDL la interacción entre los bloques y los procesos.) Hay distintos tipos de instancias:
 - Instancia principal: Es la que determina el comienzo de la descripción de la instancia.
 - Instancia eje:
 - Contiene una secuencia de eventos.
 - Se ejecuta de arriba abajo.
 - Cada eje de órdenes son eventos en el tiempo.
 - No hay tiempo pedido explícito entre los eventos en diferentes instancias (MSCs no puede expresar el tiempo absoluto).
 - Instancia final: Es aquella que determina el final de la descripción de la instancia, no la terminación de la instancia.
- Estructura es lo que representa el medio ambiente.
- Los mensajes son:
 - Asíncronos y no-reconocidos.
 - Son representados por una flecha que se inicia en el envío de la instancia (envío de mensajes) y termina en la recepción de la instancia (consumo de mensajes).
- El envío de mensajes y el consumo son los eventos de comunicación.
- Una acción describe una actividad interna de una instancia mediante un texto estructurado.
- No existe la noción del tiempo global.
- Existe dependencia entre el tiempo de las instancias (el mensaje debe ser enviado antes de ser recibido).
- La ejecución de una acción local es sólo limitada por el orden de los eventos en su propia instancia.
- Se asume que no hay un orden en los eventos de comunicación del entorno.

Un ejemplo de una aplicación de MSCs es:

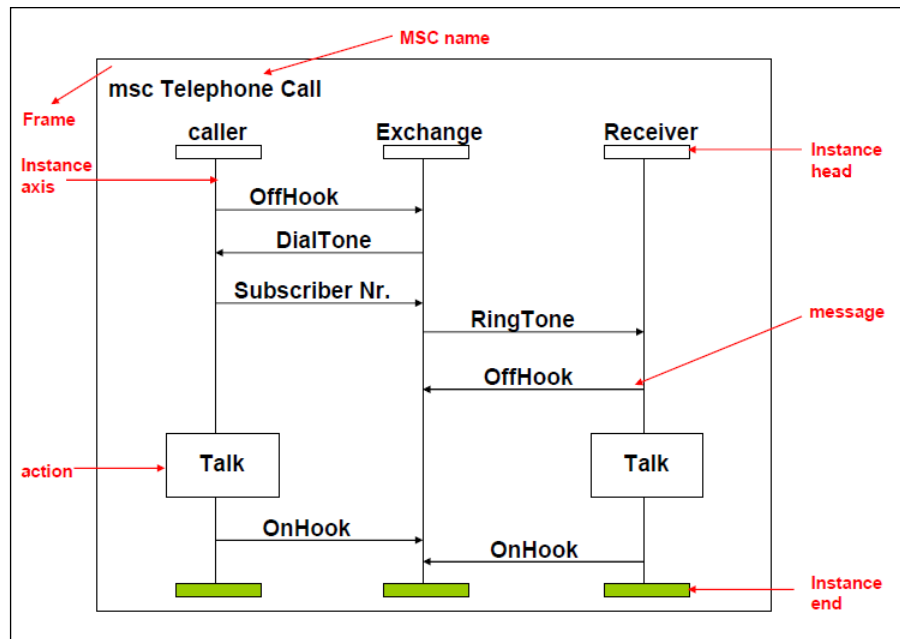


Figura 30. Aplicación del lenguaje MSCs (El Barachi, et al.).

I.- Metodología para el desarrollo de un modelo en SDL

Para modelar un sistema con el lenguaje SDL, se requiere seguir una estructura de trabajo que permita la especificación completa del sistema. Esta estructura se compone de:

1. La metodología de diseño.
2. La Validación y Verificación.
3. Las Herramientas de SDL.

I.1.- Metodología del diseño

Los aspectos que se deben seguir son:

- La especificación en SDL se realiza respecto a la estructura, el comportamiento y los datos.
- Una especificación no es completa hasta que los tres aspectos han sido completamente cubiertos.
- La especificación de un sistema con SDL puede ser desarrollada de manera gradual.
- El proceso de especificación puede ser:
 - Completo pero no formal.
 - Formal pero no completo (puede usar una herramienta de SDL).

- Formal y completo (puede usar una herramienta de SDL).
- La especificación de un proceso puede ser enriquecida por medio de:
 - La adición de nuevos estados para manejar resultados o salidas validas que no se tenían en cuenta previamente.
 - Insertando nuevas salidas, tareas, decisiones, etc., en las transiciones de estado.
- Un proceso de especificación de una etapa previa puede ser formalizada mediante la sustitución de un texto informal por una especificación formal con el lenguaje SDL.
- Extensiones temporales (Señales de retraso). Otros elementos que se pueden incorporar en el diseño de un modelo mediante SDL son las extensiones, entre ellas, las extensiones temporales tales como las Señales de retraso. Estas son útiles en una simulación basada en la programación de eventos discretos (como es el caso de SDLPS). Para definir completamente el comportamiento del modelo, el simulador debe describir los tiempos relacionados con la ejecución de cada uno de los eventos. En general, el comportamiento de estos tiempos se representa mediante distribuciones de probabilidad. En un simulador de programación de eventos, el motor gestiona el tiempo de todos los eventos, y decide donde y cuando todos los eventos deben ser enviados (a los elementos de simulación, agentes en un modelo de SDL).

