

Universitat Rovira i Virgili

ENGINYERIA TÈCNICA EN INFORMÀTICA DE SISTEMES

**ABSTRACCIÓN DE PROCESOS QUÍMICOS MEDIANTE UN
SISTEMA EXPERTO**

MEMORIA PRESENTADA POR:

DAVID SANJUAN GARCÍA

DIRECCIÓN DE PROYECTO:

**IVAN LÓPEZ ARÉVALO
DRA. AÍDA VALLS MATEU**

1	INTRODUCCIÓN	2
1.1	Planteamiento del problema:	2
1.2	Metodología del rediseño	2
1.3	Objetivo.....	3
2	REDISEÑO PROPUESTO DE PLANTAS QUÍMICAS	3
2.1	Metodología	3
2.1.1	Análisis del diagrama de flujo.....	4
2.2	Especificaciones	5
2.2.1	Representación del conocimiento obtenido por el módulo HEAD.....	5
2.2.2	Metodología de abstracción (jerarquía de niveles)	6
2.2.3	Generación de niveles de abstracción.....	9
3	DECISIONES DE DISEÑO	9
3.1	Sistemas expertos	9
3.1.1	Estructura de un sistema experto.....	10
3.1.2	Aplicaciones de un sistema experto.....	11
3.1.3	Lenguajes utilizados en sistemas expertos	12
3.1.4	Implementación de un sistema experto:	13
3.2	Jess.....	13
3.2.1	Descripción.....	14
3.2.2	El algoritmo de RETE	14
3.2.3	Formas de utilizar Jess	14
4	DESARROLLO	15
4.1	Estructura y funcionamiento de la aplicación:	15
4.2	Comunicación entre el módulo en JAVA y el sistema experto.....	16
4.3	Relación de clases JAVA más destacadas:	17
4.4	Archivos con plantillas y reglas.....	18
4.5	Abstracción	21
4.5.1	Reglas para la eliminación de reciclos	21
4.5.2	Casos de abstracción en nivel FLOW.....	22
4.5.3	Casos de abstracción en nivel PRESSURE	22
4.5.4	Casos de abstracción en nivel TEMPERATURE (coolers, heaters, air_coolers)	25
4.5.5	Casos de abstracción en nivel TEMPERATURE (Heat exchanger)	25
4.5.6	Casos de abstracción en nivel SEPARATION	28
4.5.7	Casos de abstracción en nivel REACTION.....	29
4.5.8	Reglas para la abstracción de los niveles FLOW, PRESSURE y TEMPERATURE (heaters, coolers y air coolers).....	29
4.5.9	Reglas para la abstracción del nivel TEMPERATURE (heat exchanger).	30
4.5.10	Reglas para la abstracción del nivel SEPARATION.	33
4.5.11	Reglas para la abstracción del nivel REACTION.....	34
5	EVALUACIÓN	35
5.1	Proceso amoníaco	35
5.2	Proceso dimetyl ether.....	40

5.3	Proceso ethanol.....	44
6	CONCLUSIONES	48
7	REFERENCIAS	48

1 INTRODUCCIÓN

1.1 Planteamiento del problema:

Este proyecto forma parte de una investigación realizada por la Universitat Rovira i Virgili en la que colaboran el Grupo de Investigación de Inteligencia Artificial (Banzai) del Departament d'Enginyeria Informàtica i Matemàtiques y del Grupo de Diseño y Síntesis de Procesos (GDSP) del Departament d'Enginyeria Química. La investigación trata de facilitar el rediseño de plantas químicas mediante la utilización de modelos múltiples (multimodeling) (modelos estructura, comportamiento, función y teleología), con esa finalidad están en desarrollo tres módulos HEAD, AHA y RETRO. A continuación se describen brevemente las funciones que debe cumplir de cada módulo:

1. HEAD (HYSYS™ Extraction Data). Es el módulo encargado de extraer información (Knowledge Acquisition) de procesos químicos desde el simulador de procesos HYSYS™. Este módulo ha sido desarrollado en el grupo de Diseño y Síntesis de Procesos del Departament d'Enginyeria Química de la URV.
2. AHA (Automatic Hierarchical Abstraction tool) Es el módulo para la abstracción y análisis (Data Abstraction y Data Analysis) de procesos químicos.
3. RETRO (Reverse Engineering Tool for Retrofit Options). Es el módulo que se encargará de la generación de alternativas de proceso.

1.2 Metodología del rediseño

Los procesos químicos industriales son revisados periódicamente con la finalidad de realizar mejoras en:

- Capacidad de costes de producción.
- Eficiencia, incremento en la calidad del producto.
- Consumo de energía.
- Reducción de la contaminación ambiental.
- Uso de nuevas tecnologías.
- Operatividad.
- Seguridad y control de los procesos.

La utilización de una metodología para la realización de un análisis y evaluación del proceso, se hace necesaria debido a la excesiva información y complejidad de éste.

La metodología consiste en razonar a niveles abstractos sobre la función de los equipos, facilitando así la obtención de información funcional y teleológica ya que, en cada nivel, se omiten detalles de la implementación del proceso que son irrelevantes para su estudio.

1.3 Objetivo

El objetivo de este proyecto es implementar parte del módulo AHA que se encargue de la generación automática de jerarquías de plantas químicas. La forma elegida para realizar este proceso será un sistema experto que realizará la abstracción del proceso en diferentes niveles según la funcionalidad de los equipos que haya en éste.

2 REDISEÑO PROPUESTO DE PLANTAS QUÍMICAS

Descripción de la metodología de rediseño de plantas químicas propuesta por de Diseño y Síntesis de Procesos (GDSP) del Departament d'Enginyeria Química.

2.1 Metodología

En el rediseño de plantas químicas hay que tener en cuenta varios factores:

- Que las secciones de proceso se puedan identificar de forma parecida a como se concibieron inicialmente durante su diseño.
- La representación de su diseño sea consistente con su tarea.
- Aprovechamiento de experiencias pasadas.
- Probar las alternativas más idóneas.

Las metodologías de diseño propuesta es una extensión de las metodologías de Douglas y Turton adicionando características de otras metodologías. El resultado es una metodología que es soportada por un enfoque de modelos múltiples y que cumple con los puntos expuestos anteriormente.

El enfoque de modelos múltiples permite modelar jerárquicamente el proceso químico y hacer razonamientos basándose en la funcionalidad. El objetivo es permitir una representación estructurada y jerarquizada del proceso químico en diferentes niveles de abstracción. Esto permite al que rediseña un proceso químico, la comprensión del proceso así como una obtención de sus diferentes secciones a distintos niveles de detalle.

La metodología propuesta se divide en las siguientes etapas:

1. Análisis del diagrama de flujo: La descripción del diseño se obtiene a partir del diagrama de flujo del proceso químico. La realización de un estudio de los componentes del diagrama de flujo se hace necesario para su comprensión.
2. Diagnóstico: Cuando la información del diagrama de flujo se ha representado en diferentes niveles de abstracción, se hace un estudio en busca de secciones que puedan ser sometidas a

rediseño según los nuevos requerimientos del proceso y los criterios de rediseño.

3. Obtención de alternativas: En la búsqueda de un rediseño de la planta química más acertado, en ocasiones se obtendrán secciones parecidas de otros diagramas de flujo que cumplan con los requerimientos funcionales de la sección original.
4. Evaluación de alternativas. El diagrama de flujo alternativo debe cumplir con los nuevos requerimientos impuestos al rediseño.

Este proyecto se centra en la primera de las etapas mencionadas anteriormente

2.1.1 Análisis del diagrama de flujo

La obtención de la descripción del proceso químico es el punto de partida para la tarea de rediseñar. Este proyecto realiza esta tarea por medio de modelos múltiples, lo cual permite representar el conocimiento sobre el proceso en diferentes niveles de abstracción. De esta manera se hace posible la abstracción del proceso de la manera propuesta por Douglas y Tourton.

El análisis de flujo se divide en dos etapas:

- Extracción de información: Se obtiene la información del diagrama de flujo a partir de un simulador numérico externo debido a la alta complejidad de los procesos químicos.
- Análisis funcional: Todo el conocimiento obtenido a partir del simulador numérico se modela según la funcionalidad de las unidades del diagrama de flujo. La etapa se divide en dos:
 - Identificación funcional: Se identifican las funciones de los equipos del proceso.
 - Identificación de secciones: De una forma similar a las propuestas por Douglas y Tourton, basándose en la funcionalidad de los equipos, se identifican las secciones del diagrama de flujo.

2.1.1.1 Extracción de la información

Según la metodología utilizada el diagrama de flujo obtenido debe ser por medio de un simulador de procesos químicos. Es una forma de evitar un posible fallo humano ya que el rediseñador podría introducir datos imprecisos o incorrectos.

El simulador se centra en dos puntos a la hora de obtener información de los equipos con el fin de construir el diagrama de flujo:

- Estructural, es información sobre el tipo del equipo y cómo se conecta con el resto de equipos.
- Comportamiento, el conocimiento sobre los valores de sus variables y parámetros que configuran su estado.

2.1.1.2 Análisis funcional

En el proceso de modelado jerárquico del diagrama de flujo se deben obtener las funciones de los equipos que componen el proceso. Éstas servirán para identificar las secciones funcionales del diagrama de flujo.

El análisis funcional se divide en dos partes :

1. Identificación funcional: Para la identificación de la función de un equipo se requiere del estudio de sus entradas, salidas y su entorno. Una función puede ser desarrollada por más de un equipo y por el contrario un equipo puede desarrollar más de una, por lo que las funciones se pueden clasificar en :
 - Función general: Función que puede ser desarrollada por varios equipos.
 - Función específica: Es aquella para la que se ha diseñado un equipo explícitamente.

En concreto a partir del simulador de procesos obtenemos información del tipo :

- Funcional: acerca de los roles que desempeñan las unidades en el proceso.
- Teleológica: Informa sobre la finalidad de los equipos en el proceso.

A las unidades modeladas según su estructura, comportamiento, función y teleología se las llama “units”.

2. Identificación de secciones:

Por la metodología empleada es posible la agrupación de units según su función generando bloques de units denominadas metaunits. Las metaunits pueden contener tanto units como otras metaunits. En la generación de estas metaunits se consideran los modelos funcional y teleológico de las units y metaunits involucradas.

2.2 Especificaciones

2.2.1 Representación del conocimiento obtenido por el módulo HEAD.

El conocimiento obtenido por el módulo HEAD se representa de la siguiente manera:

Conocimiento fundamental:

- Estructural, es el conocimiento acerca de la topología de un objeto. Comprende objetos físicos, nodos y conexiones.

- Comportamiento, es el conocimiento acerca de cómo operan e interactúan los dispositivos con otros dentro del mismo proceso.

Conocimiento interpretativo:

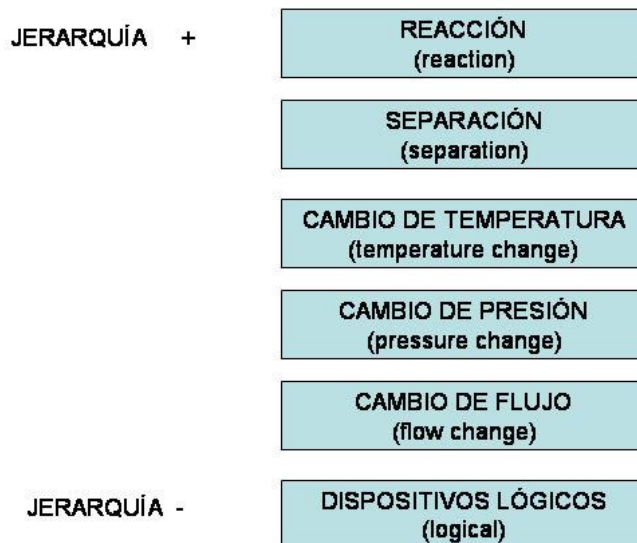
- Funcional, es el conocimiento sobre la tarea que realiza un dispositivo dentro del proceso como por ejemplo pueda ser la separación .
- Teleológico, es el conocimiento sobre la finalidad de un dispositivo asignado por el diseñador y de las condiciones operacionales.

Un proceso puede ser representado de dos formas diferentes, una a nivel de subsistemas y la otra a nivel de componentes elementales.

2.2.2 Metodología de abstracción (jerarquía de niveles)

La metodología de abstracción se basa en las reglas de precedencia y el nivel 1 de las funciones jerárquicas. Las reglas de precedencia están definidas de la siguiente forma:

Jerarquía de niveles



Siendo “Reacción” la de más alta jerarquía y “Cambio de flujo” la de menor.

Como resultado de la abstracción de dos unidades de proceso se generará una “META-Estructura”.

1.DISPOSITIVOS LOGICOS (LOGICAL).

De la información que se obtiene del diagrama de flujo de proceso (DFP) de Hysys, se cuenta con los dispositivos lógicos. Este tipo de dispositivos contienen información sobre las necesidades del proceso relacionadas con balances de materia y controlabilidad.

Level 1 General Function	Level 2 Function Required	Level 3 Broad classification of equipment
Logical	Bal-control	Adjust
		Balance
		Recycle
	Var-control	Digital Point
		PID Controller
		MPC Controller
		Transfer Function
		Selector Block
		Set
		Spreadsheet

2. CAMBIOS DE FLUJO (FLOW CHANGE).

Este tipo de dispositivos tienen como función general la mezcla o división de corrientes de proceso.

Level 1 General Function	Level 2 Function Required	Level 3 Broad classification of equipment
Flow change	Mixing	Mixer
	Splitting	Split

4. CAMBIOS DE TEMPERATURA (TEMPERATURE CHANGE).

Estas unidades de proceso tienen como función general un cambio de temperatura.

Level 1 General Function	Level 2 Function Required	Level 3 Broad classification of equipment
Temperature change	Heating	Heater
		Heat-exchanger

		Reboiler
		3Phase-reboiler
	Cooling	Air-cooler
		Condenser
		Cooler
		Heat-exchanger

5. SEPARACION (SEPARATION).

Estas unidades de proceso tienen como función general la separación de la materia mediante cambios físico-químicos.

Para este tipo de unidades de proceso existen varias posibilidades de agrupamiento:

- No hay reactores en el DFP: Los equipos de separación se agruparán en una estructura denominada META-Separación. Esta estructura será el último nivel de abstracción que se pueda obtener del DFP en cuestión.
- Si hay reactores en el DFP: Los equipos de separación se agruparán formando la estructura denominada META-Separación antes de agruparse a los sistemas de reacción.

Level 1 General Function	Level 2 Function Required	Level 3 Broad classification of equipment
Separation	Extraction	Liquid-liquid-extractor (LLE)
	Distillation	Flash
		3 Phase separator
		Trayed
		Packed
	Absorption	Vapor-absorption-tower
	Stripping	Liquid-absorption-tower
	Adsorption	Gas-adsorption-tower
		Liquid-adsorption-tower
	Crystallization	
	Leaching	
	Drying	
	Membranes	

6. REACCION (REACTION).

Estas unidades de proceso tienen como función general la transformación de la materia mediante reacciones químicas.

Los reactores formarán estructuras denominadas META-Reacción las cuales serán el último nivel de abstracción de un proceso determinado.

Level 1 General Function	Level 2 Function Required	Level 3 Broad classification of
-----------------------------	------------------------------	------------------------------------

		equipment
Reaction	Reactor	Continuous Stirred Tank Reactor
		Conversion reactor
		Equilibrium reactor
		Gibbs reactor
		Tubular Reactor

2.2.3 Generación de niveles de abstracción.

La generación de niveles de abstracción se rige por reglas de agrupamiento y una jerarquía de precedencia. Las reglas provocan que unas unidades de proceso generen metaunidades como resultado de la agrupación de dos o más de ellas.

La jerarquía de precedencia es el orden en el que se agrupan las unidades basándose en la función general que representan. En particular para este proyecto, las unidades cuya función sea reaction, tienen mayor jerarquía que las de separation y éstas a su vez mayor que las de temperature y así sucesivamente.

