XX(112568.1)

CINVESTAV

Centro de Investigación y de Estudios Avanzados del I.P.N.
Unidad Guadalajara

Modelado Multinivel de Sistemas de Procesos por Lotes

Tesis que presenta:
Norma Isabel Villanueva Paredes

para obtener el grado de:
Maestro en Ciencias

en la especialidad de:
Ingeniería Eléctrica

Director de Tesis
Dr. Luis Ernesto López Mellado

Guadalajara, Jal., Diciembre de 2003.

Modelado Multinivel de Sistemas de Procesos por Lotes

Tesis de Maestría en Ciencias
Ingeniería Eléctrica

Por:
Norma Isabel Villanueva Paredes

Licenciada en Informática
Instituto Tecnológico de Ciudad Guzmán 1994-1999

Becario del CONACyT, expediente no. **169864**

Director de Tesis
Dr. Luis Ernesto López Mellado

CINVESTAV del IPN Unidad Guadalajara, Diciembre de 2003.

Agradecimientos

A Dios por brindarme la oportunidad de superarme y dar un pequeño fruto.

A dos seres muy importantes en mi vida, Norma Angélica y Alfredo por comprender y apoyar mis decisiones en todo momento, y por el enorme cariño que les tengo por ser mis padres.

A mis hermanos por su insistente motivación.

Al Dr. Ernesto López Mellado quien dirigió esta tesis, por su paciencia, sus enseñanzas y sobre todo por su gran calidad humana.

A mi familia por tenderme la mano en momentos difíciles especialmente a mi prima Sandra Torres Paredes.

Y como olvidar a mis amigos y compañeros por su ayuda y por los ratos de convivencia.

A CONACYT por su apoyo económico.

A todos ellos les dedico este esfuerzo que se hace tangible en este trabajo.

Índice general

Introducción iv

1. Sistemas de manufactura por lotes 3
1.1. Sistemas de producción 4
1.1.1. Clasificación de sistemas de producción 4
1.1.2. Características de los diferentes tipos de procesos 5
1.1.3. Sistemas de manufactura flexible (FMS) 6
1.2. Procesos por lotes 6
1.2.1. Definiciones 7
1.2.2. Modelos de referencia 8
1.2.3. Clasificación de celdas de proceso 12
1.2.4. Recetas 14
1.3. Modelado de sistemas de manufactura por lotes 17
1.3.1. Las redes de Petri como herramienta de modelado para procesos por lotes 17
1.3.2. Modelado de componentes 19
1.4. Conclusión 23

2. Un formalismo de red multinivel 25
2.1. Motivación 26
2.2. Presentación intuitiva 26
2.3. Definición formal de n-LNS 29
2.3.1. Redes tipo 29
2.3.2. Redes de nivel i 33
2.3.3. Sistema n-LNS 34
2.3.4. Evolución del marcado en n-LNS 34
2.4. Conclusión 38

3. Metodología de modelado multinivel de procesos por lotes 39
3.1. Introducción 40
3.2. Estrategia general de modelado 40
3.2.1. Estructuración del sistema 40
3.2.2. Interacción entre componentes 41
3.3. Metodología de modelado 43
3.4. Propiedades de los módulos 47
3.5. Conclusión 48

4. Modelos de sistemas de manufactura por lotes **49**
4.1. Caso 1: Modelado con 3-LNS 50
4.1.1. Descripción del sistema 50
4.1.2. Descripción del proceso 52
4.1.3. Modelo del sistema en n-LNS 54
4.2. Caso 2: Modelando separación y mezcla de lotes con n-LNS 60
4.2.1. Descripción del proceso 60
4.2.2. Modelado del sistema en n-LNS 63
4.2.3. Conclusión 69

5. Simulación de modelos multi-nivel **71**
5.1. Introducción 72
5.2. Características de Renew 72
5.3. Edición de modelos 73
5.3.1. Modelo del ambiente 73
5.3.2. Modelo del agente 75
5.3.3. Modelo del plan de procesos 76
5.3.4. Modelos de recetas 78
5.3.5. Modelos de recursos 81
5.4. Simulación de modelos 82
5.4.1. Modelo de la planta 83
5.4.2. Modelo del lote 84
5.4.3. Modelo del plan de procesos 85
5.4.4. Modelo de recetas 87
5.4.5. Modelos de recursos 88
5.5. Conclusión 90

6. Conclusiones **93**

Referencias **94**

A. Herramienta de simulación Renew **99**

Introducción

La operación de sistemas de producción requiere de sistemas de coordinación que integren una gran variedad de funciones que incluyan la especificación y configuración del equipo utilizado, especificación de los productos y su procesamiento, la calendarización de las tareas, el control en tiempo real, el monitoreo o supervisión de la ejecución de las operaciones, y la detección de fallas en la realización de las tareas y su eventual recuperación. Las técnicas de diseño y de implementación utilizadas varían según la clase de sistemas de producción de que se trate, los cuales pueden ser clasificados como discretos, continuos, e híbridos.

Sistemas de producción por lotes

El presente trabajo trata con los sistemas de producción híbridos, en particular los llamados sistemas de procesos por lotes. Un sistema de procesos por lotes opera con materiales de cantidad finita, donde un número entero de lotes pueden estar en operación en localidades distintas dentro de la misma planta, lo que permite visualizar el comportamiento discreto del sistema; por ejemplo, un lote de líquido en un reactor puede ser considerado como una parte. También, un lote implica el manejo de números reales los cuales representan la cantidad de material y sus condiciones de operación (temperatura, presión, etc). Durante el transporte de material entre dos equipos, la naturaleza discreta del lote desaparece temporalmente y la naturaleza continua se manifiesta. Es por eso que un proceso por lotes opera como un sistema híbrido, es decir, un sistema discontinuo. El producto resultante de un proceso por lotes es llamado "lote".

Estos sistemas comúnmente son plantas de manufactura multi-propósito de la industria química, alimenticia, farmacéutica, etc.

Especificación de procesos

El desarrollo de sistemas de coordinación para el manejo de procesos complejos requiere en su etapa más temprana, de la especificación o modelado formal de las actividades a llevar a cabo en el sistema de producción. Los modelos permiten por un lado, la minimización de las ambigüedades en la descripción de los procesos y por otro lado, el análisis de los modelos detectando posibles problemas en la futura operación del sistema.

Uno de los formalismos de modelado más utilizados es las redes de Petri (RP). Las RP son una familia de formalismos que proveen un marco de trabajo extenso para la descripción, modelado, análisis, implementación y control de sistemas de producción. Su naturaleza gráfica y su simple soporte matemático, ayudan a representar comportamiento complejo de manera clara y compacta, especificando causalidad, concurrencia, paralelismo, sincronización, toma de decisiones e intercambio de información.

Desde hace más de veinte años las RP han sido utilizadas en la especificación de sistemas de control/coordinación de sistemas de manufactura [9]. En particular en la década pasada se han aplicado en el modelado, análisis y operación de procesos por lotes [1],[2],[4],[5],[6],[7], [8], [16],[17], [18], [25], [28],[29], [30].

Redes de Petri de alto nivel

Algunos de los problemas que aparecen al modelar esta clase de sistemas de producción se debe en gran medida a su tamaño y complejidad, lo cual ha motivado la creación de RP de alto nivel que permiten la obtención de modelos más compactos. Primero las redes Predicado/Transición propuestas por H. Genrich [11] y RP coloreadas propuestas por K. Jensen [15] dan una identidad a las marcas (colores) manipulando marcado simbólico. Más tarde en la implementación de un modelo de RP coloreadas reportado por E. Kasturia [17]se señala que los colores representan estructuras de datos.

En la última década, el enfoque de modelado de alto nivel aparece en especificaciones de programas orientado a objetos: conceptos de RP de alto nivel y programación orientada a objetos son mezclados, conduciendo a OOPN [22] y Cooperative Objects [26]. En estos trabajos las marcas pueden ser otras RP. Estos métodos/formalismos aproximan las especificaciones a la implementación de software perdiendo claridad en la descripción.

Recientemente, la idea de considerar RP como marcas es retomada por R. Valk [31] quien propone un formalismo de RP a dos niveles "limpio de código", independiente de lenguajes de programación. La misma noción de redes dentro de redes, es sostenida por K. Hiraishi [12] quien propone PN^2, un formalismo a dos niveles similar a la definición de Valk; también I. Lomazova [23] propone RP anidadas; En tanto, que O. Kummer [20] propone redes referencia como un soporte para una herramienta de simulación en la cual las marcas son referencias a otras redes. La propuesta de Valk es extendida por H. Almeyda [3] definiendo NS-3, un formalismo a tres niveles para modelar agentes móviles donde el nivel más bajo es similar a RP coloreadas.

Enfoque

Esta tesis se centra en el modelado de sistemas de producción por lotes utilizando RP de alto nivel. El objetivo planteado ha sido el de extender NS-3 proponiendo una definición a n niveles y de obtener una metodología para la obtención de modelos de sistemas de procesos por lotes utilizando la nueva definición.

Así, como extensión a NS-3 se propone n-LNS, un formalismo de red a n niveles que permite la especificación de sistemas de eventos discretos con entidades móviles, las cuales evolucionan en ambientes estructurados; en esta extensión se relajan algunas restricciones incluidas en la definición de NS-3. La metodología de modelado propuesta, la cual está orientada a la obtención de modelos expresados en n-LNS, se adapta a los conceptos definidos el estándar ANSI/ISA-S88.01 de la industria de procesos.

La validación de los modelos obtenidos fue realizada por simulación utilizando la herramienta computacional Renew.

Organización de la tesis

La tesis esta organizada en cinco capítulos. El capítulo 1 introduce conceptos sobre sistemas de producción, definiciones y modelos referencia del estándar ANSI/ISA-S88.01 para control de procesos por lotes, así como una revisión de enfoques para el modelado de sistemas de manufactura usando redes de Petri. En el capítulo 2 se presenta la definición formal del sistema n-LNS (n-Level Net System). El capítulo 3 propone una metodología de modelado para sistemas de procesos por lotes, basada en el formalismo n-LNS. Dos casos de estudio son analizados en el capítulo 4. Ambos tratan con plantas multi-lote de la industria química, la metodología de modelado propuesta en el capítulo 3 se aplica a estos dos casos de estudio. Finalmente, el capítulo 5 describe la simulación del primer caso de estudio usando la herramienta Renew.

Capítulo 1

Sistemas de manufactura por lotes

Resumen: En este capítulo se introducen conceptos básicos sobre sistemas de producción. También, se presentan definiciones y modelos referencia del estándar ANSI/ISA-S88.01, el cual provee la terminología manejada en la industria de procesos por lotes. Además, una breve revisión de enfoques es presentada para el modelado de sistemas por lotes usando redes de Petri. Y por último, una planta sencilla de manufactura por lotes es modelada con esta herramienta.

1.1. Sistemas de producción

El comportamiento de los sistemas de producción en la industria de manufactura o de procesos es extremadamente complejo. Los sistemas de producción desempeñan procesos de transformación de material, desde la materia prima hasta la terminación total o parcial del producto. Los procesos de transformación consumen material, energía, equipo y trabajo. El control de estos procesos implica flujos complejos de información a través de todo el sistema, esto es, desde la planta hasta los niveles más altos de organización.

1.1.1. Clasificación de sistemas de producción

La clasificación de sistemas de producción puede ser hecha usando dos criterios. El primero tiene que ver con las trayectorias del producto a través de la planta. El segundo es inherente a la naturaleza del material que será transformado. Respecto a la arquitectura del sistema de producción se consideran cuatro tipos [28]:

- Líneas de transferencia. Los procesos de producción están diseñados como una secuencia de operaciones que toman cantidades similares de tiempo para completarse. El material es movido sincronizadamente de una estación de trabajo a la siguiente, y cada estación de trabajo desempeña repetidamente la misma tarea. El grado de flexibilidad es bajo, por lo que el principal problema de optimización en líneas de transferencia es el balanceo de carga, es decir, hacer que los tiempos de operación en diferentes estaciones de trabajo sean tan similares como sea posible para reducir tiempos inactivos y para balancear la carga de trabajo de las estaciones de trabajo.

- Líneas de producción. Cuando alguna variabilidad es requerida en las estaciones de trabajo, por ejemplo, cuando algunas son operadas manualmente, las partes se mueven entre ellas asincrónicamente. En este caso el sistema no necesita ser completamente balanceado, y posiblemente pueden ser introducidos almacenes (buffers) para amortiguar las variaciones. La eficiencia de estos sistemas es afectada por el fenómeno "deadlock" y "starvation". También el problema de asignación del buffer llega a ser crucial.

- Flow Shop. Cuando algunos productos pueden ser procesados diferentemente, o aún por pasos en algunas estaciones o seguir rutas alternativas, por ejemplo, cuando se produce una única familia de productos que difiere ligeramente una de la otra, el diseño de la planta refleja claramente el flujo del material, pero no tan estricto como en líneas de transferencia o líneas de producción. El nivel de flexibilidad es medio. El problema de secuenciación trata de minimizar los costos de inventario y producción para encontrar una secuencia de partes apropiada (o lotes), que satisfaga la demanda de producción.

- Job Shop. Cuando la variedad de productos es grande, el diseño de la planta no refleja el flujo del material, porque éste puede diferir de un producto a otro. Para cada producto una ruta de producción es definida describiendo una secuencia de operaciones. En este caso el nivel de flexibilidad es alto. En principio, la gran versatilidad es pagada por una menor eficiencia: la flexibilidad de las estaciones hace que no sean tan eficientes, las máquinas pueden permanecer inactivas una parte significativa del tiempo, el inventario en proceso tiende a incrementarse, la transportación entre estaciones aparece como

un nuevo problema a ser solucionado. La automatización integrada de las estaciones, almacenes, transportación y soporte llega a ser altamente demandada, conduciendo a FMS. Para optimizar el desempeño del sistema mientras satisface las demandas de producción, los problemas de decisión complejos deben ser abordados.

Otra clasificación importante es, considerando la naturaleza del material; así que los procesos de manufactura industrial pueden ser[14],[28].

- Procesos continuos: En un proceso de este tipo el material pasa como flujo continuo a través de un equipo de procesamiento, es decir, el material no es dividido en porciones identificables. Todas las operaciones ocurren al mismo tiempo.
- *Procesos de manufactura de partes discretas*: En un proceso de manufactura de partes discretas, los productos son clasificados en lotes de producción que están basados en materia prima, requerimientos de producción e historias de producción comunes. En este tipo de procesos una cantidad especifica de producto se mueve como una unidad (grupo de partes) entre estaciones de trabajo, y cada parte mantiene una única identidad.
- *Procesos por lotes:* En este tipo de procesos, el material es operado en cantidades finitas, donde un número entero de lotes pueden estar en operación en localidades distintas dentro de la misma planta. Por lo que, se puede visualizar el proceso como un sistema de manufactura discreta, por ejemplo, un lote de líquido en un reactor puede ser considerado como una parte. Pero un lote implica el manejo de números reales que representan la cantidad de material y sus condiciones de operación (temperatura, presión, etc). Durante el transporte de material entre dos equipos, la naturaleza discreta del lote desaparece temporalmente. De esta manera, un proceso por lotes opera como un sistema híbrido, es decir, son procesos discontinuos. En un proceso por lotes el producto resultante es llamado "lote"

1.1.2. Características de los diferentes tipos de procesos

Procesos continuos

- El producto es cuantificado en números reales
- El producto es un fluído
- La producción es en línea y está limitada a unos cuantos productos diferentes
- El tipo de control es básico

Procesos de partes discretas

- El producto es cuantificado en números enteros
- El producto es una colección de entidades (partes)
- Generalmente usado en la industria de ensamblaje

- Los dispositivos de transferencia pueden ser fijos o variables

Procesos por lotes

- El producto es cuantificado en números reales y enteros
- El material puede ser líquido, gaseoso, sólido (polvo)
- Utilizan procesos discontinuos (parte discreta, parte continua)
- Los dispositivos de transferencia son fijos
- Usados en la industria de procesos químicos y alimentos
- Generalmente son plantas multi-propósito
- El tipo de control más común es por procedimiento

1.1.3. Sistemas de manufactura flexible (FMS)

Las plantas de producción son configuradas dependiendo principalmente de la variedad de productos y del volumen de producción. Mientras la uniformidad de productos y la producción en masa requiere alta eficiencia (la cual puede ser alcanzada por una rígida automatización) para alcanzar una economía de escala, la gran variedad y los volúmenes de producción bajos requieren alta flexibilidad, para ser capaces de cambiar el escenario del producto y así reaccionar a los cambios del mercado. La conciliación de flexibilidad y eficiencia, da origen al concepto de *sistema de manufactura flexible.*

Un *sistema de manufactura flexible* (FMS) se ha definido como un grupo de máquinas automatizadas capaz de procesar una variedad de productos a través de rutas de proceso diferentes, generalmente sujetos a un control por computadora. Estos sistemas están diseñados para producir simultáneamente diferentes tipos de productos. La flexibilidad es alcanzada gracias al uso de máquinas multi-operación, recursos de transportación automatizados y sistemas de computación para supervisión y administración de actividades de manufactura [28].

Para el caso de sistemas de procesos por lotes, la flexibilidad puede ser explotada, y esto no sólo por la posibilidad de cambios dinámicos en la asignación de recursos y en las trayectorias de los productos como en el caso de FMS, sino también por la posibilidad de elegir el tamaño de los lotes en un rango continuo de valores. Separar y mezclar lotes son operaciones clásicas que pueden ser aplicadas a los lotes, pero para ello se requiere llevar el balance del material mediante modelos matemáticos que describan correctamente la dinámica de los procesos por lotes.

1.2. Procesos por lotes

En años recientes, las organizaciones han hecho esfuerzos significativos en estandarización de control de procesos por lotes. Dos comités son muy conocidos, NAMUR e ISA-SP88. El comité para control por lotes NAMUR es parte de un comité de estandarización para aseguramiento e ingeniería en control, integrado con miembros de la Industria Química Alemana

(NE33 – versión 19.05. 1992). El comité ISA-SP88, está representado por organizaciones de usuarios y vendedores del Norte de América, Europa y de otras partes del mundo. El comité SP88 de estándares ISA fue establecido en 1988 para desarrollar estándares para sistemas de control por lotes. El objetivo del estándar producido por SP88 es proveer una terminología común y un conjunto consistente de modelos para definir los requerimientos de control para plantas de manufactura de procesos por lotes. El trabajo del estándar ha ido progresando y el comité ha reconocido la importancia de los conceptos orientado a objetos para simplificar y consolidar los modelos y las definiciones.

El estándar ISA-S88 ha sido dividido en dos partes. La primera parte del estándar (ISA-S88.01) habla acerca del control por lotes. Esta parte define los modelos de referencia que son usados en la industria de procesos, y la terminología que ayuda a explicar las relaciones entre estos.

El estándar también sirve como guía en el diseño de un proceso por lotes caracterizándolo y definiéndolo. El estándar provee terminología común pero no inter - operable entre diferentes sistemas. La terminología y los modelos pueden usarse como un medio consistente en la administración de recetas sobre múltiples sitios. El estándar S88 puede ayudar a enfatizar buenas prácticas no sólo para diseño, sino para operación de plantas de manufactura por lotes, y además, establece un modelo universal que ayuda a usuarios y vendedores a usar una terminología común. Esto reduce el potencial de desavenencias en la comunicación, ya que provee un lenguaje y una notación universal.

El estándar ISA-S88.01 define un conjunto de modelos y terminología para:

- Equipo y procesos por lotes
- Conceptos de control por lotes
- Funciones y actividades de control por lotes

1.2.1. Definiciones

Un *proceso por lotes* es definido como un proceso que conduce la producción de cantidades finitas de material (lotes) a un orden definido de acciones de procesamiento usando una o más piezas de equipo [14], [10]. Los procesos por lotes son procesos híbridos.

Un *lote* puede ser definido como:

1. Una entidad que representa la producción de un material en cualquier punto del proceso.
2. El material que será producido o que ha sido producido por una simple ejecución de un proceso por lotes.

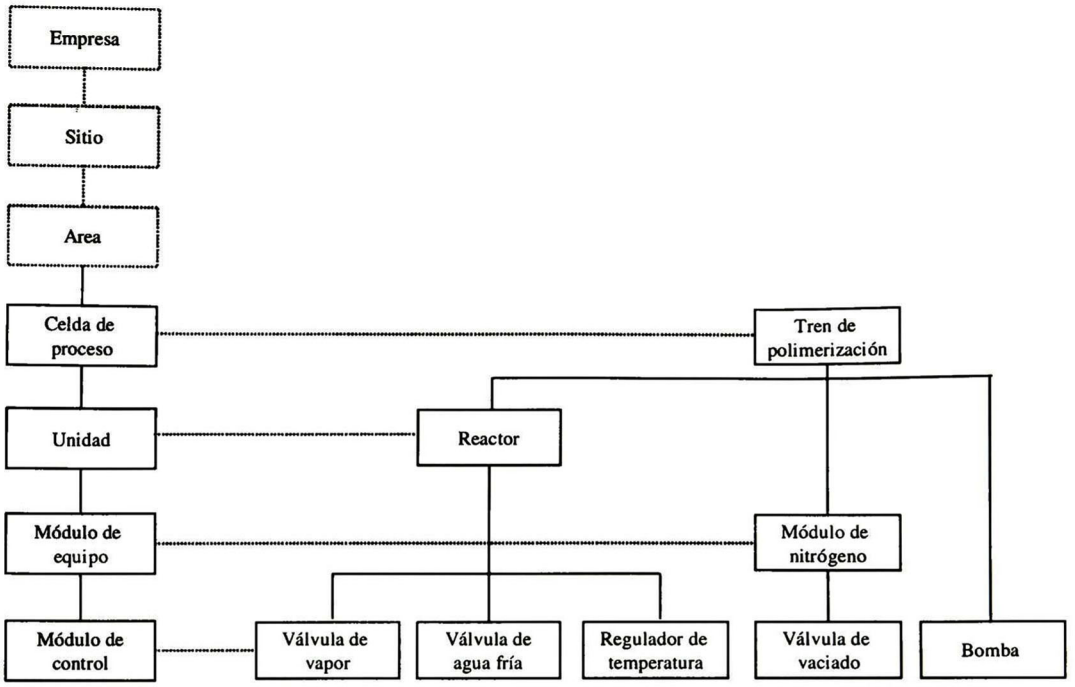


Figura 1.1: Modelo físico y un ejemplo

1.2.2. Modelos de referencia

En esta sección se describen tres modelos de referencia para plantas de manufactura por lotes: modelo físico, modelo de proceso y modelo de control de procedimiento. Los modelos presentados en esta sección presumen de ser completos, aunque algunos elementos en estos modelos pueden ser omitidos hasta el punto en que conserven su consistencia, otros elementos pueden ser agregados. siempre y cuando mantengan la integridad de la relación original.

Modelo físico

El modelo físico es usado para describir los activos físicos de una empresa. Los activos físicos son usualmente organizados de manera jerárquica. El modelo tiene siete niveles iniciando en "empresa" y finalizando con "módulo de control" La figura 1.1 ilustra este modelo. Los primeros tres niveles (línea punteada) son frecuentemente definidos respecto a un criterio organizacional más que a un criterio técnico, y no son descritos a detalle en el estándar. Los cuatro niveles inferiores de este modelo se refieren a tipos de equipo específicos. Un tipo de equipo es una colección de equipo de control y procesamiento físico agrupado para un propósito particular. Los cuatro niveles inferiores (celdas de proceso, unidades, módulos de equipo y módulos de control) son comúnmente definidos por las actividades de ingeniería. Durante estas actividades de ingeniería, el equipo de niveles más bajos es agrupado para formar un nuevo grupo de nivel más alto, y en algunos casos, un grupo dentro de un nivel puede ser incorporado en otro grupo del mismo nivel. La finalidad de hacer esto es simplificar la operación de aquellos equipos que son considerados piezas muy grandes. A continuación se describen brevemente los niveles del modelo físico [14],[10].

- *Empresa.* Es una colección de uno o más sitios; ésta puede contener sitios, áreas, celdas de proceso, unidades, módulos de equipo, y módulos de control. La empresa es responsable de determinar qué productos serán manufacturados, en qué sitios y en general cómo serán producidos.

- *Sitio.* Es una agrupación física, geográfica o lógica determinada por la empresa. Esta puede contener áreas, celdas de proceso, unidades, módulos de equipo y módulos de control.

- *Área.* Es una agrupación física, geográfica o lógica determinada por el sitio; ésta puede contener celdas de proceso, unidades, módulos de equipo y módulos de control.

- *Celda de proceso.* Es una agrupación lógica de equipo que incluye el equipo necesario para procesar uno o más lotes. La celda de proceso define el espacio de control lógico de un conjunto de equipos de procesos dentro de un área. Esta puede contener unidades, módulos de equipo y módulos de control. El dominio para un sistema de control por lotes es la celda de proceso. En muchos casos la celda de proceso es una combinación de sistemas suministrados por más de un proveedor.

- *Unidad.* Está compuesta de módulos de equipo y módulos de control. Una o más actividades de procesamiento – tales como reaccionar, cristalizar y hacer una solución – pueden ser conducidas por una unidad. Usualmente, la unidad se centra en una pieza de procesamiento, tal como un tanque mezclador o un reactor. Físicamente la unidad puede incluir o puede adquirir los servicios de todo el equipo relacionado lógicamente para completar una tarea de procesamiento específica. Una unidad puede tener cualquiera de los siguientes atributos:

  - Contiene u opera sobre un lote completo (o sobre una parte de material).
  - No opera sobre más de un lote al mismo tiempo.
  - No requiere el uso de otras unidades.
  - Opera independientemente de otras unidades.

- *Módulo de equipo.* Puede estar compuesto de módulos de control y módulos de equipo subordinado. Un módulo de equipo puede ser parte de una unidad o un equipo independiente agrupado dentro de una celda de proceso. Un módulo de equipo puede realizar un número finito de actividades menores de procesamiento específicas, tal como dosificar, pesar, filtrar, etc. Usualmente, el módulo de equipo se centra en una pieza fija de equipo de procesamiento como un filtro, una báscula, un agitador, etc.

- *Módulo de control.* Es una colección de sensores, actuadores, otros módulos de control y equipo de procesamiento que, desde el punto de vista de control, es operado como una sola entidad. Un módulo de control puede también estar compuesto de otros módulos de control. Por ejemplo, un módulo de control puede ser definido como una combinación de varios bloques de válvulas automáticas On/Off.

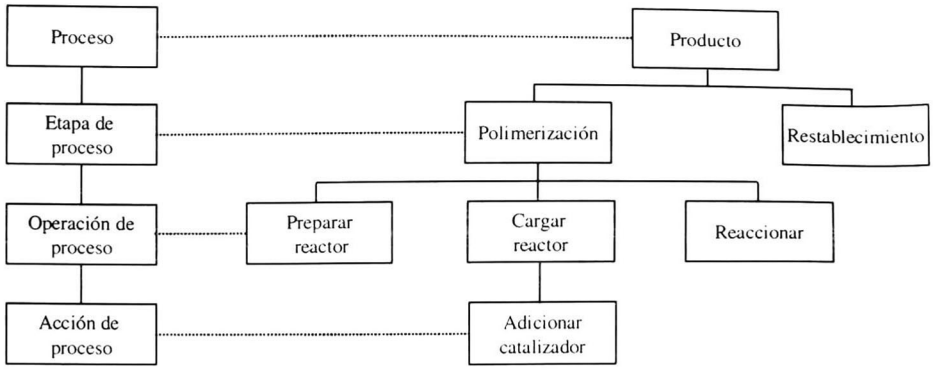


Figura 1.2: Modelo de proceso y un ejemplo

Modelo de proceso

Un proceso por lotes puede ser subdividido y organizado de una manera jerárquica [14],[10]. En la figura 1.2 se ilustra esta división.

Los componentes del esquema mostrado son descritos a continuación:

- *Proceso.* Es una secuencia de actividades químicas, físicas o biológicas, transporte para la conversión, transporte o almacenamiento de material o energía.

- *Etapa del proceso.* El proceso consiste de una o más etapas organizadas como un conjunto ordenado, el cual puede ser secuencial, paralelo, o una combinación de ambos. Una etapa del proceso es una parte que generalmente opera independientemente de otras etapas del proceso, y usualmente resulta ser una secuencia planeada de cambios físicos o químicos (polimerización, secado, etc) en el material que será procesado.

- *Operación de proceso.* Cada etapa del proceso consiste de un conjunto ordenado de uno o más operaciones. Las operaciones de proceso representan las principales actividades de procesamiento. Una operación de proceso usualmente conlleva un cambio físico o químico en el material que será manejado. Algunos ejemplos son, preparar reactor, cargar material, reaccionar substancias, etc.

- *Acciones de proceso.* Cada operación del proceso puede ser subdividida en un conjunto ordenado de una o más acciones. Una acción de proceso describe una actividad pequeña de procesamiento que es requerida para hacer una operación de proceso. Algunos ejemplos son, adicionar catalizador, calentar reactor, etc.

Modelo de control por procedimiento

El modelo de control por procedimiento, describe el control que dirige las acciones orientadas a equipo que toman lugar en una secuencia ordenada para realizar alguna tarea orientada a proceso. El control por procedimiento es el tipo de control más común en sistemas de manufactura por lotes. Su función es habilitar el equipo para desempeñar un proceso por lotes; los elementos por procedimiento están organizados de una manera jerárquica [14],[10]. Esta jerarquía se ilustra en la figura 1.3. Los niveles del modelo se describen a continuación.

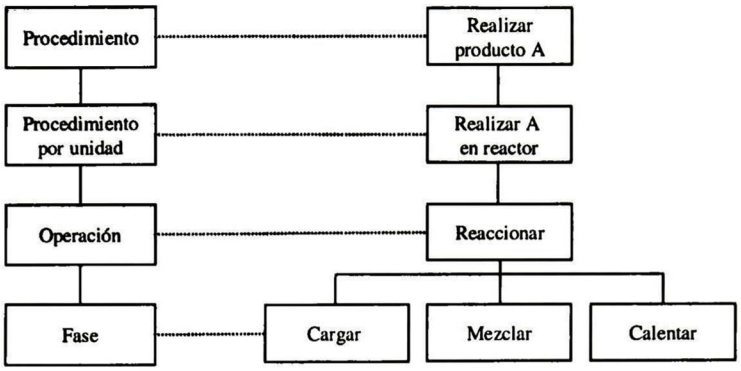


Figura 1.3: Modelo de control por procedimiento y un ejemplo

- *Procedimiento.* Es la estrategia para realizar un proceso, donde su dominio es la celda de proceso. El procedimiento es el nivel más alto en la jerarquía, y es definido en términos de un conjunto ordenado de procedimientos por unidad. Un ejemplo de un procedimiento es "hacer un producto x"

- *Procedimiento por unidad.* Consiste de un conjunto ordenado de operaciones que originan una secuencia de producción, la cual toma lugar dentro de una unidad. Un procedimiento por unidad es una estrategia para realizar viarias operaciones dentro de una unidad, pero sólo una operación puede ser ejecutada por una unidad en un instante de tiempo. Algunos ejemplos de procedimientos por unidad son: polimerización, secado de PVC, etc.

- *Operación.* Es un conjunto ordenado de fases que define una secuencia de procesamiento principal, donde el material que será procesado es tomado de un estado y llevado a otro estado; usualmente implica un cambio físico y químico sobre dicho material. Es deseable localizar límites de operación dentro del procedimiento, donde el procesamiento normal seguramente pueda ser suspendido en ciertos puntos. Ejemplos de operaciones son:

  - Preparación: Vaciar y limpiar el reactor
  - Cargar: Adicionar agua desmineralizada
  - Reaccionar: Adicionar sustancia x y un catalizador, calentar, y esperar que el reactor alcance la presión necesaria.

- *Fase.* El elemento más pequeño de control por procedimiento que puede cumplir con una tarea orientada a proceso, es una fase. Las fases pueden ser ejecutadas en forma paralela o secuencial. El diseño de una fase necesita tomar en cuenta la seguridad y las condiciones de excepción. Una fase puede ser subdividida en partes más pequeñas, y puede ser caracterizada por uno o más comandos, tales como:

  - Configurar, iniciar y cambiar alarmas y otros límites
  - Configurar y cambiar constantes, modos del controlador y tipos de algoritmos.

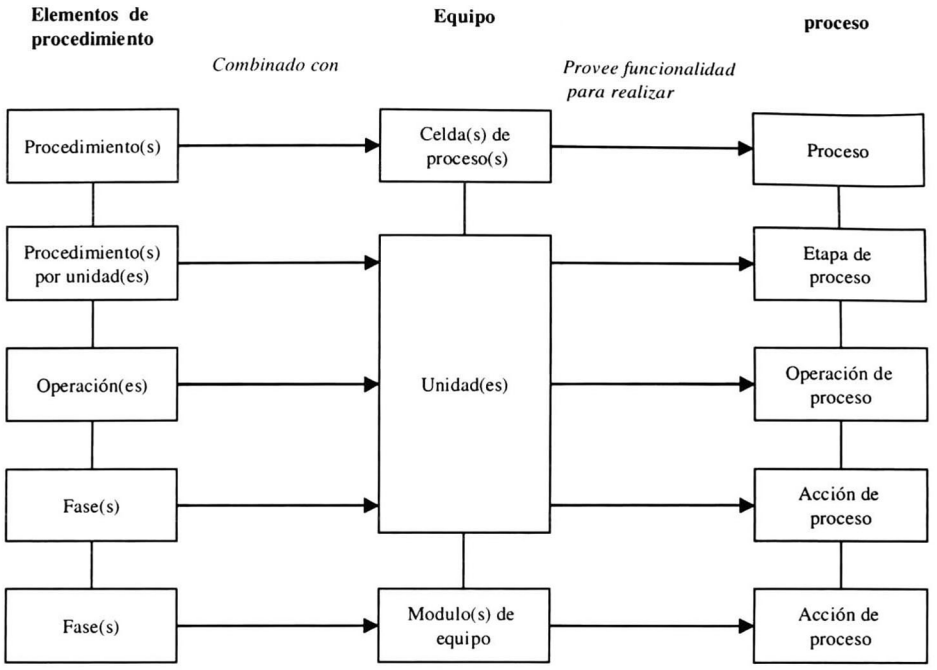


Figura 1.4: Integración de modelos

- Lectura de variables de proceso, tal como la densidad y temperatura del gas, el volumen, la masa, etc.

La finalidad de una fase es causar o definir una acción orientada a proceso, mientras el conjunto de pasos que componen una fase son específicos del equipo. Ejemplos de fases son: adicionar, calentar, mezclar, enfriar, etc.

Integración de modelos

La relación general entre el modelo físico, el modelo de procesos y el modelo de control por procedimiento, es ilustrado en la figura 1.4. Aplicar el control de procedimiento al equipo provee funcionalidad para realizar cualquier elemento descrito en el modelo de proceso [14],[10].

1.2.3. Clasificación de celdas de proceso

Las celdas de proceso pueden ser clasificadas: por número de productos y por estructura física [14],[10].

Por número de productos

Un proceso *mono-producto*, realiza el mismo producto en cada proceso. Variaciones en el procedimiento y parámetros son posibles.