SDL tiene dos estructuras principales de gestionar el tiempo, los temporizadores (timers) y los Canales de retraso (Delaying Channels) y se puede definir con tres tipos de transiciones (Bozga, et al., 2001), (i) Eager (ansiosos), (ii) Lazy (perezoso) y (iii) Delayable (retrasado). Desde el punto de vista de un simulador discreto, todas las transiciones pueden ser consideradas Delayable

Por otra parte, en SDLPS, todas las señales llevan definidos los siguientes parámetros: (i) ExecutionTime, que representa el momento en que el evento debe ser ejecutado; (ii) Priority, que hace mención a la prioridad del evento y sirve para romper una posible simultaneidad de eventos; (iii) CreationTime, que representa el momento en que el evento se crea; (iv) Id, que es un identificador del evento. (v) Time, que es el reloj del proceso y (vi) Destination, que indica el destino final de la señal (PID).

Estos parámetros permiten que SDLPS retrasar u ordenar por prioridad las distintas señales. En el contexto de SDLPS, se puede utilizar los elementos de extensión para definir estos parámetros relacionados con la señal, como se puede ver en la figura 31. No todos los parámetros de la estructura del evento deben ser definidos, sólo las necesarias para definir completamente el comportamiento del modelo.



Figura 31. Definición del tiempo de retraso y otros parámetros de la señal mediante extensiones de SDL (Fonseca i Casas, 2010).

- Respecto a la metodología de diseño, la ITU en la recomendación Z.100 [2], entrega directrices metodológicas basadas en 9 pasos, de modo que en cada uno de ellos debe:
 - Ser un punto natural para detener el proceso.
 - Producir un resultado autónomo que sea útil y que pueda ser chequeado, preferentemente por herramientas.
 - Ser independiente (o en menor medida) de las posteriores etapas.

Los 9 pasos para el diseño mencionados previamente son:

1. Límite del sistema: Permite que el sistema tenga su estructura interna definida. Para ello se debe:
 - a. Delimitar el sistema de su entorno. Buscar un nombre adecuado para el sistema. Identificar las diferentes entidades en el medio ambiente con que el sistema interactúa. Describir la finalidad y características del sistema de manera informal en un comentario.
 - b. Especificar un canal para cada entidad designada en el medio ambiente, y dar un nombre adecuado que corresponde al nombre de la entidad.
 - c. Introducir un bloque simulado dentro del sistema. Este bloque será sustituido más adelante por la estructura real del sistema.
 - d. Identificar la información que circula o fluye a través de la frontera del sistema, en términos de eventos discretos. Estos hechos constituyen el alfabeto del sistema y son modelados por las señales. Las señales para la comunicación externa se especifican a nivel del sistema. Debe explicar el objetivo de cada señal en un comentario en la especificación de la señal. Identificar la información que se transmite por cada señal e indicar el tipo de parámetros asociado (usar tipos predefinidos en la medida que le sea posible). Asociar las señales con los canales ya sea directamente o por medio de listas de la señal.
 - e. Proporcionar una especificación de la estructura por cada nuevo tipo de elemento introducido.

2. Estructura del sistema: El objetivo es identificar los bloques a nivel de sistema. El lenguaje MSCs puede ser introducido en este paso para dar una visión general de la comunicación entre los bloques. Las tareas a cumplir son:
 - a. Identificar los principales componentes conceptuales del sistema y el nombre de ellos. Estos serán los bloques del sistema. Encontrar un nombre adecuado para cada bloque y describir el bloque y sus relaciones con su entorno (la estructura envolvente) de manera informal en un comentario dentro de la especificación del bloque.
 - b. Conectar cada bloque con la frontera del sistema con los canales introducidos en el Paso 1.
 - c. Identificar el flujo de información (canales y las señales) entre los bloques. Especificar las nuevas señales introducidas a nivel del sistema, como en el paso 1.
 - d. Proporcionar una especificación de la estructura por cada nuevo tipo de elemento introducido a nivel del sistema, como en el paso 1.
3. Partición del bloque: Un bloque es como un árbol, cuya raíz es el sistema y sus particiones como las ramas y hojas. Esto se logra dividiendo cada bloque complejo en sub bloques, al igual que en el paso 2. y se repite hasta que no haya bloques complejos.
4. Componentes de los bloques: En esta etapa se pretende identificar los tipos de procesos dentro de cada bloque sin dividir. Lo que se realiza es:
 - a. Identificar las actividades dentro de cada bloque sin particiones. Estas actividades serán los procesos del bloque. Encontrar un nombre adecuado para cada proceso y describir la función, características y su relación con su entorno (el asociado al bloque) de manera informal en un comentario dentro de la especificación del proceso.
 - b. Conectar, si corresponde, el proceso a los canales en el límite del bloque con rutas de señal. Identificar el flujo de la información (rutas de señales y las señales) entre los tipos de proceso. Especificar las nuevas señales introducidas, como en el paso 1.
 - c. Proporcionar una especificación de la estructura para cada nuevo tipo de elemento introducido, como en el paso 1.
5. Especificación de la estructura del proceso: En este paso también puede utilizarse el lenguaje MSCs. Para ello, lo que debe hacer es:
 - a. Identificar casos de uso típico y describirlos.