En las abstracciones se van eliminando detalles de las unidades abstraídas. Durante estas abstracciones se generan Metaunidades que son unidades con información básica que no contiene información detallada de las unidades involucradas. Por otra parte estas metaunidades también pueden generar nuevas Metaunidades. El conocimiento funcional de la Metaunidad predomina sobre los demás, ya que en base a éste, se identifica la sección del proceso que representa dicha Metaunidad. Aunque los conocimientos estructural, de comportamiento y teleológico de una Metaunidad son diferentes de las unidades que abstraen, ésta Metaunidad conserva la misma funcionalidad de dichos equipos, es decir, representa la misma sección de proceso en el nivel más detallado.

3 DECISIONES DE DISEÑO

Como parte importante de las decisiones de diseño está el hecho de escoger un sistema experto como herramienta de generación de los niveles de abstracción del proceso. Con el objetivo de dar el porqué de su implementación en este proyecto, se procede a dar una breve explicación de estos.

3.1 Sistemas expertos

El sistema experto es un programa computacional que reproduce el proceso intelectual de un experto humano en un dominio concreto. En la siguiente tabla se muestran las diferencias entre un sistema clásico y un sistema experto.

Sistema clásico	Sistema experto
Conocimiento y procesamiento mezclados	Conocimiento y procesamiento separados
No contiene errores	Puede contener errores
Los cambios son complicados	Cambiar reglas es sencillo
No da explicaciones mientras está procesando	Capacidad de explicar su proceso de razonamiento
El sistema sólo funciona completo	El sistema puede funcionar con pocas reglas
Ejecución paso a paso	La ejecución usa heurística y lógica
Necesita información completa	Puede operar con información incompleta
Representa y usa datos	Representa y usa conocimiento

3.1.1 Estructura de un sistema experto

Los sistemas expertos están constituidos por:

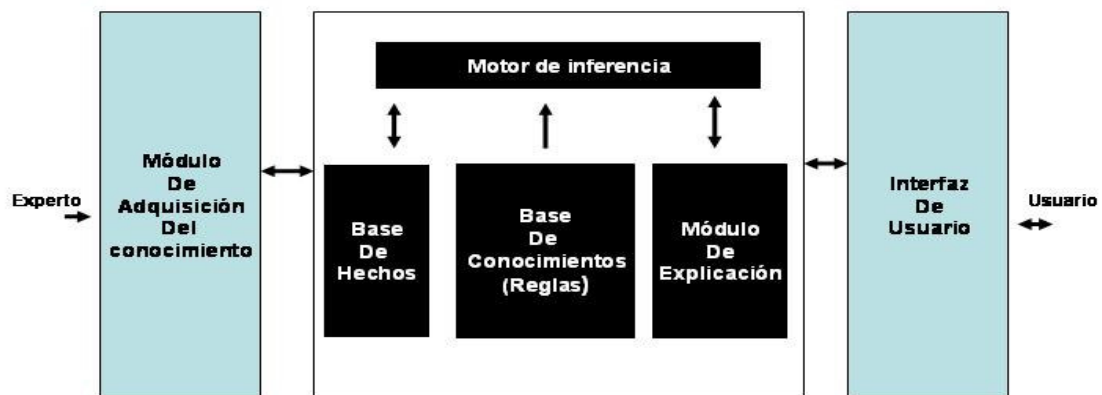
- Base de conocimiento: Es la parte que contiene el conocimiento del dominio en el que se trabaja. La representación de éste puede tomar varias formas:
 - Marcos (Frames): son estructuras de datos donde se almacena información concreta de un cierto concepto e información relacional para completar la definición del concepto. Están representados por una serie de campos y los valores asociados a los mismos, pudiendo ser otro "frame".
 - Redes semánticas: son representaciones gráficas del conocimiento, mediante nodos, que representan objetos o conjuntos de objetos, y arcos, que relacionan dichos objetos. Tanto los "frames" como las redes son representaciones descriptivas, difíciles de usar para representar el razonamiento.
 - Reglas: Es la más utilizada en sistemas expertos. Una regla es una estructura condicional que relaciona lógicamente información contenida en la parte del antecedente con otra información contenida en la parte del consecuente.
- Base de hechos: Representa el conocimiento del sistema en un instante. De hecho es la memoria de trabajo donde se almacena la información y los datos de entrada. También se registran las conclusiones intermedias y los datos generados en el proceso de inferencia, lo que facilita el poder dar una explicación del razonamiento del sistema.
- Motor de inferencia: Es el módulo que se encarga de modelar el razonamiento humano, por lo que trabaja con la información de la base de conocimiento y la base de hechos. Este módulo busca y selecciona las reglas a utilizar en el proceso de razonamiento.

El tipo de razonamiento utilizado puede ser de dos tipos:

- Encadenamiento hacia delante (forward chaining): Basada en el método deductivo, es una estrategia guiada por los datos, es decir, a partir de los datos disponibles se ejecutan reglas hasta obtener una conclusión o varias. Su esquema de funcionamiento es el siguiente:
 1. Inicializar base de hechos con los datos conocidos.
 2. Obtener hechos derivables a partir de los iniciales.

- Detectar reglas aplicables.
 - Solucionar una.
 - Añadir lo consecuente con la base de hechos
3. Repetir el paso 2 hasta que se encuentre el objetivo o hasta que no hayan más reglas por aplicar.
- Encadenamiento hacia atrás (backward chaining): Contraria a la anterior y basada en el método inductivo. Es una estrategia guiada por los objetivos, en este caso lo que se trata es de averiguar si una hipótesis cumple unas premisas.
- Módulo de explicación: Es un módulo muy útil de los sistemas expertos ya que da información al usuario del porqué el sistema ha llegado a una conclusión, mientras que para el diseñador supone una herramienta de detección de errores.
 - Interfaz con el usuario: Establece una comunicación sencilla entre el usuario y el sistema. Un usuario puede realizar consultas a través de menús, gráficos, etc.

Estructura sistema experto



3.1.2 Aplicaciones de un sistema experto

Los sistemas expertos son útiles en diversos modelos funcionales:

- Interpretación: Explicar el sentido o significado de los datos como por ejemplo en análisis financieros, análisis de imágenes, reconocimiento del habla, etc.

- **Predicción:** Obtener situaciones futuras a partir de los hechos actuales. Prevención del tráfico, evolución de la bolsa, predicciones meteorológicas.
- **Diagnóstico:** Se realizan una serie de preguntas en el sistema sometido a estudio con el fin de buscar posibles fallos a partir de las respuestas obtenidas. Algunos sistemas expertos han sido implementados para realizar diagnósticos de circuitos electrónicos. También destacan los del sector de la medicina como el famoso MYCIN realizado en 1972 para el diagnóstico de enfermedades infecciosas.
- **Diseño:** El diseño en muchas ocasiones requiere un gran conocimiento debido a las especificaciones y restricciones. Los sistemas expertos ayudan a los diseñadores a realizar el trabajo dentro de unos límites de tiempo y costes. Se usan en campos como en la electrónica diseñando circuitos, arquitectura, etc.
- **Planificación:** Cuando se quiere llegar a una meta realizando una serie de acciones. Por ejemplo en planificación militar, planes de producción de fábricas, configuración de grandes sistemas informáticos.
- **Monitorización:** Supervisión y observación del sistema. Por ejemplo en centrales nucleares, plantas químicas, etc.
- **Depuración:** Realizar una serie de acciones con el fin de hacer el sistema más eficiente. En circuitos electrónicos, en desarrollos de software, etc.
- **Reparación:** Reparar el sistemas con los datos disponibles, como los sistemas informáticos, automoción.
- **Instrucción:** Diagnóstico, depuración y corrección de una conducta. Útiles para la enseñanza.
- **Control:** Interpretando prediciendo y supervisando las acciones de un sistema. En controles aéreos, estrategias militares, etc.
- **Enseñanza:** Aprendiendo de los datos (experiencia).

3.1.3 Lenguajes utilizados en sistemas expertos

Los dos lenguajes más destacados a la hora de implementar sistemas expertos son LISP y PROLOG.

Lisp (List-processing): Fue desarrollado a finales de los años 50. Se hizo muy popular por su capacidad de manejar símbolos y fue muy utilizado en sistemas de inteligencia artificial. Durante estos años, se han ido desarrollando varios dialectos MACLISP, COMMONLISP, INTERLISP, ZETALISP, donde el COMMONLISP es el que se está imponiendo.

Conceptos característicos de lisp:

- **Listas y Átomos:** Está basado en el uso de listas. Los átomos son números, cadenas de caracteres o símbolos. Un símbolo puede tener varios valores, al igual que una variable en otros lenguajes de programación, como por ejemplo un número, o también puede ser el nombre de una función, o incluso ambos. Además a un símbolo pueden subordinarse cualidades, que además del valor del símbolo, contienen información adicional. Estas cualidades también reciben el nombre de atributos. Los átomos pueden subordinarse a cualidades.
- **La Función:** Cada función LISP y cada programa LISP tiene estructura de lista. Los programas no pueden distinguirse sintácticamente de los datos. LISP ofrece sus propias funciones básicas.

- **Forma de Trabajo:** LISP es un lenguaje funcional. Ofrece la posibilidad de realizar definiciones recursivas de funciones. La unión de procedimientos se realiza de forma dinámica, es decir en plena ejecución, y no como en otros lenguajes de programación. El sistema realiza automáticamente una gestión dinámica de memoria.

Prolog (Programming in Logic): Este es otro de los lenguajes más utilizados en inteligencia artificial. Desarrollado en Francia en 1973, que más tarde en la Universidad de Edimburgo, desarrollaría la teoría de la programación lógica. La sintaxis propuesta en Edimburgo se considera el estándar de facto de PROLOG.

Los mecanismos de PROLOG son: Recursividad, instanciación, verificación, unificación, backtracking e inversión.

3.1.4 Implementación de un sistema experto:

Existen cuatro variantes para implementar un sistema experto:

- **Lenguajes de alto nivel:** Los lenguajes de alto nivel son de uso muy extendido ya que son fáciles de aprender y presentan una gran portabilidad pero tienen un gran inconveniente a la hora de implementar sistemas expertos y es que la mayoría no están preparados para una programación basada en símbolos.
- **Lenguajes simbólicos:** Los dos lenguajes más destacados son LISP y PROLOG. Son lenguajes de alto nivel que presentan unas características que se ajustan a los problemas planteados en inteligencia artificial por su forma de obtener conclusiones y por su adaptación a la lógica de la base de conocimientos representada mediante símbolos.
- **Shells de sistemas expertos(herramientas de desarrollo):** Son programas especializados en implementar sistemas expertos. Incorporan el motor de inferencia, el módulo explicativo y a veces una interfaz de usuario, mientras que la base de conocimiento está vacía.
- En ocasiones el diseñador debe implementar parte del motor de inferencia pero por lo general solo debe preocuparse de la construcción de la base de conocimiento.
- **Entornos de desarrollo:** Añaden a los shells un entorno gráfico haciendo más fácil desarrollar un sistema experto. En contra, hay que tener en cuenta que son poco flexibles y suelen tener un alto coste.

3.2 Jess

El módulo AHA! está desarrollado íntegramente en JAVA. El hecho de decidir implementar un sistema experto que se encargue de realizar la abstracción, abre un abanico de posibilidades sobre las diferentes formas de implementación tanto en lenguajes de programación, shells y entornos de desarrollo.

Jess es una shell de sistemas expertos que además está provista de un lenguaje de script. La herramienta está escrita totalmente en el lenguaje JAVA de SUN por Ernest Friedman-Hill en Sandia Nacional Laboratorios en Livermore, Canadá. Esta situación permite diseñar aplicaciones que incluyen sistemas expertos totalmente en JAVA y es por este motivo por el cual Jess ha sido elegido para la realización del sistema experto que irá incluido en el módulo AHA!.

3.2.1 Descripción

Jess fue originalmente inspirado por CLIPS otra shell de sistemas expertos. De hecho el núcleo de Jess todavía es compatible con el de CLIPS y muchas de las scripts escritas en Jess funcionan en CLIPS i viceversa.

Jess utiliza un algoritmo llamado RETE para procesar las reglas y añade varias características respecto a CLIPS. Entre otras incluye backwards chaining, búsquedas en la memoria de trabajo y la capacidad de manipular directamente objetos JAVA. Jess es un potente entorno de scripts desde el cual se pueden crear objetos JAVA y llamar a métodos JAVA sin tener que compilar ningún código JAVA.

3.2.2 El algoritmo de RETE

La mayoría de los sistemas expertos tienen un conjunto de reglas fijas y una base de hechos que va cambiando constantemente. Sin embargo es un hecho que generalmente entre la ejecución de una regla y la siguiente la base de hechos cambia muy poco. Aunque nuevos hechos se van añadiendo a la base de hechos y otros se van eliminando, el porcentaje del número de hechos que cambian por unidades de tiempo por lo general es bastante pequeño. La implementación consiste en guardar en una lista las reglas y cíclicamente comprobar el lado condicional de cada regla y ejecutar el lado derecho de algunas. Esto resulta ser muy ineficiente ya que el resultado de dos o más ciclos consecutivos puede ser el mismo. Incluso cuando el sistema está estable muchos tests serán repetidos.

Jess en lugar de utilizar dicha metodología implementa el algoritmo llamado RETE que llega a ser la base de toda la generación de los sistemas expertos rápidos. Entre los que destacan OPS5, ART y CLIPS. Rete intenta minimizar los efectos antes mencionados recordando los resultados de los test pasados a lo largo de los ciclos. Sólo los nuevos hechos son testeados por las reglas. La principal desventaja que presenta es que utiliza bastante memoria, ya que guarda el estado del sistema , por lo cual una regla escrita deficientemente además de procesarse con lentitud, llega a consumir grandes cantidades de memoria.

3.2.3 Formas de utilizar Jess

Existen diferentes formas y grados en los que Jess se puede implementar con JAVA. De las más destacadas:

- Scripts de lenguaje de Jess.
- Scripts de lenguaje de Jess, con accesos al API de JAVA.
- Scripts en Jess con algo de código JAVA en la forma de nuevos comandos de JESS escritos en Java.
- Código JAVA que carga scripts de Jess.
- Código implementado en JAVA que manipula Jess a través del API de JAVA.

4 DESARROLLO

4.1 Estructura y funcionamiento de la aplicación:

El módulo AHA!, se encarga de realizar la abstracción del proceso y guardar los resultados obtenidos de dicho proceso. En el momento en el que el usuario ejecuta la aplicación y abre un proceso, la clase GetInformation extrae la información del proceso a partir del fichero generado con el módulo HEAD (encargado de obtener información a partir del simulador de procesos HYSYS) y toda la información del proceso tanto de los equipos como de los streams, etc, quedará guardada en los vectores de una clase llamada common.

En la clase common es el lugar donde queda almacenada la información del proceso tanto del nivel actual como de los anteriores. De hecho se encuentra un vector que almacena el resultado de todas las abstracciones. El vector al final de realizar todos los niveles de abstracción queda:

Vector : copias de los Niveles

Proceso sin abstraer
Nivel Flow
Nivel Pressure
Nivel Temperature
Nivel Separation
Nivel Reaction

También hay otros vectores que son los encargados de almacenar la configuración del proceso actual. Cada tipo de equipo tiene destinado un vector así como los material streams y los energy streams.