Un proceso *multi-producto*, realiza diferentes productos utilizando diferentes métodos de producción y control. Hay dos posibilidades:

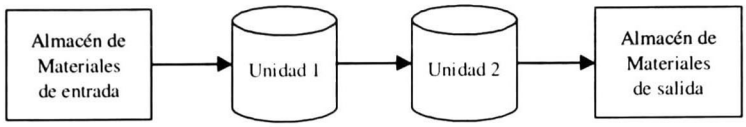


Figura 1.5: Estructura mono-ruta

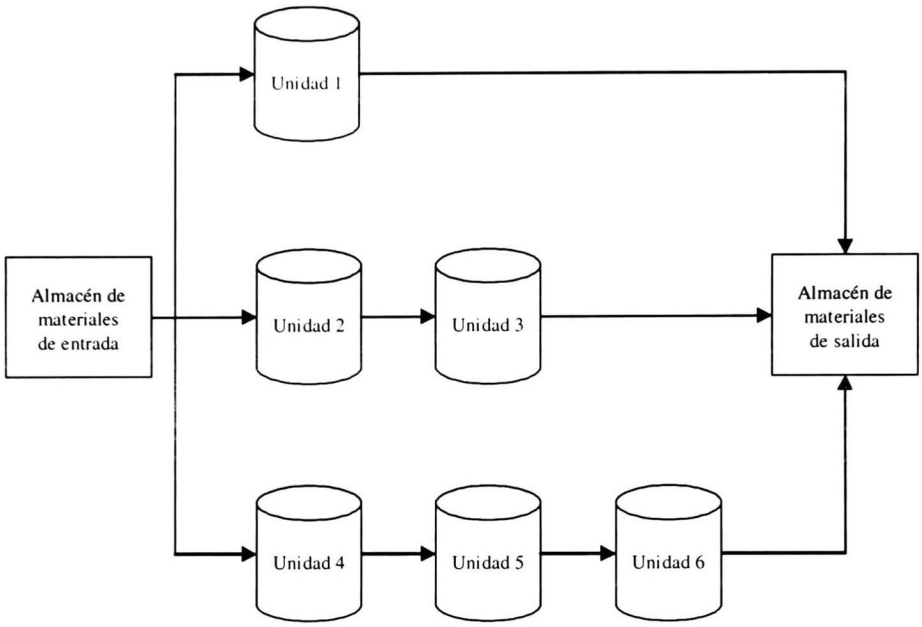


Figura 1.6: Estructura multi-ruta

1. Todos los productos son manufacturados con el mismo procedimiento usando diferentes valores en la fórmula (variando materiales y/o parámetros).

2. Los productos son manufacturados usando diferentes procedimientos.

Por estructura física

Una estructura *mono-ruta*, es un grupo de unidades a través de las cuales un lote pasa secuencialmente. Una estructura *mono-ruta* puede ser una sola unidad, como un reactor, o varias unidades en secuencia. Pueden usarse diferentes materiales de entrada, lo cual permite que múltiples productos puedan ser generados. Varios lotes pueden procesarse al mismo tiempo. Ver figura 1.5.

Una estructura *multi-ruta*, consiste de múltiples estructuras mono-ruta en paralelo donde los productos no se transfieren entre ellas. Las unidades pueden compartir fuentes de materia prima y almacenes de producto. Varios lotes pueden estar procesándose al mismo tiempo. Aunque las unidades dentro de una estructura multi-ruta pueden ser físicamente similares, es posible tener rutas y unidades dentro de una estructura multi-ruta que sean radicalmente diferentes en el diseño físico. Ver figura 1.6.

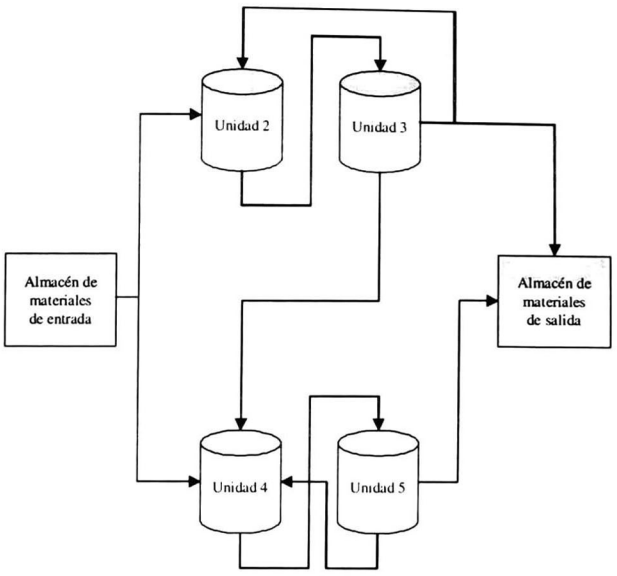


Figura 1.7: Estructura de red

Una estructura *de red,* se constituye de varias secuencias de unidades o rutas que pueden ser fijas o variables. Cuando la ruta es fija, las mismas unidades son usadas en la misma secuencia. Cuando la ruta es variable, la secuencia puede ser determinada previamente o durante la ejecución del lote. La ruta puede ser totalmente flexible, esto es, un lote no necesariamente tiene que iniciar en una unidad específica, éste puede iniciar en cualquier unidad y tomar múltiples rutas a través de las celdas de proceso. La ruta apropiada es determinada en el momento de ejecución y se basa en las restricciones de capacidad del equipo y en los requerimientos de la receta. El control de un proceso en este tipo de estructura es muy complejo, debido a la asignación y arbitración de solicitudes del equipo. En este tipo de estructura varios lotes pueden estar en producción al mismo tiempo. La figura 1.7 muestra la complejidad de esta estructura del proceso.

1.2.4. Recetas

En la actualidad la forma de visualizar los procesos por lotes ha evolucionado, se ha intentado dejar a un lado la intuición y la experiencia, lo cual da lugar a un enfoque más formal y riguroso. Este esfuerzo por crear una visión estructurada y consistente de procesos por lotes ha llevado a una revisión de paradigmas. Entre éstos, las recetas son uno de los paradigmas que más atención ha puesto la industria y las organizaciones de estandarización.

El viejo paradigma de que una receta es una lista de ingredientes con un conjunto de instrucciones escritas sobre papel, es actualmente inoperante en ambientes de trabajo informatizados. En sí, una receta puede tomar cualquier apariencia que se desee. Los datos contenidos en una receta pueden estar sujetos a un formato dentro de un reporte, sobre una pantalla, o bien, sobre una hoja de papel.

De la experiencia se puede tener una idea de qué compone una receta. Las recetas usadas en la preparación de comida dan una primera aproximación. Este paradigma es una excelente

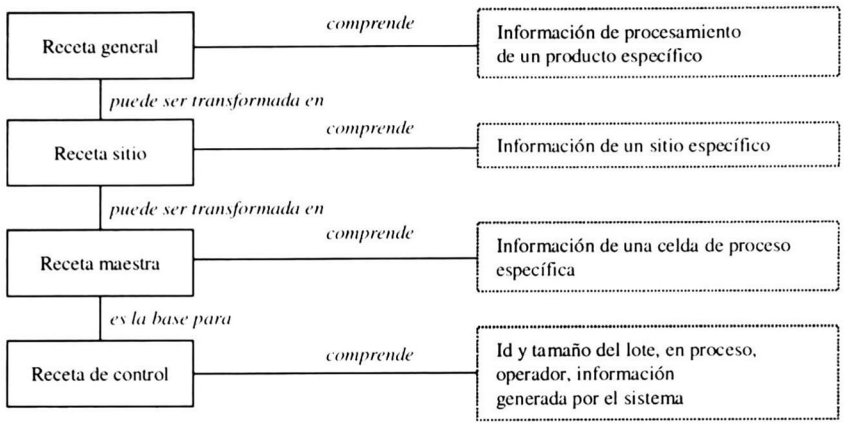


Figura 1.8: Tipos de receta

base para el entendimiento de los fundamentos de una receta por lotes, por ejemplo, en una receta de cocina, los elementos típicos son: producto terminado, su cantidad, número de porciones a servir, color, textura, pruebas, etc. Los ingredientes son listados en la receta, la cual supone un número de procedimientos estándar que aseguran la calidad deseada del producto.

El estándar revisado (S88.01-1995) describe cuatro tipos de recetas generalmente usadas en la industria de procesos, nombradas: *general*, *sitio*, *maestra* y de *control* [14], [10]. Una receta es una entidad que contiene el mínimo conjunto de información que define los requerimientos de manufactura para un producto específico. Dependiendo de los requerimientos específicos de una empresa, puede existir otro tipo de recetas. Los tipos de receta mencionados son mostrados en el esquema de la figura 1.8 y descritos brevemente a continuación.

- **Receta general.** Es una receta de nivel empresarial que sirve como base para recetas de nivel inferior. Este tipo de receta es creada sin conocimiento específico del equipo que será usado para manufacturar el producto; esta receta identifica la materia prima, sus cantidades relativas y el procesamiento requerido. La receta general es creada por personas con conocimiento en química y en ingeniería de procesos del producto en cuestión. La receta *general* no incluye especificaciones de equipo, pero la tecnología para manufacturar un producto usualmente va más allá de un laboratorio, por lo que los requerimientos de equipo pueden ser descritos con suficiente detalle. La receta *general* provee un medio para comunicar requerimientos de procesamiento para sitios de manufactura múltiple.

- **Receta sitio.** Esta receta es específica de un sitio particular; es la combinación de información específica de un sitio y de una receta *general*. La receta *sitio*, es usualmente derivada de una receta *general*, la cual se adapta a las condiciones de un sitio de manufactura específico, y provee el nivel de detalle necesario para este nivel de receta. Sin embargo, esta receta puede ser creada directamente, sin la existencia de una receta *general*. También, es posible derivar múltiples recetas *sitio* de una receta *general*.

- **Receta maestra.** Este nivel de receta está dirigido a una celda de proceso o a un subconjunto de ella. Una o varias recetas maestras pueden ser derivadas de una receta *general* o de una receta *sitio*. También puede ser creada como una sola entidad sí el creador tiene el conocimiento necesario del producto y del proceso. La receta maestra puede contener información detallada de un producto específico, tal como datos del proceso y requerimientos de equipo. Este nivel de receta es requerido, porque sin él, la receta de control no puede ser creada y por lo tanto, los lotes no pueden ser producidos. Si el equipo de manufactura por lotes es operado manualmente o automáticamente, la receta maestra puede ser un conjunto de instrucciones escritas o una entidad electrónica.

- **Receta de control.** Es una instancia de la receta maestra, es decir, una copia de una versión de la receta maestra y es modificada con información operacional para atender un lote específico. La receta de control contiene información del proceso de un producto, la cual es necesaria para manufacturar un lote particular. Esta provee el nivel de detalle para iniciar y monitorear procedimientos sobre equipo en una celda de proceso. La receta de control ha sido modificada para considerar la calidad de los materiales y el equipo que será utilizado. Las modificaciones de una receta de control pueden ser hechas sobre un periodo de tiempo basado en calendarización, equipo, e información del operador; una receta de control puede sufrir varias modificaciones durante el procesamiento de un lote. Ejemplos de modificaciones: definir el equipo que será realmente usado para la receta de control en la iniciación de un lote, adicionar o ajustar los parámetros basados sobre la calidad de la materia prima, cambiar el procedimiento basado en algunos eventos esperados. La receta general y sitio son recetas definidas independientemente del equipo; la receta maestra y de control por el contrario, dependen completamente del equipo utilizado.

Una receta contiene las siguientes categorías de información [14],[10]:

Encabezado: Incluye información de naturaleza administrativa. Esta información habla de la receta o producto, y generalmente, esta sección no participa directamente en la manufactura. Los datos que incluye el encabezado pueden variar de una compañía a otra. Algunos de ellos pueden ser, identificador de la receta, nombre del autor, estado, fecha de actualización, versión, comentarios, resumen del proceso, código de manufactura y alguna otra información administrativa.

Fórmula: Es una categoría de información dentro de la receta que incluye entradas de proceso, salidas de proceso y parámetros de proceso.

- Entrada de proceso. Se refiere a la identificación y cantidad de materia prima u otro recurso requerido para hacer el producto. Las cantidades pueden ser especificadas como valores absolutos o como ecuaciones basadas en parámetros de la fórmula. Las entradas de proceso pueden ser vistas como una lista de ingredientes.

- Salida de proceso. Se refiere a la cantidad de material y/o energía esperada como resultado de una ejecución de la receta. Estos datos pueden detallar el impacto ambiental, y también pueden contener información sobre la especificación de las salidas propuestas en términos de cantidad y beneficio.

- Parámetros de proceso. Detallan información tal como, temperatura, presión, o tiempo pertinente para el producto, pero que no cae dentro de la clasificación de entradas o salidas. Los parámetros de proceso pueden ser usados como un conjunto de puntos, comparación de valores.

Requerimientos de equipo. Restringe la elección del equipo que será eventualmente usado para implementar una parte específica del procedimiento. En las recetas general y sitio, los requerimientos de equipo son usualmente descritos en términos generales, así como los materiales permitidos y las características de procesamiento requeridas. Esto es la guía de las restricciones impuestas por los requerimientos de equipo que permitirán a la receta general y sitio eventualmente ser usadas para crear la receta maestra.

Procedimiento. Define la estrategia para realizar un proceso. Los procedimientos de la receta general y sitio son estructurados usando los niveles descritos en el modelo de proceso, ya que estos niveles permiten que el proceso sea descrito independientemente del equipo. Los procedimientos de la receta maestra y de control son estructurados usando los elementos del modelo de control por procedimiento, ya que estos elementos de procedimiento tienen una relación con el equipo.

El creador de la receta está limitado a usar los elementos de procedimiento que han sido o serán configurados y los cuales estarán disponibles para su uso en la creación del procedimiento. El creador de la receta puede usar cualquier combinación de estos elementos para definir un procedimiento. La determinación de cuál de estos elementos pueden ser parte del procedimiento, es una decisión de diseño específica de una aplicación basada en muchos factores, incluyendo las capacidades de los controles y del grado de libertad apropiado para el creador de la receta.

Otra información. Es una categoría de la receta que contiene información sobre datos de soporte, los cuales no están contenidos en otras partes de la receta. También captura comentarios sobre medidas de seguridad. Un comentario válido puede ser activar el sistema de ventilación, si el lote es sobrecalentado durante una operación reactiva. Además, puede incluir requerimientos para la generación de reportes. En la figura 1.9 se muestran los elementos que componen una receta.

1.3. Modelado de sistemas de manufactura por lotes

1.3.1. Las redes de Petri como herramienta de modelado para procesos por lotes

Desde hace más de veinte años las redes de Petri (RP) han sido utilizadas en la especificación de sistemas de control/coordinación de sistemas de manufactura. En particular en la década pasada se han aplicado en el modelado, análisis y operación de procesos por lotes. Aunque la naturaleza de este tipo de sistemas los hace particularmente complejos, su investigación ha resultado ser un reto para la industria de procesos. Desde entonces las redes de Petri han sido utilizadas en el modelado de procesos por lotes.

Existen trabajos muy interesantes al respecto, por ejemplo R. Champagnat [7],[8], presenta una integración de RP Predicado/Transición con un conjunto de ecuaciones algebraicas

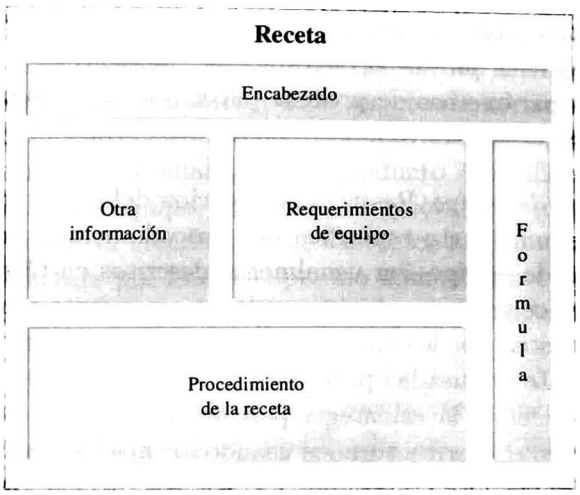


Figura 1.9: Elementos que componen una receta

diferenciales, para modelar el comportamiento híbrido de un sistema por lotes. También presenta algunos modelos formales basados en RP (temporizadas, coloreadas, híbridas, Predicado/Transición) que describen procesos por lotes.

Andreu [4] analiza como un enfoque jerárquico usado en el campo de sistemas de manufactura discreta puede ser extendido a sistemas por lotes. Este enfoque jerárquico permite la descomposición de todo el problema dentro de tres niveles diferentes: *nivel de control local*, *nivel de coordinación* y *nivel de supervisión*. La posibilidad de usar varias clases de RP en cada nivel jerárquico ofrece una ventaja para la consistencia del sistema.

Por otro lado, Valette [5],[8],[16] aborda los beneficios de un enfoque basado en RP para el modelado, análisis y simulación de sistemas híbridos, ya que las RP no sólo describen la funcionalidad del sistema (modelo comportamental), sino también capturan la estructura del sistema (modelo estructural).

Andreu y Valette [5] tratan la interacción y coordinación de un modelo discreto y un modelo continuo, sin integrar el aspecto continuo dentro de la teoría de RP. El modelo discreto lo descomponen en dos partes: un modelo de refrencia y un módulo de control. La interacción la establecen a través de un generador de eventos que se ejecuta en paralelo.

Silva [28] presenta un tutorial sobre la utilización de modelos de RP en varias etapas del ciclo de vida de un proceso por lotes.

Tittus y Akesson [2],[29],[30] introducen modelos de RP para recetas y recursos de la planta. Dos clases genéricas de recursos, nombrados dispositivos de procesamiento (tanques, reactores y otros contenedores como unidades) y dispositivos de transporte (válvulas y líneas de conexión) son considerados. Las recetas son formuladas de acuerdo a cinco elementos: secuencias de operación, movimiento de material, adición de material durante una operación, mezcla y separación de lotes. El modelo en RP para toda la planta puede ser construido por sincronización de los modelos de recursos y con el modelo de la receta.

1.3.2. Modelado de componentes

Una gran cantidad de actividades de procesos por lotes pueden ser representadas formalmente y gráficamente a través de RP, donde los nodos *lugar* son usados para representar condiciones o el estado de un recurso, y los nodos *transición* para modelar eventos, como: iniciar o finalizar operaciones, prender o apagar interruptores, abrir o cerrar válvulas, etc.

La presencia de una marca en un lugar indica si un recurso esta disponible, si una operación está en ejecución, o si una condición es verdadera. Múltiples marcas implica disponibilidad de múltiples recursos.

Modelar un sistema de manufactura por lotes implica modelar los recursos y modelar la receta.

Modelado de recursos

Una planta consiste de dos clases genéricas de recursos: *dispositivos de procesamiento* (unidades), los cuales efectúan cambios físicos y lógicos en las propiedades del producto, y *dispositivos de transporte*, que se encargan de transportar el producto entre el equipo o entre los sitios de la planta.

En la mayoría de los sistemas de manufactura, los *dispositivos de procesamiento* y los *dispositivos de transporte* tienen que cooperar para mover el material de un contenedor al siguiente contenedor, o bien de un lugar a otro. Esto conduce a la idea de definir recursos virtuales, llamados *líneas de conexión* o simplemente *líneas*. Una línea es un objeto abstracto, el cual puede ser visto como una clase de dispositivo de transporte, y su función es conectar unidades de equipo fuente con unidades de equipo destino. Así que para cada posible conexión entre cualquier par de unidades un objeto línea tiene que ser creado con la información de la topología de la planta.

Modelado de la receta

Los modelos de producto o recetas pueden ser especificados en diferentes niveles de abstracción. Esto significa que se pueden distinguir recetas *independientes de la planta*, las cuales describen las operaciones que serán aplicadas sin hacer referencia al equipo que será utilizado, y las recetas *dependientes de la planta*, las cuales describen las operaciones y rutas que sigue la materia prima a través de una planta específica.

Una receta debe ser capaz de modelar la secuencia de operaciones junto con los recursos que utiliza. Una receta debe previamente hacer la reservación de los recursos que pretende usar. Generalmente, los recursos pueden ser utilizados sólo por una receta al mismo tiempo. Además, una receta es responsable de la liberación de recursos. Si varias recetas se ejecutan en paralelo, es necesario tener una estrategia de reservación para evitar bloqueo de recursos.

Las funciones que pueden ser aplicadas a un lote son:

- Operación. Aplicar diferentes fases a un lote completo o a una parte de él.
- Mover. Trasladar el lote de una unidad a otra.
- Adicionar. Agregar material a una unidad que ya contiene parte del lote.
- Juntar. Mezclar dos partes de un lote provenientes de fuentes distintas en una unidad distinta.
- Dividir. Separar un lote en dos partes disjuntas.

La dificultad de construir un modelo para una clase de sistemas depende en gran medida de la técnica de modelado elegida.

Las RP son adecuadas para construir modelos de sistemas de producción. Se ha probado que esta herramienta es muy completa para el modelado, análisis, simulación y control de sistemas de manufactura.

Existen varias razones que justifican lo anterior:

- Poseen un fuerte poder descriptivo
- Capturan las relaciones de precedencia, sincronización y concurrencia de eventos
- Sirven como herramienta de análisis, ya que tienen un fuerte fundamento matemático.

Las RP son frecuentemente usadas para modelar aspectos discretos de procesos discontinuos, como es el caso de los sistemas de manufactura por lotes. Para intuir el modelado de este tipo de sistemas y para aplicar los conceptos definidos se ha considerado un sistema simple de manufactura por lotes basado en receta. La receta del producto deseado es muy sencilla, y consiste en: 1) Cargar un lote de materia prima (fluído) a un reactor; 2) Ejecutar una reacción con dicho material dentro del reactor; 3) Transferir el lote a un tanque de almacenamiento (buffer), y mientras es transferido el lote debe ser enfriado. 4) Almacenar y reposar el lote un cierto tiempo en un tanque; 5) Transferir el lote a otro reactor para una segunda reacción (otra operación de carácter continuo); 6) Una vez que termina esta operación, transferir el lote fuera de la planta.

Los pasos de esta receta exponen claramente la naturaleza híbrida que caracteriza a un sistema de producción por lotes. Por ejemplo, el paso 2 implica una operación de carácter discreto (reaccionar), y su duración no depende del tamaño del lote. Sin embargo, en el paso 3 la operación es continua (transferir), y su duración es proporcional al tamaño del lote, porque al pasar el lote por el dispositivo de enfriamiento el índice de flujo es constante. También, otra operación continua se ejecuta en los pasos 5 y 6.

Por lo tanto, dada la receta asumimos que el sistema de producción comprende dos reactores, $R1$ y $R2$, un tanque de almacenamiento $T1$, y un dispositivo de enfriamiento $C1$ que esta disponible a la entrada del tanque de almacenamiento. La estructura física de la planta se presenta en la figura 1.10.

La receta asignada a la planta es una *receta maestra*, la cual describe la secuencia de operaciones que deben ser ejecutadas para obtener el producto final, a partir de la materia prima, y es similar a una ruta de producción abstracta. Esta puede ser fácilmente representada por una RP, la cual se dibuja en la figura 1.11.

La receta *maestra* establece una relación con la disponibilidad de los recursos. En este caso, la disponibilidad de los dos reactores se modela en base a las operaciones que requieren su uso, por lo que las operaciones de transferencia y reacción enunciadas en la receta deben solicitar previamente su reservación. Ver figura 1.12.

También, son requeridos por la receta el tanque de almacenamiento $T1$ y el dispositivo de enfriamiento $C1$ para una operación de transferencia. Por su dependencia funcional ambos dispositivos son considerados como un solo recurso. Así, un lugar con una marca en el modelo es suficiente para indicar su disponibilidad. Ver figura 1.13.

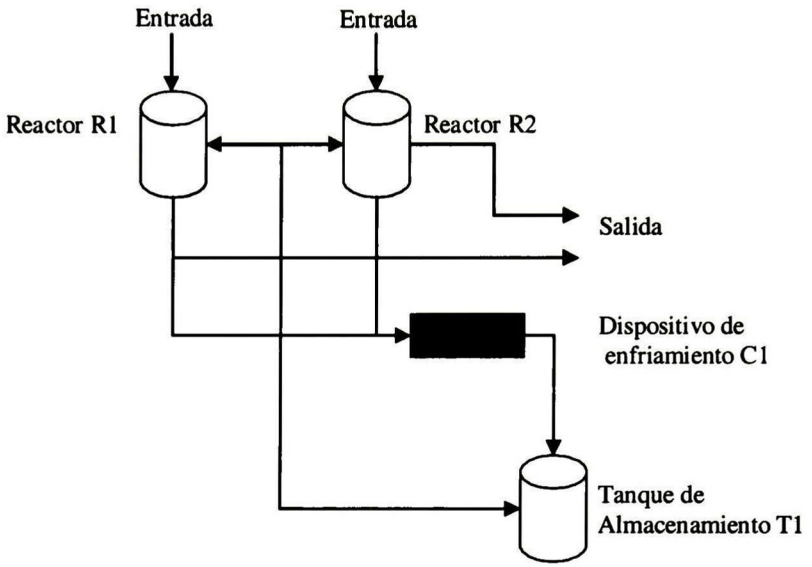


Figura 1.10: Planta de un sistema de lotes simple

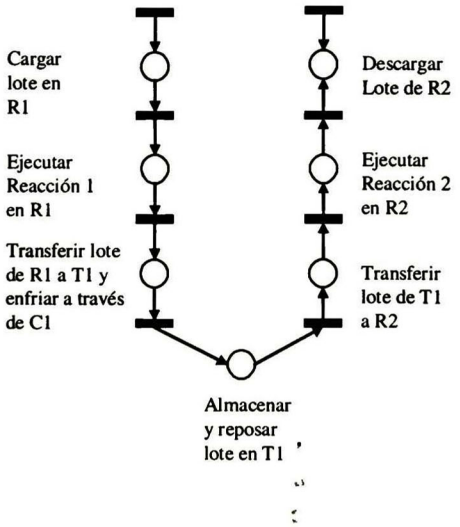


Figura 1.11: Descripción de la receta maestra

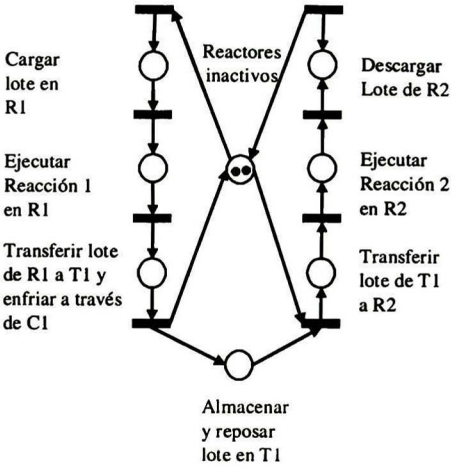


Figura 1.12: Reservación de reactores

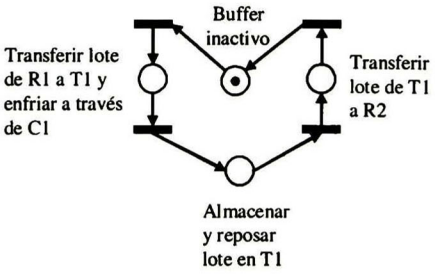


Figura 1.13: Reservación del tanque de almacenamiento (Buffer)

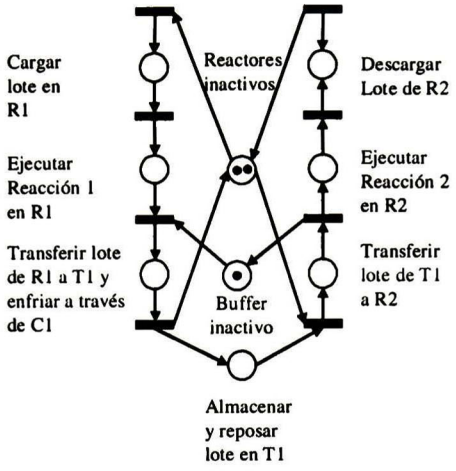


Figura 1.14: Receta maestra con reservación de recursos

La receta *maestra* es construida por la secuencia de operaciones y el equipo utilizado durante su ejecución, por lo tanto, la receta *maestra* se obtiene de la composición de los tres modelos anteriores. La figura 1.14 muestra el modelo final.

Este modelo sólo describe el aspecto discreto del sistema por lotes y descarta el aspecto continuo, pero es muy representativo para el modelado de la receta y de los recursos de una planta de manufactura.

1.4. Conclusión

Este capítulo resume una revisión bibliográfica de diversos tópicos sobre sistemas de producción, procesos por lotes, modelado de sistemas de manufactura flexible y otros temas más relacionados, que hacen uso de las RP como herramienta de soporte para el modelado, análisis (cualitativo y cuantitativo), diseño, control, planificación y calendarización e implementación de sistemas que caen dentro del dominio; claramente se puede inducir que las RP son una técnica atractiva para las áreas de ingeniería, ya que proveen entre otros, un marco formal apto para satisfacer la gran mayoría de los requerimientos industriales que exige una especificación de sistemas de producción. Sin embargo, siguen despertando interés, debido a que aún quedan problemas que atacar y mejorar, y como muestra el caso que aborda esta tesis.

Además en el capítulo se han incluido algunos conceptos, modelos referencia y terminología manejada en la industria de procesos por lotes. Esta información contenida en el estándar ANSI/ISA-S88: Control por lotes, Parte 1: Modelos y terminología, acercan las buenas prácticas para el diseño y operación de plantas de manufactura por lotes. Información muy útil para los fines que se persiguen en capítulos posteriores, ya que los modelos que propone son retomados, y los términos son adoptados para uniformizar la notación.

También, se destacan algunos trabajos importantes, enfocados particularmente a sistemas que operan con procesos basados en receta, los cuales usan las redes de Petri para modelar sus componetes. Se modela un ejemplo sencillo de una planta de manufactura por lotes para

dar una vista previa al modelado de esta clase de sistemas.

Capítulo 2

Un formalismo de red multinivel

Resumen: En este capítulo se presenta la definición del sistema $n - LNS$ (n-Level Net System), se trata de un formalismo de red a n niveles que permite la especificación de sistemas con entidades móviles con capacidad de movilidad, los cuales evolucionan en ambientes estructurados. Esta extensión de las RP permite que las marcas en los lugares sean otras RP. Una explicación intuitiva del formalismo da paso a la descripción formal del sistema $n - LNS$ y se concluye con un ejemplo.

2.1. Motivación

Las redes de Petri han sido ampliamente adoptadas por las comunidades de ciencias de la computación y de automatización como un formalismo para la especificación de sistemas grandes y complejos. Por lo que, la descripción de estos sistemas ha motivado la creación de redes de Petri de alto nivel.

Primero las redes Pr-T, propuestas por H. Genrich [11], y poco después las RP coloreadas, introducidas por K. Jensen [15] donde las marcas tienen una identidad (color) representada por símbolos. Estos trabajos son extensiones al formalismo original. Más tarde, en la implementación de un modelo por E. Kasturia [17], los colores son representados como estructuras de datos.

En la última decada, el enfoque de modelado de alto nivel aparece en la especificación de programas orientados a objetos: los conceptos de RP de alto nivel y la programación orientada a objetos se unen para dar origen a OOPN [22] y objetos cooperativos [26], donde las marcas pueden ser otras redes de Petri. Sin embargo, estos formalismos aproximan los modelos a implementaciones de software, perdiendo la claridad en la descripción del modelo.

Recientemente, la idea de considerar RP como marcas es retomada por R. Valk [31] quien propone RP a dos niveles "libres de código", es decir, un formalismo independiente del lenguaje de programación. Un formalismo similar a esta definición, es introducido por K. Hiraishi [12] quien con la noción de redes dentro de redes propone PN^2. Por otro lado, Lomazova [23] propone RP anidadas. En tanto O. Kummer [20] propone redes referencia como un formalismo de soporte para una herramienta de simulación en la que las marcas son referencias a otras redes.

Estos trabajos inspiran la creación de un sistema de red a tres niveles, llamado NS-3 Almeyda [3]. Una aportación de este formalismo con respecto al trabajo de Valk, es que agrega un tercer nivel de abstracción, y las redes definidas a este nivel son similares a las RP coloreadas, lo cual enriquece el modelo al brindar mayor compacidad que si se modelara con RP ordinarias. Además, el sistema NS-3 mantiene un modelo libre de detalles de implementación, y describe sistemas complejos sobre modelos orientado a objetos utilizando el enfoque de diseño multi-agente.

De ahí que surga el interés por adoptar este sistema y hacer una extensión a n-niveles de red, que elimine algunas restricciones impuestas por NS-3, mejorando el mecanismo de sincronización, y amplie las capacidades de especificación al no limitar la cantidad de niveles de red, aprovechando dos de los beneficios inherentes al enfoque multi-nivel: *compacidad* y *modularidad*.

La aplicación del formalismo $n - LNS$ está orientado a sistemas de manufactura por lotes, ya que generalmente los proyectos de la industria son grandes y complejos, y es donde se pueden apreciar las ventajas de un enfoque multi-nivel.

2.2. Presentación intuitiva

En esta sección se presenta una descripción informal del sistema $n - LNS$ propuesto. El sistema $n - LNS$ permite definir un número arbitrario de niveles de red. La cantidad de niveles de red que se pueden considerar para modelar un sistema, depende del grado de abstracción que se desee en el modelo, de acuerdo a la talla y complejidad del sistema a

describir.

Cada nivel de red es identificado por un número entero que está dentro del rango de 1 hasta n, donde el nivel de red más alto toma el valor de 1. La forma de establecer una jerarquía entre los niveles de red es mediante las marcas de red, esto es, las marcas a su vez son RP que están contenidas en lugares de una red de mayor nivel. Por lo tanto, cualquier red de nivel inferior puede estar contenida en una red de nivel superior, pero al revés no.

Un conjunto de redes puede ser asignado a cada nivel exceptuando el nivel de red 1, el cual consiste de una sola red.

Para construir una red de nivel i es necesario definir una red *tipo* y proporcionar un *marcado*. La red *tipo* se compone de:

- una estructura de red ordinaria,
- un conjunto finito de redes y/o símbolos permitidos para marcar los lugares de la red,
- un conjunto finito de etiquetas que serán asignadas a las transiciones que requieran ser sincronizadas
- un conjunto finito de variables asociadas a los pesos de los arcos, y
- tres funciones: una para asignar los tipos permitidos a los lugares, otra para asignar las etiquetas a las transiciones, y la última, para asociar los pesos a los arcos de la red respecto al conjunto de etiquetas.

El *marcado* está dado por una función que distribuye en los lugares de la estructura de red, un multi-conjunto de símbolos y redes de nivel inferior, cuyas redes tipo pertenecen al conjunto de tipos asignados a los respectivos lugares.

El nivel de red n, se distingue del resto de niveles por utilizar redes similares a las redes de Petri coloreadas, por lo tanto, su marcado no acepta redes marca, sino únicamente símbolos.

Example 1 *Para ilustrar lo anterior consideremos el modelo parcial de la figura 2.1; se trata de un sistema de red a cuatro niveles (n=4), donde el nivel* 1 *está representado por la red* NET_1, *el nivel* 2 *por las redes* $NET_{2,1}$, $NET_{2,2}$, *el nivel* 3 *por las redes* $NET_{3,1}$, $NET_{3,2}$, $NET_{3,3}$, $NET_{3,4}$, *y el nivel de red más bajo por las redes* $NET_{4,1}$, $NET_{4,2}$ *y* $NET_{4,3}$ *La figura muestra estructuras parciales de estas redes. La red* NET_1 *tiene como marcado dos redes de nivel* 2 : $NET_{2,1}$ *y* $NET_{2,2}$, *tres redes de nivel* 3: $NET_{3,1}$, $NET_{3,2}$ *y* $NET_{3,4}$, *una red de nivel* 4 : $NET_{4,2}$ *y los símbolos* $c1$, $c2$ *y* $c3$; *el marcado de la red* $NET_{2,1}$ *está dado por una red de nivel* 4, $NET_{4,1}$ *y el símbolo* $c3$. *La red* $NET_{2,2}$ *está marcada con una red de nivel* 3, $NET_{3,3}$, *y el símbolo* $c2$; *la red* $NET_{4,3}$ *está contenida en una red de nivel* 3, $NET_{3,3}$. *Las redes restantes sólo tienen símbolos en su marcado.*

La interacción entre componentes se realiza por medio de la sincronización de transiciones, la cual se declara mediante el etiquetado de transiciones. Dos o más transiciones que requieran sincronizarse llevarán la misma etiqueta; de otro modo la etiqueta es ε. Esta interacción es tomada en cuenta en las condiciones de habilitación y disparo de transiciones.

Cuando una transición está etiquetada con ε la habilitación y disparo de la transición se realiza de manera autónoma, es decir, la habilitación sólo depende del marcado de sus lugares de entrada y el disparo puede efectuarse al estar habilitada.

