



Lámpsakos

E-ISSN: 2145-4086

[lampsakos@amigo.edu.co](mailto:lampsakos@amigo.edu.co)

Fundación Universitaria Luis Amigó

Colombia

Parra Ortega, Carlos

MODELADO CONCEPTUAL DE LA SUPERVISIÓN DE UN SISTEMA HOLÓNICO DE  
PRODUCCIÓN CONTINUA

Lámpsakos, núm. 7, enero-junio, 2012, pp. 19-30

Fundación Universitaria Luis Amigó

Medellín, Colombia

Disponible en: <https://www.redalyc.org/articulo.oa?id=613965334007>

- ▶ Cómo citar el artículo
- ▶ Número completo
- ▶ Más información del artículo
- ▶ Página de la revista en [redalyc.org](https://www.redalyc.org)

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal  
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

## MODELADO CONCEPTUAL DE LA SUPERVISIÓN DE UN SISTEMA HOLÓNICO DE PRODUCCIÓN CONTINUA

### CONCEPTUAL MODELING OF THE SUPERVISION OF CONTINUOUS PRODUCTION HOLONIC SYSTEMS

**Ph.D. Carlos Parra Ortega**

*Universidad de Pamplona, Colombia*

*carapa@unipamplona.edu.co*

(Artículo de **Reflexión**. Recibido el 29/10/2011. Aprobado el 19/12/2011)

**Resumen.** La arquitectura de automatización PROSA es una descripción orientada hacia objetos que implementan sistemas de producción. Un enfoque basado en sistemas para eventos discretos, detección de eventos y la teoría de control supervisorio, representa el comportamiento de una unidad de producción controlada. Para obtener la descripción orientada a objetos de esta unidad, se aplicó modelado conceptual y se utilizó el Lenguaje unificado de modelado.

**Palabras clave:** Arquitectura de automatización PROSA; Control supervisorio; Detección de eventos; Modelado conceptual; Sistemas de eventos discretos; Lenguaje unificado de modelado.

**Abstract.** The PROSA Automation Architecture is an object-oriented template that implements production units. An approach based on Discrete-Event Systems, Event-Detection and theory of Supervisory Control represents the dynamical behavior of a controlled production unit described under the PROSA architecture. In order to obtain the description of this unit, conceptual modeling using Unified Modeling Language is applied.

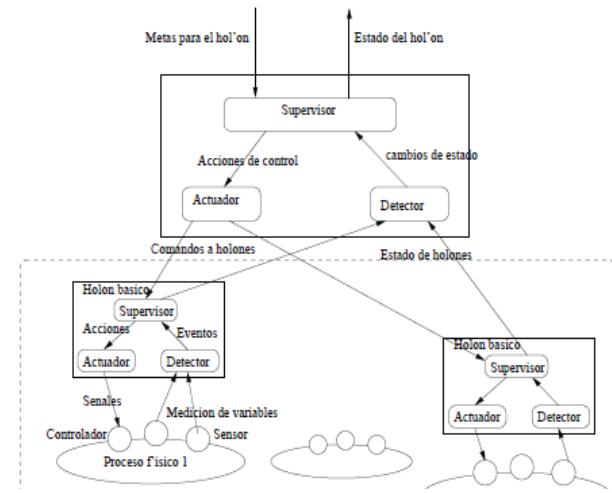
**Keywords:** PROSA Automation Architecture, Supervisory Control, Event Detection, Conceptual Modeling, Discrete-Event Systems, Unified Modeling Language.

## 1. INTRODUCCIÓN

El desarrollo de sistemas con capacidad de supervisar el comportamiento de unidades de producción sobre la base tecnológica disponible es un problema de suma actualidad. Diversos enfoques se han propuesto para aplicar los mecanismos de control, supervisión y gestión para un proceso productivo en el que estos mecanismos proyectan su funcionamiento hacia la arquitectura computacional y de comunicaciones de la empresa, cuya condición ideal de operación para estas organizaciones ocurre cuando existe la integración de estos elementos y, en su implementación, se usan las tecnologías de informática y comunicaciones de avanzada. Uno de los beneficios que resultan de integrar control e información es obtener una unidad de producción autoajustable. Una condición para ajustar un sistema de producción es la recolección oportuna de datos del piso de planta, de manera que permita determinar su estado y las decisiones para controlarlo. Además de que lo pueda comunicar a otras unidades de producción con el fin de coordinar sus actividades. En la Fig. 1 se observa un esquema general de una unidad de producción cuya función es controlar un proceso por etapas y coordina las actividades de otros componentes que, a su vez, son unidades de producción que controlan un proceso continuo atómico. Cada una de estas unidades detecta eventos cuya ocurrencia es difícil de predecir con exactitud porque implica analizar datos históricos del piso de planta. Esta detección de eventos es condición necesaria para derivar acciones que permitan mantener una trayectoria deseada en las variables continuas que se están controlando y comunicar su estado actual a la unidad de producción que la contiene. Esta unidad observa los eventos y actualiza la imagen del global y, eventualmente, decide la ejecución de algunas acciones que comunicará a sus unidades. Una unidad de producción compleja puede conformarse por medio de la composición de unidades de producción básicas y su evolución dinámica en el tiempo resulta del acoplamiento de las dinámicas de las unidades componentes.