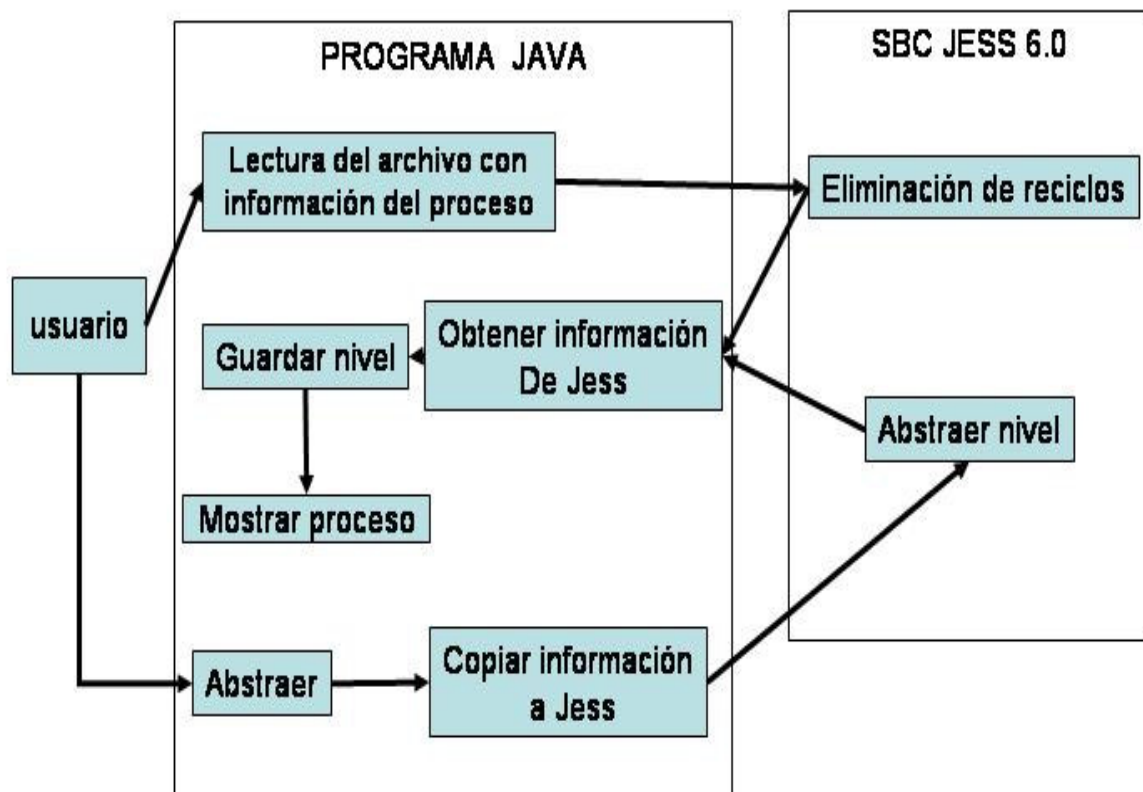
Una vez obtenida la información del proceso ésta se pasa al sistema experto con el fin de que elimine posibles reciclos en su implementación ya que por el momento no se van a tener en cuenta. Una vez eliminados, la información se vuelve a pasar a la aplicación y se guarda en el vector de common destinado a guardar todos los niveles. La información que queda en

este vector, es una copia del resto de vectores que guardan la configuración actual del proceso.

Cada vez que el usuario provoque una abstracción del proceso, la aplicación, enviará la información del proceso actual al sistema experto que realizará la abstracción y una vez terminada la información volverá a la aplicación creando una nueva copia del nivel procesado y mostrándolo por la interfaz de usuario.

En la siguiente imagen se ilustra el funcionamiento general de la aplicación:

Funcionamiento de la aplicación

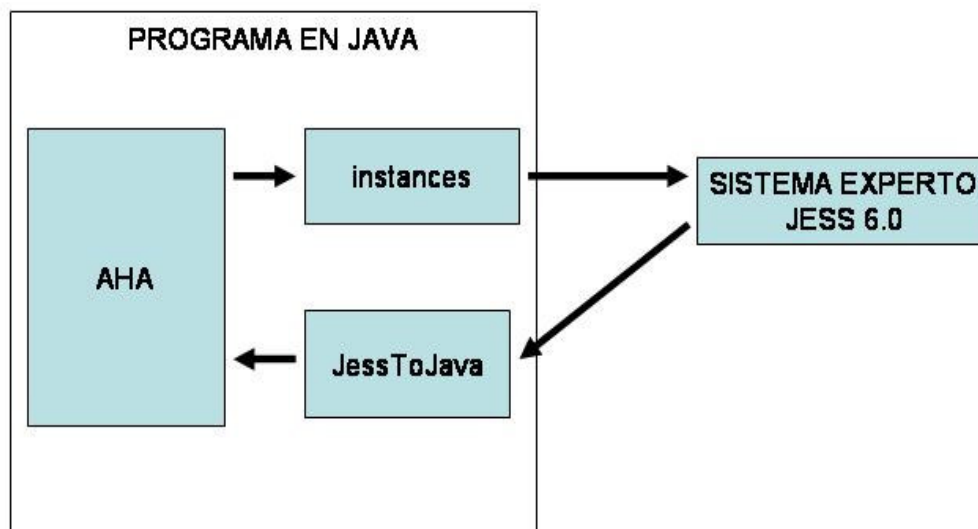


4.2 Comunicación entre el módulo en JAVA y el sistema experto

Para implementar éste módulo, como ya se ha comentado antes se ha implementado en lenguaje JAVA y con la shell de sistemas expertos Jess, la cual está totalmente integrada en JAVA.

En la siguiente imagen se muestra como se establece la comunicación entre el módulo implementado en JAVA y Jess.

COMUNICACIÓN ENTRE JAVA Y JESS 6.0



- Instances.java : Es la clase que se encarga de obtener la descripción del proceso químico y trasladarla a la base de hechos del sistema experto. Es decir copia la información de los equipos del proceso en la base de hechos de Jess. De la información copiada hay que destacar las corrientes entrantes y salientes, las funciones de los equipos y la referencia a los objetos JAVA (equipos de proceso).
- JessToJava.java: Esta es la clase que se encarga de hacer lo contrario que instances. Hace búsquedas de los distintos tipos de equipos por la base de hechos por medio de las "defqueries" que es un mecanismo de Jess que permite la realización de búsquedas directas en la base de hechos. Por cada equipo que encuentra comprueba si ha sido eliminado en la abstracción. En caso afirmativo el equipo es eliminado de los vectores common y en caso contrario se actualiza el equipo. Por último llena unos vectores utilizados por la interficie gráfica para delimitar que equipos guardan conexiones entre ellos y dibujarlos.

4.3 Relación de clases JAVA más destacadas:

- GetInformation.java: Extrae la información del proceso a estudiar a partir de un archivo creado por HEAD! que es un módulo que obtiene información a partir del simulador de procesos HYSYS.
- Common.java: Contiene:

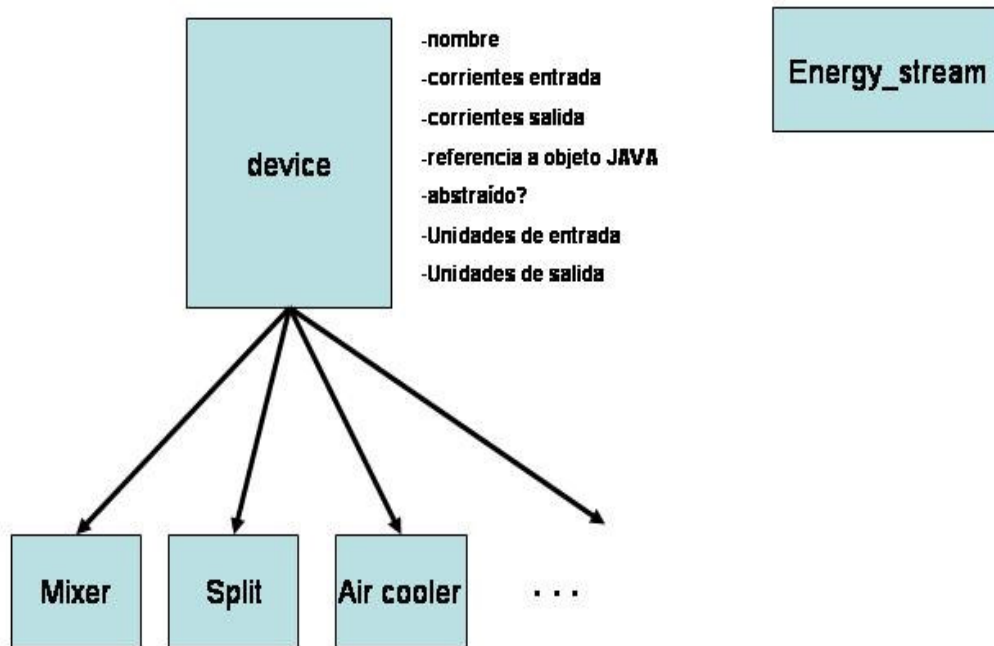
- la información actual del proceso: Los dispositivos del mismo tipo son almacenados en un mismo vector al igual que los energy streams y material streams.
- Información de los niveles abstraídos: guarda una copia de todos los vectores de cada nivel en un vector.
- GuiJess.java: Es una consola no interactiva que ha sido utilizada para desarrollar este proyecto. En la parte superior de la consola se muestran los Hechos que han sido insertados y los resultados de otro tipo de ordenes enviadas desde el módulo AHA. En la parte inferior se muestran las reglas que ya han sido disparadas.
Esta consola ha sido de gran utilidad para el desarrollo del sistema experto haciendo posible un seguimiento más cercano de las acciones emprendidas por éste.
- MyEventHandler.java: Implementa los eventos de la consola GuiJess.
- iniJess.java: Inicializa el entorno del sistema experto .
- templates.java: Clase que añade las plantillas de los equipos y las reglas a la base de conocimiento. Tanto las plantillas como las reglas se encuentran en distintos archivos.
- Instances.java y JessToJava.java: Clases empleadas en la comunicación de AHA con Jess.

4.4 Archivos con plantillas y reglas

Los siguientes archivos contienen las reglas y plantillas que implementará la base de conocimiento.

- Templates.clp: se encuentran las plantillas referentes a los equipos. La mayoría de los equipos tienen campos comunes por lo que existe una plantilla (device) de la cual heredan el resto de plantillas(equipos).

Templates.clp



-
- rulesFlow_change.clp: Regla para obtener las unidades que deben abstraerse en el nivel FLOW e iniciar la abstracción. Los equipos que se tratan son:
 - Mixer.
 - Split.
 - rulesPressure_change.clp: Reglas para obtener los equipos que se abstraen en el nivel pressure e iniciar la abstracción. Los equipos que se pueden encontrar en dicho nivel son:
 - Valve.
 - Expander.
 - Compressor.
 - Pump.
 - rulesTemperature_change.clp: En este archivo hay reglas para iniciar la abstracción de los equipos heater , coolers, air coolers y reglas para abstraer heat exchangers.

El Heat exchanger a parte de contener reglas para su abstracción, presenta otras destinadas a mantener toda su información actualizada cada vez que un equipo se abstrae con él .

Los equipos a tratar en este nivel son:

- Heater.
- Cooler.
- Heat exchanger.
- Air cooler.
- rulesSeparation.clp: Reglas para abstraer separadores. Los equipos son:
 - Liquid-liquid-extractor.
 - Flash.
 - 3 phase separator.
 - Trayed.
 - Packed.
 - Vapor absorption tower.
 - Liquid absorption tower.
 - Liquid adsorption tower.
- rulesReaction.clp: Reglas para abstraer los equipos reactores:
 - continuous stirred tank reactor.
 - Conversion reactor.
 - Equilibrium reactor.
 - Gibbs reactor.
 - Plug flow reactor.
 - tubular_reactor.
- generales.clp :

Incluye reglas que son aplicadas en varios o todos los niveles:

- Reglas destinadas a eliminar los ciclos del proceso y actualizar los equipos Heat exchanger afectados por la eliminación de estos.
- Obtener las conexiones entre equipos .Es decir saber que un equipo se conecta con otro a través de una corriente determinada .
- Obtener las unidades de entrada y de salida de cada equipo.
- Función de incrementar el contador de metaunidades.Cada vez que un equipo se abstrae con otro, se crea una metaunidad y se incrementa el contador.
- Función de construir el nombre de una metaunidad. Construye el nombre a partir de: El número de metaunidades actual, el nombre y la función del equipo que ha abstraído a otro .
- Función para saber, si un equipo es una null unit.
- Función para saber a través del nombre si es una salida de proceso
- Función para saber a través del nombre si es un energy stream.
- Regla para eliminar bucles. Corrientes que salen y entran en un mismo dispositivo.
- Función que devuelve un valor numérico según la función del equipo. El valor devuelto depende de la situación que ocupe dentro de la jerarquía de funciones.
- Función que devuelve un valor booleano y que indica si un dispositivo se abstrae a la salida por defecto.
- Reglas de abstracción empleadas en equipos de los niveles FLOW, PRESSURE y en el TEMPERATURE en los heaters , coolers y air coolers. Las reglas para abstraer a estos equipos son comunes. Las

reglas que se encuentran en los otros archivos referentes a éstos solo sirven para iniciar la abstracción, es decir hacen una copia temporal de los equipos a abstraer para que las reglas situadas en este archivo realicen la abstracción.

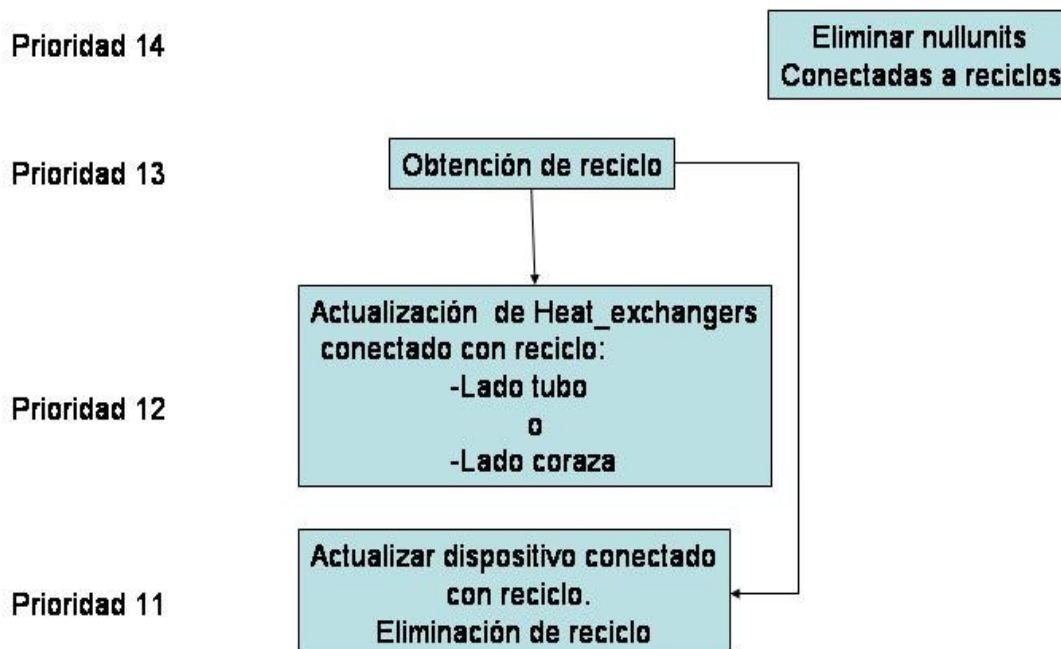
4.5 Abstracción

4.5.1 Reglas para la eliminación de reciclos

En primer lugar el sistema experto busca null units que a su entrada o salida tengan una corriente que a su vez también tengan un reciclo y otro dispositivo. Una vez encontradas las elimina . Esta situación tiene que ver con la forma de obtener la información.

El siguiente paso es obtener los reciclos que hay en el proceso y buscar si hay Heat exchangers conectados con ellos ya que en caso afirmativo es necesario actualizar la información de los campos referentes a las entradas y salidas del lado tubo y lado coraza .El último paso es buscar todos los dispositivos conectados con los reciclos y actualizar su información sobre las entradas y salidas de corrientes y por finalizar eliminar el reciclo.