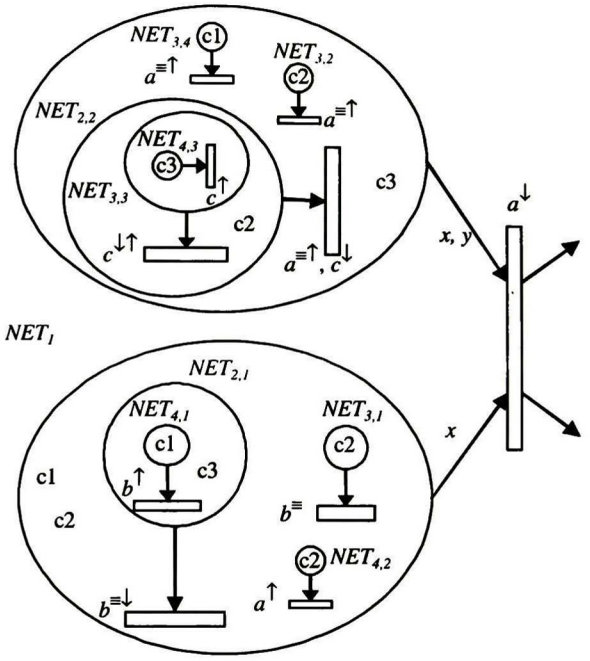


Figura 2.1: Fragmento de un módulo a 4 niveles

Pero cuando una transición está sincronizada se establece una dependencia con otras redes, la cual fija las condiciones de habilitación y disparo. Esta dependencia se define de acuerdo a tres tipos de sincronización: *sincronización local, sincronización interna y sincronización externa.*

La *sincronización local* se presenta cuando una transición de una red se sincroniza con transiciones de redes de nivel inferior y/o superior localizadas en el mismo lugar. La *sincronización interna* se presenta cuando una transición de una red es sincronizada "hacia dentro" con transiciones de redes marcando su lugar de entrada. El caso opuesto a este tipo de sincronización es, la *sincronización externa*, donde transiciones de redes son sincronizadas "hacia fuera" con una transición de salida del lugar donde las redes están contenidas.

La red de nivel 1 sólo utiliza sincronización interna. Las redes de nivel medio (nivel 2 hasta nivel $n-1$) utilizan los tres tipos de sincronización: *local, interna* y *externa.* Las redes de nivel n usan sincronización *local* y *externa.* De esta manera, una etiqueta puede estar denotada por: un etiqueta (*label*) cuando la red es de nivel 1, una tupla (*label*, *label*, *label*) cuando la red es de nivel medio, y una tupla (*label*,*label*) cuando la red es de nivel n. Cada etiqueta contenida en la tupla corresponde a un tipo de sincronización.

La figura 2.1 ilustra los tres tipos de sincronización mencionados. Para simplificar el etiquetado de la figura, la *sincronización local* se representa con el símbolo $\equiv$, la *sincronización interna* con $\downarrow$, y la *sincronización externa* con $\uparrow$. Así, la red $NET_{2,2}$ etiquetada con $c^{\downarrow}$, se sincroniza internamente con la red $NET_{3,3}$ etiquetada con $c^{\downarrow\uparrow}$, la cual a su vez se sincroniza internamente con la red $NET_{4,3}$, y externamente con $NET_{2,2}$. La red $NET_{4,3}$ etiquetada con $c^{\uparrow}$ indica que se sincroniza externamente con la red $NET_{3,3}$. Todas ellas se sincronizan respecto al mismo símbolo c.

La sincronización *local* y *externa* está presente en las redes $NET_{2,2}$, $NET_{3,2}$ y $NET_{3,4}$ las cuales se encuentran marcando un mismo lugar de la red NET_1; por lo que NET_1 se sincroniza internamente con dichas redes, y además, con la red $NET_{4,2}$. La sincronización entre ellas se da respecto al símbolo *a*.

La red $NET_{2,1}$ se sincroniza de manera interna con $NET_{4,1}$, quien al mismo tiempo se sincroniza externamente con $NET_{2,1}$, de acuerdo a la etiqueta *b*. También, $NET_{2,1}$ se sincroniza localmente con la red $NET_{3,1}$.

El mecanismo de sincronización es muy útil porque permite por un lado, controlar el comportamiento de redes subordinadas, establecer dependencia entre niveles de red, manifestar las formas en que las redes pueden interactuar; y por otro lado, es posible obtener modelos más compactos, al permitir que un conjunto de etiquetas sea asignado a una transición.

Al dar interpretación al modelo, el mecanismo de sincronización puede ser traducido como el mecanismo que controla el comportamiento de entidades, la colaboración entre ellas, la interacción que éstas tienen con el medio ambiente, la coordinación de actividades, la reservación simultánea de recursos, y otros aspectos más que cumplen con el principio de sincronización.

2.3. Definición formal de n-LNS

Un sistema de redes está formado por un conjunto de redes de diversos niveles; cada red se expresa como un *tipo* con un *marcado*.

2.3.1. Redes tipo

Definition 2 *Una red tipo es una tupla* $typenet_{i,k}=(G,\ TOKEN_{i,k},\ LABEL_{i,k},\ VAR_{i,k},\ \tau,\ \lambda,\ \pi)$ *para* $1 \leq i \leq n$, *donde:*

- G es una estructura de RP ordinaria. $G = (P, T, F)$ donde:
  - P es un conjunto finito no vacío de lugares
  - T es un conjunto finito no vacío de transiciones
  - F es una relación de flujo P x $T \cup T$ x P, tal que $P \cap T = \varnothing$.
- $TOKEN_{i,k}$ es un conjunto finito no vacío de redes tipo y símbolos permitidos dentro de los lugares de una red tipo k de nivel i:

$TOKEN_{i,k} \subseteq \{typenet_{i..j,k} \mid j = 1, .., n-i, k = 1, .., r\} \cup SYMBOLS$ donde:

$-n$ es el número de niveles de un sistema de red multi-nivel

$-r$ es el número de redes tipo diferentes permitidas dentro de los lugares de una red tipo de nivel i.

– $SYMBOLS$ es un conjunto finito de símbolos.

Así,

$TOKEN_{1,k} \subseteq \{typenet_{2,1}, ..., typenet_{n,k}\} \cup SYMBOLS$

$TOKEN_{2,k} \subseteq \{typenet_{3,1}, ..., typenet_{n,k}\} \cup SYMBOLS$

$$TOKEN_{3,k} \subseteq \{typenet_{4,1}, ..., typenet_{n,k}\} \cup SYMBOLS$$
$$\vdots$$
$$TOKEN_{n,k} \subseteq SYMBOLS$$

- $LABEL_{i,k}$ es un conjunto finito de etiquetas definidas para la red tipo k de nivel i.
- $VAR_{i,k} = \{x, y, \dots\}$ es un conjunto finito de variables definidas para la red tipo k de nivel i.

Type: $VAR_{i,k} \longrightarrow TOKEN_{i,k}$ es una función de asignación de tipos a las variables.
$Type(x) = \{typenet_{i,k} \mid typenet_{i,k} \in \tau(p)\}$ es el conjunto de tipos asociados a la variable x.

- τ: $P_{i,k} \rightarrow 2^{TOKENi,k}$ $\varnothing$ es una función de asignación de redes tipo y símbolos a los lugares.
- λ: $T_{i,k} \rightarrow 2^{LABi,k}$ $\varnothing$ es una función de asignación de etiquetas a las transiciones, donde:
  - Si $i = 1$ entonces $LAB_{1,k}$= $LABEL_{1,k} \cup \{\varepsilon\}$
  - Si $2 < i < n - 1$ entonces $LAB_{i,k}$= $(LABEL_{i,k} \cup \{\varepsilon\}) \times (LABEL_{i,k} \cup \{\varepsilon\}) \times (LABEL_{i,k} \cup \{\varepsilon\})$
  - Si $i = n$ entonces $LAB_{n,k}$= $(LABEL_{n,k} \cup \{\varepsilon\}) \times (LABEL_{n,k} \cup \{\varepsilon\})$
- π: $F_{i,k} \times LAB_{i,k} \rightarrow M_{VAR_{i,k} \cup SYMBOLS}$ es una función de asignación de pesos para cada arco respecto a las etiquetas de una transición; el peso es un multi-conjunto de variables y/o símbolos.

Si $i = n$, $VAR_{n,k}$= $\varnothing$, por lo tanto, $\pi : F_{n,k} \times LAB_{n,k} \rightarrow M_{SYMBOLS}$.
Si label $\notin \lambda(t)$, entonces $\pi((p, t), label) = \pi((t, p), label) = \varnothing$.

Una red tipo $typenet_{i,k}$ es una estructura de RP ordinaria con información adicional que declara y manipula los datos definidos en $TOKEN_{i,k}$ de acuerdo a las pre y post condiciones establecidas por la función π, y al etiquetado simbólico de transiciones especificado por λ, para la interacción entre redes.

La definición establece una jerarquía de n niveles. En cada nivel puede haber una o más redes tipo diferentes. Cada red tipo k de nivel i permite tipos definidos en $TOKEN_{i,k}$, que incluye tipos $TOKEN_{i+1,k}$, $TOKEN_{i+2,k}$, ..., $TOKEN_{n,k}$.

La función τ asigna a cada lugar de G un conjunto de redes *tipo* y/o *símbolos* que pertenecen al conjunto $TOKEN_{i,k}$, especificando que el lugar sólo puede tener como marcas *símbolos* y/o redes tipo del conjunto asignado.

La función λ asigna a cada transición de G un conjunto de etiquetas. Una etiqueta *label* puede tener asociado uno o varios tipos de sincronización dependiendo del nivel de red al que pertenece G. La ausencia de cualquier tipo de sincronización en la etiqueta se indica con el símbolo ε, implicando un paso autónomo. Así, si la red es de nivel uno, la etiqueta se representa con un sólo símbolo *label o* ε. Si la red representa el último nivel, la etiqueta es un par de la forma (*label, label*), donde el primer elemento indica *sincronización local*, y el segundo elemento *sincronización externa*; el par $(\varepsilon, \varepsilon)$ expresa que no hay sincronización

entre redes. Pero si la red es de nivel intermedio (2 hasta $n-1$), entonces la etiqueta es una tripleta (*label*, *label*, *label*), que indica los tres tipos de *sincronización: local*, *interna* y *externa*, respectivamente. Cuando la tripleta se denota (ε, ε, ε) significa que no hay ningún tipo de sincronización.

La función π establece las precondiciones y las poscondiciones de cada transición de G. Esta función asigna un peso a cada arco con respecto a una etiqueta (incluyendo ε). Determina la cantidad y el tipo de redes y/o símbolos que se necesitan dentro de los lugares de entrada para habilitar una transición, y la cantidad y el tipo de redes y/o símbolos que deben ser agregados a los lugares de salida. La función π sólo asigna un multi-conjunto de símbolos al peso de los arcos cuando la red tipo es de nivel n, por lo tanto, la red tipo es similar a una red de Petri coloreada.

Una red tipo no tiene restricción alguna sobre el peso de los arcos. Es posible especificar en el peso de los arcos de salida de una transición, símbolos y/o variables del tipo no incluido en los lugares de entrada a dicha transición, e inclusive omitir uno o varios de ellos. De esta manera, una transición puede:

- Eliminar redes y/o símbolos, como resultado de consumirlos y no agregarlos a un lugar de salida.
- Crear redes y/o símbolos, como resultado de producirlos en un lugar de salida cuando no han sido removidos.
- Clonar redes y/o símbolos que se obtienen de removerlos y sumar varias copias de estos.

A continuación se ejemplifican redes tipo de diferente nivel.

Example 3 *La figura 2.2 muestra una red tipo 1 de nivel 1,* $typenet_{1,1}$= *(G,* $TOKEN_{1,1}$, $LABEL_{1,1}$, $VAR_{1,1}$, τ, λ, π*), la cual se define como sigue: G es la estructura de red,* $TOKEN_{1,1}$ = {$typenet_{2,1}$, $typenet_{4,2}$}*, donde* $typenet_{2,1}$ *es la red tipo de la figura 2.3, y* $typenet_{4,2}$ *es la red tipo de la figura 2.5 inciso b).* $LABEL_{1,1}$ = {ε, *a, c*} *y* $VAR_{1,1}$ = {*x, y*}*, siendo x e y las variables de tipos* $typenet_{2,1}$ *y* $typenet_{4,2}$ *respectivamente. Las funciones* λ *y* τ *tienen las siguientes asignaciones:* $\lambda(t1)=\{\varepsilon\}$, $\lambda(t2)=\{a\}$, $\lambda(t3)=\{c\}$, $\tau(p1)=\{typenet_{2,1}, typenet_{4,2}\}$, $\tau(p2)=\tau(p4)=\{typenet_{2,1}\}$ *y* $\tau(p3)=\tau(p5)=\{typenet_{4,2}\}$*. Los pesos de los arcos están dados por la función* π*, y son:* $\pi((p1,t1),\varepsilon) = \pi((t3,p5),c^{\downarrow}) = x+y$, $\pi((t1,p2),\varepsilon) = \pi((p2,t2),a^{\downarrow}) = \pi((t2,p4),a^{\downarrow}) = x$, $\pi((t1,p3),\varepsilon) = \pi((p3,t3),c^{\downarrow}) = y$*. En la figura los pesos tienen la forma* [*label*$\rightarrow \pi(arco)$].

Example 4 *La figura 2.3 muestra una red tipo 1 de nivel 2,* $typenet_{2,1}$ = *(G,* $TOKEN_{2,1}$, $LABEL_{2,1}$, $VAR_{2,1}$, τ, λ, π*), G es la estructura de red,* $TOKEN_{2,1}$ = {$typenet_{3,1}$, $typenet_{4,1}$, *c1, c2*} *donde* $typenet_{3,1}$ *se representa en la figura 2.4 y* $typenet_{4,1}$ *en la figura 2.5 inciso a).* $LABEL_{2,1}$={*a, c*} *y* $VAR_{2,1}$ = {*x, y*}*, donde type(x) =* {$typenet_{3,1}$} *y type(y)=*{$typenet_{4,1}$}*. La función* τ *es:* $\tau(p1)=\{typenet_{3,1}, c1\}$, $\tau(p2)=\{typenet_{4,1}, c2\}$, $\tau(p3)=\{typenet_{3,1}, typenet_{4,1}, c2\}$ *y* $\tau(p4)=\{typenet_{3,1}, typenet_{4,1}, c1, c2\}$*. La función* λ *es:* $\lambda(t1)=\{(\varepsilon,a,a)\}$ *y* $\lambda(t2)=\{(\varepsilon,\varepsilon,c)\}$*. La función* π *es:* $\pi((p1,t1),(\varepsilon,a,a)) = x+c1$, $\pi((p2,t1),(\varepsilon,a,a)) = y+2c2$, $\pi((t1,p3),(\varepsilon,a,a)) = x+y+2c2$, $\pi((p3,t2),(\varepsilon,\varepsilon,c)) = x+y+c2$, $\pi((t2,p4),(\varepsilon,\varepsilon,c)) = x+y+c1+c2$.

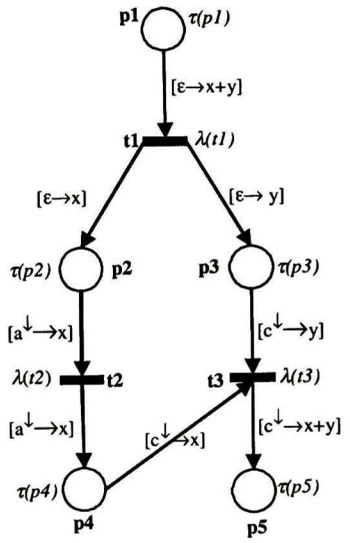


Figura 2.2: Ejemplo red-tipo de nivel 1

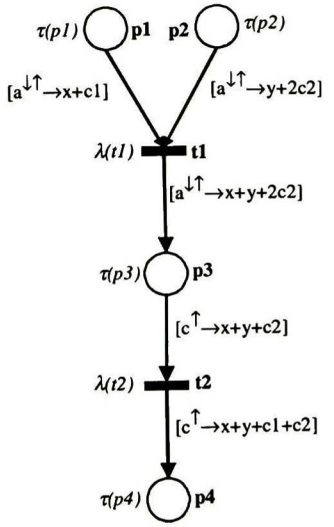


Figura 2.3: Ejemplo red-tipo de nivel 2

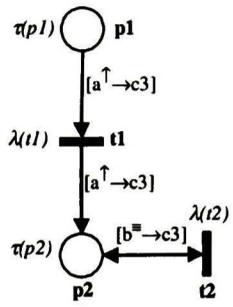


Figura 2.4: Ejemplo red-tipo de nivel 3

Example 5 *La figura 2.4 muestra una red tipo 1 de nivel 3,* $typenet_{3,1}$= *(G,* $TOKEN_{3,1}$, $LABEL_{3,1}$, $VAR_{3,1}$, τ, λ, π*), definida así: G es la estructura de red,* $TOKEN_{3,1}$= $\{typenet_{4,3}\}$, *donde* $typenet_{4,3}$ *se representa en la figura 2.5 inciso c).* $LABEL_{3,1}$ = $\{a, b\}$, $VAR_{3,1}$ = $\varnothing$. *La función* λ *esta dada por:* $\lambda(t1)=\{(\varepsilon, a, a)\}$, *y* $\lambda(t2)=\{(b, \varepsilon, \varepsilon)\}$; *la función* τ *asigna a los lugares:* $\tau(p1)=\tau(p2)=\{typenet_{4,3}\}$. *La función* π *define los pesos de los arcos así:* $\pi((p1, t1), (\varepsilon, a, a) = \pi((t1, p2), (\varepsilon, a, a)) = \pi((p2, t2), (b, \varepsilon, \varepsilon)) = \pi((t2, p2), (b, \varepsilon, \varepsilon)) = typenet_{4,3}$.

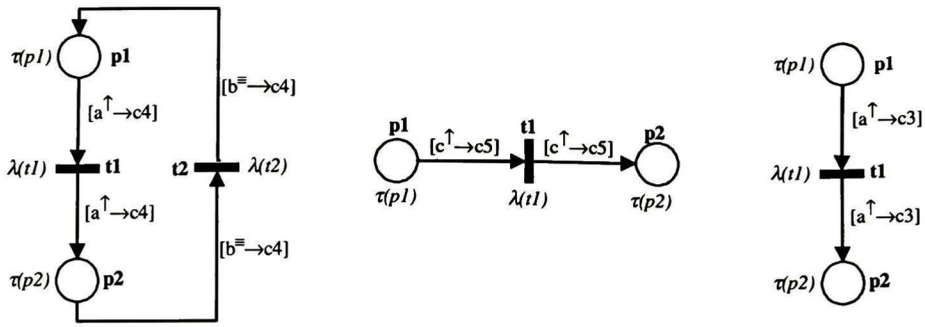
Example 6 *La figura 2.5 muestra tres redes tipo de nivel 4,* $typenet_{4,1}$, $typenet_{4,2}$, $typenet_{4,3}$. *Cada una tiene una estructura de red diferente. La red tipo* $typenet_{4,1}$, *define* $TOKEN_{4,1}=\{c4\}$, $LABEL_{4,1}=\{a, b\}$. *Las funciones* τ, λ *y* π *de la red* $typenet_{4,1}$ *están dadas así:* $\tau(p1)=\tau(p2)=\{c4\}$, $\lambda(t1)=\{(\varepsilon, a)\}$, $\lambda(t2)=\{(b, \varepsilon)\}$, $\pi((p1, t1), (\varepsilon, a)) = \pi((t1, p2), (\varepsilon, a)) = \pi((p2, t2), (b, \varepsilon))$ $\pi((t2, p1), (b, \varepsilon)) = c4$. *La red tipo* $typenet_{4,2}$, *define* $TOKEN_{4,2}=\{c5\}$ *y* $LABEL_{4,2}=\{c\}$. *Las funciones* τ, λ *y* π *de la red* $typenet_{4,2}$ *están dadas así:* $\tau(p1)=\tau(p2)=\{c5\}$, $\lambda(t1)=\{(\varepsilon, c)\}$, $\pi((p1, t1), (\varepsilon, c)) = \pi((t1, p2), (\varepsilon, c)) = c5$. *La red tipo* $typenet_{4,3}$ *define* $TOKEN_{4,3}$ = $\{c3\}$ *y* $LABEL_{4,3}$ =$\{a\}$. *Las funciones* τ, λ *y* π *de la red* $typenet_{4,3}$ *están dadas así:* $\tau(p1)=\tau(p2)=\{c3\}$, $\lambda(t1)=\{(\varepsilon, a)\}$, $\pi((p1, t1), (\varepsilon, a)) = \pi((t1, p2), (\varepsilon, a)) = c3$. *El conjunto de variables en ambas redes tipo es vacío.*

2.3.2. Redes de nivel i

Una red k de nivel i es una tupla $NET_{i,k}$ = $(typenet_{i,k}, \mu_{i,k})$ | $1 \leq i \leq n$, $k = 1, .., r$, donde:

- $typenet_{i,k}$ es una red tipo k de nivel i.
- $\mu_{i,k} : P_{i,k} \rightarrow M_{NETS_{TOKENi,k} \cup SYMBOLS}$ es una función de marcado para la red k de nivel i.

$NETS_{TOKENi,k} \subseteq \{NET_{i+1,k}, NET_{i+2,k}, \ldots, NET_{n,k}\}$ es el conjunto de todas o algunas redes de nivel inferior con marcados diferentes que están como marcas de la red de nivel i.



a) $typenet_{4,1}$ b) $typenet_{4,2}$ c) $typenet_{4,3}$

Figura 2.5: Ejemplo de redes-tipo de nivel 4

Una red k de nivel i es una red tipo $typenet_{i,k}$ con un marcado $\mu_{i,k}$. El marcado es una función que asigna a cada lugar de G un multi-conjunto de redes tipo y un multi-conjunto de símbolos, cuyas redes tipo y símbolos están en el conjunto de tipos y símbolos asignados. Así, una red tipo k de nivel i se convierte en una red k de nivel i, cuando la función $\mu_{i,k}$ le asigna un marcado.

Si las redes tipo anteriormente ejemplificadas son inicializadas con un marcado, entonces se tendrían las siguientes redes: $NET_{1,1}$=($typenet_{1,1}$, $\mu_{1,1}$), $NET_{2,1}$=($typenet_{2,1}$, $\mu_{2,1}$), $NET_{3,1}$=($typenet_{3,1}$, $\mu_{3,1}$), $NET_{4,1}$=($typenet_{4,1}$, $\mu_{4,1}$), $NET_{4,2}$=($typenet_{4,2}$, $\mu_{4,2}$) y $NET_{4,3}$=($typenet_{4,3}$, $\mu_{4,3}$) donde: $\mu_{1,1}(p1) = \{NET_{2,1}, NET_{4,2}\}$, $\mu_{2,1}(p1) = \{NET_{3,1}, c1\}$, $\mu_{2,1}(p2) = \{NET_{4,1}, c2, c2\}$, $\mu_{3,1}(p1) = \{NET_{4,3}\}$, $\mu_{4,1}(p1) = \{c4\}$, $\mu_{4,2}(p1) = \{c5\}$ y $\mu_{4,3}(p1) = \{c3\}$

2.3.3. Sistema n-LNS

Un sistema de red a n niveles es una tupla $n{-}LNS = (NET_{i,k}, |\exists i = 1..n : n \in \aleph, k = 1, .., r)$, donde:

- $NET_{1,1}$ es una red de nivel 1. Representa el nivel más alto del sistema de red.
- $NET_{i,k} = \{NET_{1,1}, \ldots, NET_{i,r}\}$ es el conjunto de redes r de nivel i. Representa las redes de nivel intermedio y nivel más bajo.

Un sistema de red $n - LNS$ es una colección de todas las instancias de red de tipos diferentes definidas en todos los niveles.

2.3.4. Evolución del marcado en n-LNS

El comportamiento de una red se manifiesta con la evolución de su marcado. Para lograr dicha evolución se requiere especificar las reglas de habilitación y disparo de las transiciones de la red.

Para definir las condiciones de habilitación y disparo es necesario introducir la noción de ligado de variables y distinguir los tres tipos de sincronización posibles en un sistema multi-nivel.

Una función de ligado b sobre un conjunto de variables $VAR_{i,k}$ es una función $b : VAR_{i,k} \rightarrow NETS_{TOKEN\ i,k}$; $b(v)$ es una red de nivel inferior cuya red tipo es $Type(v)$.

Definition 7 $b_t : \{v|v \in \bigcup_{p \in \bullet t} E_{\pi((p,t),label)}\} \rightarrow \bigcup_{p \in \bullet t} NETS_{TOKENi}$. *una función que asigna a cada variable definida en el peso de los arcos de entrada a la transición t, con respecto a la etiqueta label, redes del tipo asignado al conjunto de lugares de entrada a t.* $E_{\pi((p,t),label)}$ *denota los elementos sin repetición del multiconjunto* $\pi((p,t), label)$.

Definition 8 *m<b> denota el multi-conjunto de redes que resulta de evaluar un multi-conjunto de variables* $m = \{2x, 3y\}$ *en un ligado b.*

Definition 9 *Sincronización local. Es un tipo de sincronización que requiere la habilitación simultánea de transiciones de otras redes del mismo nivel, de nivel inferior y/o superior localizadas en el mismo lugar. La sincronización se realiza respecto a un mismo símbolo.*

Definition 10 *Sincronización interna. Es un tipo de sincronización que requiere la habilitación simultánea de transiciones con respecto a una mismo símbolo, donde la sincronización se realiza únicamente con redes de nivel inferior. Por lo tanto, una transición de la red k de nivel i se sincroniza "hacia dentro" con una transición de la red que tiene como marca.*

Definition 11 *Sincronización externa. Es un tipo de sincronización que requiere la habilitación simultánea de transiciones, con respecto a un mismo símbolo, cuya sincronización se efectúa con una red de nivel superior. Así, la transición de una red k de nivel i se sincroniza "hacia fuera" con una transición de la red en la que reside.*

Regla de habilitación

Definition 12 *Una transición t de una red k de nivel i* $NET_{i,k}$ *está habilitada con respecto a una etiqueta* $lab \in \lambda(t)$ *si:*

1. Existe un ligado $b_t : VAR_t \longrightarrow NETS_{TOKENi,k}$,donde VAR_t es el conjunto de variables que aparecen en todos los arcos de entrada a t.

2. $\forall p \in \bullet t, \pi((p,t), lab) < b_t > \subseteq \mu_{i,k}(p)$. Esta condición establece que una transición t está habilitada respecto a una etiqueta *lab*, si existe un ligado $< b_t >$ de las variables inscritas en los arcos de entrada a t, y si en cada lugar de entrada a t hay tantas redes de nivel inferior y símbolos como lo especifica la función π en el ligado b_t.El ligado $< b_t >$ es omitido cuando el nivel de red es n, ya que no tiene variables en el peso de los arcos, por lo tanto, la condición es simplificada de la siguiente manera: $\forall p \in \bullet t, \pi((p,t), lab) \subseteq \mu_{i,k}(p)$.

3. Las condiciones de uno de los siguientes casos se cumplen:

   - *Caso autónomo.* Sí $lab = (\varepsilon, \varepsilon, \varepsilon)$ entonces el disparo de t no requiere sincronización.

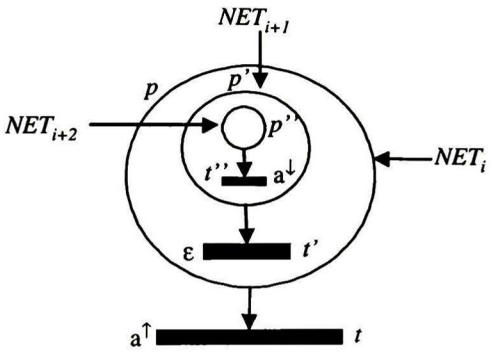


Figura 2.6: Sincronización interna no permitida

- *Caso síncrono.* Sí $lab \neq (\varepsilon, \varepsilon, \varepsilon)$ entonces se debe considerar una o una combinación de las siguientes situaciones:

i) Sí $lab = (l, \varepsilon, \varepsilon)$, requiere la habilitación simultánea de las transiciones etiquetadas con $l^{\equiv}$ que pertenecen a otras redes que se encuentran dentro del mismo lugar $p\prime$ de una red de nivel superior. El disparo de estás transiciones es también simultáneo, y las redes sincronizadas permanecen dentro del mismo lugar $p\prime$. Esto significa que no hay "transferencia"de redes, sólo evolucionan internamente.

ii) *ii)* Sí $lab = (\varepsilon, l, \varepsilon)$, requiere la habilitación simultánea de las transiciones etiquetadas con $l^{\uparrow}$ que pertenecen a otras redes de nivel inferior incluidas en los lugares de entrada a t. Al disparar estas transiciones las redes de nivel inferior y/o símbolos especificados en $\pi((p,t), lab) < b_t >$ son removidos.

iii) Sí $lab = (\varepsilon, \varepsilon, l)$, requiere la habilitación de al menos una de las transiciones $t\prime \in p'\bullet$ etiquetadas con $l^{\downarrow}$de la red de nivel superior donde la $NET_{i,k}$ está contenida. El disparo de t provoca la "transferencia"de $NET_{i,k}$ y símbolos indicados en $\pi((p\prime, t\prime), lab) < b_t >$.

Una etiqueta puede implicar una combinación de cualesquiera de estos incisos. Así, una etiqueta (l, l, l) indica que una transición debe ser sincronizada *localmente*, *internamente* y *externamente* respecto al símbolo $l^{\equiv\downarrow\uparrow}$

Estas situaciones aplican a todos los niveles de red excepto a los niveles de red 1 y n. En el caso de una red de nivel 1, sólo la *sincronización interna* es permitida, por lo tanto, las transiciones pueden ser etiquetadas con $l^{\downarrow}$ o ε, tipo de sincronización que se excluye cuando la red es de nivel n. Una red de nivel n sólo permite *sincronización local* y/o *externa* en sus transiciones, las cuales pueden ser etiquetadas con $l^{\equiv}$, $l^{\uparrow}$, $l^{\equiv\uparrow}$, ε.

Es importante señalar que la *sincronización interna* no tiene efecto cuando una red de nivel i se sincroniza internamente a través de una transición t, respecto a un mismo símbolo, con una transición t" de una red de nivel $i+2$, la cual está contenida en una red de nivel $i+1$, que se encuentra marcando $\bullet t$ de la red de nivel i, y cuyo símbolo no está asociado a la transición de salida t' Esto mismo ocurre para el caso de *sincronización externa.* La figura 2,6 ilustra esta idea.

La red NET_i no puede sincronizarse internamente con la red NET_{i+2}, a menos que la transición t' de la red NET_{i+1} esté etiquetada con $a^{\uparrow}$.

Regla de disparo

De manera análoga al mecanismo de disparo en una RP ordinaria en el sistema $n-LNS$, el disparo de una transición consume y produce marcas de red y/o símbolos; la cantidad y el tipo son especificados en el peso de los arcos de entrada y salida a la transición.

El cambio de marcado está dado por la expresión:

$$\forall p \in \bullet t \cup t\bullet \ , \ \mu'_{i,k}(p) = \mu_{i,k}(p) - \pi((p,t), label) < b_t > \cup \pi((t,p), label) < b_t >$$

donde:

- $\mu'_{i,k}$ es el marcado obtenido después de disparar la transición t.
- $\mu_{i,k}(p)$ es el marcado actual antes de disparar t.
- $\pi((p,t), label) < b_t >$ remueve redes y/o símbolos de todos los lugares de entrada (Pre).
- $\pi((t,p), label) < b_t >$ agrega redes y/o símbolos a todos los lugares de salida (Pos).

El ligado $< b_t >$ no es necesario en redes de nivel n.

Considerando las redes tipo expuestas en las figuras 2,2, 2,3, 2,4 *y* 2,5 con sus respectivos marcados asignados, podemos definir un sistema de red $4-LNS$, el cual consiste de una red de nivel uno $NET_{1,1}$, una red de nivel dos $NET_{2,1}$, una red de nivel 3 $NET_{3,1}$,y tres redes de nivel 4 $NET_{4,1}$, $NET_{4,2}$ y $NET_{4,3}$. Estas redes pueden interactuar según el etiquetado simbólico de sus transiciones de la siguiente manera:

La evolución del sistema inicia con la red $NET_{1,1}$, ya que es la única red con una transición inicialmente habilitada. El marcado dado y el ligado de las variables x, y satisfaciendo el peso indicado en el arco respecto al símbolo ε, causan la habilitación de la transición $t1$ de dicha red; su disparo no requiere ninguna condición, por lo tanto, las redes $NET_{2,1}$ y $NET_{4,2}$ que se encuentran marcando el lugar de entrada a $t1$, son removidas del lugar $p1$. Después del disparo de $t1$, la red $NET_{2,1}$ es depositada en el lugar $p2$ y la red $NET_{4,2}$ en el lugar $p3$. El marcado de la red $NET_{2,1}$ y $NET_{4,2}$ no cambia. Este es un ejemplo claro del caso autonómo.

El nuevo marcado de $NET_{1,1}$ conduce a la sincronización de transiciones. Ahora la transición $t2$ está habilitada respecto a la etiqueta $a^{\downarrow}$, la cual exige *sincronización interna* de la red $NET_{2,1}$ (red que tiene como marca). La red $NET_{2,1}$ habilitada a través de la transición $t1$ inscrita con $a^{\downarrow\uparrow}$, se sincroniza externamente con $NET_{1,1}$ e internamente con las redes $NET_{3,1}$ y $NET_{4,1}$, quienes a su vez se sincronizan externamente con $NET_{2,1}$ a través de su transición habilitada $t1$ etiquetada con $a^{\uparrow}$ La red $NET_{3,1}$ también se sincroniza internamente con la red $NET_{4,3}$ respecto al mismo símbolo. Así, el disparo simultáneo de tales transiciones implica remover de $NET_{1,1}$, la red $NET_{2,1}$ del lugar $p2$ y agregarla al lugar $p4$; remover $NET_{3,1}$, $c1$ del lugar $p1$, y $NET_{4,1}$, $2c2$ del lugar $p2$ y sumar ambas redes más dos símbolos $c2$ al lugar $p3$ de la red $NET_{2,1}$; remover $NET_{4,3}$ del lugar $p1$ y sumarlo al lugar $p2$ de la red $NET_{3,1}$; remover $c4$ del lugar $p1$ y agregarlo al lugar $p2$ de la red $NET_{4,1}$.Y por último, remover $c3$ del lugar $p1$ y agregarlo al lugar $p2$ de la red $NET_{4,3}$. La interacción de estas redes exhibe una combinación de tipos de sincronización: *interna* y *externa.*

El caso de *sincronización local*, se aprecia en las redes $NET_{3,1}$ y $NET_{4,1}$. Con el marcado actual, tanto una como otra red, tienen la transición $t2$ habilitada respecto a la etiqueta $b^{\equiv}$

Esta sincronización puede suceder en el lugar $p3$ o $p4$ de la red $NET_{2,1}$.El disparo simultáneo de ambas transiciones origina un cambio de marcado en la red $NET_{4,1}$ al remover el símbolo $c4$ del lugar $p2$ y regresarlo al lugar $p1$. El marcado de la red $NET_{3,1}$ permanece igual, el símbolo $c3$ queda en el mismo lugar $p2$.

Por último, las redes $NET_{2,1}$ y $NET_{1,1}$ son sincronizadas respecto al símbolo c. La red $NET_{2,1}$ se sincroniza externamente con la red $NET_{1,1}$ respecto a la etiqueta $c^{\uparrow}$ asociada a la transición $t2$, por el otro lado, $NET_{1,1}$ se sincroniza internamente con $NET_{2,1}$ respecto a la etiqueta $c^{\downarrow}$ asignada a la transición $t3$.

2.4. Conclusión

En este capítulo se presentó la definición formal del sistema $n-LNS$. Un formalismo de red a n niveles que permite la especificación de sistemas de eventos discretos que incluyen componentes móviles que evolucionan en ambientes conocidos.