- b. Tomar las decisiones adicionales que sean necesarias relativas a cómo modelar el comportamiento. Esto puede requerir la introducción de nuevas señales y ordenamientos, según lo especificado en el paso 1.
 - c. Escribir una especificación de la estructura del proceso que abarque los casos de uso típico, pero que no consideren la combinación de estos todavía.
 - d. Considerar el uso de procedimientos para ocultar los detalles y temporizadores para la supervisión del tiempo. Introducir sinónimos externos para los valores sin especificar.
6. Especificación informal del proceso: Las especificaciones informales del proceso enriquecen las especificaciones de la estructura del proceso del paso 5. Para ello debe:
- a. Tener en cuenta la combinación de los casos típicos que se suelen utilizar y describirlos.
 - b. Identificar la información que necesita ser almacenada en los procesos. Introducir tareas y decisiones usando sólo texto informal en esta etapa. Introducir cualquier nuevo estado que se requiera.
 - c. Escribir una estructura de especificación para cada procedimiento e indicar el tipo de parámetros del procedimiento.
 - d. Indicar el tipo de parámetros formales del proceso. Especificar cada nuevo tipo introducido, como en el paso 1.
7. Completar la especificación del proceso: En este paso se revisan la especificación del proceso, aunque aún de manera informal. Debe considerar los casos excepcionales como posibles situaciones de error. Completar la especificación del procedimiento según corresponda. Comprobar que todas las combinaciones de señales de estado están cubiertas.
8. Especificación formal del proceso: Se deben estudiar las especificaciones de tipo semi formal y formal. Las tareas a realizar son:
- a. Identificar tipo de información almacenada. Especificar la clase y usar un texto informal para formar las ecuaciones que sean necesarias.
 - b. Especificar las variables para la información almacenada y los parámetros de salida. Especificar formalmente los parámetros del proceso.
 - c. Cambiar el texto informal en tareas, decisiones y respuestas a las asignaciones y expresiones.

- d. Agregar parámetros de entradas, salidas, creaciones y llamadas a procedimientos.
9. Especificaciones de tipo formal: El resultado en este paso debe ser la especificación completa y formal del proceso. Para ello debe formalizar las ecuaciones construidas (para operación, asignación u otras tareas), mediante la sustitución de las ecuaciones informales por ecuaciones formales. Además debe completar el tipo de especificaciones con ecuaciones.

I.2.- Validación y Verificación

I.2.1.- Validación

La validación consiste en determinar si el modelo teórico es correcto. Esto implica:

- Evaluar si la especificación satisface las necesidades de los usuarios y el medio ambiente de los sistemas. Debe haber una correcta secuencia de las señales, limitaciones de tiempo, estados invariantes, etc.
- Realizar una simulación:
 - Investigar el comportamiento del sistema mediante el envío de señales al sistema y la observación de las respuestas.
 - Chequear si el comportamiento esperado está incluido en la especificación realizada.
- Validar las especificaciones SDL en contra de MSCs (en caso de utilizar ambos lenguajes).

I.2.2.- Verificación

La verificación consiste en determinar si la implementación informática es correcta. Esto implica:

- Determinar los comportamientos incorrectos (o fuera de lo esperado) se debe evitar en todo tipo de especificaciones:
- Bloqueos mutuos, conocido como Interbloqueo o deadlock (es decir, dos procesos están en espera de señales entre sí. Cada proceso está esperando una señal del otro proceso).
- Comportamientos inesperados: por ejemplo, combinaciones poco frecuentes de las señales y tiempo de espera, livelock (lo que implica que un proceso específico no está procesando), etc.
- Revisar también algunas propiedades específicas del sistema definidas por el diseñador del sistema en SDL.

- Utilizar técnicas para reducir Estados-Espacios. Esto permite explorar problemas con los sistemas grandes y complejos (la explosión de estados). Intentar con métodos heurísticos, exploración parcial, etc.

J.- Las Herramientas de SDL y MSC.

SDL cuenta con una serie de herramientas que se han desarrollado durante los últimos 10 años y que son altamente convenientes para el uso efectivo de SDL y MSC. Algunas de estas herramientas y el desglose de sus características son:

- Editor gráfico (Graphical Editor):
 - Fue el primero en surgir.
 - Su principal objetivo es manejar los diagramas SDL, cuya creación y actualización requiere un gran esfuerzo.
- Analizador estático (Static analyzer):
 - Revisa la sintaxis y la exactitud estática usando un texto formal.
 - La conversión desde la representación gráfica a la textual es normalmente parte del analizador estático.
- Generador de documentos (Document Generator):
 - Las especificaciones SDL son sólo un subconjunto de la documentación completa de un sistema.
 - Convierte las especificaciones SDL en documentos de acuerdo a las normas internas de la organización.
- Conversor de Sintaxis (Syntax Converter):
 - El convertidor de sintaxis transforma la sintaxis gráfica por una representación textual.
 - Esta representación textual es usada como un formato estándar para otros lenguajes o herramientas.
 - La mayoría de los fabricantes de herramientas de diseño de sistemas no ofrecen esta función.
- Simulador (Simulator):
 - Realiza una ejecución interactiva.
 - Simula una especificación SDL con el fin de obtener una mejor comprensión de las propiedades dinámicas del sistema.
- Analizador Dinámico (Dynamic Analyzer / Validation):
 - Comprueba si algunas de las propiedades dinámicas de un determinado sistema de SDL se cumplen.
- Generador de código (Code Generator):

- Su principal énfasis es la producción de instancias del sistema.
- Es un pre-requisito para el uso industrial de SDL.
- Produce programas en el lenguaje de programación para una especificación SDL.

- Generación de Casos de Prueba (Test Cases Generation):
 - Genera casos de prueba para evaluar la validez de las especificaciones SDL hechas en el sistema

Otras herramientas que también se puede aplicar son ObjectGEODE (VERILOG SA), TAU (Telelogic AB), Cinderella (CINDERELLA SOFTWARE, 2007), entre otras.

BIBLIOGRAFÍA

- Adan I and Vissers J** Patient mix optimisation in hospital admission planning: a case study [Journal] // International Journal of Operations & Production Management. - 2002. - 4 : Vol. 22. - pp. 445-461.
- Ahmed Mohamed A and Alkhamis Talal M** Simulation optimization for an emergency department healthcare unit in Kuwait [Journal] // European Journal of Operational Research. - October 2009. - Vol. 198. - pp. 936-942.
- Asby Ross William** An introduction to Cybernetics [Book]. - London : Chapman & Hall, 1956. - Segunda : pp. 39-41.
- Ashby Marshall [et al.]** Discrete event simulation: optimizing patient flow and redesign in a replacement facility [Conference] // Winter Simulation Conference. - 2008.
- Ballard Sarah M Sarah M and Kuhl Michael E** The use of simulation to determine maximum capacity in the surgical suite operating room [Conference] // Winter Simulation Conference. - 2006.
- Banks Jerry** Handbook of Simulation: Principles, Methodology, Advances, Applications and Practice [Book]. - Georgia : John Wiley & Sons, 1998.
- Baumgart Andre [et al.]** Using Computer Simulation in Operating Room Management: Impacts on Process Engineering and Performance [Conference] // International Conference on System Sciences. - 2007.
- Bernal García Juan, Martínez María Dolores Soledad and Sanchez García Juan** Simulación del funcionamiento de la lista de espera quirúrgica en el Servicio Murciano de Salud [Report] / Departamento de Métodos Cuantitativos e Informáticos ; Universidad Politécnica de Cartagena. - Murcia : [s.n.], 2003.
- Blake John and Donald Joan** Mount sinai hospital uses integer programming to allocate operating room time [Journal] // Interfaces. - March-April 2002. - 2 : Vol. 32. - pp. 63-73.
- Blake John, Dexter Franklin and Donald Joan** "Operating room manager's use of integer programming for assigning block time to surgical groups: A case study [Journal] // Anesthesia and Analgesia. - 2002. - Vol. 94. - pp. 143-148.
- Bozga Marius [et al.]** Timed Extensions for SDL [Journal] // Proceedings of SDL-Forum'01. - Copenhagen : [s.n.], 2001.
- Brade Dirk** Enhancing modeling and simulation accreditation by structuring verification and validation results [Conference] // Winter Simulation Conference / ed. Joines J. A. [et al.]. - 2000.
- Cardoen Brecht and Demeulemeester Erik** Evaluating the capacity of clinical pathways through discrete-event simulation [Report] / Faculty of Economics and Applied Economics, Department of Decision Sciences and Information Management ; Katholieke Universiteit Leuven,. - Leuven-Belgium : [s.n.], 2007.
- Cardoen Brecht, Demeulemeester Erik and Beliën Jeroen** Operating room planning and scheduling: A literature review [Journal] // European Journal of Operational Research. - March 2010. - Vol. 201. - pp. 921-932.
- Cardoen Brecht, Demeulemeester Erik and Beliën Jeroen** Optimizing a multiple objective surgical case scheduling problem [Report] / Faculty of Economics and Applied Economics, Department of Decision Sciences and Information Management ; Katholieke Universiteit Leuven. - Leuven-Belgium : [s.n.], 2006.
- Cerda Emilio, De pablos Laura and Rodríguez María** La gestión de las listas de espera quirúrgica en España [Report] / Ministry of Economy and Finance Institute of Fiscal Studies. - Spain : [s.n.], 2001.
- CINDERELLA SOFTWARE** Cinderella SDL [Online]. - 2007. - 03 31, 2009. - <http://www.cinderella.dk>.
- Corporation, Palisade** Guía para el uso de @RISK [Online] // Bestfit 5.5. - February 2010. - http://www.palisade.com/downloads/manuals/ES/RISK5_ES.pdf.
- Daiki Min and Yuehwern Yih** An elective surgery scheduling problem considering patient priority [Journal] // Computers & Operations Research. - June 2010. - 6 : Vol. 37. - pp. 1091-1099.
- De la Fuente Pablo** Universidad de Valladolid [Online] // Laboratorios Docentes. - 12 2010, 01. - http://jair.lab.fi.uva.es/~pablfue/leng_simulacion/materiales/v_v_0405.pdf.
- Denton Brian [et al.]** Optimal allocation of surgery blocks to operating rooms under uncertainty [Journal] // Operations Research. - July-August 2010. - 4 : Vol. 58. - pp. 802-816.
- Denton Brian [et al.]** Simulation of a multiple operating room surgical suite [Conference] // Winter Simulation Conference. - 2006.
- Dexter Franklin [et al.]** Statistical method to evaluate management strategies to decrease variability in operating room utilization: application of linear statistical modeling and Monte Carlo simulation to operating room management [Journal] // Anesthesiology. - July 1999. - 1 : Vol. 91. - pp. 262-274.
- Dexter Franklin [et al.]** Use of linear programming to estimate impact of changes in a hospital's operating room time allocation on perioperative variable costs [Journal] // Anesthesiology. - March 2002. - 3 : Vol. 96. - pp. 718-724.