REGLAS: Eliminación de reciclos

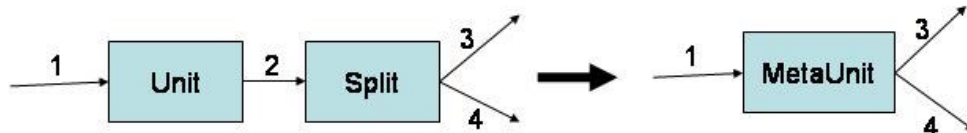
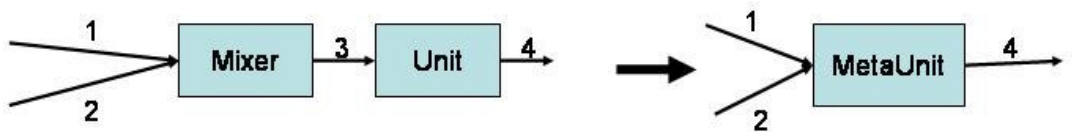


4.5.2 Casos de abstracción en nivel FLOW

Los equipos a abstraer en este nivel son los mixer y los split. Descripción del proceso de abstracción de cada dispositivo:

- Mixer: Presenta varias entradas y una salida. Será agrupado al equipo con que se encuentre conectado a la corriente de salida.
- Split: Presenta una entrada y varias salidas. Será agrupado al equipo con que se encuentre conectado a la corriente de entrada.

Casos nivel FLOW



4.5.3 Casos de abstracción en nivel PRESSURE

Los equipos que deben abstraerse son: Valves, expander, compressor, Pump.

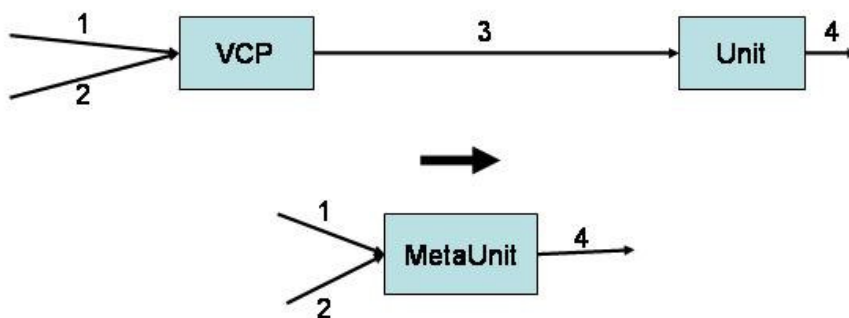
- Valve: Sin haber absorbido ningún equipo del nivel anterior dispone de una entrada y una salida. Por lo general este dispositivo será absorbido por el dispositivo que haya a su salida a no ser que en la salida se encuentre un output process (salida de proceso), en ese caso la valve se abstrae a la entrada.
- Expander: Éste dispositivo tiene una entrada sin haber sufrido absorciones. Es absorbido por el equipo que conecta con la corriente de entrada.
- Compressor: Presenta una corriente de salida sin haber absorbido equipos. Éste dispositivo es absorbido por el equipo que se encuentre en su salida. En el caso de que a la salida haya un output process, el equipo se abstrae a la entrada.

- Pump: Presenta una corriente de salida sin haber absorbido equipos. Éste dispositivo es absorbido por el equipo que se encuentre en su salida. En el caso de que a la salida haya un output process, el equipo se abstraee a la entrada.

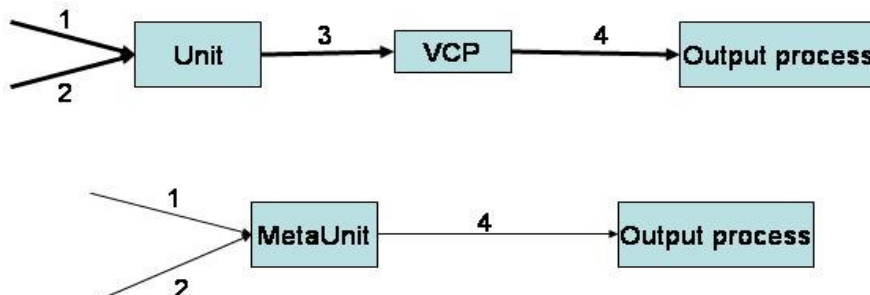
Casos nivel Pressure

VCP => equipos valve , compressor o pump.

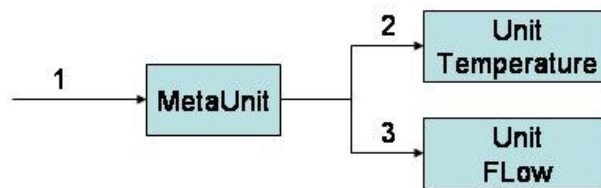
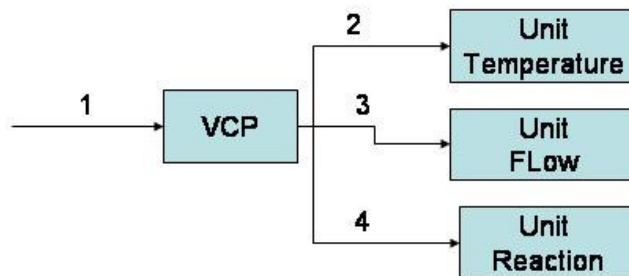
Caso: El equipo tiene una única corriente de salida y no conecta con una output process.



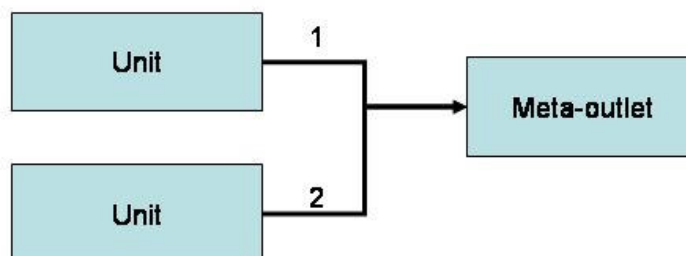
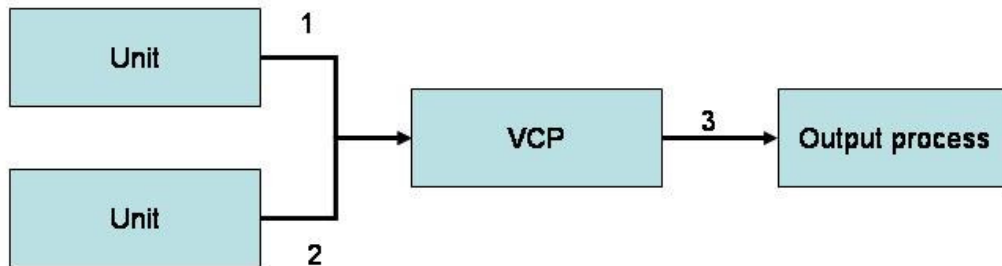
Caso: El dispositivo tiene a la salida un output process y tiene una sola entrada



Caso: El dispositivo tiene varias salidas (ha absorbido a otros dispositivos).
En ese caso se abstrae con el de mayor jerarquía

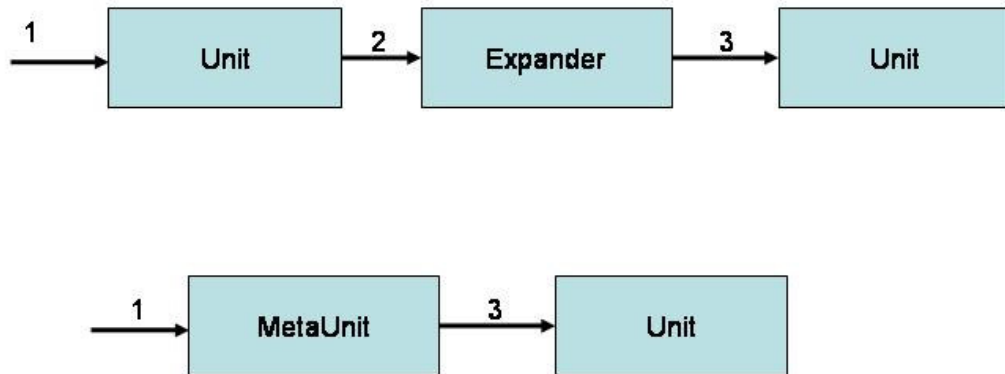


Caso: Hay un out process a la salida del dispositivo y hay varias entradas.



Expander

Caso: Una corriente de entrada en el dispositivo expander



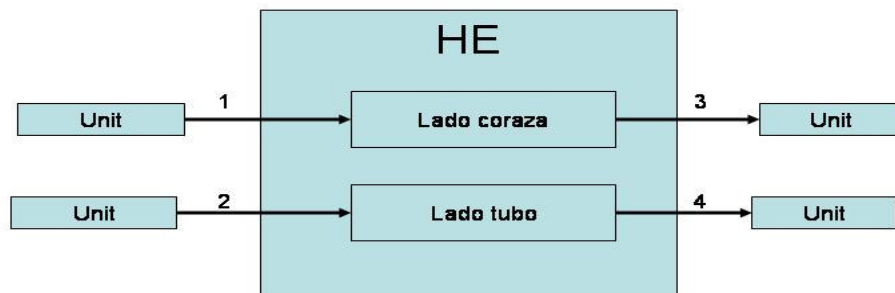
4.5.4 Casos de abstracción en nivel TEMPERATURE (coolers, heaters, air_coolers)

La forma en la que se abstraen los coolers, heaters y air coolers del nivel temperature es idéntica a los valves, compressors y pumps. Por lo que utilizan las mismas reglas para abstraerse con otros dispositivos.

4.5.5 Casos de abstracción en nivel TEMPERATURE (Heat exchanger)

Los HE (heat exchanger) constan de dos partes el lado tubo y el lado coraza. Cada vez que se realiza una abstracción estas partes se tratan como si fueran dos unidades en lugar de una .

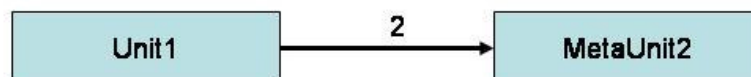
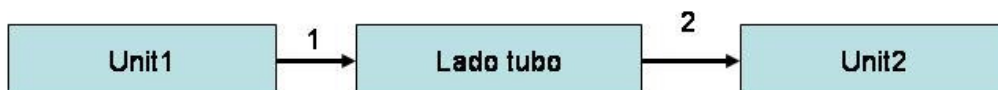
HEAT EXCHANGER



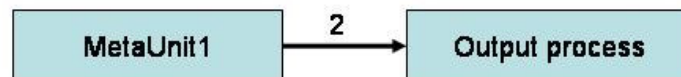
En las siguientes figuras se muestra como se comportan a la hora de abstraerse. Tanto el lado tubo como coraza se comportan de igual forma, por lo que sólo se representa uno de los dos en las siguientes imágenes.

CASOS DE ABSTRACCIÓN DEL HEAT EXCHANGER

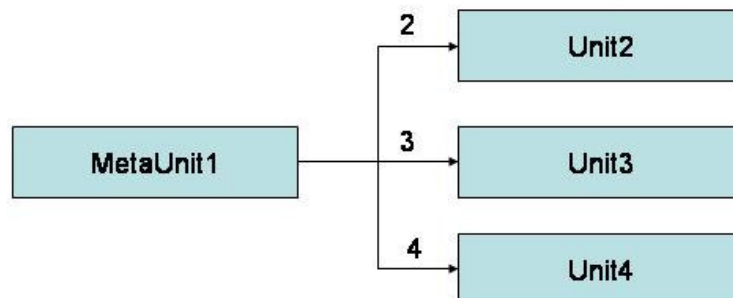
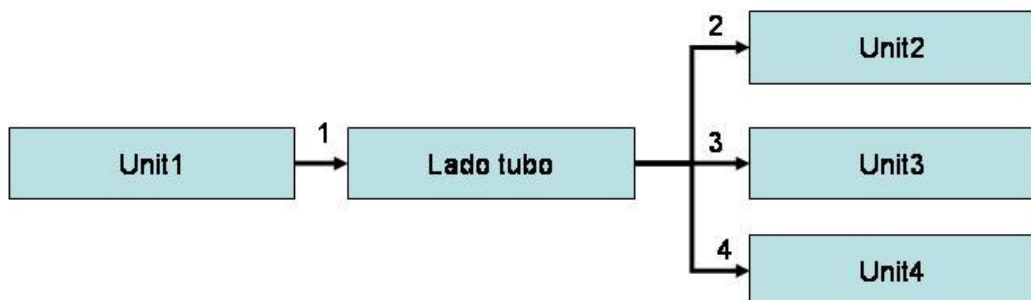
Caso: una entrada y una salida en el lado tubo



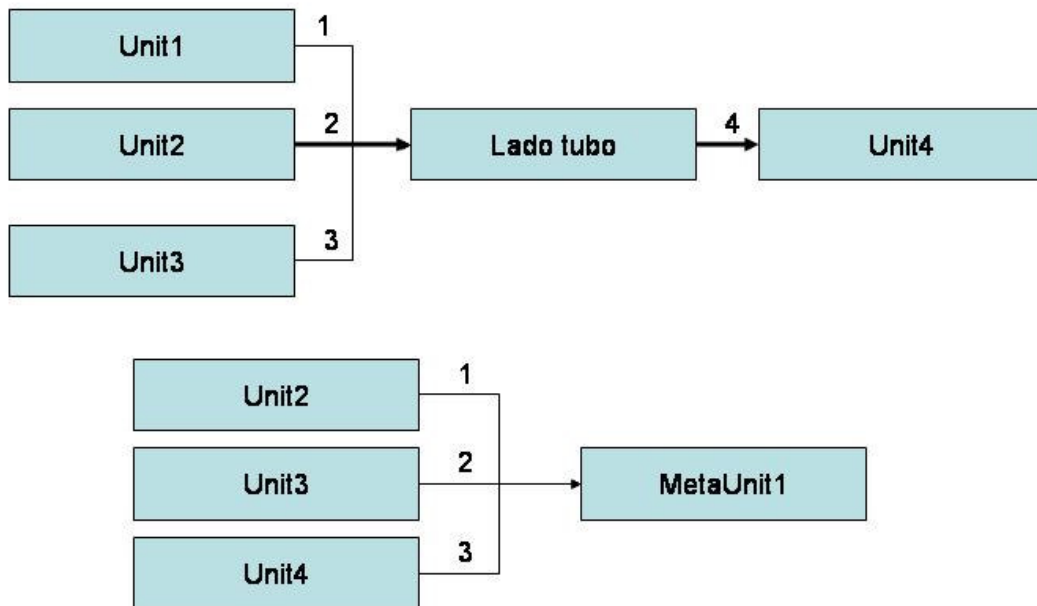
Caso : Una entrada y una salida a un output process



Caso: 1 entrada y varias salidas



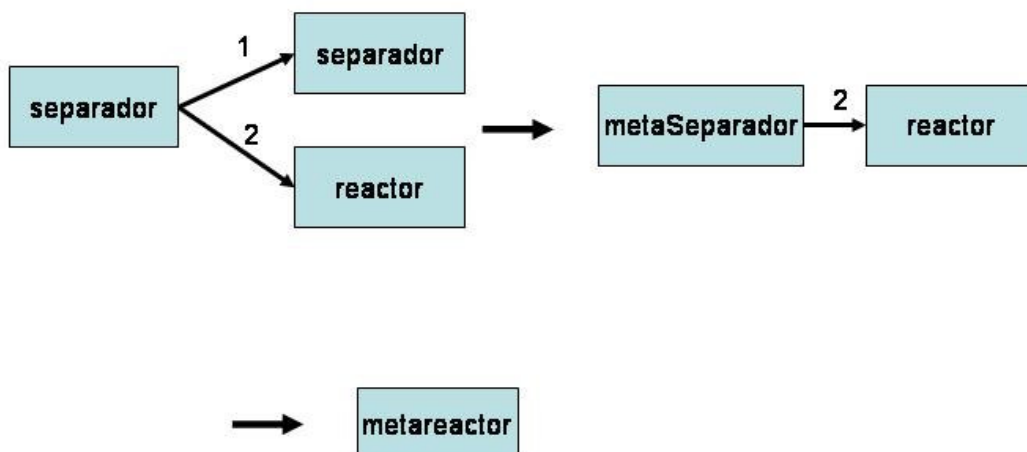
Caso: varias salidas y una entrada



4.5.6 Casos de abstracción en nivel SEPARATION

A estas alturas ya sólo deben quedar separadores y reactores. En este nivel primero se absorben los separadores entre ellos y después absorben a los reactores.