El formalismo hace una extensión a n niveles de red, elimina algunas restricciones impuestas por el formalismo NS-3, mejora el mecanismo de sincronización, y amplia las capacidades de especificación al no limitar la cantidad de niveles de red, aprovechando dos de los beneficios inherentes al enfoque multi-nivel: *compacidad* y *modularidad*, lo que conduce a modelos más manejables.

La aplicación del formalismo $n-LNS$ como se verá en el siguiente capítulo está orientado a sistemas de manufactura por lotes, ya que generalmente los proyectos de la industria son grandes y complejos; la utilidad del formalismo es apreciada en el modelado de esta clase de sistemas.

Capítulo 3

Metodología de modelado multinivel de procesos por lotes

Resumen: Este capítulo propone una metodología de modelado para la especificación de sistemas de manufactura por lotes. La metodología combina el formalismo $n - LNS$, el enfoque orientado a objetos, el paradigma multi-agente, y el estándar ANSI/ISA-S88.01 de la industria de procesos. Esta combinación hace del método propuesto un instrumento práctico, útil y novedoso en el diseño y modelado de sistemas de manufactura por lotes.

3.1. Introducción

El objetivo principal de este capítulo es, ofrecer una solución al planteamiento ¿Cómo modelar un sistema de producción por lotes que reúna las siguientes características?

- Modularidad
- Alta compacidad en la abstracción del modelo
- Reducción en la complejidad del modelo
- Diseño y modelado orientado a objetos
- Representación de comportamiento multi-agente
- Modelado sin detalles de implementación
- Sencillez para modelar causalidad, concurrencia y sincronización de procesos
- Transparencia en el mecanismo de interacción
- Eliminación de restricciones en marcado
- Clara interpretabilidad en el modelo
- Modelado que adopte recomendaciones del estándar ISA-S88.01.

Este planteamiento es resuelto mediante una serie de pasos cuidadosamente definidos en la metodología que se propone. Esta estrategia se basa en el formalismo $n - LNS$, lo que permite describir los componentes de un sistema de manufactura de procesos por lotes en varios niveles de abstracción.

3.2. Estrategia general de modelado

3.2.1. Estructuración del sistema

Un sistema de producción por lotes, se compone de un conjunto de elementos, los cuales pueden variar de un ambiente a otro, no obstante, la existencia de 5 de ellos es indispensable. Estos son: *planta*, *lote*, *plan de procesos*, *receta* y *recursos*. La estrategia propuesta consiste en modelar cada uno de los elementos con una red, lo que permite describir inicialmente en forma individual cada componente. Los modelos correspondientes a cada elemento son:

- Modelo "*planta*". Este modelo describe la estructura geográfica de la planta que involucra la producción de lotes. La estructura representa un espacio físico o una agrupación lógica del ambiente, este puede ser una *celda de proceso*, un *área*, un *sitio* o una *empresa* [14], dependiendo de los requerimientos de cada organización. El modelo también describe la interconexión entre los espacios físicos o agrupaciones lógicas del ambiente.

- Modelo *"lote"*. Modela el material del lote como una entidad móvil con cierto comportamiento autónomo que tiene asignado un objetivo, el cual consiste en coordinar la ruta y el proceso de manufactura que debe seguir dentro de la planta. Para alcanzar dicho objetivo el lote requiere un plan de procesos y una o varias recetas.

- Modelo *"plan de procesos"*. Describe las trayectorias o rutas que guían al lote dentro de la planta, y las recetas que son aplicadas para transformar el material que lo compone en un producto final. La metodología redefine el plan de procesos tradicional caracterizándolo como *no-lineal* y *multi-receta*. Esto permite rutas alternativas para manufacturar un mismo producto y que un producto pueda ser terminado usando una o varias recetas.

- Modelo *"receta"* Describe la elaboración detallada de un producto, es decir, la secuencia de operaciones que serán aplicadas sobre el lote para manufacturar el producto deseado. Además, modela la reservación y liberación de recursos. Un modelo receta puede incluir la especificación total del producto o sólo una parte de ella. Por lo tanto, la especificación del producto puede estar repartida en varios modelos receta.

- Modelo *"recursos"* Modela la disponibilidad o estado de uso de los recursos. Los estados pueden ser: disponible, reservado, en uso, en mantenimiento, etc.

3.2.2. Interacción entre componentes

Un sistema de manufactura por lotes puede ser visto como un sistema distribuido en el cual numerosos componentes interactúan de diferentes formas dentro de un ambiente. Algunos componentes sean de hardware o software, pueden ejecutar varias acciones de manera conjunta, o bien de manera independiente. La naturaleza de este tipo de sistemas de manufactura hace que la mayoría de sus componentes sean entidades heterogéneas, algunas con la capacidad de desplazarse dentro de su entorno.

La coordinación entre los componentes de un sistema de manufactura es un medio indispensable para lograr un objetivo común. Esta se origina cuando se realiza un intercambio de información entre entidades que integran el sistema, cuando existe colaboración de tareas entre ellas, cuando se sincronizan acciones antes de iniciar una actividad, cuando se requiere resolver un conflicto y se recurre a la negociación, y en una serie de situaciones más.

Esta forma de visualizar un sistema de manufactura es similar a la forma en que se concibe un sistema multi-agente, esto es, las entidades móviles que autónomamente desempeñan una función pueden ser consideradas agentes, lo cual implica que tengan un comportamiento propio y la propiedad de movilidad.

Por lo tanto, la metodología propone modelar un sistema de manufactura por lotes como un sistema multi-agente, donde:

- El *lote* es una entidad móvil, heterogénea y autónoma, que puede ser representada por un agente.

- La *planta* representa el ambiente donde se desenvuelve el agente.

- El *plan de procesos* representa las metas trazadas que tiene el agente por alcanzar.

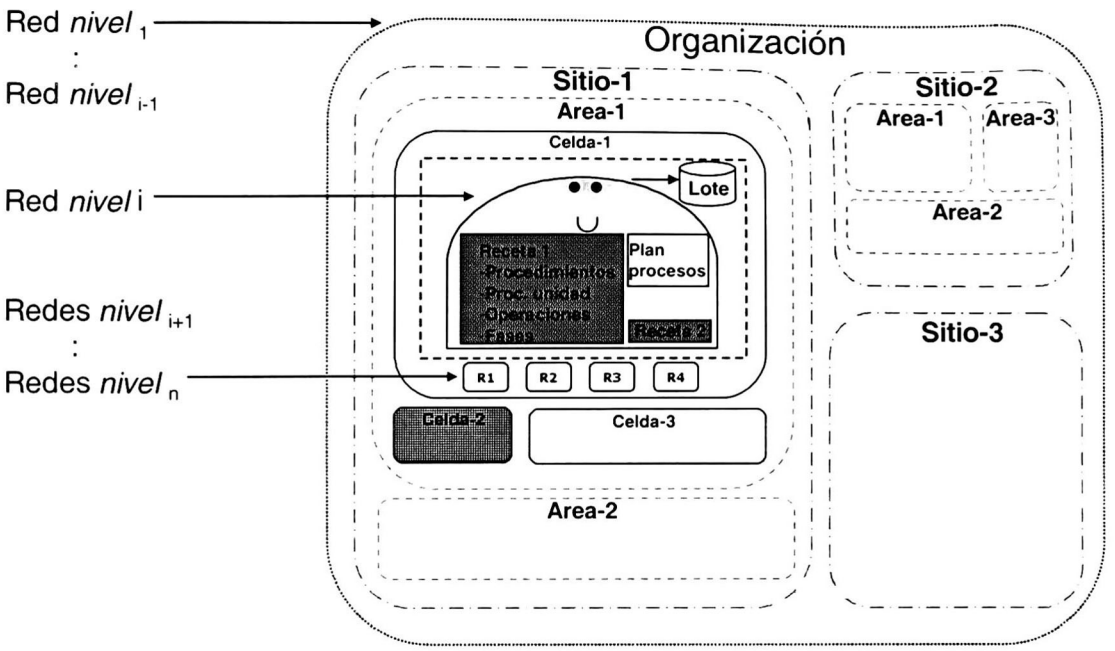


Figura 3.1: Estrategia de modelado multinivel

- La *receta* representa las tareas que debe desempeñar el agente para cumplir su objetivo
- Los *recursos* los elementos que necesita eventualmente el agente para realizar su función.

La coordinación entre los componentes se consigue mediante un mecanismo de interacción basado en sincronización de actividades. En este sentido,

- El lote puede interactuar con el ambiente al navegar dentro de un espacio de trabajo.
- El lote puede interactuar con el plan de procesos para determinar la ejecución de una receta.
- El lote puede interactuar con otro lote para mezclarse o bien para negociar sobre el uso de algún recurso.
- El lote y el plan de procesos pueden interactuar con la receta para controlar el inicio y fin de la secuencia de operaciones.
- La receta puede interactuar con los recursos para solicitar su uso.

Estas y otras tantas formas de interacción son las más importantes.

La estrategia general de modelado se ilustra de manera sencilla en la figura 3.1, donde se representa una jerarquía de contención de modelos.

La estrategia consiste en modelar los elementos del sistema por separado, estableciendo una jerarquía entre ellos, de acuerdo a un esquema de descomposición del ambiente de la planta y del proceso de la receta.

La figura es muy intuitiva en la descomposición del ambiente. En ella, se pueden apreciar cuatro estructuras: *organización*, *sitio*, *área* y *celda de proceso*, estructuras que la metodología propone para modelar el ambiente de la planta. Cada estructura pertenece a un nivel de red distinto. El nivel de red más alto es la *organización*, el nivel de red más bajo es la *celda de proceso*. De igual forma, el proceso de la receta cosidera cuatro niveles de descripción: *procedimientos*, *procedimiento por unidad*, *operaciones* y *fases*. El nivel más alto lo ocupan los *procedimientos* y el nivel más bajo las *fases*.

La estrategia integra estos niveles con un nivel más, para obtener un enfoque de modelado multinivel. En la figura 3.1 se muestra la jerarquía propuesta. Los niveles superiores, $i-1$, $i-2$, hasta 1, están reservados para modelar el ambiente de la planta; el nivel i se reserva para modelar el comportamiento general del lote; los niveles inferiores $i+1$, $i+2$, hasta n, están destinados al modelado de la receta, el plan de procesos y los recursos.

La estrategia es bastante flexible en la definición de niveles de red. El hecho de que proponga varios niveles de red no significa que no se pueda prescindir de cualquiera de ellos, particularmente si son niveles de refinamiento. No obstante, es necesario consentir un mínimo de niveles de red para garantizar la integridad multi-nivel del sistema. Al menos 3 niveles deben ser considerados, uno para el ambiente de la planta, uno para el lote, y otro para la receta, el plan de procesos y los recursos. Los niveles extras pueden omitirse si el sistema no es muy complejo.

La idea de la estrategia es que el lote esté acompañado por un agente durante el proceso de manufactura y que éste pueda navegar de una estructura a otra, llevando consigo un plan de procesos, y una o varias recetas como meta, haciendo uso de los recursos de la planta.

3.3. Metodología de modelado

El formalismo $n-LNS$, conceptos de diseño orientado a objetos, y teoría multi-agente se combinan para crear una metodología de modelado que oriente la especificación de sistemas para la coordinación de manufactura por lotes.

La metodología se apoya en el estándar industrial ISA-S88.01, y ha sido definida de manera ordenada en 11 pasos, los cuales se explican a continuación:

Paso 1. Identificar los elementos del sistema involucrados en el proceso de manufactura por lotes, como son: la *planta*, el *lote*, la especificación del producto o *receta*, el *plan de procesos* y los *recursos*.

Paso 2. Aplicar una técnica de descomposición (*top-down*) basada en principios de manufactura modular para extraer las entidades físicas y lógicas del sistema. La descomposición consiste en fragmentar el ambiente de la planta en pequeñas estructuras y desdoblar el proceso de manufactura de la receta.

Se proponen cuatro estructuras para modelar el ambiente de la planta: *organización*, *sitio*, *área* y *celda de proceso*; y para la receta propone cuatro niveles de descripción: *procedimientos*, *procedimiento por unidad*, *operaciones*, y *fases*.

Paso 3. Definir el número de niveles que tendrá el modelo del sistema, lo que conlleva a fijar el valor de n. Se sugiere un máximo de nueve niveles de red, de los cuales cuatro están designados a la concepción física del ambiente, uno a representar el comportamiento general del lote, y los últimos cuatro a la concepción del proceso. El nivel más alto lo ocupa la *organización*, el nivel medio el *lote* y el nivel más bajo la receta por *fases*.

Estos niveles son suficientes para modelar los elementos de un sistema de manufactura por lotes.

Obviamente, la cantidad de niveles de red que se definan, va depender de los elementos del sistema, de la descomposición física de la planta, de la subdivisión del proceso y del grado de abstracción del modelado.

La descomposición de la planta se hace en forma descendente pero la definición de niveles de red se hace en forma ascendente. Por lo tanto, en caso de tener un sólo nivel para representar el ambiente, se tomaría la estructura *celda de proceso*. Para modelar la receta con más de un nivel el orden sería *receta por fases*, *receta por operaciones*, *receta por procedimiento por unidad* y finalizaría con *receta por procedimientos*. Pero en la receta el orden es menos estricto, incluso todos los niveles pueden ser descritos en uno sólo si así se desea.

Paso 4. Determinar y ordenar las actividades o las tareas de acuerdo a las relaciones de precedencia establecidas.

Paso 5. Definir los modelos de acuerdo a $n - LNS$. Esto implica, construir una estructura de red para cada elemento del sistema, dibujar los nodos y la conexión entre ellos, expresando la causalidad y concurrencia de actividades conforme a las relaciones de precedencia definidas previamente.

Paso 6. Asignar a cada nodo del modelo una interpretación, según el contexto de cada módulo.

- Modelo "planta" Los *lugares* expresan un espacio físico, o agrupación lógica en la cual reside el lote. Las *transiciones* especifican la entrada y salida del lote a cualquiera de los espacios físicos o agrupaciones lógicas.

- Modelo "lote" Los *lugares* representan el estado general del lote en un punto del proceso, esto es, el plan de procesos que está en operación, la receta que utiliza y el estado de ejecución de ambos. Las *transiciones* representan el inicio o el fin del plan de procesos o de la recetas que se ejecutan, y el desplazamiento del lote a los espacios de trabajo permitidos.

- Modelo "plan de procesos" Los *lugares* representan la receta que está en ejecución, y el área física en la que es desempeñada. Las *transiciones* representan el inicio o fin de ejecución de una receta y el cambio de lugar del lote en la planta.

- Modelo "receta" Los *lugares* indican que un proceso, etapa, operación o acción del producto está en ejecución. También describe si un recurso está reservado o libre, y el lugar físico o lógico donde es aplicada la receta. Las *transiciones* indican el comienzo o término de una actividad, la reservación y liberación de un recurso, el desplazamiento del lote dentro de la planta, y la conexión entre recetas.

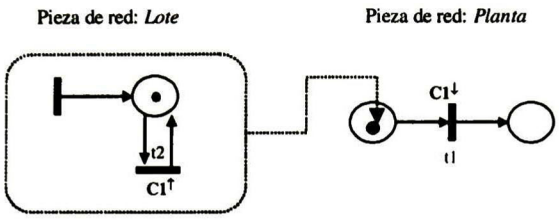


Figura 3.2: Sincronización: t1 y t2 con respecto a la etiqueta C1.

- Modelo "recursos" Los *lugares* representan el estado de uso de los recursos: reservado, libre, en mantenimiento, etc. Las *transiciones* cambian la disponibilidad del recurso.

Paso 7. Inscribir en cada arco del modelo las precondiciones y poscondiciones que controlen la ocurrencia de un evento del sistema. Estas son inscritas mediante la función de asignación de pesos de $n - LNS$. Algunas de las precondiciones y poscondiciones que deben ser establecidas en los modelos son utilizadas para:

- Expresar las restricciones de movilidad de uno o varios lotes entre los espacios de trabajo de la planta.
- Indicar la requisición de un plan de procesos y una receta para su ejecución.
- Solicitar el uso de recursos.
- Terminar una receta para iniciar otra.
- Determinar si un lote se divide o se mezcla.

Paso 8. Establecer las relaciones de interacción entre los modelos, si es que existen actividades que requieran coordinarse, entidades que necesiten cooperar entre sí para realizar una tarea en común, o solicitar la reservación de recursos.

El mecanismo de interacción definido en $n-LNS$ permite sincronizar los nodos transición de los modelos. Un ejemplo de interacción es cuando la red que representa el lote, considerada un agente, interactúa con su medio ambiente para su movilidad; para lo cual es necesario que la red lote tenga conocimiento de su entorno y que su desplazamiento entre celdas, sitios o áreas se coordine con la red que representa la geografía de la planta. Para lograr esto, se requiere etiquetar respecto a un mismo símbolo, una transición de la red lote con una transición de la red planta, especificando el tipo de sincronización entre ambas. La figura 3.2 muestra este ejemplo.

Paso 9. Especificar el marcado inicial en los modelos de red, de acuerdo al estado inicial del sistema modelado. La metodología propone el siguiente marcado para cada uno de los modelos:

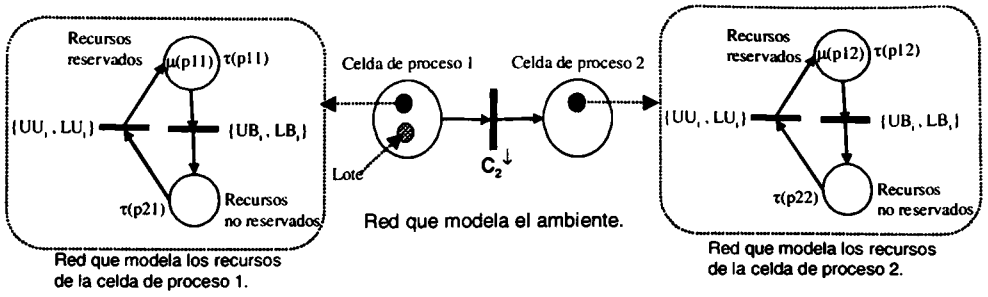


Figura 3.3: Marcado sugerido para la red que representa el ambiente de la planta

- El modelo *planta* tendrá dos tipos de marcas red. Un tipo representará el lote y el otro tipo representará los recursos. La marca de red que representa el lote es completamente dinámica a diferencia de la marca de recursos que permanece en el mismo lugar durante la evolución de la red. Esto porque el lote se mueve dentro de la planta y los recursos de un sistema de manufactura por lotes generalmente se caracterizan por ser fijos. Es conveniente definir un modelo de recursos para cada espacio de la planta, que contenga exclusivamente recursos locales. De esta manera, a cada lugar de la red que representa el ambiente de la planta le corresponde una marca de red recursos. Esta marca evoluciona por si misma cambiando el estado de disponibilidad de los recursos. En el caso de que hubiera recursos que se desplazaran de un espacio a otro, estos podrían ser parte de un modelo de recursos "móviles", el cual podría evolucionar de un lugar a otro.

La figura 3.3 muestra la asignación de los dos tipos de marcas. La marca red que representa el lote puede estar distribuida en cualquiera de los lugares de la red que modela la planta. En este caso se asigno a la celda de proceso 1, lugar donde inicia el proceso de manufactura del lote. La marca red que representa los recursos es asignada a cada celda de proceso.

- El modelo *lote* también tendrá 2 tipos de marcas-red. Una red representará la receta y la otra el plan de procesos. Las redes que representan la receta y el plan de procesos se conciben en el mismo nivel de red, aunque en estructura sean redes totalmente diferentes. En condiciones iniciales su asignación es disjunta. La marca-red receta se deposita en el lugar nombrado Recetas$_T$ la red plan de procesos se coloca en el lugar nombrado "Plan procesos" Ambos lugares forman parte de la red que modela el lote.

- El modelo *plan de procesos* no contempla marcas-red; su marcado está dado por símbolos (un sólo color). Inicia su evolución a partir del primer lugar de la red teniendo un símbolo como marca.

- El modelo *receta*, al igual que el plan de procesos, sólo maneja marcas de un sólo color. El marcado inicial figura con un símbolo en el primer lugar de la red.

- El modelo *recursos* es la red de más bajo nivel, por lo tanto los lugares sólo pueden contener símbolos. Cada símbolo identifica un recurso del ambiente. Todos los símbolos inicialmente se encuentran en estado *no reservado*, el cual es representado por un lugar.

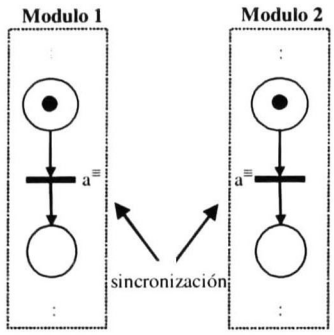


Figura 3.4: Condición de vivacidad trivial

Paso 10. Integrar los modelos constituyendo así el modelo global del sistema para su ejecución. La ejecución de la red está asociada a los 2 casos de disparo definidos en el formalismo. En el caso autónomo la transición del modelo puede ser disparada independientemente de cualquier otro modelo, sólo es necesario que se cumplan las precondiciones establecidas. Para el caso interactivo, la transición requiere el disparo simultáneo de las transiciones de otros modelos, etiquetadas respecto a un mismo símbolo. Pueden darse 3 tipos de sincronización; *local*, *interna* y *externa*.

Paso 11. Aplicar un método de verificación para hacer corresponder el modelo con el sistema real. Para ello es importante considerar algunas propiedades, tales como vivacidad, acotamiento y reversibilidad, tres propiedades que generalmente deben ser verificadas en los sistemas de producción.

3.4. Propiedades de los módulos

La metodología no aborda el análisis de propiedades, sin embargo, con la noción de algunas técnicas de síntesis para redes de Petri se podrían establecer ciertas condiciones básicas que garanticen las propiedades de los modelos cuando incurren en un tipo de interacción.

De esta manera, una condición de vivacidad puede cumplirse si:

- Dos modulos elementales que pertenecen a redes distintas, los cuales comparten un paso de sincronización son vivos por si solos, y
- Además las transiciones involucradas en el paso de sincronización se encuentra etiquetadas respecto un mismo símbolo.

Esto daría como resultado que la unión de modulos, por la sincronización de sus transiciones, preservara también la propiedad de vivacidad. La figura 3.4 ilustra un caso de vivacidad trivial

Desde luego, formalizar las condiciones para analizar las propiedades del formalismo $n - LNS$, quedaría como trabajo futuro.En el siguiente capítulo se presentan dos casos de estudio de manufactura por lotes que implementan la metodología de modelado descrita arriba.

3.5. Conclusión

La complejidad de los sistemas de manufactura por lotes hace que el modelado sea una tarea difícil. De ahí que la metodología tenga como principal objetivo facilitar esta tarea, definiendo la especificación del sistema en varios niveles de red. La definición de niveles que propone es flexible, libre pero limitada; el grado de abstracción que se decida tener en el modelo va depender del diseñador y del desarrollador de software. Pueden construirse modelos generales o detallados, jerárquicos o planos, modulares o no-modulares, según se desee, pero aprovechar la docilidad que ofrece la metodología sobre estos aspectos no debe ser descartada.

La metodología resulta ser un instrumento práctico, útil y novedoso en el diseño y modelado de sistemas de manufactura por lotes. Permite fácil modificación y extensibilidad en el modelado.

Construir una metodología basada en el formalismo $n - LNS$ es sólo una manera de explotar su aplicación, pero éste puede ser empleado en otro tipo de sistemas o incluso en otras áreas de eventos discretos.

Capítulo 4

Modelos de sistemas de manufactura por lotes

Resumen: Este capítulo se presenta la aplicación de la metodología de modelado propuesta a través de dos casos de estudio, los cuales tratan con sistemas de producción por lotes de la industria química. El primer caso describe el funcionamiento de una planta química que opera en modo multi-lote. El segundo caso considera la misma planta utilizando una receta que incluye operaciones de separación y mezcla.

Como se ha visto en en el capítulo anterior, la metodología involucra varios aspectos que facilitan la construcción del modelo. Uno de ellos es el estándar S88.01, el cual es muy conocido y aceptado en la industria de procesos por lotes. De ahí que la terminología y los modelos de referencia que plantea hayan sido tomados en cuenta para acercar los modelos del sistema a la forma en que son organizados los componentes en plantas de manufactura real.

Desde el punto de vista físico, el estándar propone fragmentar una planta en pequeños módulos y organizarlos jerárquicamente. Desde el punto de vista del proceso, sugiere subdividirlo en procesos, etapas, operaciones y acciones. Además el estándar define 4 tipos de recetas para la descripción de requerimientos de manufactura. De esta manera, se tiene un enfoque modular, lo cual resulta ser muy congruente con el enfoque multi-nivel del formalismo $n - LNS$, otro aspecto considerado en la metodología.

Los casos de estudio son variantes de ejemplos extraídos de la literatura de sistemas de procesos por lotes[10][13].

4.1. Caso 1: Modelado con 3-LNS

4.1.1. Descripción del sistema

En este caso de estudio se considera una planta sencilla de procesos químicos que opera en modo multi-lote. La planta consiste de 4 celdas de proceso, las cuales se componen de una colección de recursos, tales como:

Celda de proceso 1

- 3 tanques de almacenamiento ST11, ST21, ST31
- 1 reactor R11
  1 condensador CD11
  1 mezclador con agitador M11
- 7 líneas de conexión ST11-R11, ST21-R11, ST31-M11, R11-M11, R11-CD11, CD11-R12, M11-R13

Celda de proceso 2

2 tanques de almacenamiento ST12, ST22
1 tanque de desbordamiento OT12
1 reactor R12
1 centrifuga CN12
6 líneas de conexión ST12-R12, ST22-R12, CN12-OT12, R12-CN12, CN12-R13, CN12-C14

Celda de proceso 3

1 tanque de almacenamiento ST13
1 reactor R13
1 calentador PH13
1 bomba P13
- 4 líneas de conexión ST13-PH13, PH13-P13, P13-R13, R13-C14

Celda de proceso 4

2 tanques de almacenamiento ST14, ST24
1 mezclador con agitador M14

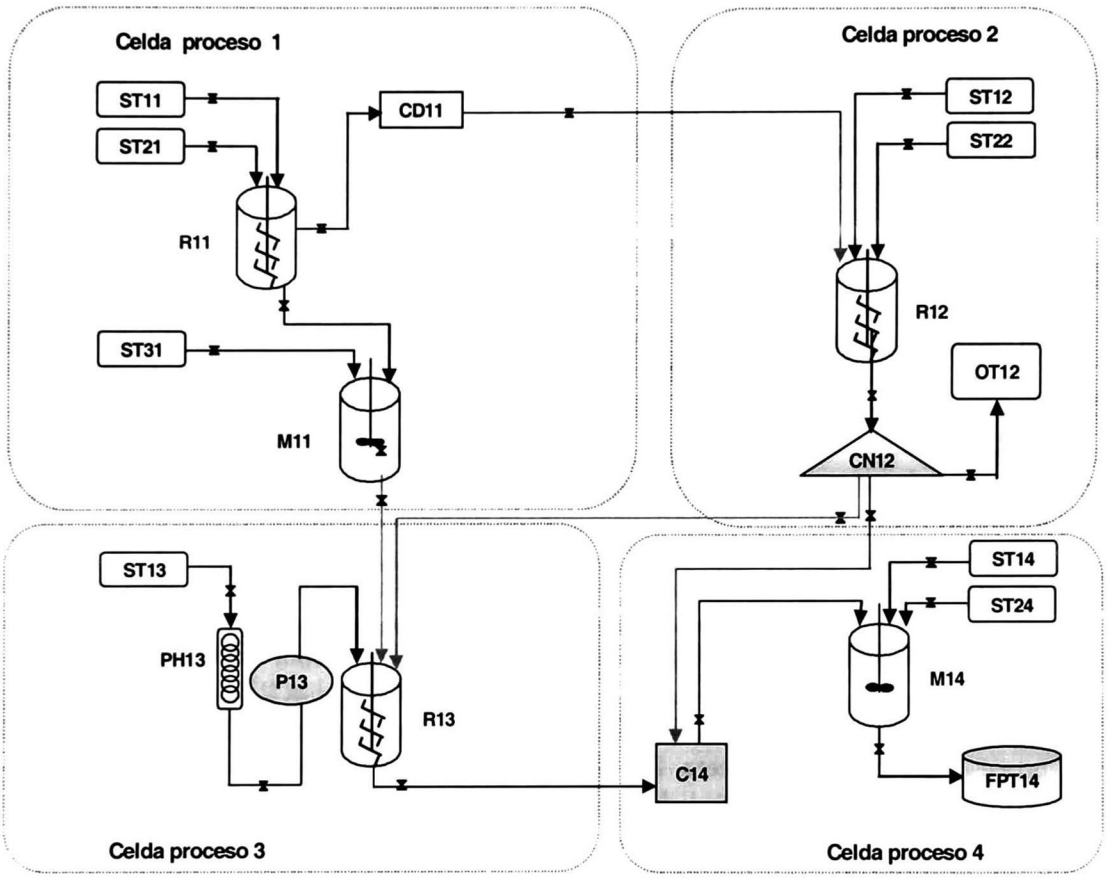


Figura 4.1: Planta multi-propósito de procesos químicos por lotes

1 dispositivo de enfriamiento C14
1 tanque de producto final FPT14
- 4 líneas de conexión ST14-M14, ST24-M14, C14-M14, M14-FPT14

Las celdas de proceso están interconectadas mediante líneas de conexión, las cuales se denotan por un par de recursos fuente-destino. En la planta pueden ser localizadas 5 líneas de conexión: CD11-R12, M11-R13, CN12-R13, CN12-C14 y R13-C14, donde:

CD11-R12 y M11-R13 conectan la celda de proceso 1 con la celda de proceso 2 y 3 respectivamente.

- CN12-R13 y CN12-C14 conectan la celda de proceso 2 con la celda de proceso 3 y 4 respectivamente.

- R13-C14 conecta la celda de proceso 3 con la celda de proceso 4.

De esta manera, cada lote se transfiere de una celda a otra para recibir diferentes tratamientos. El esquema de la planta se muestra en la figura 4.1.

4.1.2. Descripción del proceso

Aunque la planta está diseñada para manufacturar productos de manera concurrente, sólo un tipo de producto será descrito en este caso de estudio. SULF es el nombre del producto resultante. Sus ingredientes, las reacciones químicas que sufre y los pasos que implica su transformación son meramente ilustrativos.

El producto SULF tiene asociada una especificación de requerimientos de manufactura, la cual es descrita a manera de receta. La receta se encarga de definir la secuencia de operaciones que será aplicada al lote, así como la solicitud de recursos que se necesitan usar para completar dichas operaciones.

Respecto a la extensión que se hizo a la definición de plan de procesos, es posible asignar varias recetas a un lote. Este caso de estudio soporta un plan de procesos *multi-receta*, que designa dos de tres recetas para transformar la materia prima del lote SULF en un producto terminado; las recetas son receta 1 y receta 3, o bien, receta 2 y receta 3; éstas representan dos rutas alternativas para manufacturar el mismo producto, lo cual origina un plan de procesos *no lineal*. La receta 1 y 2 inician su ejecución en la celda de proceso 1, una vez terminada su ejecución en dicha celda, la receta 1 se ejecuta en la celda de proceso 2, y la receta 2 se ejecuta en la celda de proceso 3, concluyendo la manufactura del producto en la celda de proceso 4 con la receta 3.

El plan de procesos asignado describe el itinerario del lote de materia prima del producto SULF a lo largo del proceso, desde su inicio hasta su etapa final. A continuación se describen las recetas involucradas:

Receta 1:

1. Llenar los tanques ST11 y ST21, con H_2O y ácido tipo C, respectivamente.
2. Cargar y reaccionar H_2O y ácido C en el reactor R11 para formar el lote P.
3. Condensar el flujo del lote P en el dispositivo CD11 para obtener el líquido tipo Q.
4. Transferir el líquido del lote Q a la celda de proceso 2.

5. Llenar el tanque ST12 con la sustancia Z y ST22 con carbonato.
6. Cargar y reaccionar la sustancia Z y carbonato en el reactor R12 con el líquido tipo Q para obtener el lote L.
7. Separar el lote L en L1 y L2 en la centrifuga CN12
8. Drenar el lote L2 al tanque de sobre-flujo OT12.
9. Transferir L1 a la celda de proceso 4.

Receta 2:

1. Llenar los tanques ST11, ST21 y ST31 con H_2O, ácido D y sustancia O, respectivamente.
2. Cargar y reaccionar H_2O y ácido D en el reactor R11 para formar el lote U.
3. Mezclar el lote U y la sustancia O en el mezclador M11 para obtener el lote W.
4. Transferir el lote W a la celda de proceso 3 y limpiar el mezclador M11.
5. Llenar el tanque ST13 con la sustancia S.
6. Precalentar la sustancia S en el dispositivo PH13.
7. Bombear la sustancia S recalentada a través de la bomba P13 al reactor R13.
8. Reaccionar el lote W y la sustancia S para formar el lote L1
9. Transferir el lote L1 a la celda de proceso 4.

Receta 3:

1. Llenar el tanque ST14 con sal.
2. Enfriar el lote L1 en el dispositivo de enfriamiento C14.
3. Mezclar la sal y el lote L en el mezclador M14 para dar como resultado SULF
4. Almacenar el producto SULF en el tanque de producto final.

Con la información descrita anteriormente es posible modelar los elementos principales de un sistema de manufactura por lotes. La idea es construir un modelo en $3-LNS$ basado en la metodología de modelado para procesos por lotes. Esto implica abstraer el sistema a 3 niveles de red, donde el nivel superior será destinado a modelar la estructura de la planta; el nivel medio será asignado al lote; y el nivel inferior modelará la receta, el plan de procesos y los recursos. En este ejemplo, por convención se llamará a la red de nivel 1 red *ambiente*, a la red de nivel 2, red *agente*, y a las redes de nivel 3, redes *objeto*.

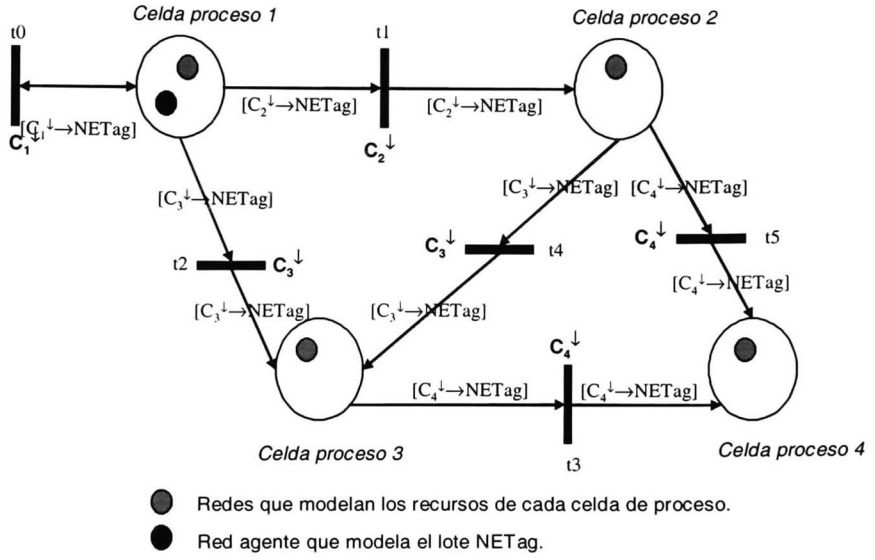


Figura 4.2: Modelo de la red ambiente

4.1.3. Modelo del sistema en n-LNS

Red ambiente

Como se puede apreciar en la figura 4.1 la planta ha sido estructurada en celdas de proceso. Estas celdas representan la agrupación física más alta dentro de la planta, por lo que sólo se requiere una red para modelar el ambiente físico del sistema. Las celdas de proceso son representadas por lugares en el modelo de red; así que hay tantos lugares en la red como celdas de proceso tiene la planta. La planta tiene líneas de conexión que comunican las celdas de proceso permitiendo realizar la transferencia del lote. En el modelo son representadas mediante la relación de flujo lugar-transición, transición-lugar.