Dexter Franklin and Tinker John Analysis of strategies to decrease postanesthesia care unit costs [Journal] // *Anesthesiology*. - 1995. - 1 : Vol. 82. - pp. 94-101.

Dexter Franklin and Traub Rodney How to schedule elective surgical cases into specific operating rooms to maximize the efficiency of use of operating room time [Journal] // *Anesthesia and Analgesia*. - 2002. - Vol. 94. - pp. 933-942.

Dexter Franklin Operating room utilization: Information management systems [Journal] // *Anaesthesiology*. - December 2003. - Vol. 16. - pp. 619-622.

Dexter Franklin, Macario Alex and O'Neill Liam Scheduling surgical cases into overflow block time - Computer simulation of the effects of scheduling strategies on operating room labor costs [Journal] // *Anesthesia and Analgesia*. - 2000. - Vol. 90. - pp. 980-988.

Dexter Franklin, Macario Alex and Traub Rodney An operating room scheduling strategy to maximize the use of operating room block time: Computer simulation of patient scheduling and survey of patients' preferences for surgical waiting time [Journal] // *Anesthesia and Analgesia*. - 1999. - Vol. 89. - pp. 7-20.

Dexter Franklin, Macario Alex and Traub Rodney Enterprise-wide patient scheduling information systems to coordinate surgical clinic and operating room scheduling can impair operating room efficiency [Journal] // *Anesthesia and Analgesia*. - 2000. - Vol. 91. - pp. 617-626.

Dexter Franklin, Macario Alex and Traub Rodney How to release allocated operating room time to increase efficiency: Predicting which surgical service will have the most underutilized operating room time [Journal] // *Anesthesia and Analgesia*. - 2003. - Vol. 96. - pp. 507-512.

Dexter Franklin, Macario Alex and Traub Rodney Operating room utilization alone is not an accurate metric for the allocation of operating room block time to individual surgeons with low caseloads [Journal] // *Anesthesiology*. - 2003. - 5 : Vol. 98. - pp. 1243-1249.

Dexter Franklin, Macario Alex and Traub Rodney Optimal sequencing of urgent surgical cases [Journal] // *Journal of Clinical Monitoring and Computing*. - May 1999. - 3-4 : Vol. 15. - pp. 153-162.

Dexter Franklin, Macario Alex and Traub Rodney Which Algorithm for Scheduling Add-on Elective Cases Maximizes Operating Room Utilization? [Journal] // *Anesthesiology*. - 1999. - 5 : Vol. 91. - pp. 1491-1500.

Dexter Franklin, Willemsen-Dunlap Ann and Lee John Operating room managerial decision-making on the day of surgery with and without computer recommendations and status displays [Journal] // *Anesthesia and Analgesia*. - 2007. - Vol. 105. - pp. 419-429.

Doldi Lauren Validation of Communications Systems with SDL: The Art of SDL Simulation and Reachability Analysis [Book]. - [s.l.] : John Wiley & Sons, Inc., 2003. - p. 310. - ISBN: 978-0-470-85286-6.

El Barachi May, Khendek Ferhat and Dssouli Rachida Introduction to SDL and MSCs [Online]. - July 2010. - http://users.ens.concordia.ca/~elbar_m/invitedTalks/Introduction%20to%20SDL%20and%20MSCs.pdf.

Erdogan Ayca and Denton Brian Industrial & Systems Engineering [Online] = Surgery Planning and Scheduling: A Literature Review // North Carolina State University. - November 1, 2010. - <http://www.ise.ncsu.edu/bdenton/Papers/pdf/Erdogan2009.pdf>.

Ernest Edward [et al.] Operating room scheduling by computer [Journal] // *Anesthesia and Analgesia*. - 1977. - Vol. 56. - pp. 831-835.