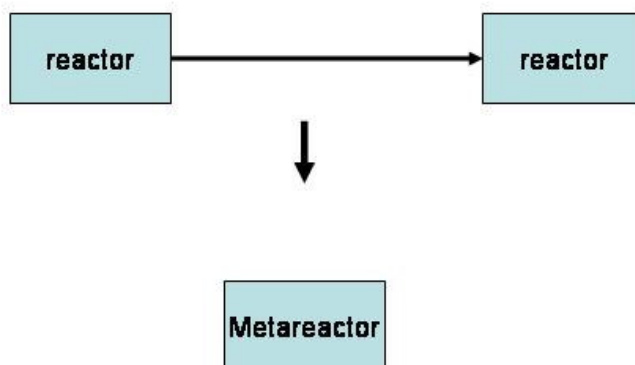
Casos nivel Separation



4.5.7 Casos de abstracción en nivel REACTION

Último nivel de abstracción. En esta fase de la abstracción todos los equipos excepto los reactores ya han sido absorbidos en niveles anteriores por lo que solamente es posible que los reactores se agrupen con otros reactores.

Casos nivel Reaction

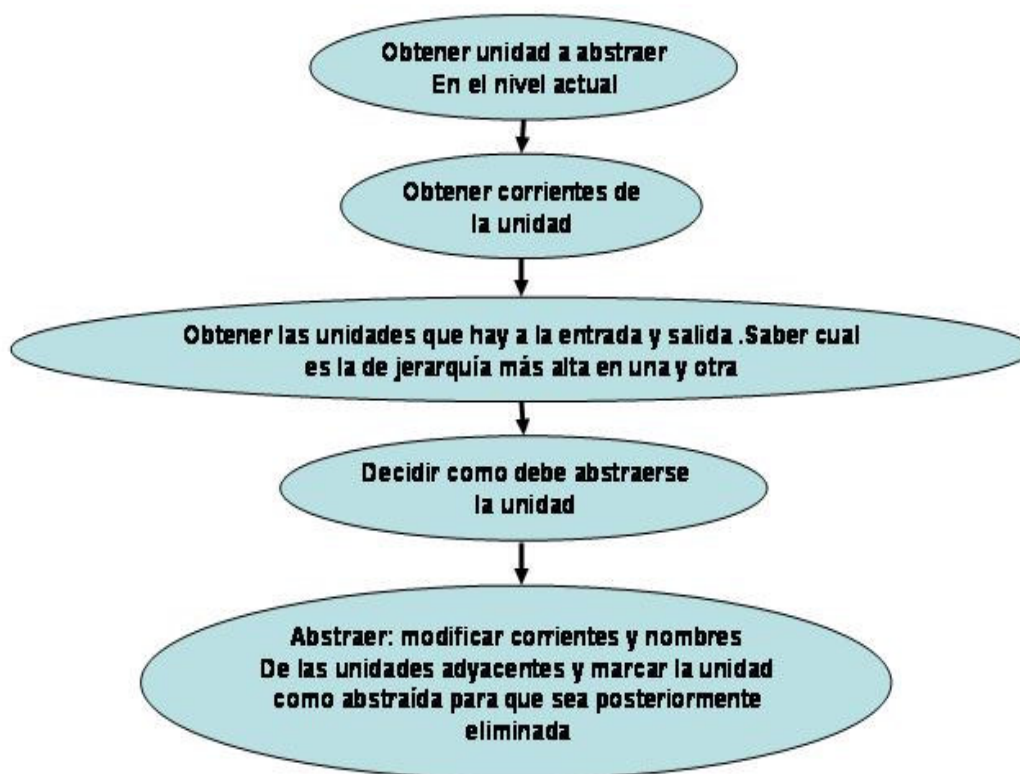


4.5.8 Reglas para la abstracción de los niveles FLOW, PRESSURE y TEMPERATURE (heaters, coolers y air coolers).

En primer lugar se obtienen las unidades que van a abstraerse en el nivel que toca. A partir de ese momento se obtienen las corrientes de la unidad tanto de la entrada como de la salida que servirán para saber cuales son las unidades adyacentes y que jerarquía tienen. Conociendo todas las unidades adyacentes y su jerarquía y teniendo en cuenta el tipo de dispositivo a abstraer, el sistema experto decide como abstraer la unidad.

Por último en el caso de que haya algún heat exchanger que haya sido afectado por la abstracción, debe ser actualizado para que la información que posee sea coherente.

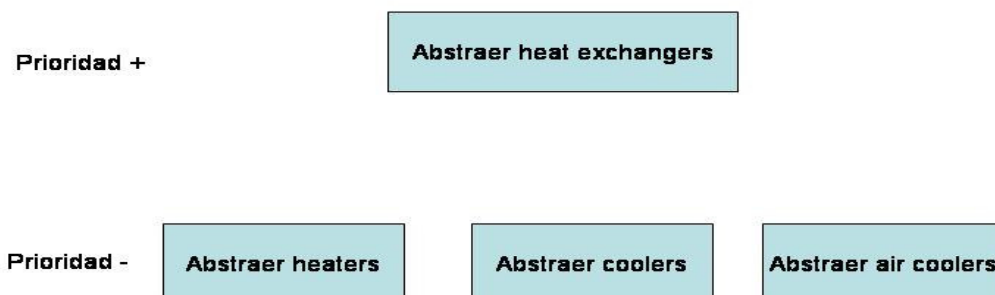
Esquema operativo del sistema experto en los niveles flow y pressure



4.5.9 Reglas para la abstracción del nivel TEMPERATURE (heat exchanger).

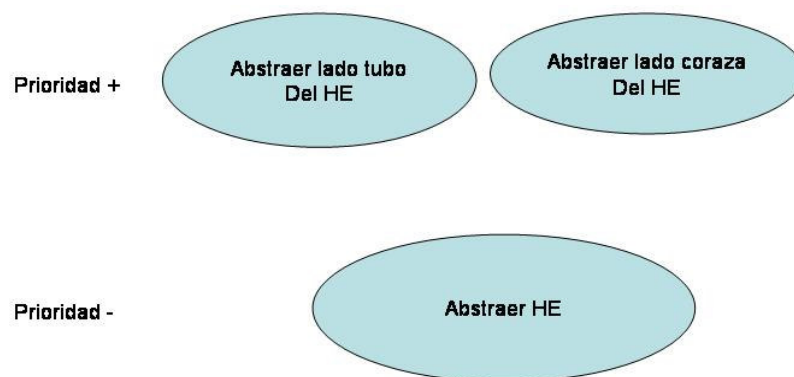
En este nivel, primero se abstraen los dispositivos heat exchangers y después el resto, los heaters, los coolers y los air_coolers. El esquema operativo de estos últimos ya ha sido comentado en el apartado anterior.

Esquema operativo del sistema experto en el nivel temperature



En la siguiente figura se muestra como se abstrae un HE.

Abstracción del HE por las reglas



Cuando los dos lados del HE quedan abstraídos, se procede a abstraer al HE.

Para abstraer a los heat exchangers las reglas tienen en cuenta el número de entradas y salidas tanto del lado coraza como del lado tubo. El tratamiento de las reglas de las diferentes formas de abstracción es la siguiente:

- Una entrada y una salida (en lado tubo o coraza): Tienen prioridad los equipos que a su entrada no tienen un HE respecto a los que si lo tienen. Se cambia del dispositivo de entrada la corriente de salida que conecta con el HE, por la corriente de salida del HE (del mismo lado ya sea tubo o coraza). Finalmente en función de si a la salida hay o no hay una output process la metaunidad se formará en el dispositivo de salida o de entrada.

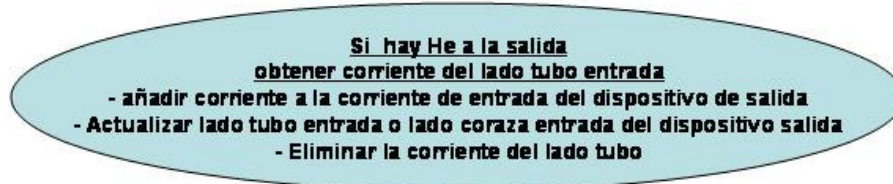
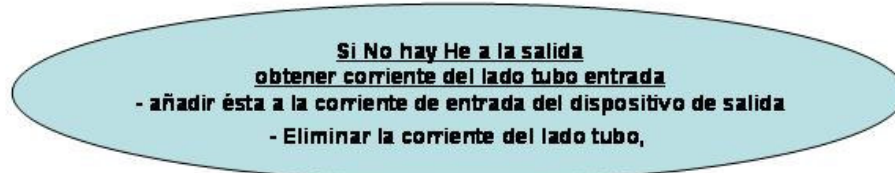
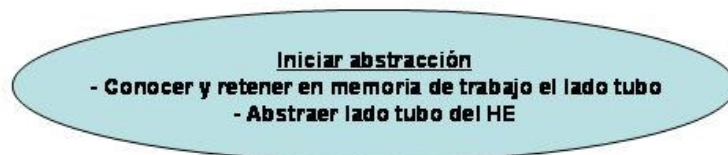
Abstracción del lado tubo. Caso 1 entrada 1 salida => 2 reglas



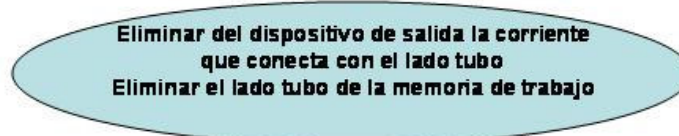
- Más de una entrada y una salida (en lado tubo o coraza): Se obtiene el lado tubo o coraza del He y se marca como abstraído. El sistema experto retiene la información de éste antes de ser abstraído. Después conecta cada entrada del HE (lado tubo o coraza) con el dispositivo de salida. Cuando se han conectado todas, eliminar del dispositivo de salida la corriente que le llegaba desde el He (lado tubo o coraza).

Abstracción del lado tubo. Caso varias entradas y 1 salida => 4 reglas

Prioridad +



Prioridad -



- Una entrada y más de una salida.(lado tubo o coraza). Se obtiene el lado (tubo o coraza) y se marca en el HE como leído. Conectar cada salida con la entrada. Cuando todas las salidas hayan conectado con la entrada, al equipo de entrada se elimina la corriente de salida que conectaba con HE.

Abstracción del lado tubo. Caso 1 entrada y varias salidas => 4 reglas

Prioridad +

Iniciar abstracción

- Conocer y retener en memoria de trabajo el lado tubo
- Abstraer lado tubo del HE

Si No hay He a la entrada obtener corriente del lado tubo salida

- añadir ésta a la corriente de salida del dispositivo de entrada
- Eliminar la corriente del lado tubo,

Si hay He a la entrada obtener corriente del lado tubo salida

- añadir ésta a la corriente de salida del dispositivo de entrada
- Actualizar lado tubo salida o lado coraza salida del dispositivo entrada
- Eliminar la corriente del lado tubo

Prioridad -

- Eliminar del dispositivo de entrada la corriente
que conecta con el lado tubo
- Eliminar el lado tubo de la memoria de trabajo

4.5.10 Reglas para la abstracción del nivel SEPARATION.

Esquema de la base de conocimiento para el tratamiento del nivel SEPARATION

abstracción nivel separation

Prioridad +

Abstraer separador con separador

- Crear meta-separador
- Marcar un separador como abstraído

Prioridad -

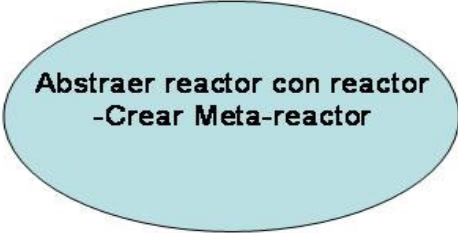
Abstraer separador con reactor

- Crear meta-reactor
- Marcar separador como abstraído

4.5.11 Reglas para la abstracción del nivel REACTION

Una única regla para abstraer reactores. Se crean metareactores hasta que

Abstracción nivel REACTION



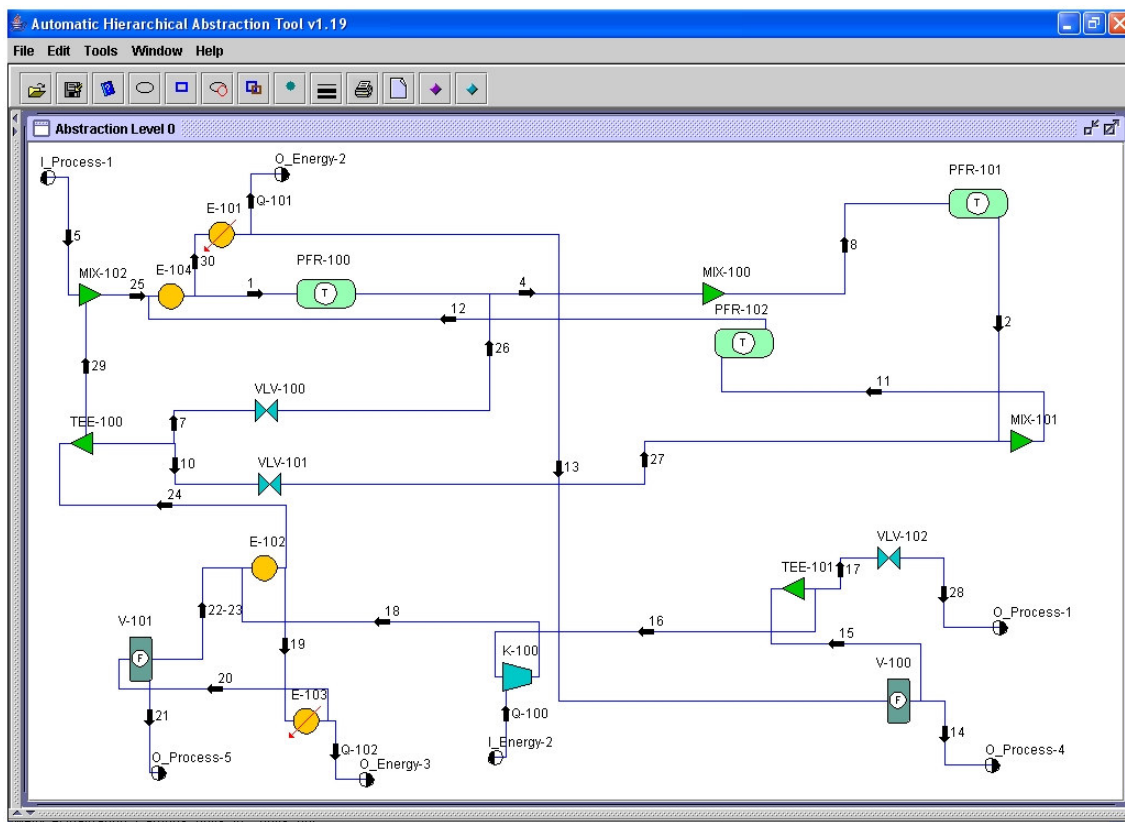
Abstraer reactor con reactor
-Crear Meta-reactor

5 EVALUACIÓN

El sistema experto ha sido probado con tres procesos diferentes que ya han sido validados.