La figura 4.2 ilustra la red de la planta, la cual tiene 4 lugares nombrados: *celda de proceso1*, *celda de proceso 2*, *celda de proceso 3* y *celda de proceso 4*, uno por cada celda de la planta; seis transiciones son dibujadas junto con sus respectivos arcos de entrada/salida, donde cada una modela una línea de conexión de la planta. De esta manera, el lugar *celda de proceso 1* se conecta con el lugar *celda de proceso 2* a través de la transición t1; esta conexión corresponde a la línea CD11-R12 de la planta. Las líneas de conexión restantes también son representadas en el modelo.

El lote es considerado un agente móvil que se desplaza entre las celdas de manufactura. Los lugares que forman la red ambiente pueden contener uno o varios agentes, donde cada agente es representado con una marca-red. En este caso la planta comienza su producción con un lote, por lo que un agente debe ser definido como marcado inicial; este agente acompañará al lote hasta que el material que lo compone sea transformado en el producto SULF. El modelo muestra la estancia de un agente en la celda de proceso 1; también otro tipo de marcas-red han sido asignadas a cada uno de los lugares; estas marcas representan los recursos propios de cada celda de proceso y se caracterizan por ser marcas-red estáticas, es decir, las

transiciones de la red ambiente no las consume/produce, ya que los recursos en un sistema de procesos por lotes son generalmente fijos.

Por ser el nivel de red más alto en el modelo del sistema, la red ambiente sólo tiene una forma de interacción. Esta interacción se efectúa con las redes contenidas en sus lugares, de ahí que derive en interacción interna.

La interacción se manifiesta en pasos sincronizados, donde un paso implica el disparo simultáneo de transiciones de la red ambiente, con cualquier otra red que sea parte de su marcado. El disparo está condicionado al sistema de etiquetado, el cual exige que las transiciones involucradas coincidan en la inscripción de un mismo símbolo. En el modelo esto queda expresado para toda transición, pues no existe el caso de un disparo autónomo.

Por supuesto, la movilidad del lote dentro de la planta implica una interacción con el ambiente para su acceso a celdas vecinas, para lo cual utiliza un mapa que contiene información total o parcial de la red que modela el ambiente. El acceso entre celdas es restringido, y es precisamente con la asignación de etiquetas como se consigue representar este control de acceso en los modelos de red. Por ejemplo, el lote no puede pasar de la celda 2 a la celda 3, sin que el agente tenga un plan de procesos y una receta definidos, y sin tener una reservación anticipada de la línea de conexión CN12-R13 para efectuar su trasferencia. Esto equivale en el modelo a sincronizar la red ambiente, con la red agente y con la red objeto que representa los recursos.

Red agente

El comportamiento de un agente está estrechamente relacionado con los objetivos que persigue. De acuerdo con la metodología, el lote es visto como un agente móvil que tiene un comportamiento autónomo. Su objetivo es transformar la materia prima que lo compone en un producto terminado. Para alcanzar su objetivo requiere la descripción de un plan de procesos, una o varias recetas, ciertos recursos y un mapa.

Estos elementos detallan el comportamiento general de un agente. En la figura 4.3 se presenta el modelo del agente. Este modelo describe una entidad que coordina la ruta y el proceso de un lote dentro de la planta. Esto queda representado con dos lugares nombrados Plan de procesos y Recetas respectivamente.

La red agente tiene como marcado inicial, una marca-red que representa el mapa, una marca-red que representa el plan de procesos, y tres redes marca más que representan las recetas; este marcado corresponde a la requisición de manufactura para el producto SULF descrito en la planta. Aunque el lugar nombrado plan de procesos puede albergar más de una marca, sólo una es asignada, ya que por definición a cada agente le corresponde uno y sólo un plan de procesos. En el caso de las recetas no hay restricción, por eso más de una marca es asignada; también por definición se permite manufacturación multi-receta, sin olividar que el número de recetas está sujeto a la concepción modular del diseñador. Un solo mapa de navegación es necesario.

Todo lote tiene que ser acompañado a lo largo del proceso por un agente, entonces un agente tiene la capacidad de realizar un producto a la vez, pero concluido éste, puede ser reutilizado para manufacturar el mismo producto u otro totalmente distinto. Esto no quiere decir que el modelo del sistema tenga que ser mono-agente, sino que la misma red-tipo puede ser usada con un marcado diferente para fabricar cualquier otro producto. De ahí que la red

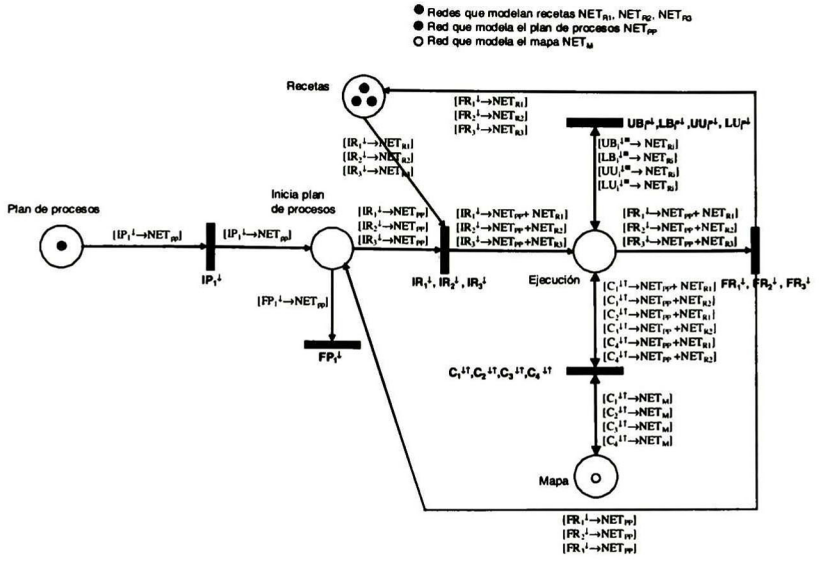


Figura 4.3: Modelo de la red agente

agente fuera concebida y diseñada de manera muy general. Esta ventaja se percibe cuando, se tienen varios productos que manufacturar dentro de la misma planta.

Por su nivel jerárquico, la red agente puede interactuar en tres formas distintas, y son: local, interna y externa. Esta última, aplica cuando hay sincronización entre redes del mismo o diferente nivel, ubicadas en un mismo lugar de red. Forma de interacción que se presenta cuando hay sincronización con las redes recursos localizadas en la red ambiente. La sincronización local también puede darse si existe un agente más con el cual cooperar en la misma celda de proceso, situación que no sucede en este caso de estudio.

En la red agente las etiquetas $C_1^{\downarrow\uparrow}$, $C_2^{\downarrow\uparrow}$, $C_3^{\downarrow\uparrow}$, $C_4^{\downarrow\uparrow}$ asignadas a la transición $t4$, permiten la interacción con la red ambiente a través de la red que modela el mapa; si el lote localizado en la celda de proceso 1 tiene que pasar a la celda de proceso 2, debe sincronizar la transición $t4$ de la red agente con la transición $t5$ de la red ambiente, respecto a la etiqueta $C_2^{\downarrow\uparrow}$ para lograr tal acción. Por otro lado, el agente interactúa de manera interna con la red plan de procesos a través de la transición $t1$ etiquetada con $IP_1^{\downarrow}$ para su inicio, y con la transición $t2$ respecto a $FP_1^{\downarrow}$ para su finalización. Además, interactúa con las redes receta, mediante la transición $t3$ con cualquier etiqueta del conjunto $\{IR_1^{\downarrow}, IR_2^{\downarrow}, IR_3^{\downarrow}\}$ para su inicio y con la transición $t5$ respecto al conjunto $\{FR_1^{\downarrow}, FR_2^{\downarrow}, FR_3^{\downarrow}\}$ para terminar su ejecución.

La red agente tiene la peculiaridad de eliminar marcas red mediante la transición $t2$. Esta situación se da cuando el plan de procesos ha sido concluido satisfactoriamente y es desechado por el agente. Con las recetas no sucede lo mismo, su ciclo es repetitivo, esto significa que una vez que finalizan "regresan" al lugar de donde fueron tomadas para ser reutilizadas.

Redes objeto

Plan de procesos

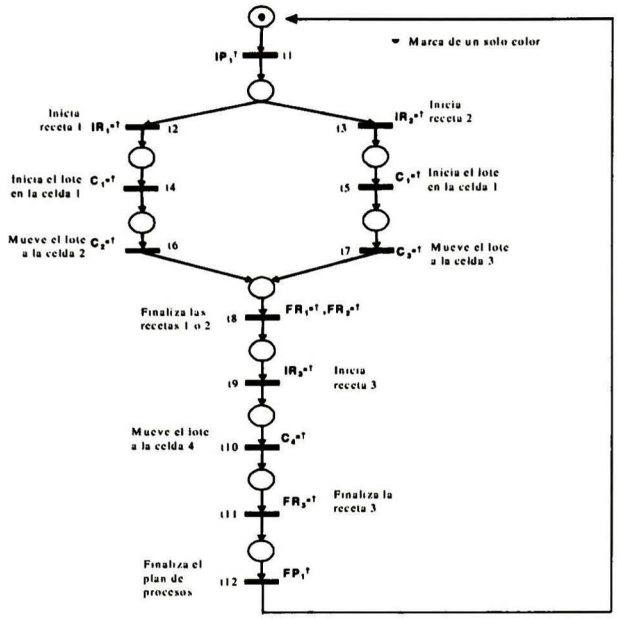


Figura 4.4: Modelo del plan de procesos

El lote debe ser guiado por un itinerario previamente definido, que le indique cuales celdas de la planta debe visitar y que recetas tendrá que aplicar en esas celdas de proceso.

El plan de procesos concentra esa información y la muestra en la red de la figura 4.4, la cual modela el plan de procesos para el producto SULF. En ella se puede observar una situación de toma de decisión para elegir la ruta del lote. De cualquier forma, sea cual sea la ruta a tomar, el resultado será el mismo producto, sólo que en una ruta se ejecuta la receta 1 en las celdas de proceso 1 y 2, y en la otra se ejecuta la receta 2 en equipos que pertenecen a las celdas de proceso 1 y 3. Esta bifurcación de rutas hace que el plan de procesos se torne no lineal.

Lo que se busca construyendo planes de proceso no lineales, es mostrar la flexibilidad del modelado para especificar concurrencia en las operaciones, o bien para considerar alternativas en caso de fallas.

El hecho de que la red pertenezca al nivel de red 3, el nivel más bajo de modelado, la restringe a establecer interacción externa y local, y a supeditar su marcado al conjunto de símbolos.

El disparo de todas sus transiciones está cuidadosamente regulado por la red agente. La interacción que mantiene con ella es de tipo externo. Con la redes receta establece interacción local, y se da cuando se encuentra junto con alguna red receta marcando uno de los lugares de la red agente. El marcado inicial de la red que describe el plan de procesos se compone de un símbolo.

Recetas

En un proceso por lotes, un producto se caracteriza por su receta, la cual define la secuencia de operaciones que serán desempeñadas sobre el lote. Existen diferentes niveles de abstracción para describir la especificación del producto de acuerdo al estándar ISA-S88.01, pero el nivel de receta maestra es el más apropiado porque hace referencia al equipo de

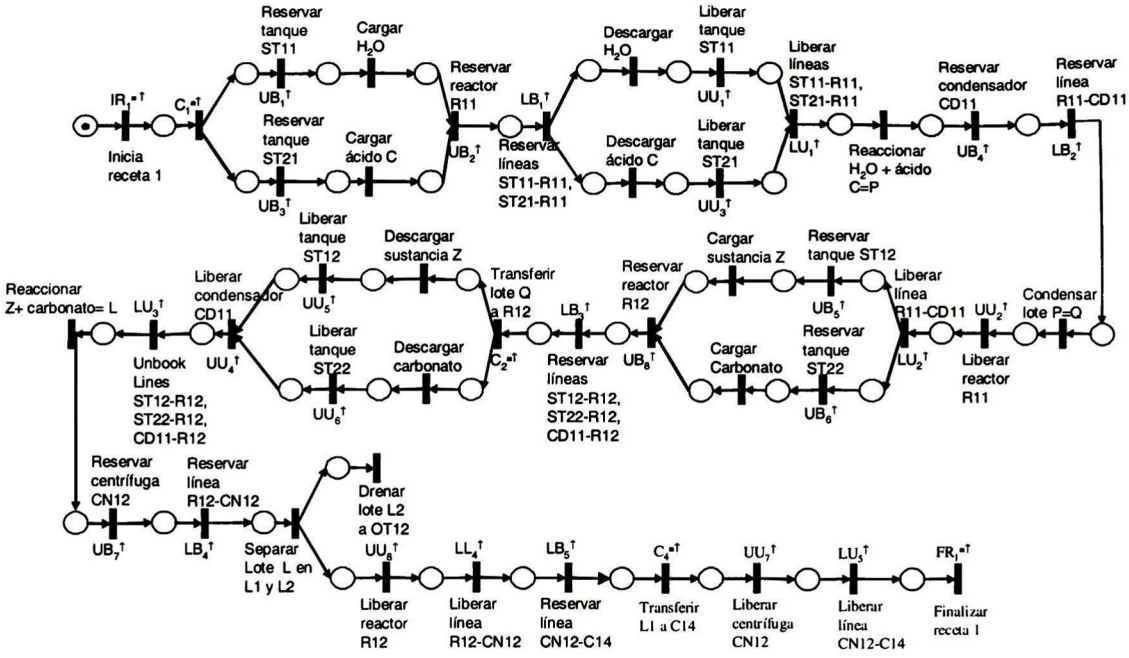


Figura 4.5: Red que modela la receta 1

operación de la planta, lo que permite obtener una receta detallada y completa, además de un modelo del producto más explicito. Esto se constata con el modelo de red mostrado en la figura 4.5.

Este modelo de red corresponde a la receta 1. La red es muy simple y su estructura ha sido construída secuencialmente con excepción de algunas actividades definidas en paralelo. La semántica es muy clara, los nodos *lugar* representan las etapas previas o posteriores a las operaciones y el estado de disposición de los recursos (reservado/no reservado); los nodos *transición* indican las acciones a ejecutar.

Al igual que el plan de procesos, la receta acompaña al lote en todo momento, desde el comienzo hasta su término. La receta es controlada por el agente y en cooperación con el plan de procesos guían la manufactura del producto. Por eso, ambos elementos forman parte del marcado de la red agente, al igual que el mapa del ambiente.

Las redes receta evolucionan entre sí de manera independiente, pero esto no las excluye de la interacción obligada que deben establecer con redes de más alto nivel, e incluso con redes de su propio nivel; tal es el caso de la red plan de procesos. La manera en como interactúan ya ha sido mencionada, y es a través de las etiquetas comunes entre nodos transición como pueden sincronizar sus acciones.

Un punto que debe ser destacado, es que bastaría una sola receta para producir SULF, pero esto implicaría tener un modelo de receta bastante extenso, Por lo que en principio, la especificación del producto fue dividida en 3 partes: receta 1, receta 2 y receta 3. El plan de procesos realiza la coordinación entre recetas, de tal forma que la especificación del producto sea ininterrumpida. Aún cuando particionar la especificación conlleve un aumento inherente de modelos de red y su interacción, el costo es mínimo si los modelos obtenidos son más

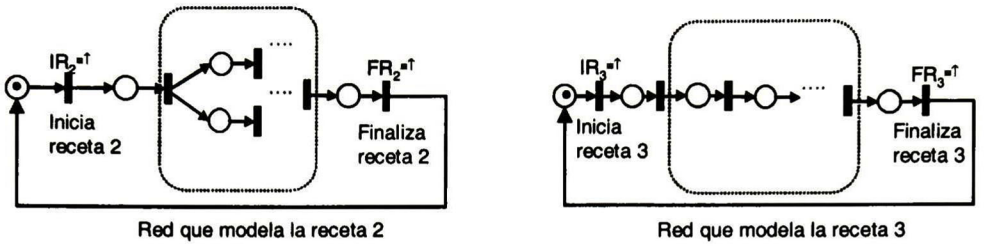


Figura 4.6: Redes que modelan parcialmente las recetas 2 y 3.

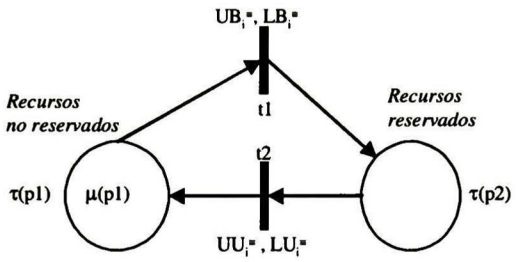


Figura 4.7: Modelo de recursos de la celda de proceso 1

compactos y manejables.

En esta sección tres modelos de red receta son presentados. Por razones de similitud con la receta 1, las recetas 2 y 3 no serán presentadas completamente, sino de manera parcial; estos modelos parciales son presentados en la figura 4.6.

Recursos

La manipulación del lote forzosamente requiere el uso de equipo. La planta considerada tiene una moderada colección de recursos, los cuales pueden ser fácilmente identificados y categorizados en dispositivos de *procesamiento* o *transporte*. Por cada celda de proceso se definió un modelo de red recursos, construido sobre dos estados de disponibilidad: *reservado* y *no-reservado*. Por lo tanto, dos lugares son suficientes en el modelo. La figura 4.7 muestra la red que modela la disponibilidad de los recursos de la celda de proceso 1.

Los recursos son modelados como un conjunto de símbolos. Por cada recurso en la planta un símbolo es agregado al marcado del modelo. La planta inicia operación con todos los recursos disponibles. La red como se muestra, representa su estado inicial. El marcado concentra todos los recursos en el lugar p1.

La red recursos evoluciona de manera sincronizada con el modelo agente a través de la receta, ya que es la responsable de dosificar el uso de recursos en la planta. Si una receta utiliza una pieza de equipo es porque previamente la reservó. En el modelo se puede observar un juego de etiquetas $\{UB_i^{=}, LB_i^{=}\}$ asignadas a la transición t1, y $\{UU_i^{=}, LU_i^{=}\}$ asignadas a t2. El primer juego reserva con la etiqueta $UB_i^{=}$ una unidad, y con $LB_i^{=}$ una línea de transporte. Contrariamente, el segundo par de etiquetas $UU_i^{=}$ y $LU_i^{=}$ liberan una unidad y una línea de transporte, respectivamente. El subíndice i denota un conjunto de etiquetas, donde cada etiqueta sirve para reservar o liberar un recurso de la planta.

Entonces, con 6 unidades de procesamiento y 7 líneas de conexión instaladas en la celda de proceso 1, la transición t1 en realidad tiene asignado el conjunto de etiquetas $\{UB_1^{\equiv}, UB_2^{\equiv}, ..., UB_6^{\equiv}, LB_1^{\equiv}, LB_2^{\equiv}, ..., LB_7^{\equiv}\}$, y la transición t2 el conjunto $\{UU_1^{\equiv}, UU_2^{\equiv}, ..., UU_6^{\equiv}, LU_1^{\equiv}, LU_2^{\equiv}, ..., LU_7^{\equiv}\}$.

La descomposición física de la planta puede variar en la cantidad de niveles de red a considerar en el modelo del sistema. En el caso anterior la planta modelada es muy sencilla, pero puede haber plantas que por su organización y tamaño requieran niveles de agrupación física más altos, como puede ser a nivel área o sitio, o bien dividir un procedimiento en procedimiento por unidad, operaciones o fases que faciliten su representación. Esto no es más que un refinamiento en el modelado del sistema, el cual puede ser representado con el formalismo $n - LNS$.

4.2. Caso 2: Modelando separación y mezcla de lotes con n-LNS

Este caso de estudio considera la misma planta de procesos químicos de la figura 4.1, pero utiliza una descripción de proceso que incluye operaciones de separación y mezcla de lotes.

4.2.1. Descripción del proceso

La planta debe manufacturar dos productos distintos: producto A y producto B. Ambos productos son el resultado de aplicar una secuencia de operaciones que serán descritas en forma de recetas.

Inicialmente la materia prima del producto A es tratada en la celda de proceso 1 y la materia prima del producto B en la celda de proceso 2. Concluido el tratamiento de cada lote en sus respectivas celdas, el lote de la celda 1 es transferido a la celda 3, mientras el lote de la celda 2 se divide en dos parte, donde una parte se transfiere a la celda 3 y la otra parte a la celda 4. Después de aplicar una serie de operaciones los dos lotes de la celda 3 se mezclan para obtener un lote intermedio a lo que será el producto A. El lote de la celda 4 también sufre un proceso de transformación que lo lleva a obtener el producto B. Cuando éste es almacenado en el tanque de producto final de la celda de proceso 4, el lote intermedio contenido en la celda de proceso 3 se transfiere a la celda de proceso 4 para su posterior almacenamiento. El producto A espera a que el producto B se extraiga de la planta para depositarse temporalmente en el tanque de producto final.

Los requerimientos de manufactura del producto A están definidos en la receta 1, y los del producto B en la receta 2 y 3. La receta 1 ha sido definida en operaciones y fases.

Receta 1: *Operaciones*

1. Llenar tanques ST11, ST21 y ST31 con concentrado B8, ácido tipo A y fructosa
2. Reaccionar sustancias concentrado B8 y ácido tipo A en el reactor R11 para obtener lote R
3. Mezclar lote R y fructosa en el mezclador M11 para obtener lote R-azucarado

4. Transferir lote R-azucarado a la celda de proceso 3
5. Llenar tanque ST13 con conservador D7
6. Preparar conservador D7 para ser mezclado
7. Reaccionar conservador D7 con R-azucarado y lote S1 para obtener producto A
8. Transferir lote producto A a celda de proceso 4

Receta 1: *Fases*

1. *Llenar tanques ST11, ST21 y ST31*
   a) Limpiar tanques
   b) Secar tanques
   c) Cargar tanques
2. *Reaccionar*
   a) Preparar reactor R11
   b) Cargar reactor con sustancias
   c) Calentar solución
   d) Vaciar lote a mezclador
3. *Mezclar*
   a) Limpiar mezclador
   b) Descargar lote R a M11
   c) Batir sustancias
4. *Transferir lote a celda 3*
   a) Vaciar lote al reactor R13
   b) Eliminar residuos del mezclador
5. *Llenar tanque ST13*
   a) Limpiar tanque
   b) Secar tanque
   c) Cargar tanque con conservador D7
6. *Preparar conservador D7*
   a) Precalentar conservador
   b) Bombear conservador a reactor R13

7. *Reaccionar*

   a) Agitar conservador D7 con R-azucarado

   b) Fusionar lote S1

8. *Transferir lote producto A*

   a) Enfriar y vaciar lote al mezclador M14

   b) Reposar lote

   c) Almacenar en tanque FPT14

Receta 2:

1. Llenar tanques ST12 y ST22 con sustancia tipo G y conservador D3
2. Reaccionar sustancia tipo G y conservador D3 en el reactor R12 para obtener lòte S
3. Separar lote S en S1 y S2-denso mediante la centrífuga CN12
4. Transferir lote S1 a la celda de proceso 3
5. Transferir lote S2-denso a la celda de proceso 4

Receta 3:

1. Llenar tanque ST14 con almidón
2. Mezclar lote S2-denso con almidón para obtener producto B en M14
3. Almacenar el producto B

Esta especificación del sistema es modelada con 4 niveles de red. El nivel 1 lo ocupa el modelo físico de la planta, organizado en cuatro celdas de proceso. En el nivel 2 se definen los modelos que representan el comportamiento de los lotes de los productos A y B. Tres modelos son requeridos: uno para modelar el comportamiento del lote del producto A antes y después de la operación de fusión; uno para modelar el comportamiento del lote del producto B antes de la operación de división y otro para después de ella. Los dos modelos para manufacturar el producto B son necesarios debido a que el material que lo compone es dividido en dos lotes. Ambos son tratados independientemente. Uno de ellos se fusiona con el lote del producto A, y el otro es transformado en el producto B.

Conforme a la metodología cada lote debe ser acompañado por un agente y cada agente requiere un plan de procesos, por lo tanto, se requieren tres planes de proceso, uno por cada lote. Estos son modelados en el nivel 4 al igual que las recetas 2 y 3 y los recursos. La receta 1 descrita por operaciones se modela en el nivel 3 y en el nivel 4 se describe por fases.

De esta manera, tenemos un modelo del sistema a 4 niveles (4-LNS).

4.2.2. Modelado del sistema en n-LNS

Red planta

La figura 4.8 ilustra la red que modela el ambiente de la planta; se compone de 4 celdas de proceso, cada una representada por un lugar. Las transiciones y los arcos definen la comunicación entre ellas.

La estructura de la red es similar a la red ambiente del caso anterior pero sufrió un ligero cambio, el cual no afecta la conexión entre celdas y con él se aprecia mejor la división y fusión de lotes. Ahora el marcado inicial incluye marcas-red de nivel 2 y nivel 4. Las redes de nivel 2 modelan el comportamiento de cada lote, y éstas son: NET1 y NET2. La marca-red NET1 es asignada a la celda de proceso 1 y NET2 a la celda de proceso 2. Las redes de nivel 4 representan los recursos propios de cada celda; éstas marcas no se mueven, razón por la cual no están inscritas en el peso de los arcos.

La red de la planta evoluciona de manera interna con las redes contenidas en su marcado. Una marca-red que representa el lote se mueve de un lugar a otro cuando alguno de los lotes tiene que ser transferido de una celda a otra. Obviamente, la movilidad del lote a las celdas indicadas en su plan de procesos queda restringida por el mecanismo de sincronización; esto es, cuando el lote localizado en la celda 1 es transferido a la celda 3, este debe solicitar al ambiente acceso para su movilidad, lo cual se consigue a través de la transición etiquetada con $C_3^{\cdot}$.

Las operaciones de separación y mezcla de lotes repercuten en el marcado de la red ambiente, tal es el caso considerado en la especificación de manufactura, donde la división de lotes toma lugar en la celda de proceso 2 y el lote representado por la red NET2 es dividido en dos partes. La primera parte del lote se transfiere a la celda de proceso 3 para posteriormente ser fusionado con el lote representado por la red NET1. Esto conlleva a eliminar la red NET2, la cual ha cumplido junto con el plan de procesos 2 y la receta 2 su misión en la celda correspondiente. En la figura se puede observar que la transición t5 elimina la red NET2 del marcado. Ahora la red NET1 se encarga de modelar el lote fusionado con el mismo plan de procesos y con la misma receta que le fue asignada inicialmente. La segunda parte del lote representada por la red NET3 es tratada en la celda de proceso 4. Esta red es creada cuando el lote se divide, incluye en su marcado un nuevo plan de procesos y una receta.

Redes lote

Partiendo de la idea de que un agente acompaña al lote hasta que la misión que tiene asignada especifica lo contrario, se requiere para este caso de estudio tres agentes para manufacturar los productos A y B. Estas redes se ilustran en la figura 4.9, 4.10 y 4.11.

El marcado inicial de estás redes se componen de tres tipos de marcas-red: una marca-red para el plan de procesos, otra marca-red para la receta y la última marca-red para el mapa.

El agente NET1 inicia su misión en la celda 1, después navega a la celda 3 para terminar en la celda 4 con el producto A. El agente NET2 comienza su misión en la celda 2 y finaliza en la celda 3 con un lote listo para fusionarse. El agente NET3 tiene como objetivo el proceso final del producto B, así que su misión inicia y concluye en la celda 4 con el producto esperado. Las tres redes interactúan con la red ambiente a través de la red mapa incluida en su marcado. La interacción interna la establecen cuando inician y finalizan su respectivo plan de procesos y receta.

Redes plan de procesos

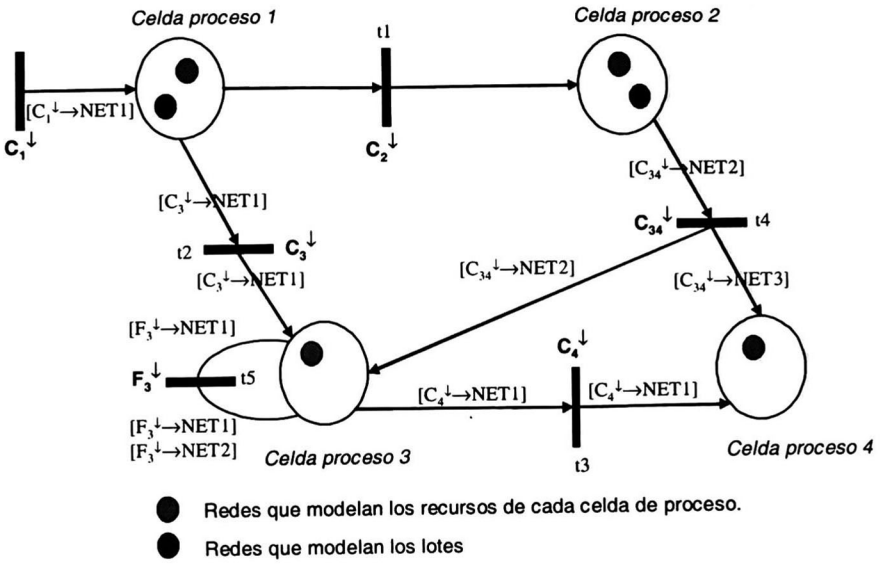


Figura 4.8: Red que modela el ambiente de la planta

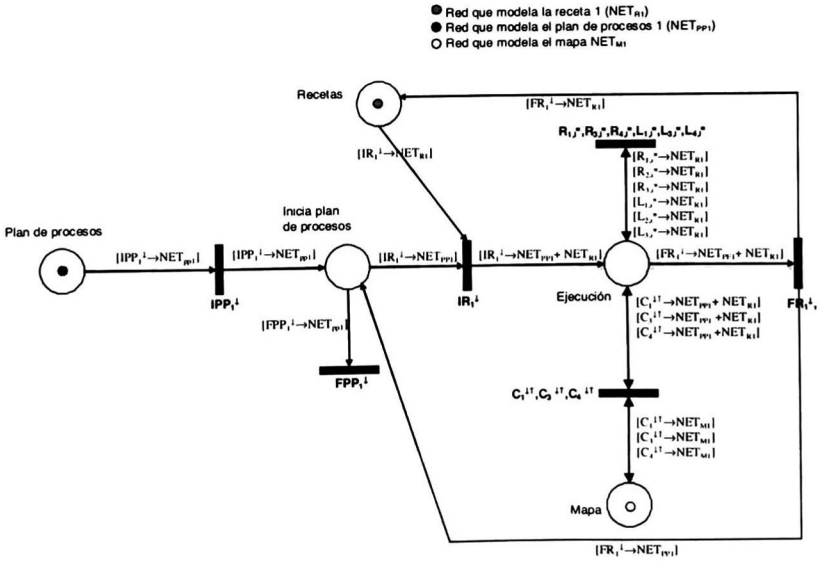


Figura 4.9: Red que modela el agente 1

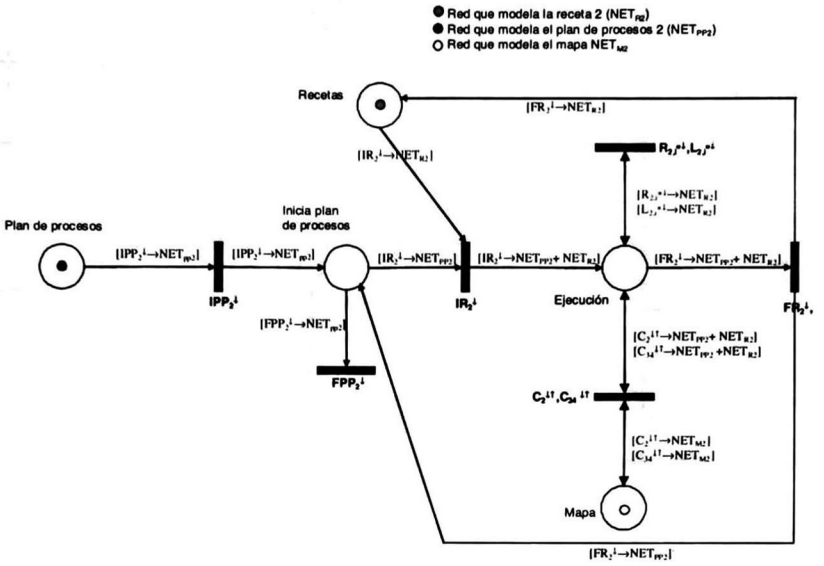


Figura 4.10: Red que modela el agente 2

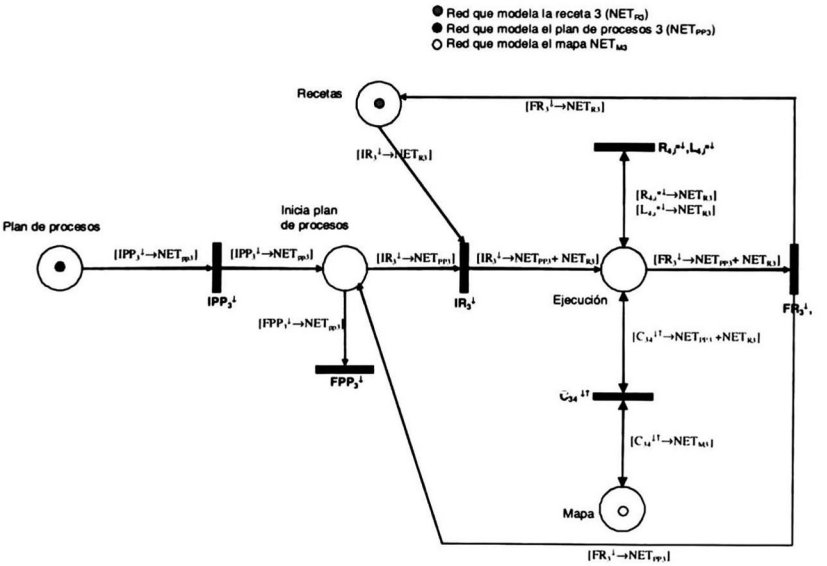


Figura 4.11: Red que modela el agente 3

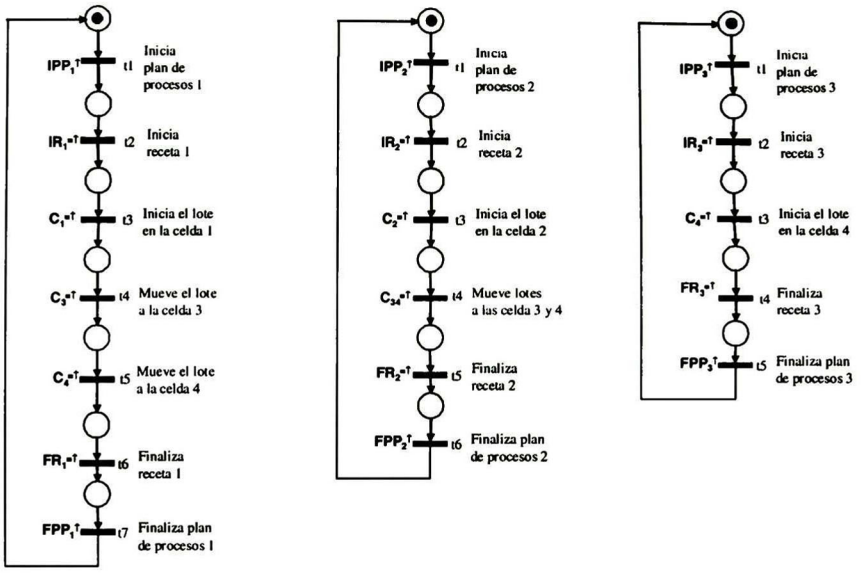


Figura 4.12: Redes que modelan los planes de proceso 1, 2 y 3

En este caso los planes de proceso son *lineales* y *mono-recetas*, esto es, ninguno utiliza más de una receta, y sólo tienen una secuencia de producción. Los planes de proceso se muestran en la figura 4.12.

Los planes de procesos 1, 2 y 3 describen las rutas que deben seguir y las recetas que deben aplicar las redes NET1, NET2 y NET3 respectivamente.