Esteban Sastre David Introducción a SDL [Online] // Unidad docente de Sistemas, Facultad de Informática. Universidad Politécnica de Madrid. - December 2010. - http://pegaso.ls.fi.upm.es/~lmengual/telelogic/INTRODUCCION_SDL.pdf.

Fei Hongying, Meskenst Nadine and Chu Chengbin An operating theatre planning and scheduling problem in the case of a block scheduling strategy [Conference] // International Conference on Service Systems and Service Management. - 2006.

Fernandez de Cañete Javier Introducción al Modelado de sistemas [Online] // Departamento de Ingeniería de Sistemas y Automática. - 2006. - December 2010. - [http://www.isa.uma.es/C17/Presentaciones%20de%20Clase%20\(ppt\)/Document%20Library/INTRODUCCION%20AL%20MODELADO%20DE%20SISTEMAS.pdf](http://www.isa.uma.es/C17/Presentaciones%20de%20Clase%20(ppt)/Document%20Library/INTRODUCCION%20AL%20MODELADO%20DE%20SISTEMAS.pdf).

Fonseca i Casas Pau SDL distributed simulator [Conference] // Winter Simulation Conference 2008. - Miami : INFORMS, 2008. - <http://wintersim.org/abstracts08/POS.htm#fonsecaicasasp84590>.

Fonseca i Casas Pau SDL, a graphical language useful to describe social simulation models [Conference] // 2nd Workshop on Social Simulation and Artificial Societies Analysis (SSASA'08). - 2008.

Fonseca i Casas Pau Using Specification and Description Language to define and implement discrete simulation models [Journal] // The 2010 Summer Computer Simulation Conference. - Ottawa : [s.n.], 2010.

Galarza Jorge, Pessana Franco and Armentano Ricardo Razonamiento Cualitativo y Simulación Cualitativa [Conference] // XV Congreso Argentino de Bioingeniería: 114MS. - 2005.

Grote R [et al.] Operation room management: From degree of utilization to distribution of capacities [Journal] // *Anaesthesist*. - September 2008. - 9 : Vol. 57. - pp. 882-892.

Guasch Antoni [et al.] Modelado y simulación. Aplicación a procesos logísticos de fabricación y servicios. [Book]. - Barcelona : UPC, 2002. - p. 15.

Guerriero Francesca and Guido Rosita Operational research in the management of the operating theatre: a survey [Journal] // *Health Care Management Science*. - November 2010. - Vols. DOI 10.1007/s10729-010-9143-6.

Hogrefe Belina SDL-88 Tutorial [Online] // [sdl-forum.org](http://www.sdl-forum.org). - June 2010. - <http://www.sdl-forum.org/sdl88tutorial/index.html>.

IBM TELELOGIC [Online]. - 2009. - 03 31, 2009. - <http://www.telelogic.com/>.

International Telecommunication Union Series Z.100: Languages and general software, Aspects for telecommunication systems. Specification and Description Language (SDL) [Report]. - 2002.

Jansson Axel and Delgado Carolina Optimización del Proceso de cirugía en hospitales públicos, una aplicación de la modelación matemática entera en la prestación de atención quirúrgica en el Hospital del salvador, Santiago de Chile [Journal] // *Pharos Ciencia Arte y Tecnología*. - May-June 2000. - Vol. 7. - pp. 31-62.

Jeang Angus and Chiang An-Jen Economic and Quality Scheduling for Effective Utilization of Operating Rooms [Journal] // *Journal of medical systems*. - 2010. - Vol. 34. - f.

Jebali Aïda, Hadj Alouaneb Atidel and Ladeta Pierre Operating rooms scheduling [Journal] // *International Journal of Production Economics*. - January-February 2006. - Vol. 99. - pp. 52-62.

Jones Alfred, Sahney Vinod and Kurtoglu Ali A discrete event simulation for the management of surgical suite scheduling [Conference] // 16th Annual Symposium on Simulation. - 1983.

Kahrs L [et al.] Planning and simulation of microsurgical laser bone ablation [Journal] // *International Journal of Computer Assisted Radiology and Surgery*. - March 2010. - 2 : Vol. 5. - pp. 155-162.

Kuo Paul [et al.] Optimization of operating room allocation using linear programming techniques [Journal] // *Journal of the American College of Surgeons*. - 2003. - 6 : Vol. 197. - pp. 889-895.

Kusters Rob and Groot Petra Modelling resource availability in general hospitals – Design and implementation of a decision support model [Journal] // *European Journal of Operational Research*. - 1996. - Vol. 88. - pp. 428-445.

Lamiri Mehdi [et al.] A stochastic model for operating room planning with elective and emergency demand for surgery [Journal] // *European Journal of the Operational Research*. - 2008. - 3 : Vol. 185. - pp. 1926-1037.

Lamiri Mehdi, Xie Xiaolan and Grimaud Frédéric Optimization methods for a stochastic surgery planning problem [Journal] // *International Journal of Production Economics*. - August 2009. - 2 : Vol. 120. - pp. 400-410.