5.1 Proceso amoníaco

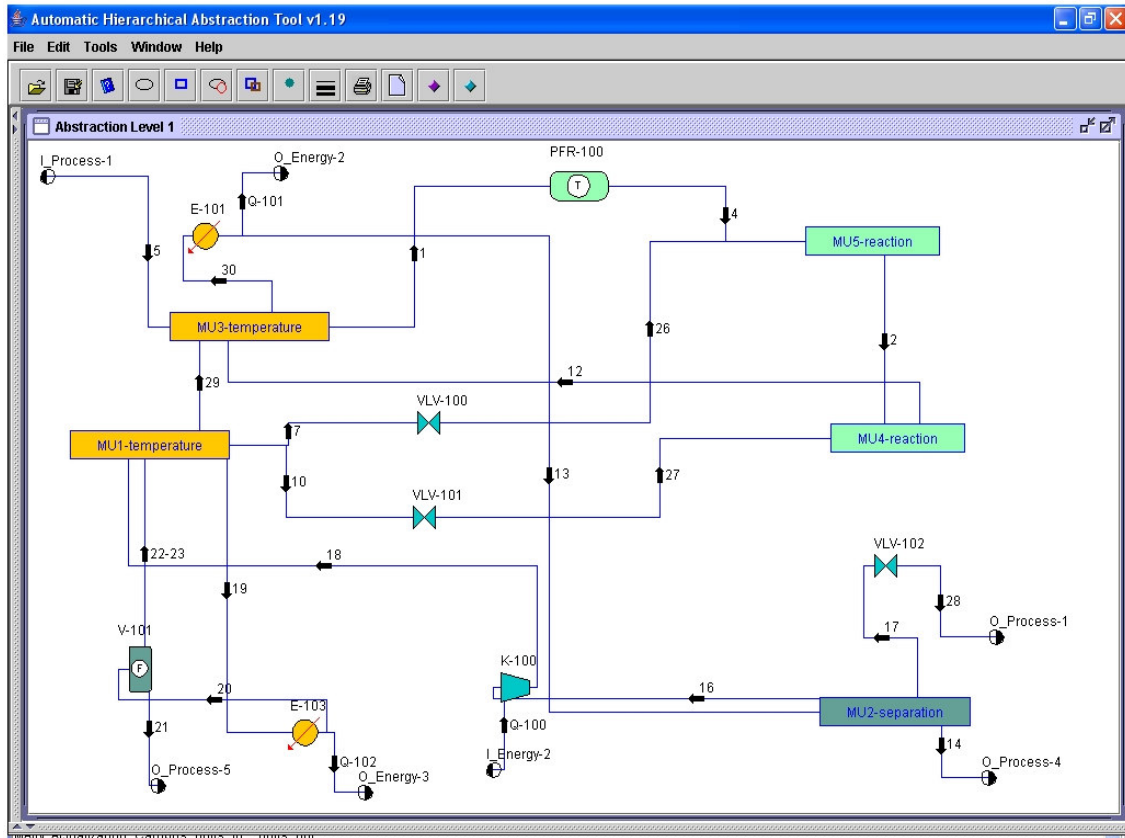
Proceso sin abstracción ni reciclos.



Abstracción nivel flow:

Abstracción de mixers y splits. Se eliminan tres mixers y dos splits, por lo que se forman tres Metaunidades.

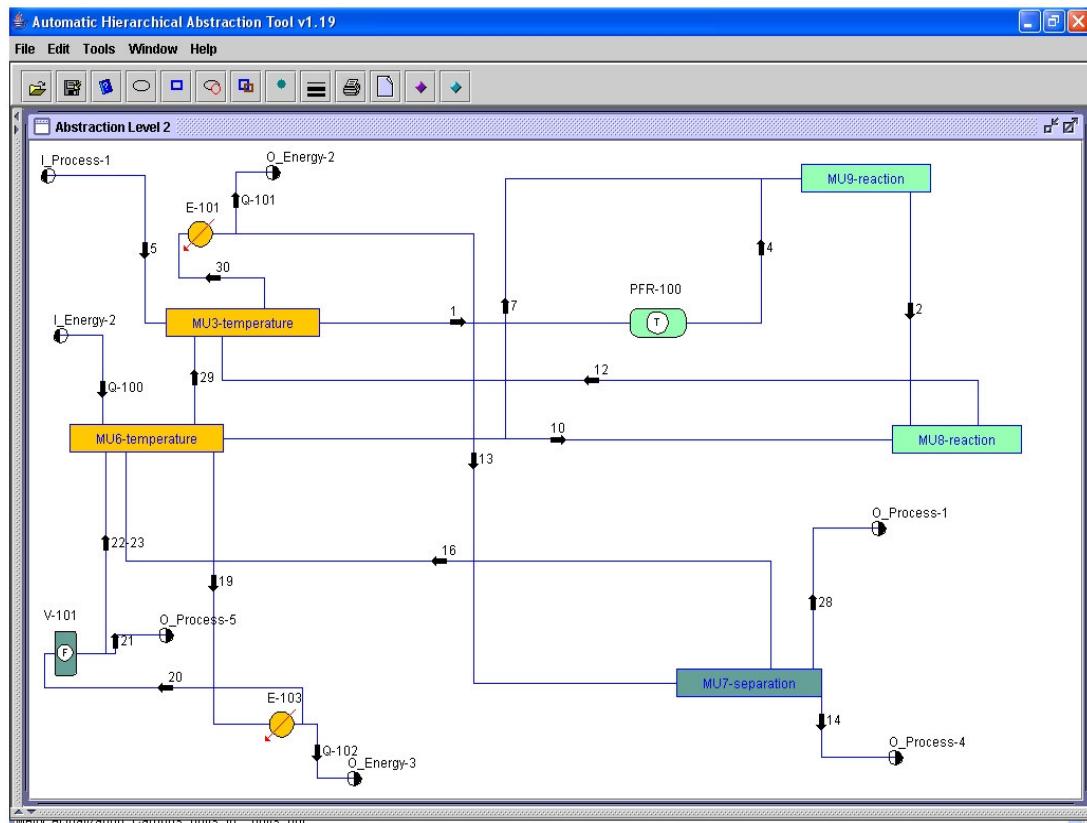
- El mix-100 se abstrae con el tubular reactor PFR-100.
- El split TEE-101 se abstrae con el flash V-100.
- EL mix-101 se abstrae con el PFR-102
- El mix-102 se abstrae con heat exchanger E-104.
- El split TEE-100 se abstrae con el heat exchanger E-102.



Abstracción nivel pressure

Quedan eliminadas tres valve y un compresor:

- La valve VLV-100 se abstrae con la Metaunidad MU5-reaction creando la nueva Metaunidad MU9-reaction.
- La valve VLV-101 se abstrae con la Metaunidad MU4-reaction creando la nueva Metaunidad MU8-reaction.
- La valve VLV-102 se abstrae con la metaunidad MU2-separation formando la nueva Metaunidad MU7-separation.
- El compresor K-100 se abstrae con MU1-temperature y foman la Metaunidad MU6-temperature.

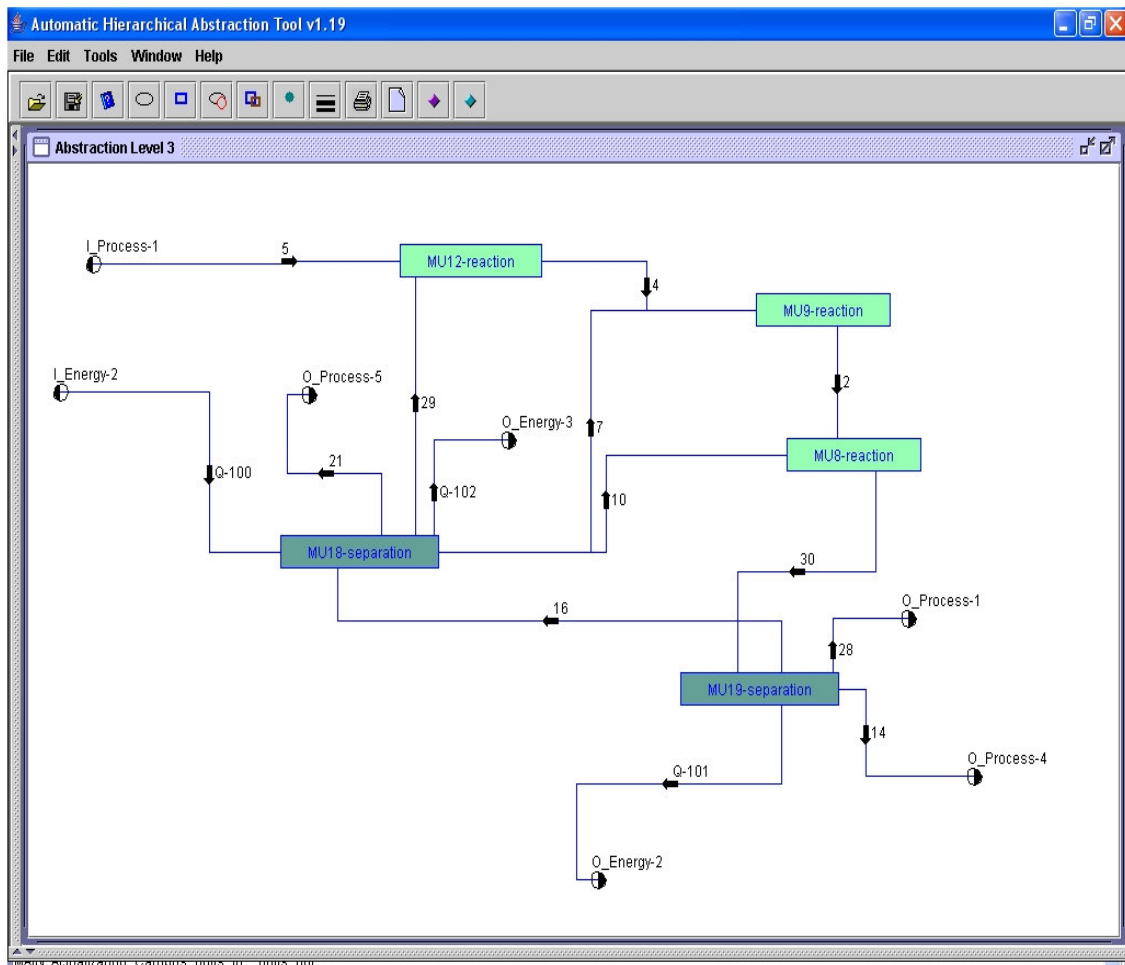


Abstracción nivel temperature

Se abstraen dos heat exchanger y dos coolers:

- El cooler E-101 se abstrae con la Metaunidad MU7-separation para formar la Metaunidad MU19-separation.
- El cooler E-103 se abstrae con la Metaunidad V-101 para formar la Metaunidad MU18-separation.
- Se abstrae el heat exchanger MU6-temperature :
 Corrientes de entrada lado tubo: 16, Q-100.
 Corrientes lado tubo salida: 19
 Corrientes de entrada lado coraza: 22-23
 Corrientes de salida lado coraza: 10, 29, 7
 El lado tubo conecta con el E-103 por lo que se formará una metaunit con el E-103 y ésta más tarde formará otra nueva metaunit que será la antes mencionada MU18-separation. El lado coraza se abstrae a la entrada formando metaunit a la entrada con el v-101.
- Se abstrae el heat exchanger MU3-temperature :
 Corrientes de entrada lado tubo: 12
 Corrientes lado tubo salida: 30
 Corrientes de entrada lado coraza: 29, 5
 Corrientes de salida lado coraza: 1

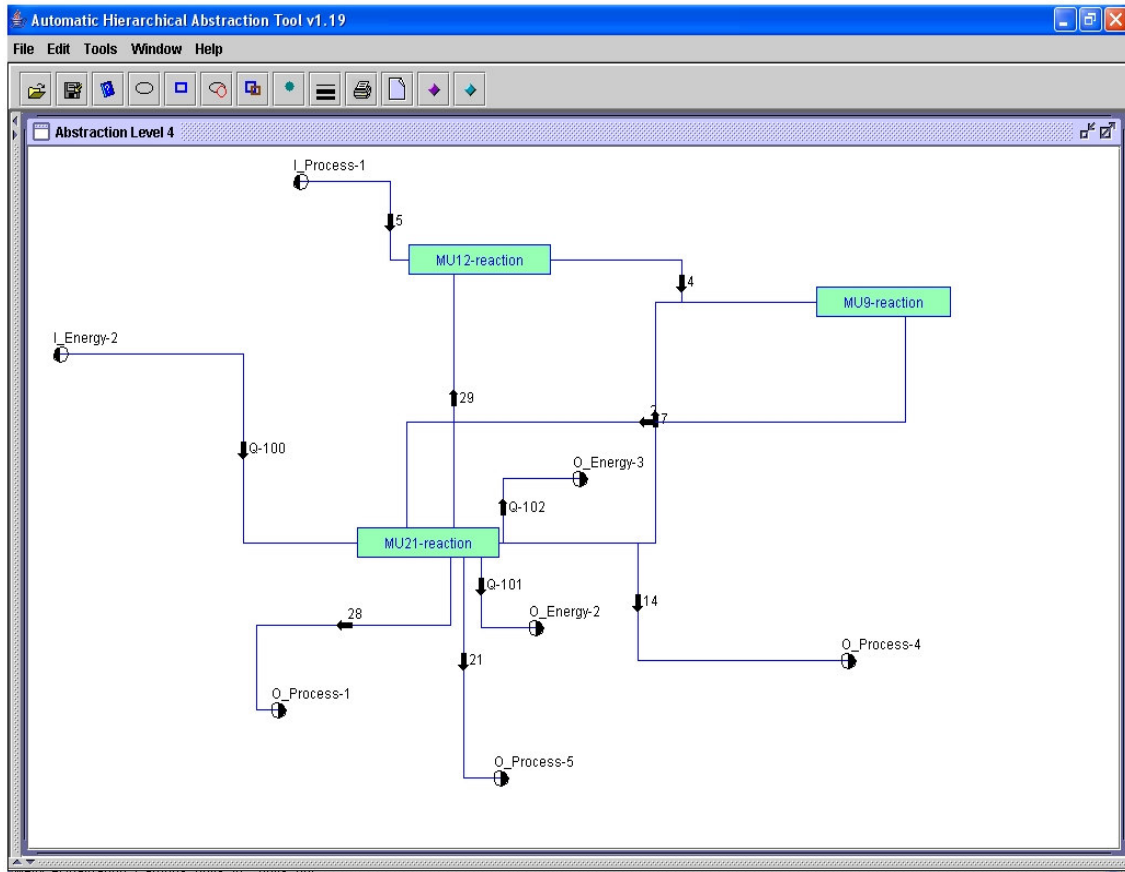
En el lado tubo conectará el MU8-reaction con el cooler E-101 formando metunidad. En el lado coraza conecta con el PFR-100 formando metaunidad.



Abstracción nivel separation.

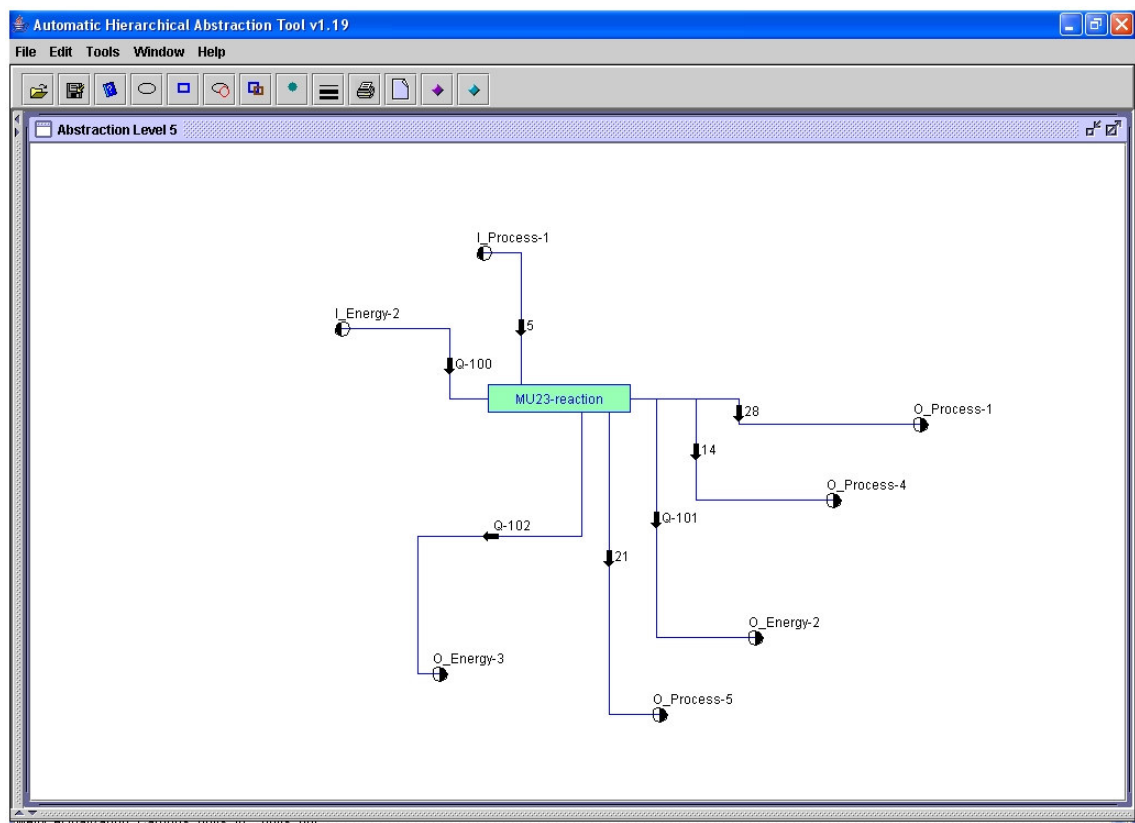
Quedan dos separadores y tres reactores.