Redes recetas

La figura 4.13 muestra el modelo de red de la receta 1 (general). En él se han definido las operaciones principales que debe realizar el agente que coordina el lote del producto A. La red tiene como marcado inicial una marca-red que representa la secuencia detallada de las operaciones que la describen. La figura 4.14 ilustra la receta detallada por fases. La receta 1 controla la ejecución de la receta detallada a través de la sincronización de transiciones. La receta 1 interactúa en las tres formas posibles de sincronización. Localmente se sincroniza con el plan de procesos, de manera interna con la receta detallada, y externamente con la red que modela el lote del producto A.

Los modelos de la redes receta 2 y 3 se ilustran en las figuras 4.15 y 4.16 respectivamente. Un sólo símbolo forma el marcado inicial de estas redes.

Redes recursos

Los recursos son agrupados por celda de proceso y modelados por separado, resultando cuatro modelos de red por las cuatro celdas de proceso que constituyen la planta.

El estado de disponibilidad de los recursos es modelado con dos lugares. Un lugar para los recursos reservados y otro para los recursos que están disponibles.

Las redes que modelan los recursos ocupan el último nivel de modelado del sistema, por lo tanto, el marcado se compone únicamente de símbolos. Cada recurso es representado por un símbolo. Inicialmente todos los recursos se encuentran en estado no-reservado. En este caso la receta detallada controla la reservación y liberación de recursos en sincronización con la red agente NET1. Cuando la receta detallada requiere el uso de alguno de ellos y éstos están

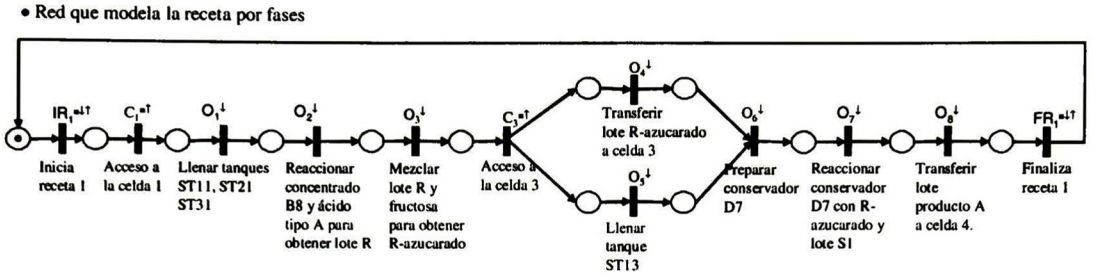


Figura 4.13: Red que modela la receta 1 (general)

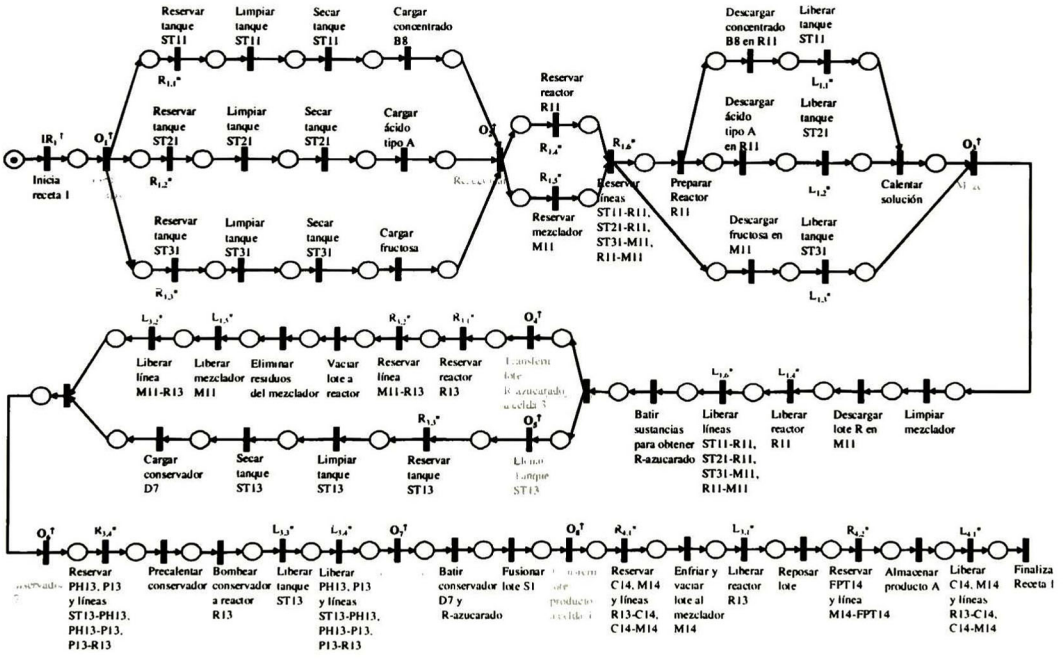


Figura 4.14: Red que modela la receta 1 (detallada)

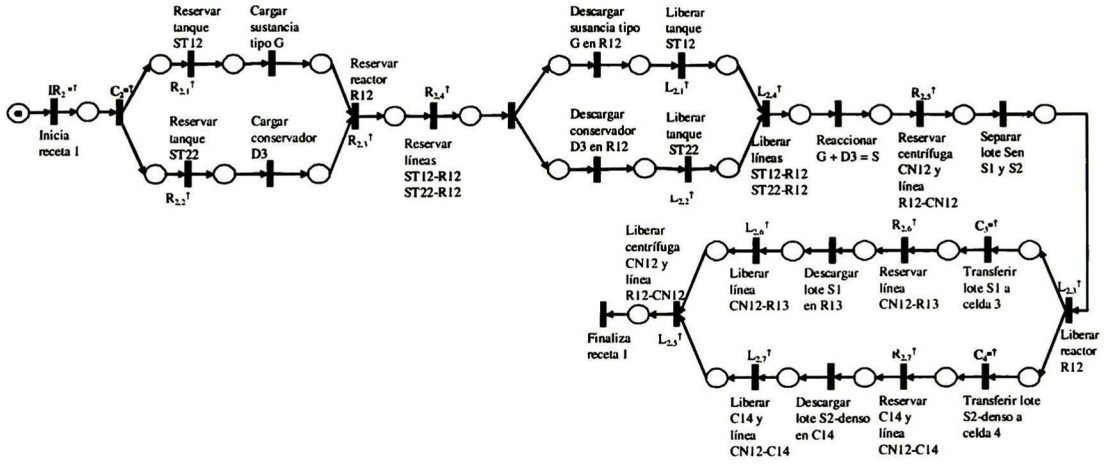


Figura 4.15: Red que modela la receta 2

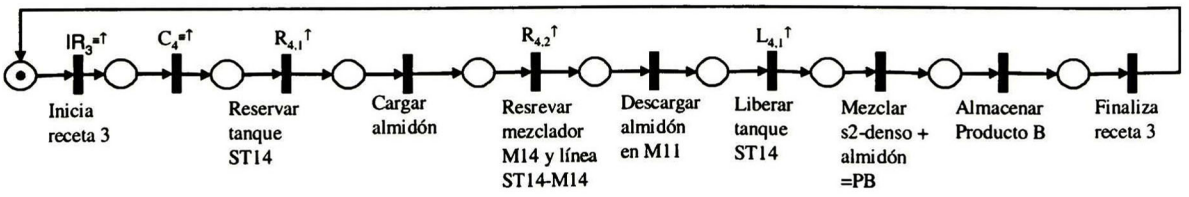


Figura 4.16: Red que modela la receta 3

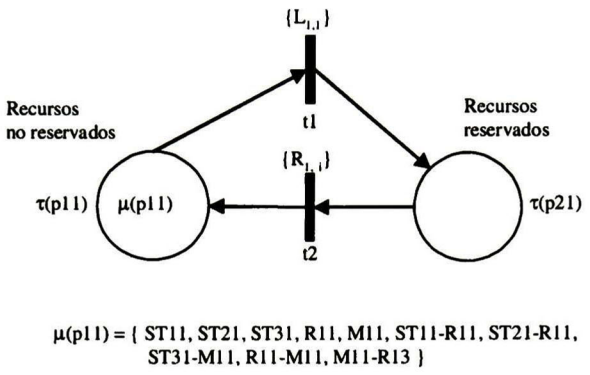


Figura 4.17: Red que modela los recursos de la celda de proceso 1

disponible, entonces el recurso cambia a estado reservado. La figura 4.17 muestra la red que modela los recursos de la celda de proceso 1. Por la similitud que existe entre los modelos de red que modelan los recursos de cada celda se han omitido los modelos de recursos de la celda 2, 3 y 4.

4.2.3. Conclusión

En este capítulo se ha presentado la aplicación de la metodología de modelado basada en el formalismo n-LNS para la especificación de sistemas de procesos por lotes.

Los dos casos de estudio modelados demuestran claramente las ventajas de modularidad y compacidad del enfoque multi-nivel propuesto. Ambos casos son ejemplos académicos variantes de casos de estudio tomados de la literatura de procesos por lotes.

Aunque el presente trabajo no tiene como objetivo el análisis de los modelos del sistema, es importante destacar que estos mantienen dos propiedades: vivacidad y acotamiento. Dos propiedades íntimamente relacionadas con los sistemas de producción.

Capítulo 5

Simulación de modelos multi-nivel

Resumen: En este capítulo se presenta la edición y simulación de modelos de sistemas de manufactura por lotes, usando la herramienta Renew. Los modelos están basados en el formalismo n-LNS y son adaptados al simulador para su implementación. Se incluye un caso de estudio del capítulo anterior.

5.1. Introducción

La simulación es un recurso frecuentemente utilizado en la validación de modelos. Ayuda a predecir el desempeño de sistemas nuevos. Es usada como herramienta de análisis para detectar problemas o evaluar cambios en sistemas ya existentes. De ahi que en esta tesis se pretenda brindar no solamente una metodología de modelado para sistemas de manufactura por lotes, sino asistir dicha metodología con una herramienta de simulación que permita observar el comportamiento de esta clase de sistemas y determinar algunas de sus propiedades. Una herramienta de simulación bien adaptada al enfoque multi-nivel propuesto es Renew. Renew es un simulador de redes de Petri de alto nivel, escrito en Java, que ofrece un esquema de modelado flexible basado en redes referencia. El formalismo de redes referencia provee la creación dinámica de instancias de red, las cuales se convierten en referencias de red cuando son parte del marcado de otra red, por lo que soporta el enfoque "redes dentro de redes"

Así, una red referencia es una red de Petri coloreada extendida, la cual inscribe sus elementos de red con expresiones en lenguaje Java. El mecanismo de comunicación entre las instancias de red se establece a través de canales síncronos.

5.2. Características de Renew

En esta sección se enuncian las características principales de la herramienta de simulación, Renew. Información más detallada sobre Renew se encuentra resumida en el ápendice A.

- Renew es una herramienta que puede ejecutarse en cualquier sistema operativo moderno sin hacer cambios, siempre y cuando una máquina virtual de Java esté disponible. Incluye OS/2, Apple Macintosh, varias versiones de Unix como Solaris o Linux, Windows 95/98/NT, y muchos más.
- Renew permite el modelado orientado a objetos, ya que las redes referencia conceptualizan el término red (clase) e instancias de red (objetos).
- Renew soporta canales síncronos como medio de comunicación entre redes, lo cual provee un mecanismo de abstracción poderoso.
- Renew utiliza Java como lenguaje de inscripción para las redes referencia.
- Renew puede hacer uso de cualquier clase de Java sin restricciones. Estas clases se acceden desde las redes referencia.
- El motor de simulación de Renew es capaz de simular concurrencia, donde las transiciones son ejecutadas con un disparo extendido con otras transiciones, si es requerido.
- La interfaz gráfica de usuario orientada a objetos es muy simple y puede ser extendida.
- Renew es una herramienta gratuita, equipada con archivos fuente, lo que permite libremente extender y mejorar sus algoritmos.
- Renew puede ser usada en aplicaciones que requieran:
    - Creación rápida de prototipos.

- Modelado basado en objetos.
- Modelado de "flujos de trabajo"
- Es una herramienta práctica para hacer experimentos con formalismos de red.

El uso de esta herramienta contribuye en gran medida a satisfacer los requisitos de una implementación que esté dentro de los puntos anteriores, ya que viene acompañada del código fuente, al cual se le pueden hacer cambios y mejoras. Sin embargo, hay que considerar que Renew:

- Actualmente cuenta con herramientas de análisis muy rudimentarias. La exploración de propiedades de una red esta confiada enteramente a la simulación, donde dinámicamente se puede visualizar su estado.
- No permite modificar el marcado actual de una red durante la simulación.
- El formalismo no soporta la noción de probabilidades de disparo o prioridades.
- Es una herramienta académica.

5.3. Edición de modelos

Renew requiere la edición previa de cada uno de los modelos que forman el sistema a simular. Para dibujar los modelos la herramienta ofrece un editor de red, el cual contiene una serie de elementos de dibujo con una interpretación semántica para el simulador.

De los dos casos de estudio modelados en el capítulo anterior, el primer caso es considerado para la simulación. Este caso de estudio consiste de una planta química de manufactura por lotes sencilla que opera en modo multi-lote. De él se derivan siete modelos de red, los cuales son jerarquizados en tres niveles de red. El primer nivel de red está representado por la red ambiente; el segundo nivel, por la red agente; el tercer y último nivel, lo representan las redes plan de procesos, receta 1, receta 2, receta 3 y recursos.

Los modelos obtenidos son editados en Renew. Como el modelado del sistema es modular, los modelos son dibujados en ventanas separadas.

Existen algunas diferencias entre los modelos editados y los modelos originales debido a que son adaptados al simulador, y el simulador obviamente está basado en un formalismo distinto al que fueron construidos. Tales diferencias serán resaltadas en cada uno de los modelos.

Los modelos son presentados de acuerdo a un orden de precedencia. Iniciando con el primer nivel de red.

5.3.1. Modelo del ambiente

La figura 5.1 ilustra la red *ambiente* que representa la estructura física de la planta, la cual está constituida por cuatro celdas de proceso, cada una representada por un lugar. Las conexiones entre las celdas están explícitamente representadas por arcos y transiciones. Como se puede observar la red difiere del modelo original en cuatro nodos de red: el lugar

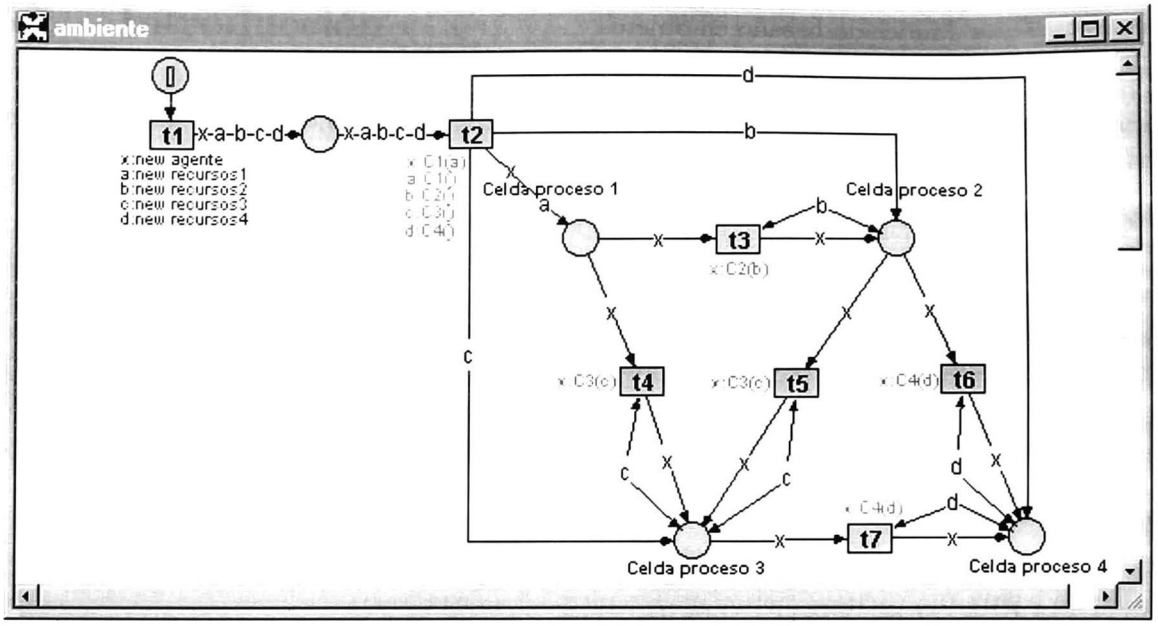


Figura 5.1: Red ambiente

marcado con la tupla vacía "[]", la transición t1 con varias inscripciones de creación, el lugar de salida de t1, y la transición t2 inscrita con múltiples canales. La inclusión de estos nodos en el modelo es para generar el marcado inicial de la red en tiempo de simulación. El marcado inicial del modelo consiste de 5 marcas de red, donde cada marca de red es una referencia a una instancia de red, que solamente puede ser creada cuando el simulador se ejecuta; por esta razón, al disparar la transición t1 se crean dichas instancias de red, las cuales son depositadas a manera de referencias en el lugar de salida de t1.

Todas las instancias de red generadas son diferentes: la inscripción *x:new agente* crea una instancia de la red *agente* en la variable x, las inscripciones *a:new recursos1*, *b:new recursos2*, *c:new recursos3* y *d:new recursos4* crean instancias de las redes recursos en las variables a, b, c y d respectivamente.

La transición t2 se encarga de hacer la distribución de las referencias de red de acuerdo al marcado inicial, el cual establece que el lugar "celda proceso 1" debe contener la red recursos1, el lugar çelda proceso 2" la red recursos2, el lugar "celda proceso 3" la red recursos3 y el lugar "celda proceso 4" la red recursos4. Así al disparar la transición t2, el peso de los arcos de salida cumplen tal especificación. Las inscripciones asociadas a la transición t2 sincronizan estas redes para su inicialización. El paso de parámetros permite utilizar las mismas instancias de red creadas en la red ambiente, en las redes recetas, que es donde se efectúa la reservación de los recursos.

En la red también se han dibujado *arcos de reserva*, etiquetados con las variables b, c, y d. Su función es reservar una marca durante el disparo de la transición, esto permite que la instancia de red asignada a la variable pueda ser enviada como parámetro a través del canal que le corresponde. Esta es una clara modificación de los modelos originales para adaptarlos

al uso de Renew.

Todas las transiciones de la red ambiente están etiquetadas, lo cual implica que la red no puede evolucionar de manera autónoma, sino que requiere sincronización con la red agente y con las redes recursos.

En la sincronización siempre debe existir un iniciador, es decir una red que explícitamente solicite la sincronización de transiciones. En este caso la red *ambiente* es quien lo hace, y para ello se han definido cuatro canales de comunicación: x:C1(), x:C2(), x:C3() y x:C4(). De esta manera, la red *agente* contenida en la red *ambiente* se sincroniza para solicitar su movilidad a cualquiera de las celdas de proceso vecinas, y las redes recursos para generar su marcado inicial.

Una vez generado el marcado inicial en la red, sólo se puede remover o producir la referencia de la red *agente*, esto se indica en el peso de los arcos al inscribir la variable x. Las referencias de red, *recursos1*, *recursos2*, *recursos3* y *recursos4* permanecen en el mismo lugar durante toda la evolución de la red.

5.3.2. Modelo del agente

El modelo de la figura 5.2 representa el comportamiento general del agente que requiere de un plan de procesos y de una o varias recetas para cumplir su objetivo. Esta especificación está dada en el marcado inicial de la red, el cual es generado en tiempo de simulación al disparar las transiciones t1 y t2. Ambas transiciones están equipadas con inscripciones de creación. La transición t1 con *x:new planprocesos*; la transición t2 con *y: new receta1*, *w:new receta2* y *z:new receta3*. El disparo de t1 produce una referencia de red en el lugar nombrado "Plan procesos" El disparo de t2 produce tres referencias de red que son depositadas en el lugar llamado "Recetas" La habilitación de las transiciones t1 y t2 está dada por tuplas vacías [].

Los arcos están etiquetados con variables. Las variables x, w, y, z tienen asignadas instancias de red distintas que pertenecen al mismo nivel de red (nivel 3). El peso de cualquier arco de la red remueve o produce solamente referencias de red, después de que es generado el marcado inicial.

La red *agente* se sincroniza con la red ambiente a través de los canales :C1(), :C2(), :C3(), :C4() asignados a las transiciones t8, t10, t9 y t11 respectivamente. Además, se sincroniza con las referencias de red contenidas en su marcado. Por lo tanto, la red *agente* se sincroniza con la red *planprocesos* mediante x:IP() para iniciar el plan, x:FP() para finalizar el plan, x:IR1() para iniciar la receta 1, x:IR2() para iniciar la receta 2, x:IR3() para iniciar la receta 3, x:C1(a) para direccionar a la celda 1, x:C2(b) para direccionar a la celda 2, x:C3(c) para direccionar a la celda 3, x:C4(d) para direccionar a la celda 4, y x:FR() para terminar cualquiera de las tres recetas.

La red *receta1*, usa los canales y:IR1() para iniciar la receta; y:C1(a) para ejecutar la receta a la celda 1 usando la red *recursos1* asignada a la variable a; y:C2(b) para ejecutar la receta a la celda 2 usando la red *recursos2* asignada a la variable b.

La red *receta2*, utiliza los canales w:IR2() para iniciar la receta; y:C1(a) para ejecutar la receta en la celda 1 usando la red *recursos1* asignada a la variable a; w:C3(c) para ejecutar la receta a la celda 3 usando la red *recursos3* asignada a la variable c.

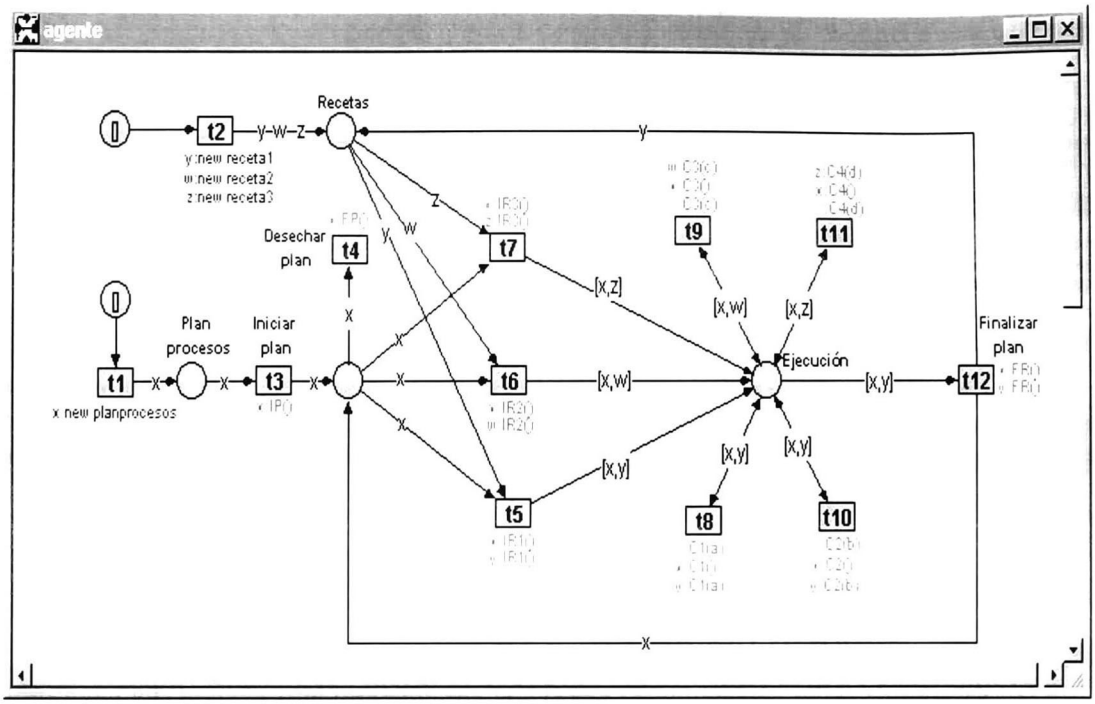


Figura 5.2: Red agente

La red *receta3*, usa los canales z:IR3() para iniciar la receta; z:C4(d) para ejecutar la receta a la celda 4 usando la red *recursos4* asignada a la variable d.

Y por último, la inscripción y:FR() finaliza la receta en ejecución.

La interacción de la red *agente* con la red *ambiente* está supeditada a una sola transición, tal transición está etiquetada con <C1,C2,C3,C4> (ver figura 4.3). No obstante, en la simulación del modelo es imposible sincronizar la red *agente* con la red *ambiente* como lo dicta el modelo original, usando una sola transición. Renew no permite que una transición tenga múltiples canales de comunicación que esperan ser invocados. Esto por supuesto limita la representación del modelado del sistema para su simulación. Sin embargo, el problema puede ser solucionado definiendo una transición para cada canal. En la figura se pueden apreciar claramente cuatro transiciones, t8, t9, t10, t11, donde cada una por separado sincroniza el acceso a la celda que le corresponde.

5.3.3. Modelo del plan de procesos

La figura 5.3 muestra la red *planprocesos*. Se puede observar que la red corresponde en estructura al modelo original, esto es, no requiere inscripciones de creación debido a que es una red de nivel tres, contiene un único color en su marcado, el cual es indicado con la tupla vacía [], por consiguiente no es necesario incluir nodos de inicialización.

La red *planprocesos* está controlada por la red *agente*, por lo tanto, ésta es quien inicia la sincronización. Así, cada canal inscrito en cualquiera de sus transiciones espera que la red *agente* solicite la sincronización con respecto al mismo canal. Entonces para sincronizar la

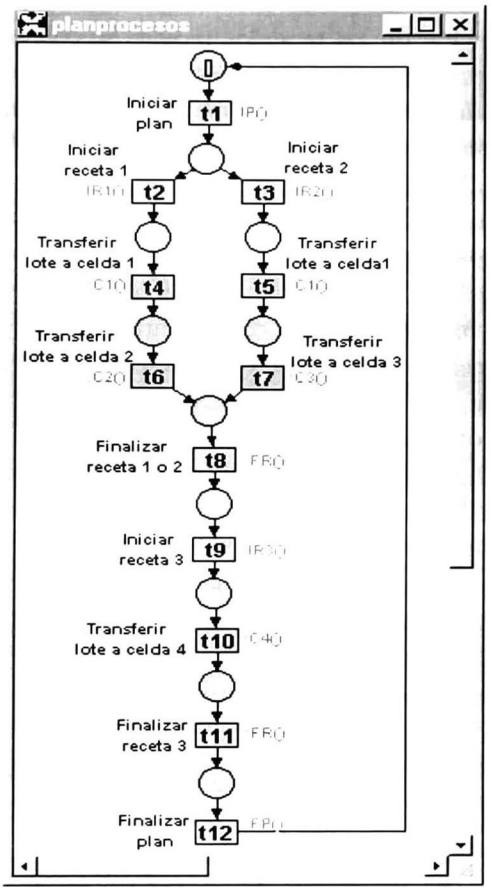


Figura 5.3: Red plan de procesos

red *planprocesos* con la red *agente*, se usan los canales, :IP() para iniciar el plan de procesos, :IR1(), :IR2() e IR3() para iniciar la receta que corresponda con el canal, :FR() para finalizar la ejecución de cualquiera de las tres recetas.

La red *planprocesos* se encarga de guiar al agente dentro de la planta, estableciendo las rutas que debe tomar, por lo tanto, requiere sincronizarse con la red *agente* y la red *ambiente*. Los canales :C1(), :C2(), :C3(), y :C4() lo permiten.

5.3.4. Modelos de recetas

En la simulación de los modelos recetas se consideran modelos parciales, debido al tamaño y a la similitud que existe entre las recetas. Las tres recetas mostradas en las figuras 5.4, 5.5, y 5.6 simulan la reservación de recursos, y únicamente la receta 1 describe parte de la secuencia de operaciones que deben ser aplicadas al lote para su transformación.

El marcado inicial de las recetas consiste de un sólo color, el cual es representado con la tupla vacía [], pero una vez iniciado el proceso de simulación el marcado cambia.

En el caso de la red *receta1*, se generan dos referencias de red, una hacia la red *recursos1* y otra hacia la red *recursos2*. Mientras la red *receta1* es ejecutada en la celda de proceso 1, la referencia de red *recursos1* reside como marca. Al ejecutarse en la celda de proceso 2, la referencia de red *recursos1* se desecha y la referencia de red *recursos2* se produce. De esta manera, se garantiza que la receta 1 exclusivamente podrá utilizar los recursos de la celda donde se encuentra operando.

A diferencia de los otros modelos, las referencias de red no son creadas dentro de la red, sino que se reciben como parámetros. En la figura 5.4, el disparo de t2 asigna a la variable x, la referencia de red *recursos1*, la cual evoluciona hasta que llega al disparo de t15, donde el lote debe ser transferido a la celda de proceso 2. En ese momento la referencia de red *recursos1* desaparece, y la referencia de la red *recursos2*, asignada a la variable y continúa hasta el termino de la receta 1. Entonces los canales :C1(x) y :C2(y) se encargan de sincronizar con la red *planprocesos* el movimiento del lote. El canal :C1(x) a la celda de proceso 1, el canal :C2(y) a la celda de proceso 2.

La red *receta1* se sincroniza localmente con la red *planprocesos*, y externamente con la red *agente* a través de los canales :IR1(), :FR(). El canal :IR1() incia la receta 1 y el canal :FR() finaliza la receta. También, la red receta1 se sincroniza con la red *recursos1*, mediante los canales x:RU1(), x:RU2(), x:RU3(), x:RL1(), x:LU1(), x:LU2(), x:LL1(), los cuales reservan y liberan las unidades de procesamiento y las líneas de conexión. Los canales y:RU4(), y:LU4 permiten que la red *receta1* se sincronice con la red *recursos2*.

La red *receta2* se ejecuta primero en la celda de proceso 1, y después en la celda de proceso 3, por lo tanto, requiere utilizar la red *recursos1* y la red *recursos3*. Ambas redes constituyen el marcado de la red en forma de referencias.

La referencia de red *recursos1* se recibe como argumento en la variable x, cuando se dispara la transición t2. La referencia de red *recursos3* se recibe como parámetro en la variable y, al disparar la transición t5. La referencia de red *recursos1*, prevalece como marca en la red mientras la receta 2 se aplica a la celda de proceso 1. En el momento que el lote se transfiere a la celda de proceso 3, la marca es sustituida por la red *recursos2*. Los canales :C1(x) y :C3(y) inscritos a dichas transiciones sincronizan el movimiento del lote.

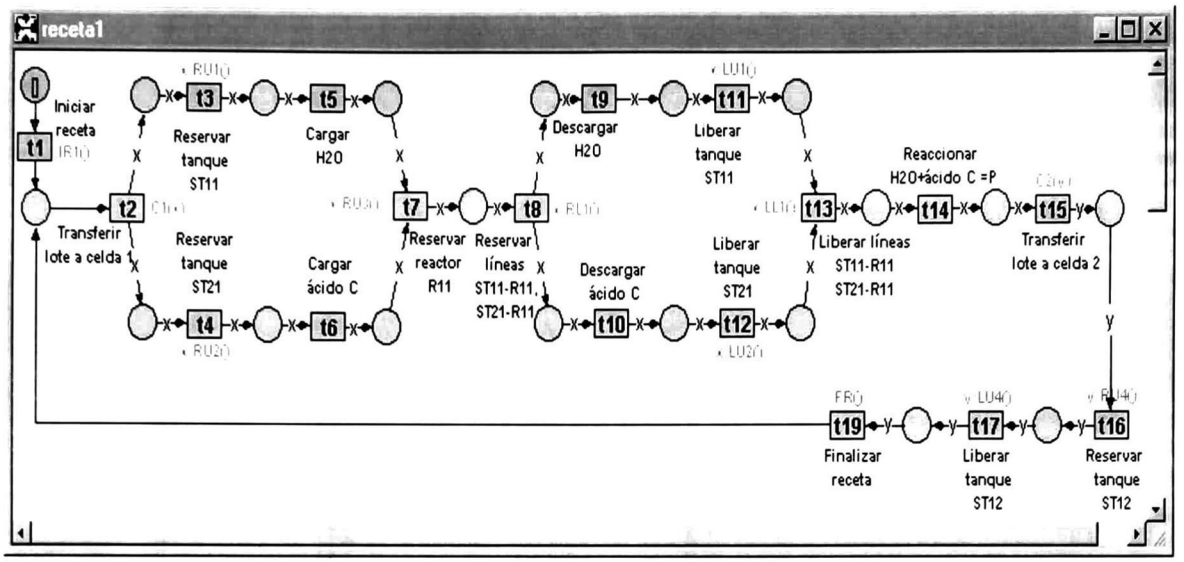


Figura 5.4: Red receta 1

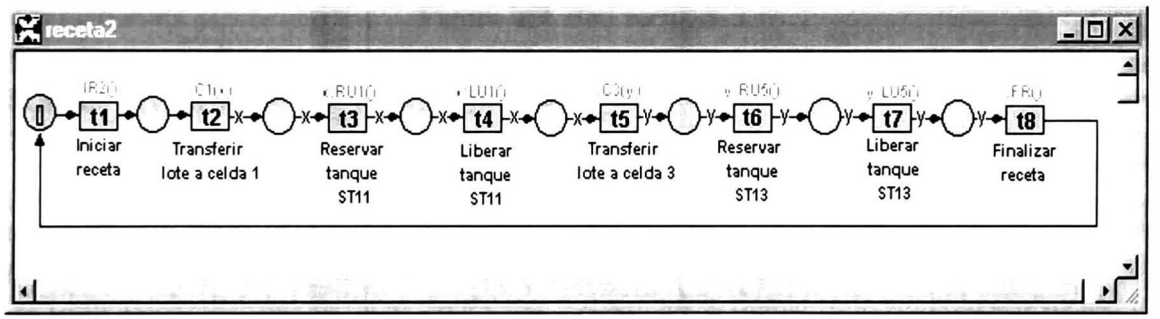


Figura 5.5: Red receta 2

Los canales :IR2(), :FR() permiten sincronización con la red *planprocesos*. El primer canal inicia la receta 2 y el segundo canal, la finaliza. Los canales x:RU1(), x:LU1(), reservan y liberan los recursos de la red *recursos1*; los canales y:RU5(), y:LU5() reservan y liberan los recursos de la red *recursos3*.

La red *receta3* es exclusiva de la celda de proceso 4, por lo que se requiere para su ejecución la red *recurso4*. La figura 5.6 muestra la red *receta3*. El disparo de la transición t1 inicia la receta respecto al canal :IR3(); La transición t2 inscrita con el canal :C4(x) transfiere el lote a la celda de proceso 4. La transición t5, a través del canal :FR(), termina la receta. Estás transiciones están sincronizadas con la red *planprocesos*. Las transiciones t3 y t4 inscritas con los canales x:RU6() y x:LU6() se sincronizan con la red *recursos4*.

El marcado de la red está dado por la referencia de red *recursos4*, la cual es evaluada en la variable x.