Lamiri Mehdi, Xie Xiaolan and Zhanga Shuguang Column generation for operating theatre planning with elective and emergency patients [Journal] // *IIE Transactions*. - September 2008. - 9 : Vol. 40. - pp. 838 - 852 .

Law Averill and Kelton David Simulation Modeling and Analysis [Book]. - New York : McGraw Hill, 2001. - Tercera : p. 4.

Lemke Heinz and Michael Vannier The operating room and the need for an IT infrastructure and standards [Journal] // *International Journal of Computer Assisted Radiology and Surgery*. - 2006. - 3 : Vol. 1. - pp. 117-121.

Lowery Julie and Davis Jennifer Determination of operating room requirements using simulation [Conference] // Winter Simulation Conference. - 2009.

Lucas CE [et al.] Mathematical modeling to define optimum operating room staffing needs for trauma centers [Journal] // *Journal of the American College Surgeons*. - May 2001. - 5 : Vol. 192. - pp. 559-565.

Marcon Eric [et al.] Determining the number of beds in the postanesthesia care unit: A computer simulation flow approach [Journal] // *Anesthesia and Analgesia*. - 2003. - Vol. 96. - pp. 1415-1423.

Marcon Eric and Kharraja Saïd Models and strategies of operative programming [Journal] // *Journal Europeen des Systemes Automatisés*. - 2003. - Vol. 37. - pp. 687-716.

Ministerio de Salud de Chile Compromisos de Gestión [Report]. - Viña del Mar : [s.n.], 2009. - p. 7.

Ministerio de Salud de Chile Cuenta Pública [Report] / Hospital Dr. Gustavo Fricke. - Viña del Mar : [s.n.], 2009.

Ministerio de Salud de Chile Producción Anual [Report] / Hospital Dr. Gustavo Fricke. - Viña del Mar : [s.n.], 2009.

Ministerio de Salud de Chile Registro y gestión de información de espera de atenciones de salud [Report] / Division of Management Assistance Network ; Undersecretary of Assistance Networks. - 2009.

Murray Calichman Creating an optimal operating room schedule [Journal] // *AORN Journal*. - 2005. - Vol. 81. - pp. 580-588.

Niu Qing [et al.] Performance analysis of the operating room using simulation [Conference] // The Canadian Design Engineering Network (CDEN) and the Canadian Congress on Engineering Education (CCEE). - Winnipeg : [s.n.], 2007. - Vol. 38.

Panayiotopoulos JC and Vassilacopoulos G Simulating hospital emergency departments queuing systems [Journal] // European Journal of the Operational Research. - 1984. - Vol. 18. - pp. 250-258.

Persson Marie and Persson Jan Analysing management policies for operating room planning using simulation [Journal] // Health Care Management Science. - December 2009. - Vol. 13. - pp. 182-191.

Persson Marie and Persson Jan Health economic modelling to support surgery management at a Swedish hospital [Journal] // Omega. - August 2009. - 4 : Vol. 37.

Persson Marie and Persson Jan Optimization modelling of hospital operating room planning: analyzing strategies and problem settings [Conference] // EURO Working Group on OR Applied to Health Services 31 Annual Meeting. - Southampton : [s.n.], 2005.

Reed Rick International Telecommunication Union [Online] = SDL-2000 for New Millennium Systems. - 01 13, 2010. - <http://www.itu.int/itudoc/itu-t/com17/tutorial/78255.html>.

Reindl Sonja [et al.] Modeling and simulation of cataract surgery processes [Conference] // Winter Simulation Conference. - Austin : [s.n.], 2009.

Rico Rocío and Martínez Saraid Gestipolis.com [Online] // El sistema de actividad humana como un lenguaje de modelación. - June 2008. - July 1, 2010. - <http://www.gestipolis.com/administracion-estrategia/lenguaje-de-modelacion-en-sistemas.htm#mas-autor>.

Rosen J [et al.] Markov modeling of minimally invasive surgery based on tool/tissue interaction and force/torque signatures for evaluating surgical skills [Journal] // IEEE Trans Biomed Eng. - May 2001. - 5 : Vol. 48. - pp. 579-591.

Ryu Jun-Hyung Modeling supply chain operations as multi-level programming problems and their parametric programming based computation methodology [Journal] // Korean Journal of Chemical Engineering. - September 2010. - 6 : Vol. 27. - pp. 1681-1688.

Sandrila Ltd Sandrila [Online] = Example diagrams-Sandrila SDL. - December 2010. - <http://www.sandrila.co.uk/visio-sdl/index.php>.

Sargent Robert G. Verification and validation of simulation models [Conference] // Proceedings of the 2007 Winter Simulation Conference / ed. Henderson S. G. [et al.]. - Washinton : IEEE, 2007.

Schlesinger S Terminology for Model Credibility [Journal] // Simulation. - 1979. - 3 : Vol. 132. - pp. 103-104.

Schmidt J.W and Taylor R.E Simulation and analysis of industrial systems [Book]. - [s.l.] : Irwind Richard, 1970.

Schmitz Home and Kwak N The Monte Carlo simulation of operating-room and recovery-room usage [Journal] // Operations Research. - 1972. - 6 : Vol. 20. - pp. 1171-1180.