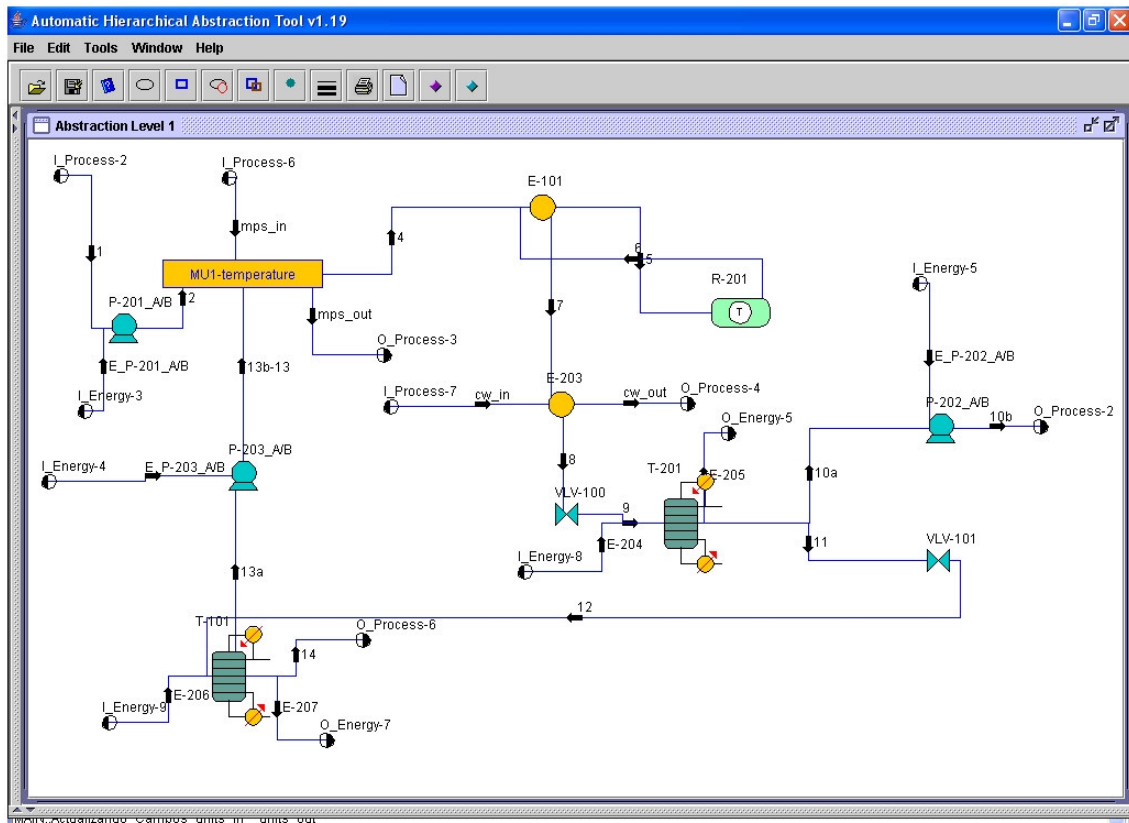
- La MU18-separation y la MU19-separation se abstraen y una vez agrupadas estas se agrupan al reactor MU8-reaction formando la Metaunidad MU21-reaction.



Abstracción nivel reaction:

Se abstraen los tres reactores quedando únicamente una Metaunidad, la MU23-reaction





Abstracción nivel pressure:

Se abstraen tres bombas y dos válvulas:

- La VLV-100 se abstrae con el trayed T-201 formando la metunidad MU6-separation.
- La VLV-101 se abstrae con el trayed T-101 formando la metunidad MU5-separation
- La pump P-202_A/B se abstrae a la salida con una salida de proceso, en concreto con la O-process-2.
- La pump P-201_A/B se abstrae a la salida con un heat exchanger el MU1-temperature para formar la metaunidad MU4-temperature.
- La pump P-203_A/B se abstrae a la salida con un heat exchanger el MU1-temperature para formar la metaunidad MU4-temperature

Hay tres heat exchanger que son abstraídos:

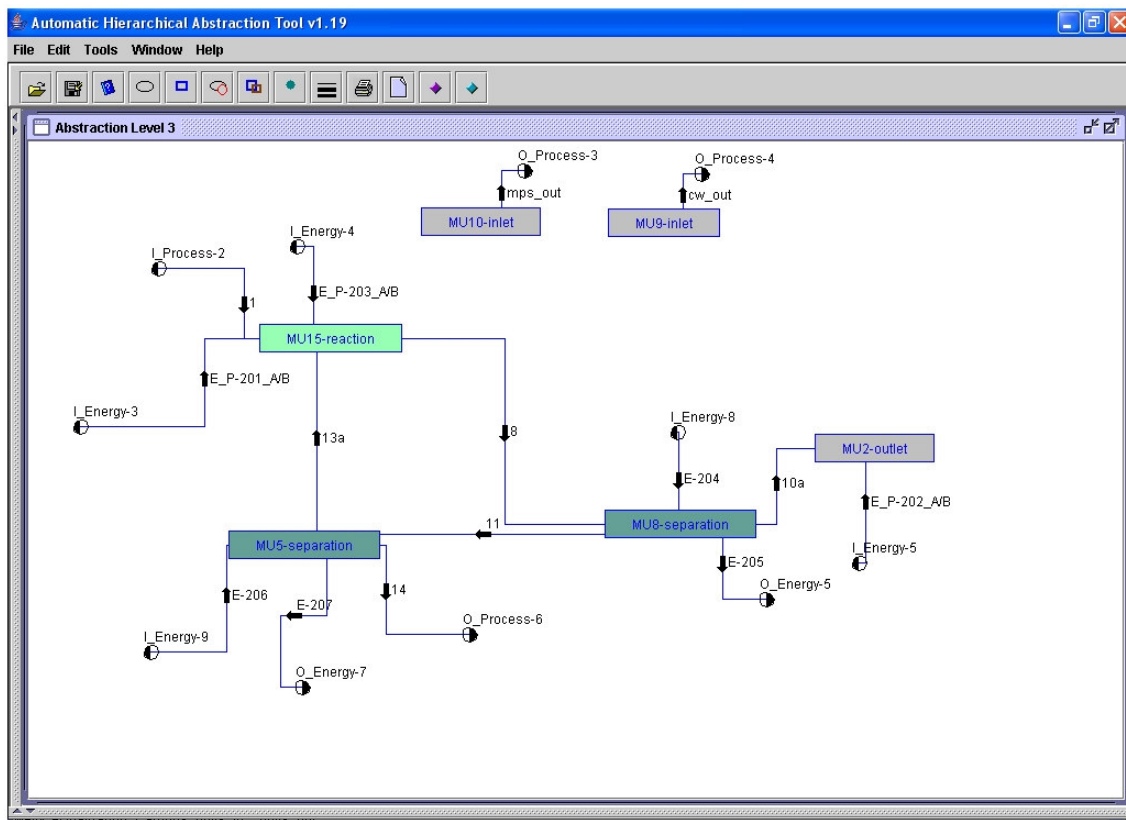
- El MU4-temperature se compone de :
 - Corrientes lado tubo entrada: E_P-203_A/B,13^a
E_P-201_A/B, 1
 - Corrientes lado tubo salida: 4
 - Corrientes lado coraza entrada: mps_in
 - Corrientes lado coraza salida: mps_out

El lado coraza se abstrae a la entrada porque tiene una única entrada y a la salida hay una salida de proceso. Se forma la Metaunidad MU10-inlet.

- El E-101 se compone de :
 - Corrientes lado tubo entrada: 4
 - Corrientes lado tubo salida: 5
 - Corrientes lado coraza entrada: 6
 - Corrientes lado coraza salida: 7
- El E-203 se compone de :
 - Corrientes lado tubo entrada: 7
 - Corrientes lado tubo salida: 8
 - Corrientes lado coraza entrada: cw_in
 - Corrientes lado coraza salida: cw_out

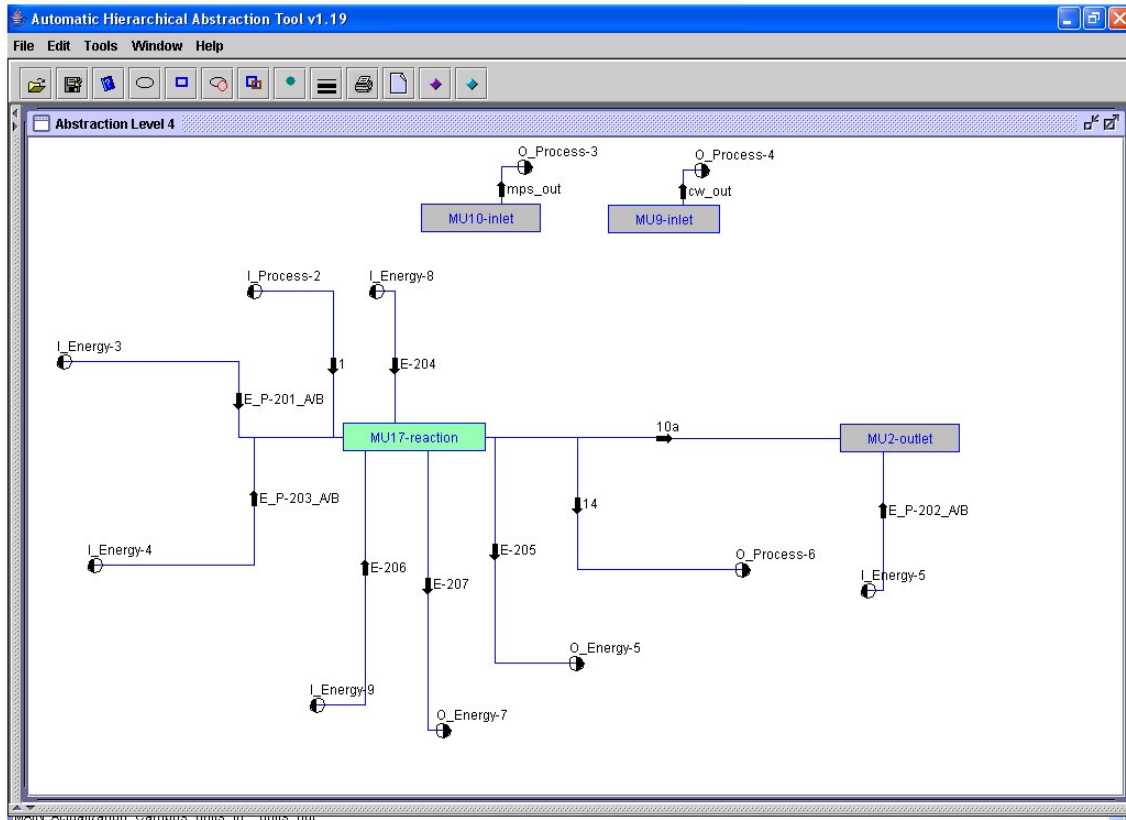
En este caso (E-203) el lado coraza consta de una entrada y una salida a un output process por lo que la abstracción se realiza a la entrada formando la metaunidad MU9_inlet.

El lado coraza del E-101 y el lado tubo del E-203 están conectados, por lo que quedan conectados según las reglas de abstracción de los HE, los equipos R-201 y MU6-separation, formando éste último, la Metaunidad MU8-separation. Por último queda por tratar el lado tubo de E-201 y el lado tubo del MU4-temperature que están conectados y que por lo tanto se forma una Metaunidad con la unidad R-201 y que pasará a ser la MU15-reaction.



Abstracción nivel separation

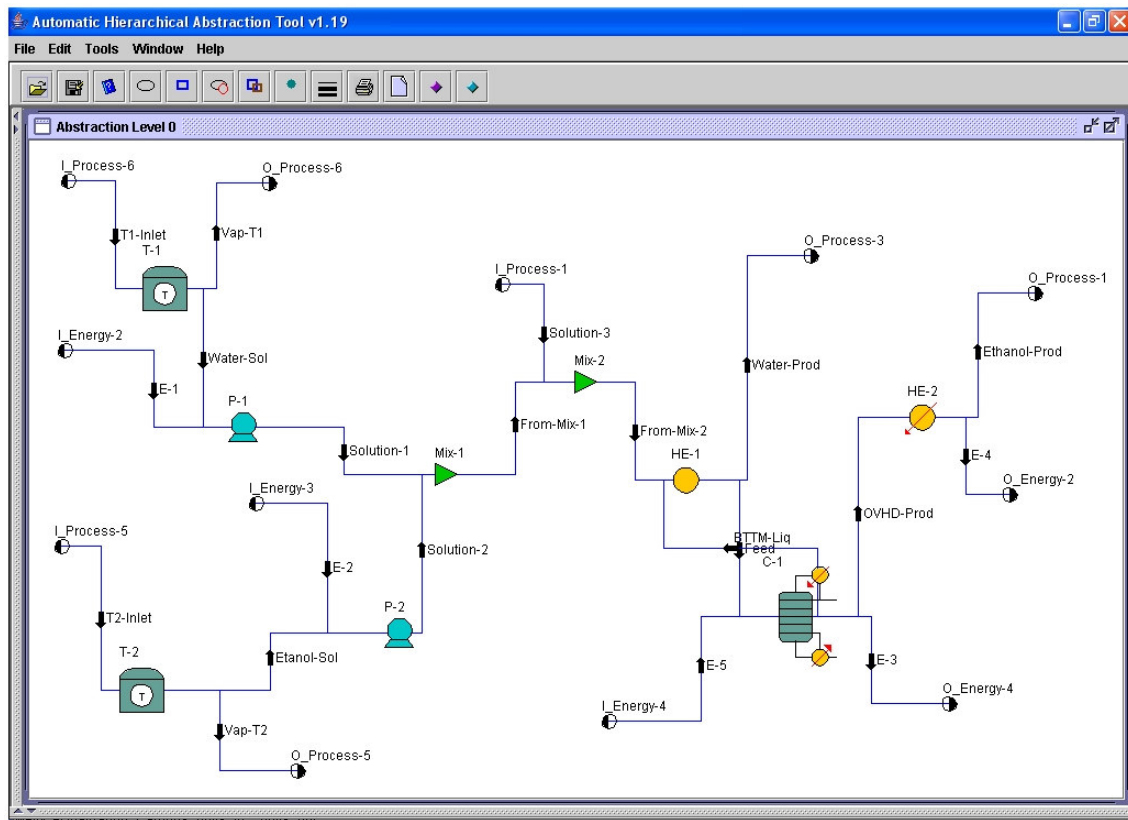
En esta fase un separador absorbe al otro y finalmente la metaunidad de separadores es absorbida por el reactor.



El proceso ha llegado al final de la abstracción Ya no se puede seguir abstrayendo ya que solo quedan entradas salidas de proceso y un reactor.