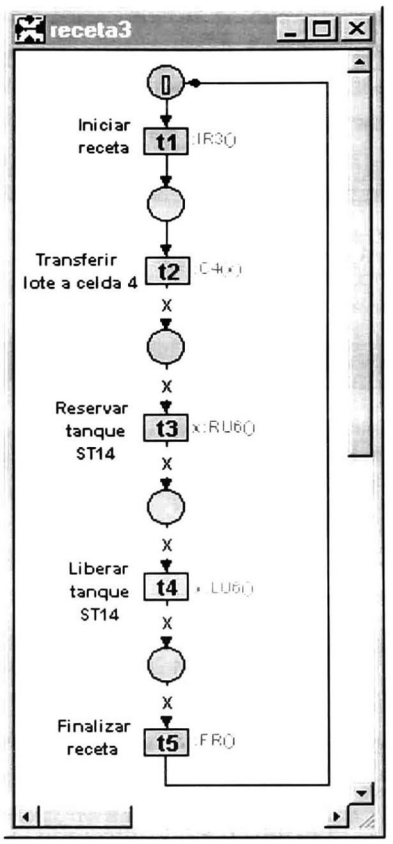


Figura 5.6: Red receta 3

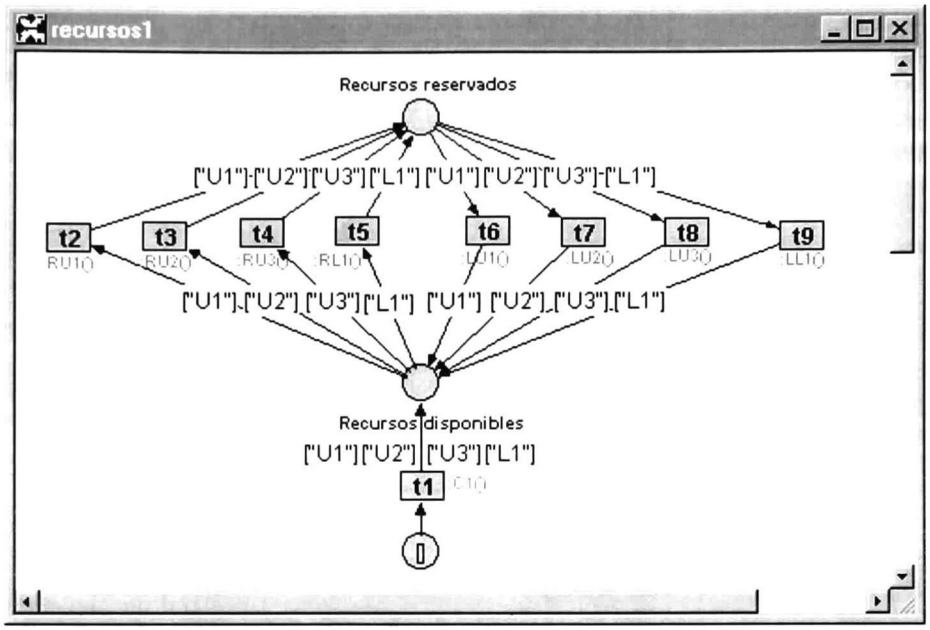


Figura 5.7: Red recursos 1

5.3.5. Modelos de recursos

Los recursos de la planta han sido clasificados de acuerdo a la celda de proceso que pertenecen. En la planta hay cuatro celdas de proceso, por lo tanto hay cuatro redes que representan la disponibilidad de los recursos de cada una de ellas. Estas redes son: *recursos1*, *recursos2*, *recursos3* y *recursos4*.

El marcado de estas redes se constituye por símbolos, donde cada símbolo representa un recurso. Para efectos de simulación no se han considerado todos los recursos.

La red *recursos1* se muestra en la figura 5.7, la cual controla la disponibilidad de los recursos de la celda de proceso 1. En la red se han incluido dos nodos de inicialización: el lugar marcado con la tupla vacía [] y la transición t1. Estos nodos generan el marcado inicial de la red *recursos1*, a través de sincronizar la transición t1 mediante el canal :C1() con la red *ambiente*.

La red tiene en su marcado los símbolos U1, U2, U3 y L1. Los tres primeros símbolos corresponden a tres unidades de procesamiento, y el último a una línea de conexión.

Debido a la imposibilidad de asociar a una transición más de un canal en espera de ser sincronizado, se ha separado la sincronización en varias transiciones. La cantidad de transiciones depende de la cantidad de recursos. Para cada recurso dos transiciones son necesarias. Una transición que reserve y otra que libere el recurso. De esta manera, si la receta solicita la reservación de la unidad de procesamiento ST11 la transición t2, inscrita con el canal :RU1(), tiene que ser disparada. Esto hace que la marca se consuma del lugar etiquetado "Recursos disponibles" y se produzca en el lugar nombrado Recursos reservados" Una vez que el recurso ya no es utilizado, la transición t6 inscrita con el canal :LU1() tiene que ser disparada para cambiar su estado a disponible, devolviendo el símbolo a su lugar inicial.

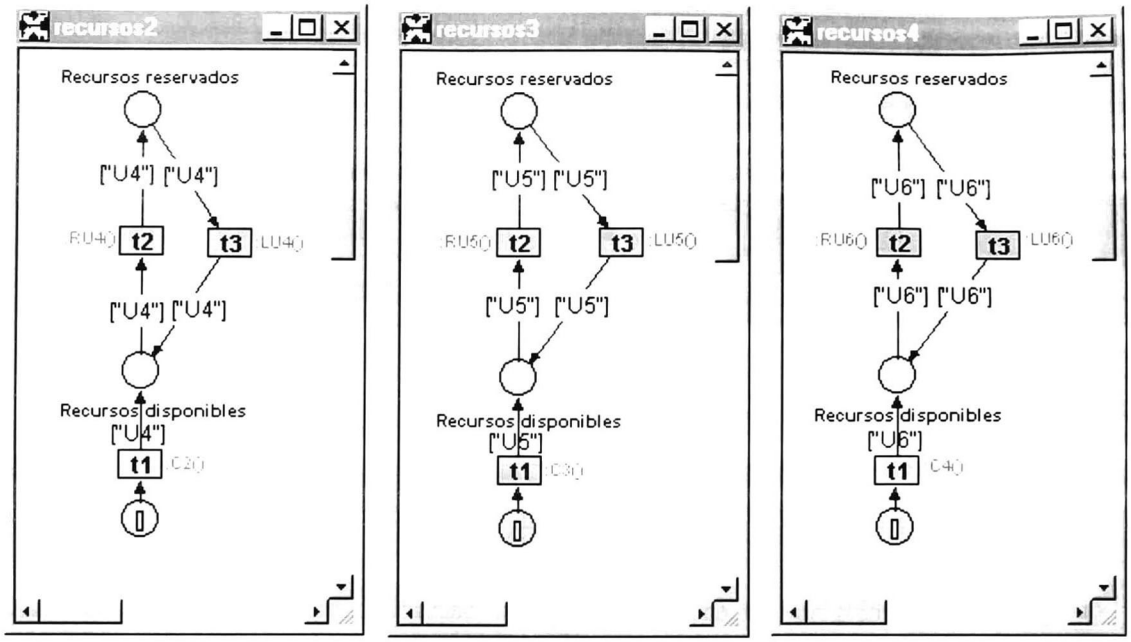


Figura 5.8: Redes recursos1, recursos2 y recursos3.

La figura 5.8 ilustra las tres redes de recursos restantes. En su marcado contienen un sólo símbolo, es decir se limitan a reservar una sola unidad de procesamiento. Estas redes se sincronizan con la red *ambiente* y con la red receta que solicite su servicio.

5.4. Simulación de modelos

Una vez editados los modelos el proceso de simulación puede iniciar. Para apreciar la evolución de los modelos, la simulación del sistema se hará paso a paso. El comando **Simulation Step** permite este modo de simulación.

Una consideración que no puede ser olvidada antes de iniciar la simulación, es la carga en memoria de todos los modelos de red creados. Por lo tanto, las ventanas nombradas: *ambiente*, *agente*, *planprocesos*, *receta1*, *receta2*, *receta3*, *recursos1*, *recursos2*, *recursos3* y *recursos4* deben estar abiertas al momento de comenzar la simulación. La ventana que contiene la red que será instanciada inicialmente, debe estar activa. En este caso, la ventana activa será la que contiene la red *ambiente*.

Las instancias de red son desplegadas en ventanas con fondo de otro color. Color que caracteriza a las ventanas que participan en la simulación del sistema. El título de cada ventana tiene un nombre y un número. El nombre está relacionado con la red que representa, y el número es asignado por el simulador, por lo tanto, puede variar de una simulación a otra.

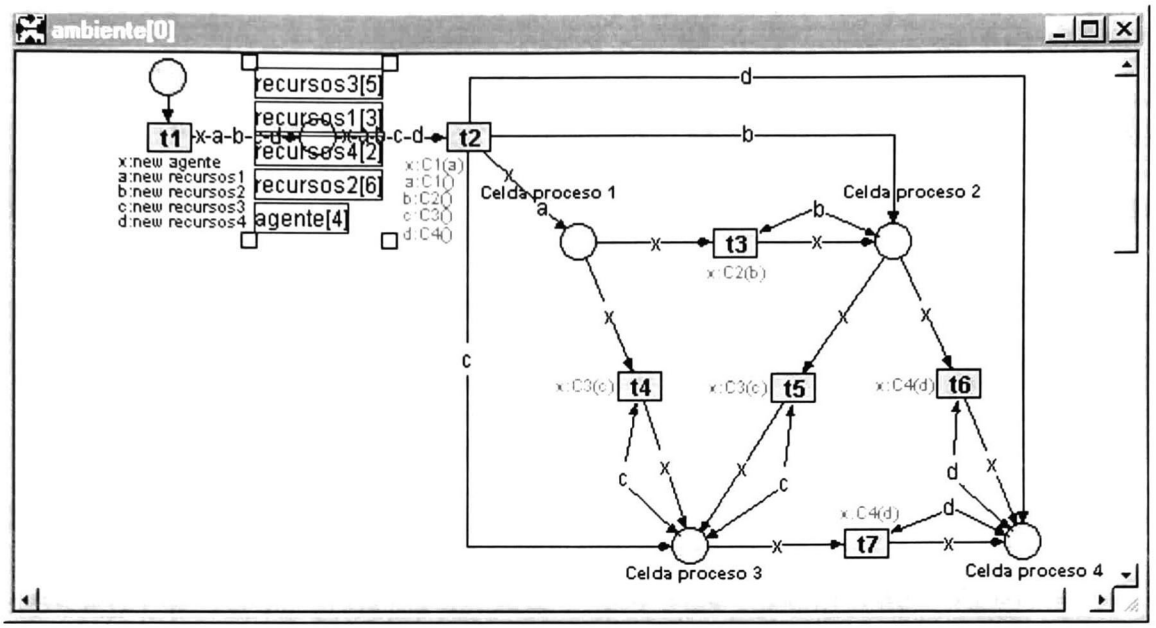


Figura 5.9: Instancia de red del ambiente después de crear las referencias de red en su marcado

5.4.1. Modelo de la planta

Al ejecutar la simulación, la instancia de la red *ambiente* es creada con el nombre ambiente[0]. Ver figura 5.9. Inicialmente la transición t1 está habilitada respecto al marcado asignado. El disparo de t1 crea 5 referencias de red. La variable x evalúa la referencia de red agente[4]; las variables a, b, c, y d, evalúan las referencias de red recursos1[3], recursos2[6], recursos3[5], y recursos4[2], respectivamente.

Estas referencias son depositadas en el lugar de salida de t1. Las instancias pueden ser visualizadas al dar doble click sobre las referencias de red contenidas en el lugar.

El marcado actual de la instancia de red ambiente[0] cumple las precondiciones establecidas para disparar t2, sin embargo esto no es suficiente, se requiere sincronizar t2 con: la transición t8 de la instancia de red agente[4] a través del canal :C1(), la transición t1 de la instancia de red recursos1[3] inscrita con el canal :C1(), la transición t1 de la instancia de red recursos2[6] mediante el canal :C2(), la transición t1 de la instancia de red recursos3[5] a través del canal :C3(), y la transición t1 de la instancia de red recursos4[2] respecto al canal :C4().

Después del disparo de t2, las referencias son distribuidas de acuerdo al marcado inicial del modelo de red, por lo tanto, las referencias de red, agente[4] y recursos1[3] son colocadas en el lugar "celda de proceso 1"; la referencia de red recursos2[6] se produce en el lugar "celda de proceso 2"; la referencia de red recursos3[5] se deposita en el lugar "celda de proceso 3"; y el lugar "celda de proceso 4" se marca con la referencia de red recursos4[2]. La figura 5.10 muestra esta distribución de marcado.

La habilitación de las transiciones t3 y t4 de la red ambiente[0] depende de la instancia de

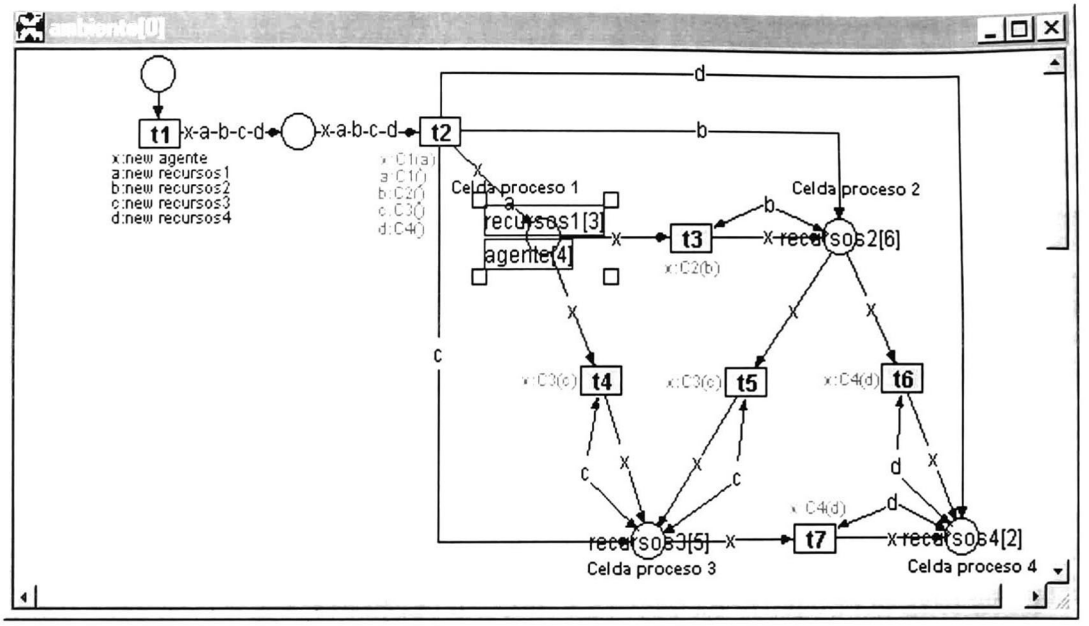


Figura 5.10: Instancia de red del ambiente después de la distribución de las referencias creadas

red agente[4]. Disparar la transición t5 de la instancia de red agente[4], habilita la transición t3, pero disparar t6 en la instancia de red agente[4] habilita t4. Esto determina la ruta que sigue el agente según la receta que ejecuta.

El disparo de las transiciones t3 y t4 requiere sincronización de la red agente[4]. La transición t3 se sincroniza con la transición t10 a través del canal :C2(b), y la transición t4 con t9 respecto al canal :C3(c). Durante el disparo de t3 la instancia de red recursos2[6] es reservada, y la referencia de red agente[4] se mueve al lugar "celda de proceso 2" En el caso de la transición t4 la instancia reservada es recursos3[5], y la referencia agente[4] se deposita en el lugar "celda de proceso 3"

Las transiciones t6 y t7 son habilitadas, después de disparar la transición t7 de la instancia agente[4]. Ambas transiciones se sincronizan con la transición t11 de la instancia agente[4]. Esta sincronización corresponde al canal :C4(d).

Al terminar la simulación, la instancia de red ambiente[0] debe tener en el lugar "celda de proceso 4", la referencia de red agente[4] y la referencia de red recursos4[2].

5.4.2. Modelo del lote

La instancia de la red *agente* es identificada como agente[4]. La figura 5.11 ilustra la instancia de red con su marcado inicial.Las transiciones t1 y t2 están inicialmente habilitadas; su disparo produce las referencias de red planprocesos[26], receta1[17], receta2[19] y receta3[18].

El marcado actual habilita la transición t3, la cual se sincroniza con la transición t1 de la instancia planprocesos[26] a través del canal :IP(). Este cambio de estado de la instancia habilita la transición t5 y t6. La ocurrencia de cualquiera de ellas determina la secuencia de

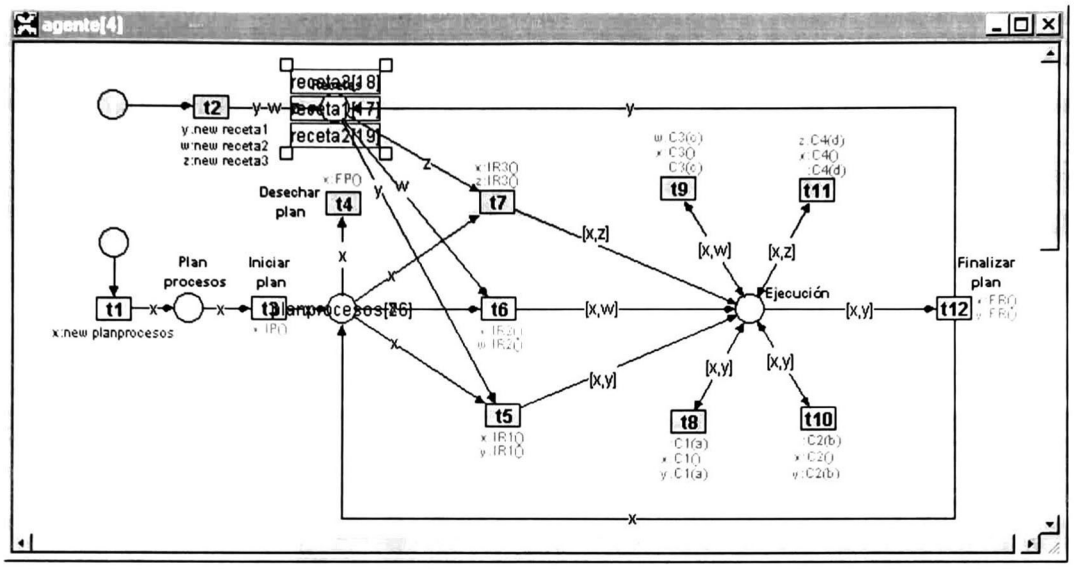


Figura 5.11: Instancia de red del agente

transiciones que serán disparadas posteriormente. Por lo que, disparar t5 lleva al disparo de t8, t10 y t12, mientras que el disparo de t6 provoca el disparo de t8, t9 y t12. Tanto t5 como t6 están sincronizadas; su disparo involucra simultáneamente el disparo de transiciones de otras instancias de red. Así, el disparo de t5 implica el disparo de la transición t2 en la instancia de red planprocesos[26], y de t1 en la instancia de red receta1[17]. Tal sincronización se establece por el canal :IR1(). Por otro lado, el disparo de t6 implica el disparo de t3 en la instancia de red planproceso[26], y t1 en la instancia de red receta2[19]. Las transiciones t6, t3 y t1 se sincronizan mediante el canal IR2().

Con el disparo de t5, las referencias de red receta1[17] y planprocesos[26] son depositadas en el lugar nombrado "ejecución". En caso de que t6 se dispare, el marcado de dicho lugar lo ocupan las referencias receta2[19] y planprocesos[26].

La transición t7 no se habilita sino hasta que la instancia de red receta1[17] o la instancia de red receta2[19] hayan sido finalizadas. Esto sucede con el disparo de la transición t12, la cual se sincroniza con la transición t11 de la instancia de red planprocesos[26]; dependiendo de cual sea la receta en ejecución, también t12 se sincroniza con t19 de la instancia de red receta1[17] o con t8 de la instancia de red receta2[19]. El canal :FR() permite tal sincronización.

Las transiciones t8, t9, t10 y t11 se sincronizan con la instancia de red ambiente[0], a través de los canales :C1(a), :C2(b), :C3(c) y :C4(d), respectivamente.

La simulación termina cuando la instancia de red agente[4] ha concluido el plan de procesos, y éste es desechado por el disparo de la transición t4.

5.4.3. Modelo del plan de procesos

La figura 5.12 ilustra la instancia de red planprocesos[26]. La evolución de la instancia inicia cuando la transición t1 es disparada. Su disparo está condicionado al disparo de la

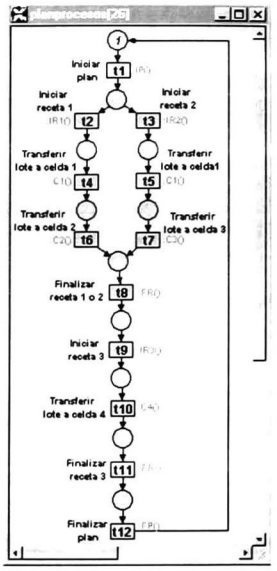


Figura 5.12: Instancia de red del plan de procesos

transición t3 de la instancia de red agente[4]. El marcado resultante en la red planprocesos[26] habilita cualquiera de las transiciones t2 y t3. Disparar la transición t2 implica, disparar las transiciones t5 de la instancia de red agente[4], y t1 de la instancia de red receta1[17]. El disparo de t3 implica, el disparo de la transición t6 de la instancia de red agente[4], y el disparo de t1 de la instancia de red receta2[19].

Las transiciones t4 y t5 se sincronizan con la instancia de red agente[4]. Estas se disparan respecto al canal :C1() inscrito en las transición t8 de dicha red. La transición t6 se dispara cuando la transición t10 de la red agente[4] se dispara. Lo mismo sucede con la transición t7, pero ésta se dispara con la transición t9 de la red agente[4].

Independientemente de cual sea la receta elegida, ésta debe finalizar. El disparo de las transiciones t8 y t11 llevan la receta a su término. Por un lado, la transición t8 se sincroniza con la transición t19 de la instancia de red receta1[17], y por otro, se sincroniza con la transición t8 de la instancia de red receta2[19]. La transición t11 se sincroniza con t5 de la instancia de red receta3[18]. La sincronización se efectúa respecto al mismo canal :FR().

Iniciar la receta 3 conlleva el disparo de t9. Esta transición espera ser sincronizada por la transición t7 de la instancia de red agente[4]. También se sincroniza al mismo nivel con la transición t1 de la instancia de red receta3[18]. El canal :IR3() es quien establece esta sincronización.

La transición t10 se dispara cuando la transición t11 de la instancia de red agente[4] inicia la sincronización por medio del canal :C4(). Y por último, la transición t12 se sincroniza con la transición t4 de la instancia de red agente[4].

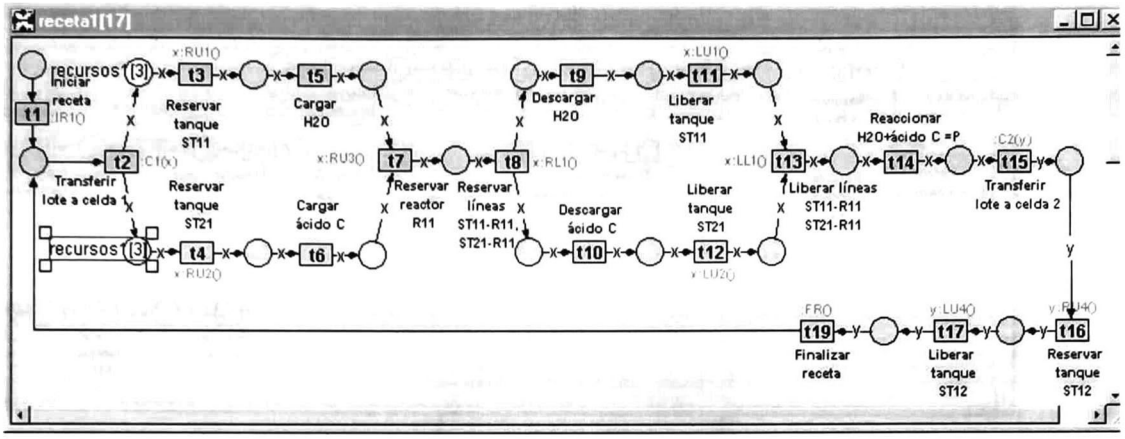


Figura 5.13: Instancia de red de la receta 1 marcada con la referencia de red recursos 1

5.4.4. Modelo de recetas

La figura 5.13 muestra una instancia de la red receta1, la cual es identificada en la simulación por receta1[17]. La evolución de esta instancia de red inicia cuando se dispara la transición t1, simultáneamente con la transición t2 de la instancia de red planprocesos[26] y con la transición t5 de la instancia de red agente[4]; todas ellas sincronizadas respecto al canal IR1(). El disparo de t2 deposita una red recursos1[3] en los lugares de salida.

Una vez incluida la referencia de la instancia de red recursos1[3] en la receta, se puede hacer la reservación y liberación de recursos de la celda de proceso 1. Las transiciones t3, t4, t7 y t8 de la instancia de red receta1[17] reservan los recursos cuando se sincronizan con las transiciones t2, t3, t4 y t5 de la instancia de red recursos1[3], respectivamente; y el cambiar su estado a no-reservado depende de las transiciones t11, t12 y t13, las cuales se sincronizan con las transiciones t6, t7, t9 de la instancia de red recursos1[3]. Esta instancia forma el marcado de la red hasta que la transición t15 es disparada. En la figura 5.14 se aprecia el nuevo marcado. Ahora la instancia de red receta1[17] se sincroniza con la instancia de red recursos2[6].Al disparar la transición t19 la instancia de red termina su ejecución.

La instancia de red de la recetas 2 se muestra en la figura 5.14; su nombre dentro de la simulación es receta2[19]. La transición t1 de la red se dispara cuando la transición t6 de la instancia de red agente[4] y la transición t3 de la instancia de red planprocesos[26] se disparan respecto al canal IR2(). Este cambio de estado en la red indica que la receta 2 ha iniciado operación en la celda de proceso 1. El canal inscrito en la transición 2 recibe como parámetro la instancia de red recursos1[3]. Con esta red se sincroniza para administrar el uso de recursos a través de las transiciones t3 y t4, mientras la receta 1 se aplica a la celda de proceso 1. La manipulación de los recursos es representativa en la instancia receta2[19]; sólo un recurso es reservado y liberado, aún cuando hay más recursos que utiliza la receta.

Suponiendo que la receta ha terminado su operación dentro de la celda de proceso 1, el disparo de la transición t5 mueve el lote a la celda de proceso 3, lo cual implica utilizar sus recursos. La disponibilidad de los recursos de la celda de proceso 3 se modelan en la instancia de red recursos3[5]. Esta instancia es recibida como parámetro después del disparo de la transición t5 y permanece como marca dentro de la instancia receta2[19] hasta que

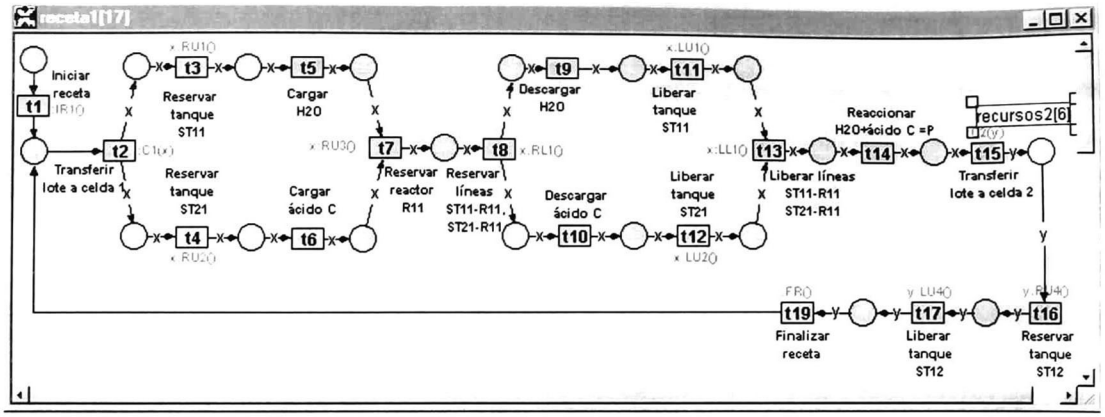


Figura 5.14: Instancia de red de la receta 1 marcada con la referencia de red recursos 2

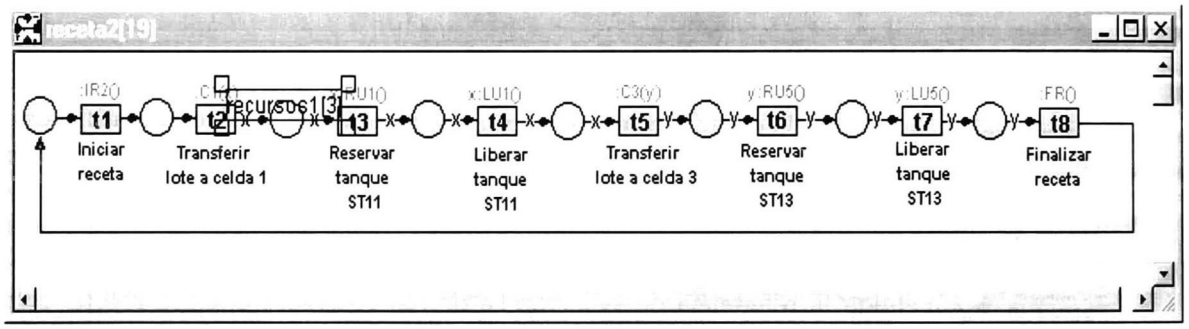


Figura 5.15: Instancia de red de la receta 2

termina su ejecución con el disparo de t8. Las transiciones t6 y t7 se encargan de reservar y liberar un recurso de la celda de proceso 3.

Por último, la instancia de la receta 3 se ilustra en la figura 5.15. Esta instancia es identificada como receta3[18]. El disparo de la transición t1 inicia la receta junto con el disparo de la transición t9 de la instancia de red planprocesos[26] y con el disparo de la transición t7 de instancia de red agente [4]. Esta sincronización es respecto al canal IR3(). La instancia de red recursos4[2] se recibe como parámetro en el marcado después del disparo de t2. La instancia de red receta3[18] puede reservar y liberar recursos a traves de las transiciones t3 y t4 respectivamente. Ambas son sincronizadas con las transiciones t2 y t3 de la instancia de red recursos4[2].

5.4.5. Modelos de recursos

La figura 5.16 muestra la instancia de red recursos1[3] y la figura 5.16 muestra las tres instancias de red recursos2[6], recursos3[5] y recursos4[2].

La instancia de red recursos1[3] se sincroniza con las instancias de red receta1[17] y receta2[19]. Alguna de las transiciones t2, t3, t4 y t5 se dispara cuando una instancia de

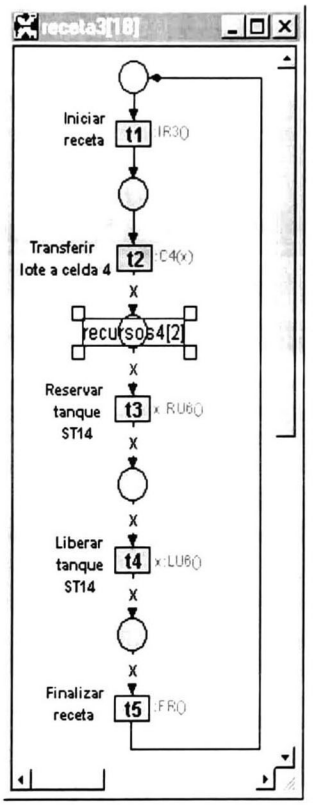


Figura 5.16: Instancia de red de la receta 3

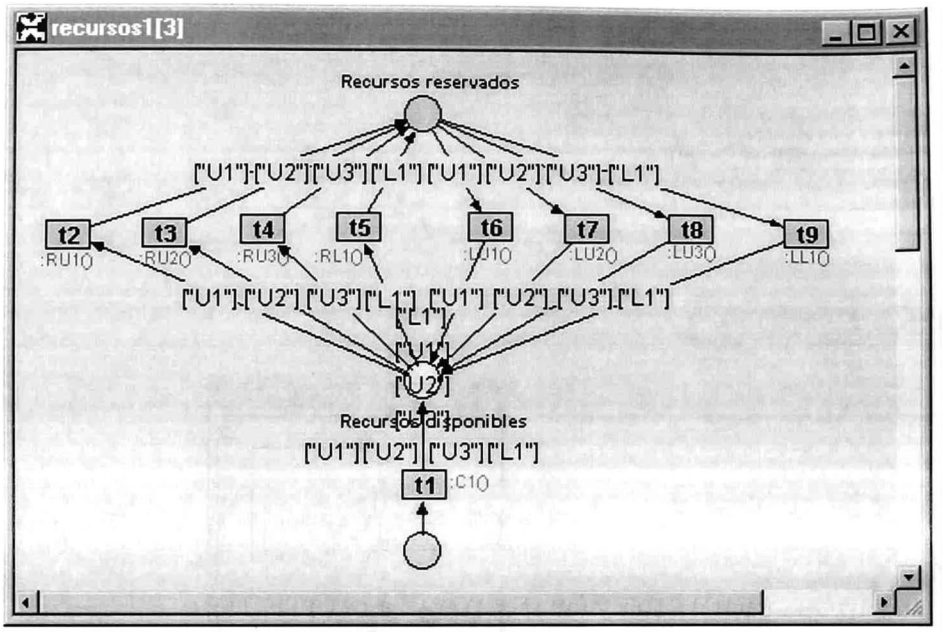


Figura 5.17: Instancia de red de recursos 1

receta mencionada solicita el uso de un recurso. Cada transición cambia el estado de un sólo recurso, el cual está incluido en el marcado como un símbolo. Cuando el recurso es desocupado por la receta que lo solicitó, entonces alguna de las transiciones t6, t7, t8 y t9 puede ser disparada respecto al canal que le corresponde.

De igual manera, las instancias de red recursos2[6], recursos3[5] y recursos4[2] son simuladas. La instancia de red recursos2[6] se sincroniza con la instancia de red receta1[17], mientras que la instancia de red recursos3[5] se sincroniza con la instancia de red receta2[19]. Finalmente, la instancia de red recursos4[2] se sincroniza con la instancia de red receta3[18].

5.5. Conclusión

En este capítulo se presentó la simulación de modelos basados en la metodología de modelado para sistemas de procesos por lotes. La herramienta de simulación utilizada fue Renew, porque se adapta bien al enfoque multi-nivel propuesto.

La simulación consistió en crear dinámicamente instancias de red de los modelos obtenidos y establecer comunicación entre ellas de acuerdo a canales síncronos.

La simulación efectuada permite observar el comportamiento de esta clase de sistemas, lo cual ayuda a su validación.

Obviamente, una implementación real requiere de un estudio más profundo, pero para efectos de esta tesis la simulación resulta ser una alternativa muy útil para el análisis.

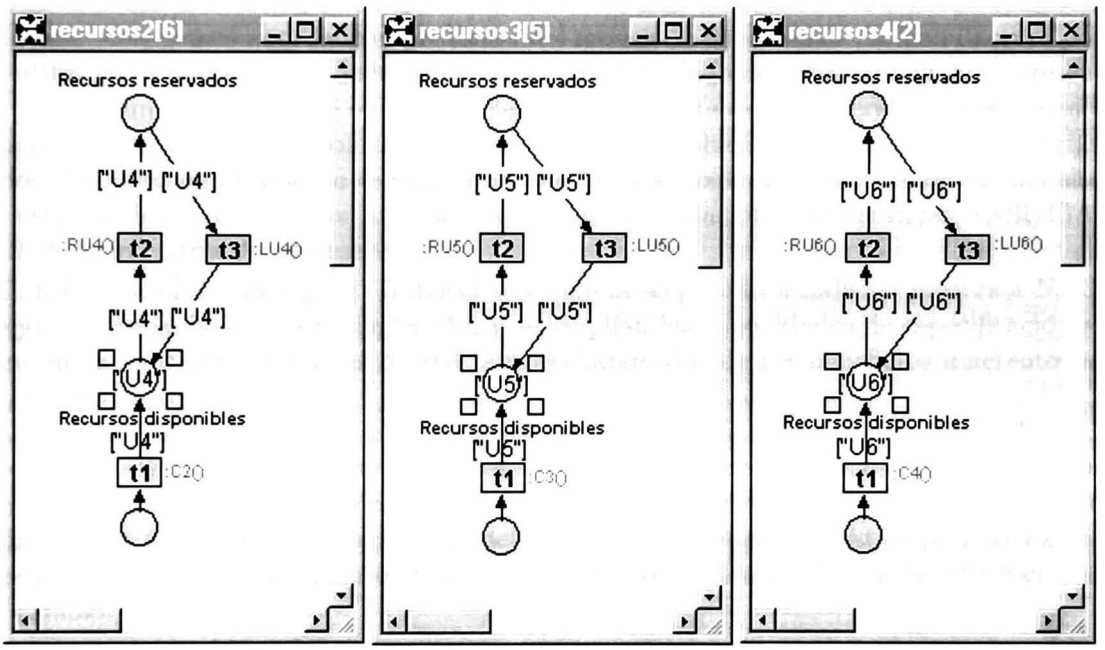


Figura 5.18: Instancias de red de recursos 2, recursos 3 y recursos 4

Capítulo 6

Conclusiones

En esta tesis hemos abordado el problema del modelado de sistemas de manufactura por lotes utilizando una extensión de las redes de Petri. El trabajo de investigación aquí presentado propone un formalismo de red a n niveles el cual permite expresar modelos modulares y compactos de sistemas que involucran entidades móviles evolucionando en ambientes estructurados. Complementariamente se ha definido una metodología de construcción de modelos para sistemas de producción por lotes acorde a las recomendaciones del estándar ANSI/ISA-S88.01 de la industria de procesos.