Sciomachen Anna, Tanfani Elena and Testi Angela Simulation models for optimal schedules of operating theatres [Journal] // International Journal of Simulation. - 2005. - 12-13 : Vol. 6. - pp. 26-34.

Sepúlveda Juan Amélioration de l'agenda des patients ambulatoires dans un hôpital public à Santiago du Chili [Conference] // Conference GISEH08. - Lausanne : [s.n.], 2008.

Shaw B and Marshall AH Modelling the flow of congestive heart failure patients through a hospital system [Journal] // Journal of the Operational Research Society. - 2007. - Vol. 58. - pp. 212-218.

Sibbel Rainer and Urban Christoph Agent-Based Modeling and Simulation for Hospital Management [Report]. - [s.l.] : Kluwer Academic Publishers, 2001.

Steelman Victoria Mathematical Modeling To Define Optimum Operating Room Staffing Needs For Trauma Centers [Journal] // Journal of the American College of Surgeons. - May 2001. - Vol. 192. - pp. 559-565.

Steins Krisjanis and Persson Fredrik Increasing Utilization in a Hospital Operating Department Using Simulation Modeling [Journal] // Simulation. - August 2010. - 8-9 : Vol. 86. - pp. 463-480.

Strum David, Vargas Luis and ay Jerrold Surgical subspecialty block utilization and capacity planning: A minimal cost analysis model [Journal] // Anesthesiology. - 1999. - Vol. 35. - pp. 1176-1185.

Su Ying and Shen Ningqiao "Modeling the Effects of Information Quality on Process Performance in Operating Room [Conference] // 12th International Conference on Computer Modelling and Simulation. - Cambridge : [s.n.], 2010.

Suzuki Naoki and Asaki Hattori The road to surgical simulation and surgical navigation [Journal] // Journal Virtual Reality. - December 2008. - 4 : Vol. 12. - pp. 281-291.

Tan Yin Yin [et al.] Mathematical programming for the scheduling of elective patients in the operating room department [Conference] // The Canadian Design Engineering Network (CDEN) and the Canadian Congress on Engineering Education (CCEE). - Winnipeg : [s.n.], 2007.

Technologies MathWave Distribution Fitting Tutorials [Online] // EasyFit 5.4. - June 2010. - http://www.mathwave.com/articles/distribution_fitting.html.

Telecommunication standardization sector of ITU Specification and Description Language (SDL) [Online] = ITU-T Z.100. 1999 // Series Z: Languages and general software aspects for telecommunication systems.. -

International Telecommunication Union, 1999. - April 2008. - <http://www.itu.int/ITU-T/studygroups/com17/languages/index.html>.

Telelogic AB Introduction to Telelogic Tau SDL Suite [Online]. - 06 2011. - <http://www.eit.lth.se/fileadmin/eit/courses/ets150/IntroSDLSuite.pdf>.

The International Engineering Consortium Web ProForum Tutorials [Online] // www.iec.org. - January 2010. - http://www.aws.cit.ie/personnel/dpesch/notes/msc_sw/sdl.pdf.

Tucker James [et al.] Using queueing theory to determine operating room staffing needs [Journal] // Journal of Trauma-Injury Infection & Critical Care. - 1999. - 1 : Vol. 46. - pp. 71-79.

Tyler Donald, Pasquariello Caroline and Chen Chun-Hung Determining optimum operating room utilization". [Journal] // Anesthesia and Analgesia. - 2003. - Vol. 96. - pp. 1114-1121.

Van Oostrum JM [et al.] A simulation model for determining the optimal size of emergency teams on call in the operating room at night [Journal] // Anesthesia and analgesia. - November 2008. - 5 : Vol. 107. - pp. 1655-1662.

VanBerkel Peter and Blake John A comprehensive simulation for wait time reduction and capacity planning applied in general surgery [Journal] // Health Care Management Science. - 2007. - Vol. 10. - pp. 373-385.

Velasquez R and Melo M A set packing approach for scheduling elective surgical procedures [Journal] // Operations Research Proceedings. - 2006. - Vol. 2005. - pp. 425-430.

VERILOG SA ObjectGEODE SDL Process to UML State Chart Translator User's Guide [Online]. - 06 2011. - http://www.mikro.fh-konstanz.de/inhalte/informationsquellen/online_manuals/ogpdf/sdl_uml.pdf.

Wright Ian [et al.] Statistical modeling to predict elective surgery time. Comparison with a computer scheduling system and surgeon-provided estimates [Journal] // Anesthesiology. - December 1996. - 6 : Vol. 85. - pp. 1235-1245.

Zeigler Bernard Theory of modeling and simulation [Book]. - Ney York : Wiley & Sons, 1976.

Zhang B [et al.] A mixed integer programming approach for allocating operating room capacity [Journal] // Journal of the Operational Research Society. - 2009. - 5 : Vol. 60. - pp. 663-673.

Zheng Qian [et al.] Simulation Modeling of the Operating Room Based on SIMIO [Journal] // Applied Mechanics and Materials. - 2010. - Vols. 37-38. - pp. 1162-1166.