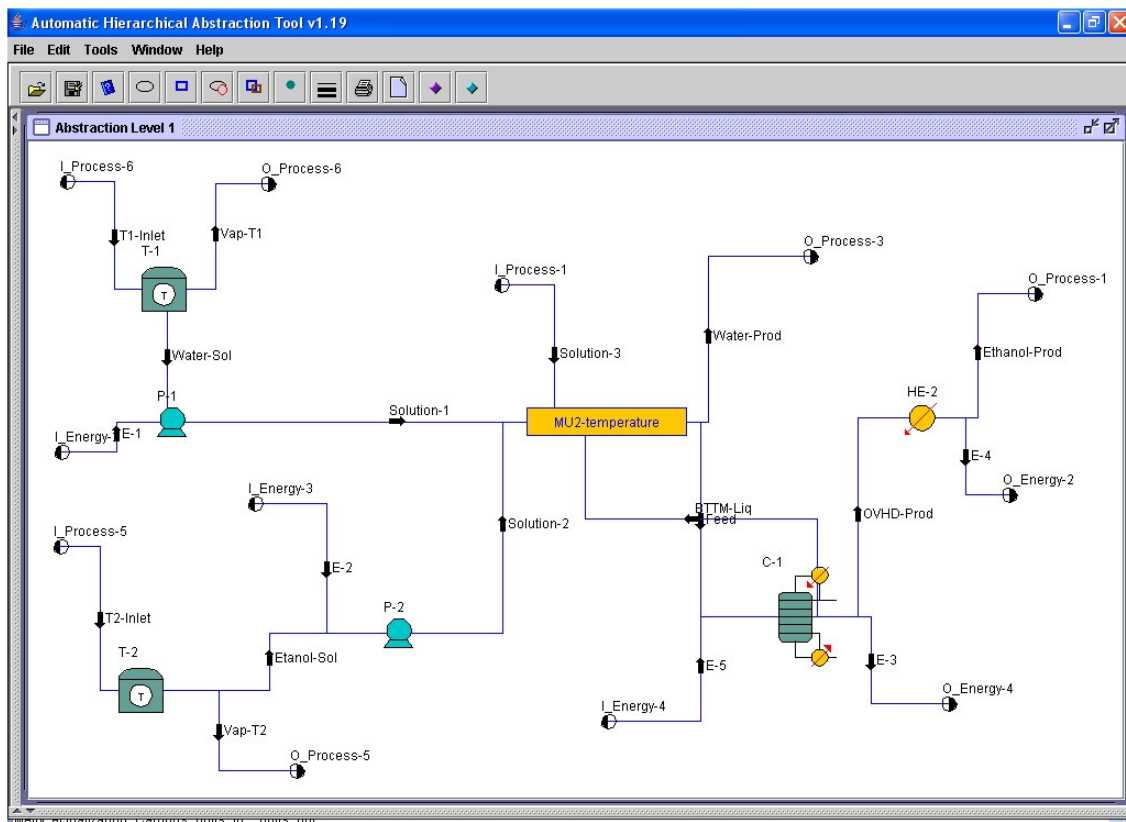
5.3 Proceso ethanol.

Sin abstracción ni reciclos



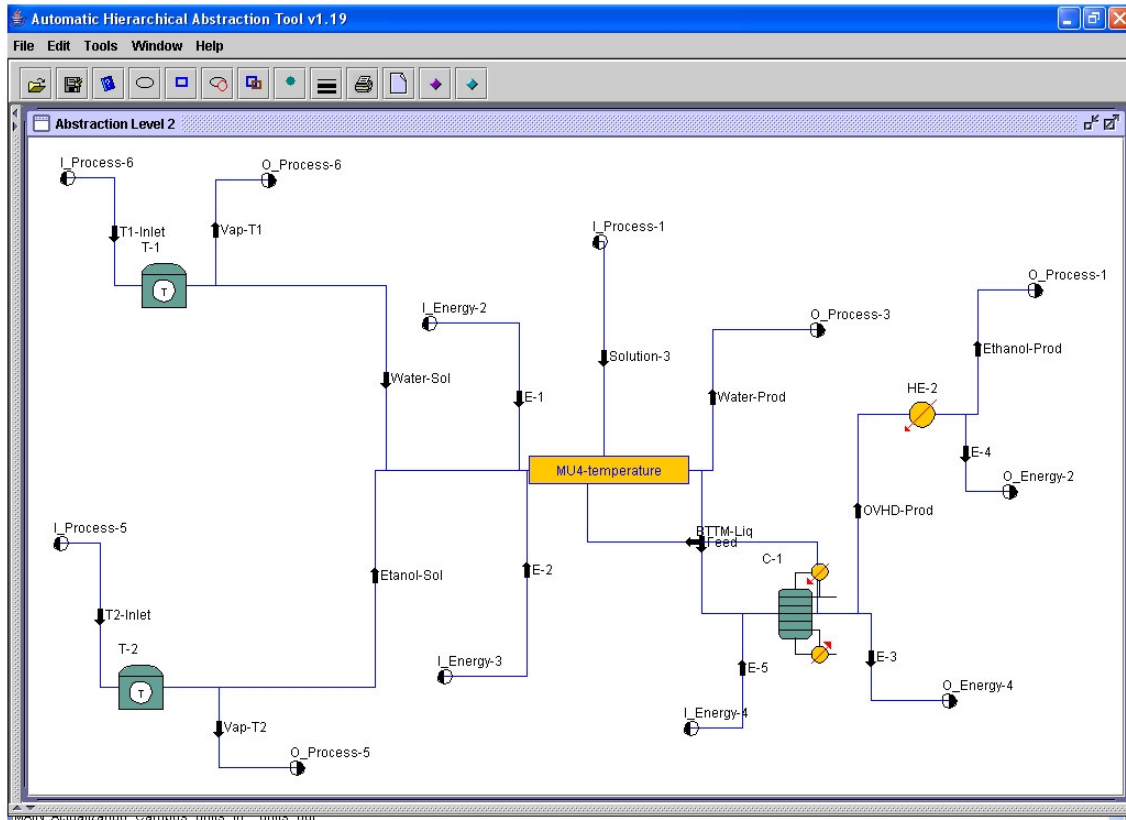
Abstracción nivel flow

EL mix-1 es absorbido por el mix-2 y finalmente la metaunidad de mixers es absorbida por heat exchanger HE-1 formando la Metaunidad MU2-t.emperature



Abstracción nivel pressure

El proceso que da igual que estaba al no presentar equipos que se abstraigan en este nivel.



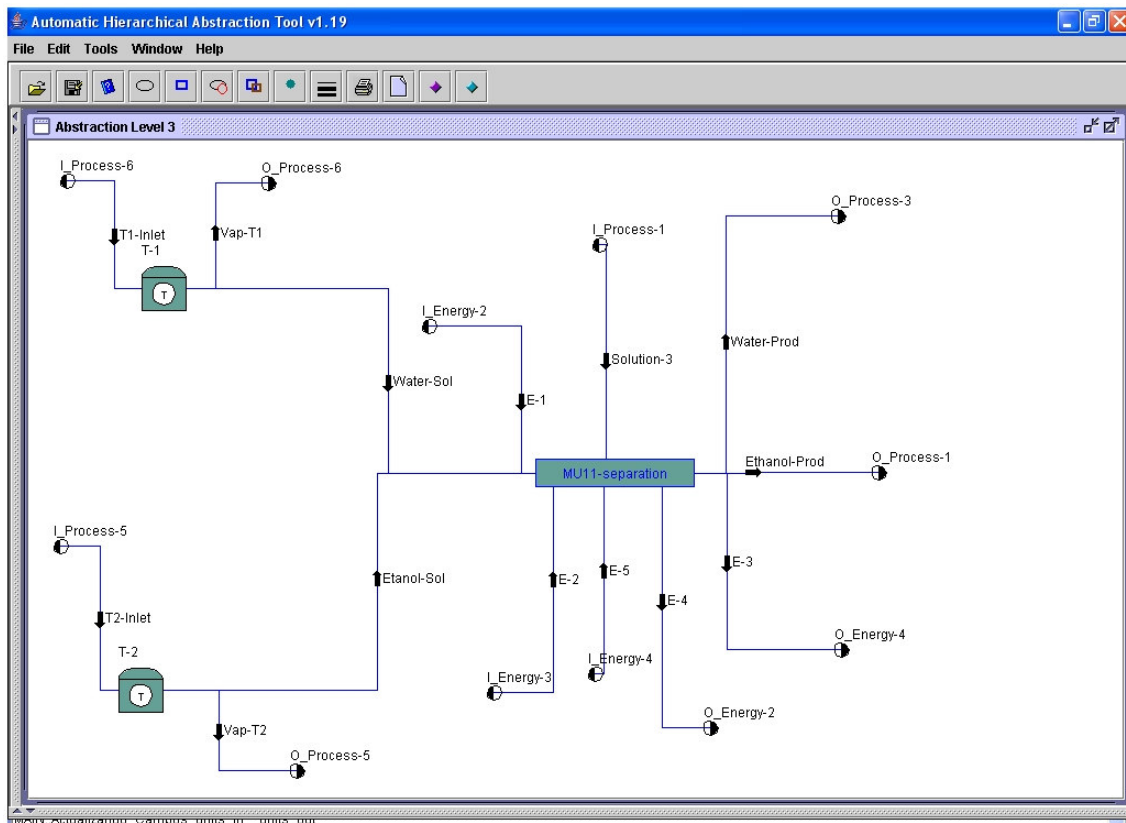
Abstracción nivel temperature

El cooler se abstrae a la entrada por tener a su salida una salida de proceso y un energy stream, por lo que forma una metaunidad con el trayed c-1.

El HE se compone de :

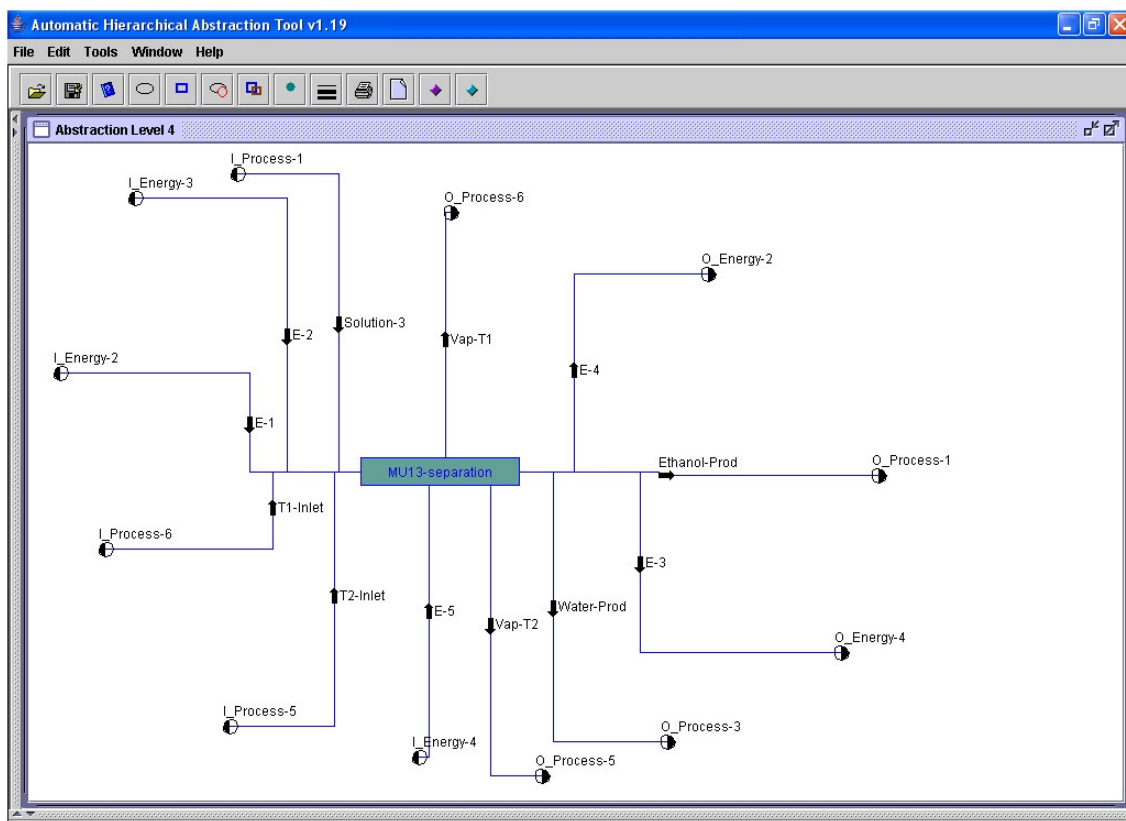
- Lado entrada tubo: E-2, Etanol-Sol, Solution-3, Water-Sol, E-1.
- Lado salida tubo: Feed.
- Lado entrada coraza: BTM-Liq.
- Lado salida coraza: Water_prod.

El lado coraza tiene una entrada y una salida que es salida de proceso por lo que la abstracción se realiza con la unidad de entrada c-1. El lado tubo conecta las entradas del HE con el equipo c-1 por lo que se forma una nueva Metunidad con el equipo c-1.



Abstracción nivel separation

Ya solo quedan equipos de separación por lo que se irán absorbiendo formando metaunidades hasta que solo quede una única metaunidad.



Sólo quedan entradas y salidas de proceso y un equipo separador por lo que se ha llegado al final de la abstracción.

6 CONCLUSIONES

La elección de implementar un sistema experto como encargado de realizar el proceso de abstracción, es acertada por su manera de representar el conocimiento y por la forma en que se separa dicho conocimiento del resto de la aplicación. Esta situación permitirá en el futuro implementar cambios en la metodología de abstracción causando un mínimo impacto en la aplicación.

De las ventajas de Jess más obvias y que han resultado de mayor utilidad para la realización de este proyecto ha sido su integración en java y el hecho que desde el módulo AHA, se pudiera acceder a la base de hechos simplemente importando una librería y utilizando unos métodos. Otra ventaja ha destacar y que también ha sido empleada en la elaboración del proyecto ha sido la utilización de las “defqueries”, que son mecanismos para hacer una búsqueda directa en la base de hechos. En concreto, cada vez que en Jess se finaliza una abstracción de proceso, la aplicación hace una búsqueda directa de todos los dispositivos que se encuentran en la base de hechos.

Jess como sistema experto en comparación con otros quizá resulte menos potente pero el hecho de poder acceder a objetos y métodos JAVA directamente hace que aumenten sus capacidades en gran medida. En contra y es algo que hay que destacar, es la poca documentación que hay respecto al tratamiento de objetos JAVA desde Jess. En su manual de referencia considero que es insuficiente la información sobre este aspecto. Y Realizando diferentes búsquedas por internet la información encontrada al respecto suele ser repetitiva que suele aportar poco o nada.

7 REFERENCIAS

- [1] López-Arévalo I. , Metodología de rediseño de procesos químicos basada en modelos múltiples. Proyecto de tesis, Universitat Politècnica de Catalunya, Departament de Llenguatge i Sistemes Informàtics. Barcelona 2003.
- [2] Rodríguez Martínez A., Metodología para el retrofit de procesos químicos basada en una representación jerárquica, Escola Tècnica d'Enginyeria química.2002 Tarragona.
- [3] <http://herzberg.ca.sandia.gov/jess>
- [4] http://ingenieroseninformatica.org/recursos/tutoriales/sist_exp

- [5] Alonso Jiménez J., Gutierrez Naranjo M. A. , Introducción a los sistemas basados en el conocimiento, Universidad de Sevilla
<http://www.cs.us.es/cursos/iic/temas/tema-02-iic04.pdf> .
- [6] Mamani Sánchez L. , Jess un Shell para Sistemas Expertos y el Algoritmo de Rete, artículo Número 2. Mayo-Junio de 2003 , Sociedad Peruana de computación.
<http://www.publicaciones.spc.org.pe/SPCMagazine/editions/II-2/html/articulo2.htm>
- [7] Ernest J. Friedman-Hill. Jess, The Expert System Shell for the Java Platform.
- [8] Universidad nacional de Colombia, Herramientas para el desarrollo de un SBC.
http://www.virtual.unal.edu.co/cursos/economicas/92211/lecciones/unidad2/ia/pagina6_5.htm.
- [9] Sistemas expertos, Universidad de Sevilla
<http://www.esi2.us.es/~dco/sistemas.htm#INDICE>
- [10] Manual online de Jess,
<http://www.cs.vu.nl/~ksprac/2002/doc/Jess60/index.html>