En n-LNS se eliminan algunas restricciones impuestas por el formalismo precursor NS-3, se mejora el mecanismo de sincronización, y se amplían las capacidades de especificación al no limitar la cantidad de niveles de red, aprovechando dos de los beneficios inherentes al enfoque multi-nivel: modularidad y compacidad, lo que conduce a modelos más entendibles y manejables. Una ventaja de usar el enfoque modular multi-nivel es que permite la construcción más cómoda de un modelo complejo definiendo modelos parciales correspondientes a componentes físicas o lógicas del sistema para después interrelacionarlos con etiquetas o incluirlos como marcas dentro de otros modelos parciales. Esta característica permite extender, reducir, o modificar fácilmente un modelo afectando únicamente las partes involucradas en el cambio.

La metodología de modelado propuesta intenta explotar las características de n-LNS adaptándola a los conceptos y técnicas incluidas en el estándar S88.01. Esta metodología, la cual sigue una estrategia ascendente (bottom-up), constituye una excelente guía para obtener los modelos expresados en n-LNS; el sistema de manufactura estructurado según el estándar, sugiere de una manera natural la descomposición identificando fácilmente módulos y niveles de abstracción. Así, las redes correspondientes a recetas, planes de proceso, recursos, estructuras de áreas, etc, son rápidamente definidos. Los ejemplos incluidos ilustran el carácter práctico de esta guía de modelado.

El presente trabajo constituye el primer paso de una investigación orientada a obtener alternativas de solución al problema de control de los sistemas de manufactura por lotes. Los casos de estudio abordados, si bien están inspirados en sistemas de la industria química, son de tipo académico, y el análisis realizado a los modelos ha sido mediante simulación utilizando la herramienta Renew. De aquí que podamos enunciar entre otros temas a profundizar, el análisis de los modelos obtenidos y el diseño e implementación de sistemas de control distribuido. El problema del análisis de propiedades, el cual se vislumbra difícil se podría abordar con una estrategia ya conocida de divide y vencerás en la que sabiendo que los

componentes poseen ciertas propiedades, éstas se preservan en el modelo global respetando ciertas condiciones de interacción, en nuestro caso, el etiquetado. El diseño de sistemas para control de procesos requiere un estudio profundo sobre sistemas reales, los cuales presentarán otras problemáticas seguramente no abordadas en este trabajo y que requerirán soluciones más detalladas en cuanto al tratamiento de los niveles más bajos en la implementación de las recetas y manejo de equipo.

Bibliografía

[1] Akesson, K. , M. Fabian, A. Vahidi. Coordination of Batches in Flexible Production. IEEE American Control Conference, Chicago, Illionis, 2000.

[2] Akesson, K. "Recipe Coordination in Chemical Batch Processes" Thesis for the Degree of Licentiate of Engineering. Chalmers University of Technology, Göteborg, Sweden 1999.

[3] Almeyda, H. I. "Un Sistema de Red a Tres Niveles para el Modelado de Agentes Móviles" Tesis para obtener el grado de M. C. Cinvestav-IPN, Guadalajara, México, Septiembre 2002.

[4] Andreu, D., J.C. Pascal, H. Pingaud, R.Valette. "Batch Process Modelling Using Petri Nets". IEEE International Conference on Systems, Man and Cybernetics, San Antonio, USA October 2-5, 1994, pp. 314-319.

[5] Andreu, D., J.C. Pascal, R. Valette. "Interaction of Discrete and Continuous parts of a Batch Process Control System". Workshop on Analysis and Design of Event Discrete Operations in Process Systems (ADEDOPS), Imperial College, London, 10-11 April 1995 (8p).

[6] Bauer, N., S. Kowalewski, G. Sand, T. Löhl. A Case of Study: Multi Product Batch Plant for the Demonstration of Control and Scheduling Problems. ADPM 2000.

[7] Champagnat, R., P. Esteban, H. Pingaud, R. Valette. "Petri Net Based Modelling of Hybrid Systems". ICIMS-NOE ASI'96 Conference Lyfe Cycle Approaches to Production Systems, Management, Control, Supervision, Toulouse France June 2-6 1996, pp.53-60.

[8] Champagnat, R., R. Valette, J. C. Hochon, H. Pingaud. "Modelling, Simulation and Analysis of Batch Production Systems. Discrete Event Dynamic Systems" Theory and Application, Vol. 11, n.1/2, January/April 2001, p.119-136. Kluwer Academic Publishers, ISSN 0924-6703.

[9] DiCesare, F. G.Harhalakis, J.M.Proth, M.Silva, F.B.Vernadat. "Practice of Petri Nets in Manufacturing" Chapman & Hall. 1993.

[10] Fleming, D. W., V. Pillai. "S88 Implementation Guide. Strategic Automation for the Process Industries" Ed. McGraw-Hill. 1999.

[11] Genrich, H.J., Lautenbach, K. "The Analysis of Distributed Systems by Means of Predicate/Transition Nets". Semantics of Concurrent Computation. Evian 1979, G.Kahn (ed). Lecture Notes in Computer Science, Vol. 70 Springer Verlag 1979, pp. 123-146.

[12] Hiraishi, K. "A Petri-net-based model for the mathematical analysis of multi-agent systems" Proceedings of the IEEE International Conference on Systems, Man & Cybernetics, Nashville, Tennessee, USA. October 2000, pp. 3009-3014.

[13] International Society for Measurement and Control (ISA). "Batch Control. Practical Guides for Measurement and Control" A. E. Nisenfeld, Editor.1996.

[14] International Society for Measurement and Control (ISA). "Standard ISA-S88.01, Batch Control, Part 1: Models and Terminology" 1995.

[15] Jensen, K. "Coloured Petri Nets and the Invariant Method" Theoretical Computer Science Vol. 14 pp. 317-336 1981.

[16] Julia, S., R. Valette. Real Time Scheduling of Batch Systems. Simulation Practice and Theory, Elsevier, 8(2000) pp. 307-319.

[17] Kasturia, E. Dicesare, F., Desrochers, A. "Real Time Control of Multilevel Manufacturing Systems Using Colored Petri Nets" IEEE Int. Conf. on Robotics and Automation, Philadelphia, PA USA, May 1987. pp. 1114-1119.

[18] Kitajima, T., T. Kodama, Y. Shimizu. A Colored Petri Net Model for Analyzing Batch Sequential Control System. PSE Asia 2000, December 6-8, Kyoto, pp. 279-284.

[19] Köhler, M., D. Moldt, H. Rölke. "Modelling the Structure and Behaviour of Petri Net Agents". Application and Theory of Petri Nets 2001: 22nd International Conference, ICATPN 2001 Newcastle upon Tyne, UK, June 25-29, 2001, Proceedings.

[20] Kummer, O. "Introduction to Petri nets and Reference nets" Sozionikaktuell 1:2001. pp.7-16.

[21] Kummer, O., F. Wienberg, M. Duvigneau. Renew" The Reference Net Workshop. http://www.informatik.uni-hamburg.de/TGI/renew/renew.html.

[22] Lakos C. "From Coloured Petri Nets to Object Petri Nets" Proc. of 16th International Conference on the Application and Theory of Petri Nets, Lecture Notes in Computer Science 935, pp 278-297, Torino, Italy, Springer-Verlag (1995).

[23] Lomazova, I. "Nested Petri nets –a formalism for specification and verification of multi-agent distributed systems". Fundamenta informaticae 43 (2000) pp. 195-214. IOS Press.

[24] Neuendorf, K., D. Kiritsis, T. Kis, P Xirouchakis. "Two-Level Petri Net Modelling for Integrated Process and Job Shop Production Planning" Workshop Algorithmen und Werkzeuge für Petrinetze, Oct. 2000, Koblenz, Germany, pp. 90-95.

[25] Niebert, P., S. Yovine. Computing Optimal Operation Schemes for Chemical Plants in Multi-Batch Mode. In Proceedings of "Hybrid Systems: Computation and Control, HSCC'00" Pittsburgh, PA, USA, March 23-25, 2000. Lecture Notes in Computer Science 1790, Springer-Verlag.

[26] Sibertin-Blanc, C. "Cooperative Nets" Proceedings of the 15th International Conference on Application and Theory of Petri Nets, LNCS 815, Zaragoza, Spain. June 1994, pp. 471-490.

[27] Silva, M. "Las redes de Petri: en la Automática y la Informática" Ed. AC. Madrid, Spain. 1985.

[28] Silva, M., E. Teruel, R. Valette, H. Pingaud. "Petri Nets and Production Systems" In Lectures on Petri nets II: applications, Lectures notes in Computer Science 1492, Springer Verlag 1998, pp.85-124.

[29] Tittus, M., K. Akesson. "Discrete Event Models in Batch Control" 2nd Mathematical and Computer Modelling of Dynamical Systems (MATHMOD), Viena, Austria, February 1997.

[30] Tittus, M. K. Akesson. "Modular Supervisor for Deadlock Avoidance in Batch Processes" Proc. of the IEEE Int. Conf. on Systems, Man and Cybernetics, San Diego, USA Oct. 1998.

[31] Valk, R. "Petri Nets as Token Objects", Lecture Notes in Computer Science, Vol. 1420: 19th International Conference on Application and Theory of Petri Nets, ICATPN'98, Lisbon, Portugal, Springer-Verlag, June 1998, pp. 1-25.

[32] Valk, R. Çoncurrency in Communicating Object Petri Nets". In F. De Cindio G. Agha and Rozenberg, editors, Advances in Petri Nets on Concurrent Object-Oriented Programming and Petri Nets, Lecture Notes in Computer Science, Springer-Verlag, LNCS 2001, pp.164-195.

Apéndice A

Herramienta de simulación Renew

Elementos de red

Las redes referencia tienen lugares, transiciones y arcos. Hay muchos tipos de arcos:

Arcos de entrada y salida. Se comportan exactamente como en redes de Petri ordinarias, remueven y depositan marcas en un lugar.

Arcos de reserva. Reservan una marca durante el disparo de la transición.

Arcos de prueba. Una sola marca puede ser examinada por varios arcos de prueba a la vez. Esto es importante porque un periodo extendido de tiempo puede ser necesario antes de que una transición pueda completar su disparo.

En la figura A.1 se muestra una red que usa todos los elementos de red mencionados. El lugar p esta rodeado de seis transiciones. Inicialmente el lugar no esta marcado. Supongamos que la transición a se dispara, entonces una marca es depositada en el lugar p, por lo que todas las transiciones excepto c están ahora habilitadas. La transición c esta inhabilitada porque reserva dos marcas de p mientras se dispara. Por otro lado, la transición e puede dispararse porque esta permitido examinar una marca dos veces. Si la transición a se dispara, entonces la transición c puede habilitarse, porque una segunda marca esta disponible. El disparo de las transiciones b,c,e y f no cambian el marcado actual. La transición d remueve una marca de p cuando se dispara.

Cada elemento de red puede tener inscripciones semánticas. Los lugares pueden tener un tipo de lugar (opcional) y un número arbitrario de expresiones de inicialización. Las expresiones de inicialización son evaluadas y los valores resultantes sirven como marcado inicial de los lugares. En una expresión la tupla vacía representada por el par de corchetes [] denota una sola marca. Por defecto, un lugar inicialmente no esta marcado.

Los arcos pueden tener inscripciones de arco (opcional). Cuando una transición se dispara la expresión en el arco es evaluada y las marcas son movidas de acuerdo al resultado.

Las transiciones pueden ser equipadas con una variedad de inscripciones:

- *Inscripciones de expresión.* Son expresiones ordinarias que son evaluadas mientras el simulador de red busca un ligado de la transición. El resultado de esta evaluación es descartada, pero en tales expresiones es posible usar el operador "=" para influenciar el ligado de las variables que son usadas en otra parte.

- *Inscripciones de guarda.* Son expresiones que tienen como prefijo la palabra reservada guard. Una transición es disparada sólo si todas las expresiones de guarda evalúan

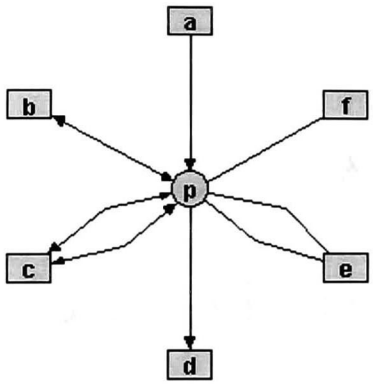


Figura A.1: La red elementos

a verdadero. Hasta aquí se han cubierto las expresiones utilizadas en redes de Petri coloreadas.

- *Inscripciones de acción.* Son expresiones precedidas por la palabra reservada action. Contrariamente a las inscripciones de expresión, las inscripciones de acción garantizan que serán evaluadas exactamente una vez durante el disparo de una transición. Las inscripciones de acción no pueden ser usadas para calcular el ligado de variables que son usadas en los arcos de entrada, porque las expresiones de los arcos de entrada deben ser evaluadas antes del disparo de la transición. Las inscripciones de acción pueden ayudar a calcular las marcas de salida.

- *Inscripciones de creación.* Conducen a la creación de instancias de red y canales síncronos.

La figura A.2 ilustra las inscripciones de arco y la asignación de tipos a los lugares. Un lugar es declarado como tipo int, lo cual significa que éste sólo puede contener enteros como marcas. En este caso, el lugar tiene como marcado inicial una marca de un entero 42. El resto de los lugares no tienen asignado un tipo, ni tampoco marcas. La transición t1 toma el valor de 42 del lugar de entrada, y deposita un 4 y un 2 en los respectivos lugares de salida. La transición t2 toma en x el valor 4 y lo produce en los dos lugares de salida. La transición t3 es similar, pero aquí la igualdad de las variables de los arcos de entrada y salida es establecida por la inscripción de la transición x=xx. La transición t4 tiene una guarda que asegura que x sea diferente de y, escrita como x!=y. Por lo tanto, esta consume un 2 y un 4.

El lenguaje de inscripción

Las redes del formalismo son similares a las redes de Petri coloreadas, por lo tanto, el simulador es quien determina que clase de marca debe ser movida por cada arco. Las clases de marcas son valores y referencias de red. Por omisión, un arco puede transportar una marca simple, denotada por []. Pero si se agrega una *inscripción de arco* a un arco, dicha inscripción debe ser evaluada y el resultado determinar que clase de marca es movida.

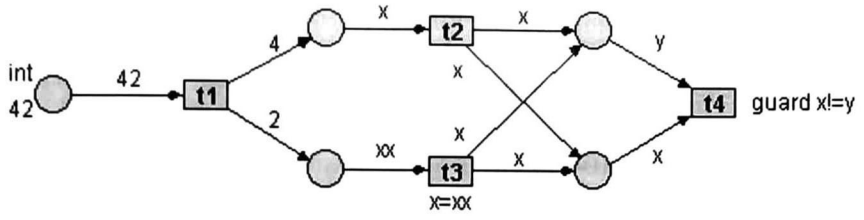


Figura A.2: La red coloreada

Expresiones y variables. Las inscripciones de arco son expresiones simples de java, pero con varias diferencias. La primera diferencia tiene que ver con los operadores binarios && (and) y || (or) usados en las expresiones. Esto porque si el resultado del operando izquierdo determina el resultado del operador, el operando derecho no es evaluado, lo cual implica un orden de ejecución que se trata de evitar en el formalismo. De aquí que los operadores no sean implementados. Posiblemente, puedan ser usados en versiones posteriores de Renew.

Tipos. En redes referencia, los tipos juegan dos roles. Por un lado, un tipo puede ser una inscripción de un lugar, lo cual significa que el lugar puede tener sólo valores de ese tipo. Y por otro lado, a las variables se les puede asignar un tipo.

En Java toda variable debe ser declarada. Sin embargo, generalmente las redes de Petri pueden ser escritas sin tener que declarar las variables. Las redes referencia proveen una opción para crear un *nodo declaración.* En el *nodo declaración* un número arbitrario de sentencias de cabecera y declaraciones de variables son permitidas. Pero, si un nodo declaración esta presente, entonces todas las variables deben ser declaradas. Entonces es importante elegir entre, si declarar o no una variable. Para las variables no declaradas su tipo es indefinido.

Los lugares que tienen un tipo asignado, permiten al simulador detectar situaciones de error, cuando las transiciones tratan de depositar marcas de tipo equivocado en los lugares.

Para los arcos de entrada a un lugar, el tipo de la inscripción de los arcos debe ser comparable al tipo del lugar. Para los arcos de salida de un lugar, se requiere que el tipo de la inscripción se ajuste también al tipo del lugar. Esto es importante, porque los valores de las expresiones de salida pueden ser sólo determinadas durante el disparo de una transición, cuando es demasiado tarde para declarar la transición inhabilitada. Para los arcos de entrada simplemente se ignora cualquier ligado que resultaría en una marca de tipo erróneo.

Tuplas y unificación. El lenguaje de inscripción de redes referencia ha sido extendido a incluir tuplas. Una tupla es denotada por una lista de expresiones separadas por comas encerradas entre corchetes; por ejemplo, [1,."bc", 1.0]. Las tuplas son útiles para almacenar un grupo total de valores relacionados dentro de una simple marca, y por consiguiente, en un simple lugar.

En Java no hay tuplas. Si se quiere almacenar un grupo de valores simplemente se requiere crear un grupo de variables, cada una de las cuales guarda un valor. En redes de Petri es posible almacenar arbitrariamente muchas marcas en un lugar.

Las tuplas son débilmente tipeadas, porque sus elementos no tienen un tipo asignado. Por lo tanto, un elemento de una tupla guarda un valor primitivo o una referencia de red.

La unificación es la única operación sobre tuplas. Las tuplas se unifican a través de una especificación de igualdad. Ejemplo, $[x, y, z]=t$ significa que t debe ser una 3-tupla, donde x es el primer elemento de t, y el segundo, y z el tercero.

Una simple marca es denotada por la tupla vacía [], entonces una marca es simplemente una tupla sin elementos (0-tuple).

Las tuplas pueden ser anidadas, por ejemplo, [[1,2], [3,4,5]], es una 2-tupla, que tiene una 2-tupla en su primer elemento y una 3-tupla en su segundo elemento. Esto puede ser útil si los elementos son jerárquicamente estructurados.

Instancias de red y redes referencia

Cuando la simulación de una red es iniciada, el simulador crea una instancia de la red simulada. Una red que es dibujada en el editor es una estructura estática. No obstante, una instancia de red tiene un marcado que puede cambiar en el tiempo. Al momento de iniciar una simulación, una instancia de red puede ser creada.

La mayoría de los formalismos llegan hasta aquí. Crean una instancia de la red y la simulan. Renew permite crear muchas instancias de una sola red. Cada instancia viene con su propio marcado y puede dispararse independientemente de otras instancias.

Cada red tiene un nombre. El nombre es derivado del archivo donde la red es guardada.

Las instancias de red son creadas por las transiciones que tienen *inscripciones de creación*, las cuales consisten de una variable, seguida de dos puntos (:), la palabra reservada *new* y el nombre de la red. Ejemplo, x:new othernet, donde a la variable x se le asigna una instancia de la red *othernet*.

Las figuras A.3, y muestran un simple ejemplo. Cuando se inicia la simulación de la red *creator*, al disparar la transición t1, se crean dos instancias de la red *othernet*, una es asignada a la variable x y la otra a la variable y. Estas instancias se convierten en referencias de red cuando son depositadas en los lugares p2 y p3 de la red *creator*. Ahora se tienen tres transiciones activadas, las dos transiciones de las dos instancias de red, y la transición t2 de la red *creator*. La *inscripción de guarda* (guarda x!=y) es satisfecha, porque las dos *inscripciones de creación* son diferentes al crear distintas instancias de red. Nunca se crea la misma instancia dos veces.

El orden de ejecución es indefinido. Esto es, puede ser que la transición de la red *creator* se dispare primero, aún cuando las transiciones de las instancias de la red *othernet* permanezcan habilitadas.

Una red no desaparece simplemente porque esta no es referenciada. Si una instancia no es refrenciada y ninguna de sus transiciones llega a estar habilitada, entonces ésta es sometida al colector de basura.

En Java la palabra reservada **this** denota el objeto cuyo método es actualmente ejecutado. En redes referencia **this** denota la instancia de red en la cual una transición se dispara.

Las instancias de red son tratadas como objetos. Tienen una identidad y tienen un estado que puede cambiar sobre el tiempo, donde los lugares parecen corresponder a los atributos. Las instancias de red también encapsulan datos. Ellas pueden ser referenciadas de otras instancias de red. Además, pueden ser equipadas con métodos. La comunicación entre redes es manejada por canales síncronos.

Canales síncronos

En los formalismos de redes de Petri, la clase de comunicación es usualmente capturada

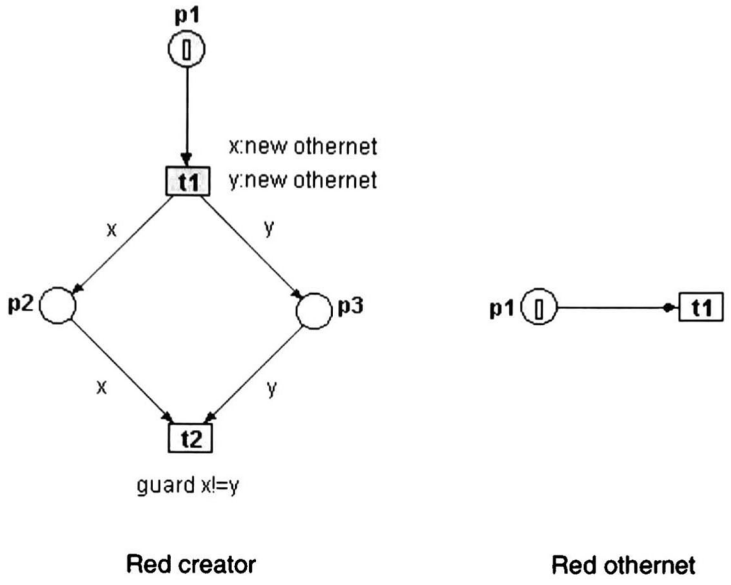


Figura A.3: Red con inscripciones de creación

por fusión de lugares. Las redes referencia implementan una clase de comunicación por canales síncronos. Esto permite modelos más expresivos comparados con el paso de mensajes, porque se oculta parte de la complejidad inherente de sincronización.

Los canales síncronos fueron considerados primero por las redes de Petri coloreadas [Christesen, Hansen]. Ellos sincronizan dos transiciones, donde ambas se disparan automáticamente al mismo tiempo. La condición es que las dos transiciones estén de acuerdo en el nombre del canal y en un conjunto de parámetros, antes de involucrarse en la sincronización. De aquí que se haya generalizado el concepto, permitiendo que transiciones de diferentes instancias de red puedan sincronizarse. Para esto se requiere que el iniciador de la sincronización conozca la otra instancia de red con la que pretende establecer sincronización.

La transición que inicia la sincronización debe tener una inscripción especial, llamada *downlink*. Un *downlink* hace una solicitud a una red subordinada. Un *downlink* consiste de, un expresión que debe evaluar a una referencia de red, seguido de dos puntos(:), el nombre del canal y entre paréntesis una serie de argumentos. Ejemplo, net:ch(1,2,3), el cual trata de sincronizar una transición de la red denotada por la variable net, donde el nombre del canal es ch y los parámetros son 1,2, y 3.

Por otro lado, la transición de la red subordinada debe ser inscrita con un *uplink*. Un *uplink* sirve las solicitudes de cualquier red. Una transición que es llamada a través de un *uplink*, no necesita saber la identidad del iniciador. Un ejemplo de *uplink* es, ch:(x,y,z), que indica que el nombre del canal es ch y que los tres parámetros deben corresponder al ligado de las variables x, y, y z.

Los uplinks y downlinks de una transición pueden estar dados como inscripciones individuales o como una lista separada por punto y coma en una misma inscripción. La lista también puede incluir inscripciones de creación y de acción simultáneamente, junto con las invocaciones de canal.

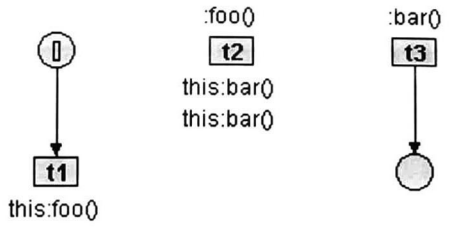


Figura A.4: Red con múltiples canales síncronos

Figura A.5: Redes con parámetros

Generalmente, las transiciones con un *uplink* no pueden dispararse sin ser explícitamente solicitadas por una transición con un *downlink* correspondiente. Una transición sin un *uplink*, es una transición espontánea.

Es permitido que una transición tenga múltiples *downlink*. También es permitido que una transición tenga un *uplink* y varios *downlink*. Sin embargo, no es posible tener una transición con múltiples *uplink*. Esto es ejemplificado en la figura A.4.

La transición t1 inicia la sincronización con la llamada this:foo(). La transición t2, inscrita con el canal :foo(), sirve la solicitud de la transición t1, al mismo tiempo que solicita sincronización con la transición t3 al invocar el canal bar dos veces.

En general, múltiples niveles de sincronización son sospechados desde el punto de vista metódico, sin embargo, algunos deben ser evitados, por ejemplo, las llamadas recursivas o ciclos; no es deseable usar los canales para implementar estructuras de control.

Los canales síncronos simulan el paso de mensajes, por lo que pueden manejar una lista de parámetros. Aunque hay una dirección de invocación, esta dirección no necesariamente debe coincidir con la dirección de la información que es transferida. Por lo tanto, es posible que una sincronización simple transfiera información en ambos sentidos. La figura A.5 muestra una aplicación, donde la transición de la izquierda consulta una tabla que es administrada por la instancia de red de la derecha. Los parámetros (x,y) y (a,b) corresponden x=a y y=b.

Los dos ejemplos previos ilustran sincronización local dentro de una red (dos piezas de red editadas sobre una misma ventana); pero es posible, editar por separado las estructuras de red, y establecer una comunicación entre ellas. Como ejemplo, se considera la figura A.3.

Usando Renew

Renew ofrece una interfaz gráfica de usuario multi-ventana, muy amigable para dibujar

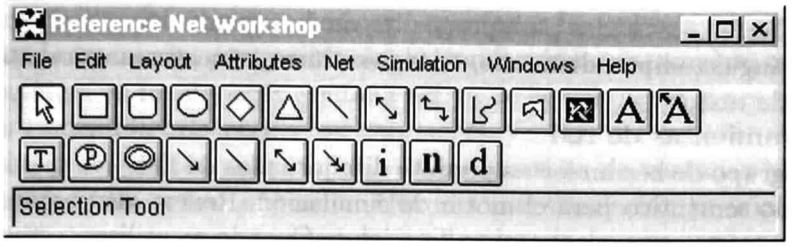


Figura A.6: Barra de herramientas de Renew

Tipo de figura	Doble click	Botón derecho
rectángulo, elipse ...	seleccionar	seleccionar y arrastrar
texto	seleccionar	editar texto
texto conectado	seleccionar nodo	editar texto
transición	seleccionar inscripciones	agregar inscripción :s()
lugar	seleccionar inscripciones	agregar inscripción []
lugar virtual	seleccionar lugar asociado	agregar inscripción []
arco	seleccionar inscripciones	agregar inscripción x
declaración	seleccionar	editar texto
inscripción, nombre, etiqueta	seleccionar nodo	edita texto
transición (instancia)	abrir ventana de ligado	disparo ligado arbitrario
lugar (instancia)	seleccionar marcado	abrir ventana de marcado actual
marcado (cardinalidad)	seleccionar lugar (instancia)	mostrar marcado (marcas)
marcado (marcas)	seleccionar lugar (instancia)	mostrar marcado (cardinalidad)

Figura A.7: Tabla: operaciones de selección

redes referencia y elementos gráficos auxiliares. El editor de red de Renew esta acompañado de una serie de herramientas.

Las capacidades de dibujo han sido tomadas de la librería JHotDraw de Java, y las figuras del editor y las herramientas han sido implementadas por el equuipo de Renew.

Renew trabaja sobre dibujos. Un dibujo consiste de varios elementos, llamados figuras. Cada dibujo es desplegado en una ventana por separado. Diferentes ventanas pueden estar abiertas al mismo tiempo, pero una sola ventana de dibujo puede estar activa.

Renew tiene una barra de herramientas (ver figura A.6) que le permiten crear, editar y manipular cierta clase de figuras. La barra de herramientas agrupa las figuras en tres clases de herramientas: *herramientas de selección*, *herramientas de dibujo*, y *herramientas de red.*

Herramientas de selección

No están relacionadas a ningún tipo de figura. Cualquier tipo de figura puede ser seleccionada y movida a través de esta herramienta. El sumario de operaciones de selección se muestra en la tabla A.7.

Herramientas de dibujo

Crean y manipulan figuras. Estas figuras no tienen significado semántico para el simulador

de red. Pueden ser usadas para propósitos de documentación e ilustración. Las herramientas son: rectángulo, elipse, diamante, triángulo, línea, conectores, garabato, polígono, imágen y cuadros de texto.

Herramientas de red

Este grupo de herramientas permite dibujar redes de Petri, lo que implica que tengan un significado semántico para el motor de simulación. Renew, distingue un rectángulo de una transición, aunque pueden parecer lo mismo. Cuando se utilizan este tipo de herramientas, algunas restricciones sintácticas son analizadas.

- *Transición.* Son creadas con un tamaño fijo. Una transición puede crear nuevos arcos de salida, desde el centro de la figura (pequeño círculo azul).

- *Lugar.* Es analógo a la transición, sólo que crea arcos de entrada, desde el círculo azul ubicado en el centro de la figura.

- *Lugar virtual.* Es usado para crear copias virtuales de un lugar. Esto mejora la confiabilidad y la apariencia de las redes, cuando ciertos lugares son usados por muchas transiciones. Sí el contenido de un lugar es necesario para varias transiciones, la confiabilidad de la red decrece por el cruce de arcos.

- *Arco.* Existen diferentes tipos de arco: arco normal, arco de prueba, arco de reserva, arco flexible, arco, limpiador, arco inhibidor.

- *Inscripción.* Las inscripciones son un ingrediente importante en los formalismos de redes de Petri de alto nivel. Una inscripción es una pieza de texto que es conectada a un elemento de red (lugar, transición, o arco). Los lugares pueden ser inscritos con tipos y marcados iniciales. Las inscripciones en los arcos determinan el tipo de marcas que serán movidas. Las transiciones pueden inscribir guardas, acciones, downlinks, uplinks, y expresiones. Múltiples inscripciones pueden ser asignadas a un sólo elemento de red, separadas por puntos y comas.

- *Nombre.* El nombre sólo puede ser asignado a lugares y transiciones, y es reconocido por la letra estilo bold. La idea de un nombre es aumentar la confiabilidad de la red cuando esta es simulada. Si una transición se dispara, su nombre es impreso en la consola. De igual forma, si un lugar remueve o pone marcas en un lugar su nombre se despliega en la ventana de la consola. Además, el nombre de un lugar es usado en el título de la ventana del marcado actual, y el nombre de una transición aparece en la ventana de ligado.

- Declaración. Una declaración es necesaria sólo si se decide usar tipos. Cada figura debe tener a lo más una declaración. El texto de la declaración tiene un significado para el simulador y es usado para declarar variables y sentencias. El contenido de una declaración es verificado tan pronto como se termina de editar.

Simulación de redes

Durante la simulación hay un retro-alimentación gráfica y textual. En la consola de Java se imprime un archivo bitácora de la simulación. Este archivo muestra exactamente que transiciones fueron disparadas y que marcas fueron consumidas y producidas. Pero también, se puede visualizar el estado de varias instancias de red gráficamente e influenciar la simulación.

Antes de que la simulación inicie, es necesario que todas las redes involucradas en el sistema sean cargadas en memoria, para evitar que la simulación no se complete correctamente. Otro punto importante, es mantener la ventana activa de la red que se desea instanciar.

Ventanas de instancias de red

La retro-alimentación gráfica de instancias de red consiste de ventanas especiales, que contienen instancias de las redes referencia. Cuando una simulación se inicia, la primera instancia de la red referencia que es generada, es desplegada en una ventana de instancia de red. En el archivo de bitácora de la simulación, el nombre de una instancia de red esta compuesto por, el nombre de red junto con un número entre corchetes. Por ejemplo, Net [1]. La ventana de instancia de red puede ser reconocida por su color de fondo (lila), lo cual impide confundirla con la ventana donde la red fue editada.

En una ventana de instancia de red, no es posible editar la red, y mucho menos seleccionar los elementos de red.

La red es una çapa de fondoτ sólo los objetos relevantes de la simulación pueden ser seleccionados, como el marcado actual de los lugares y las instancias de las transiciones. Los lugares en la ventana de instancia de red son anotados con el número de marcas que contienen. Hacer doble click en un marcado, muestra una ventana con información detallada acerca de las marcas.

Es posible desplegar el contenido del marcado actual directamente dentro de la ventana de instancia de red.

Ventana del marcado actual

Una ventana de marcado actual muestra el nombre del lugar correspondiente en el título de la ventana (nombre de la instancia más el nombre del lugar, ejemplo: Net[1].Place).

Hay una función especial para accesar a otras instancias de red. Si una marca contiene una instancia de red, un marco azul aparece alrededor del nombre de la instancia de red. Por lo tanto, al dar click dentro del marco, una nueva ventana de instancia de red es abierta, o la instancia de red correspondiente es activada, si ésta ya existe. Esto también aplica para las referencias de red contenidas dentro de una tupla.

Control de la simulación

En un sistema concurrente, muchas transiciones pueden ser activadas a la vez. Normalmente, el motor de la simulación decide cuales de éstas transiciones se disparan en el siguiente paso de simulación.

La simulación interactiva es posible. Se puede forzar el disparo de una transición habilitada, en dos maneras:

1. Botón derecho sobre la transición. El disparo es manual y el motor de simulación aún decide no deterministicamente los ligados de las variables.

2. Doble click sobre la transición. En este caso la ventana de selección de ligados aparece, dando la oportunidad de elegir entre los ligados. El título de la ventana esta compuesto por: nombre de la instancia de red, nombre de la transición seguido de possible bindings. Cada instancia de transición que participa en el ligado es mostrada en una sola línea, listando aquellas variables que ya están ligadas.

Hay situaciones donde una transición no puede ser disparada manualmente, aunque este activada. Esto ocurre para todas las transiciones inscritas con un *uplink*, lo cual se debe a

que una transición con un *uplink* esta esperando una solicitud de sincronización de cualquier otra transición con el correspondiente *downlink*.

Una simulación inicia con el comando Run Simulation, y termina con el comando **Terminate Simulation**. Si se desea simular paso a paso, entonces utilizar el comando **Simulation Step**.

Centro de Investigación y de Estudios Avanzados del IPN
Unidad Guadalajara

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, aprobó la tesis:

MODELADO MULTINIVEL DE SISTEMAS DE PROCESOS POR LOTES

del (la) C.

Norma Isabel VILLANUEVA PAREDES

el día 15 de Diciembre de 2003.

Dr. Antonio RAMIREZ TREVIÑO
Investigador Cinvestav 2B
CINVESTAV GDL
Jalisco

Dr. Félix Francisco RAMOS CORCHADO
Investigador Cinvestav 2B
CINVESTAV GDL
Jalisco

Dr. María Elena MEDA CAMPAÑA
Profesor-Investigador Titular A --
Universidad de Guadalajara,
CUCEA, Dpto. de Sistemas de Información
Jalisco