Otro esquema alterno obvia la coordinación por parte de holones agregados, y en su lugar establece una relación de colaboración entre las unidades de producción que corresponden a etapas diferentes de producción. No existe una coordinación explícita sino un conjunto de diálogos entre holones básicos de producción de forma que cada uno solicita y ofrece recursos y productos terminados. Es necesario aclarar que un producto final de una unidad de produc-

ción se concibe como un recurso para una unidad de producción diferente, y así sucesivamente.



**Fig. 1. Esquema general de unidad de producción supervisada**

Es conveniente definir los conceptos de supervisión y de planta industrial. Por esta última se entiende que es el sistema que se ha de controlar, incluye la producción física, los insumos, el método de producción y los controles de equipamiento. El supervisor se define como el controlador de la planta a alto nivel, quien previene que la misma llegue a unas condiciones de operación no deseada. El proceso de la supervisión está asociado con la detección de eventos porque es menester analizar si ocurrió un evento que implique cambios en el estado en el que se encuentra la planta, las mediciones de sus variables o su estado actual. La dinámica del supervisor y la de la planta se describen por medio de un Sistema de Eventos Discretos (SED, de aquí en adelante) y las secuencias de eventos asociadas con estos SED conforman el lenguaje de la planta y del control supervisorio. En la Fig. 2 se aprecia la dinámica de un proceso supervisado.

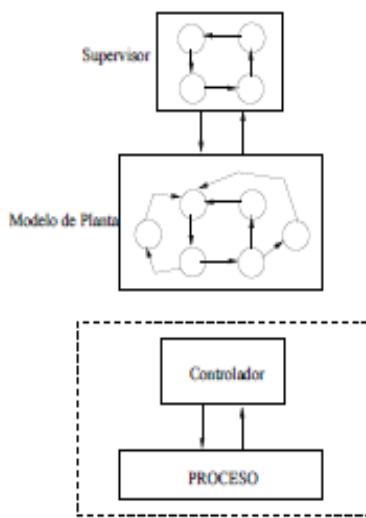


Fig. 2. Dinámica de un supervisor

El presente artículo es una propuesta para obtener el Modelo de estado de un sistema de manufactura holónico, con el fin de facilitar la implementación de supervisores para las unidades de producción en su forma más cercana al piso de planta (la más básica).

Antes de continuar, es necesario definir el concepto de Estado del proceso, que se asume como un valor de ciertas características en un tiempo determinado. Como varios actores observan y requieren información diferente, son necesarias varias perspectivas para modelar conceptualmente este estado. Las siguientes preguntas surgen como justificación para investigar sobre este tópico de modelado conceptual: ¿qué características deben asignarse a la descripción del estado del proceso? ¿Cómo debe ser el repositorio de datos? ¿Es factible una descripción general de este repositorio? ¿Cómo debe aplicarse un modelo general del repositorio en un caso práctico? Para responderlas, primero se modela el comportamiento del sistema supervisado, y luego se determina la estructura computacional estática de la unidad de producción que contiene al estado del proceso. Posteriormente, se detalla el metamodelo del producción.

## 2. ESTADO DEL ARTE

Dentro de los sistemas holónicos se han llevado a cabo proyectos de modelado conceptual. Algunos de ellos se han enfocado principalmente hacia la des-

cripción del estado del producto como en el proyecto HoMUCs [5], otros hacia la autosimilaridad de tres componentes fundamentales: Recurso, Producto y Orden como aparece planteado en PROSA, propuesto por Valckenaers, Van Brussel y otros investigadores [12], [10]. Chacón, Besembel y Hennet proponen la supervisión de unidades de producción con holones definidos con la arquitectura PROSA para gestionar un proceso industrial continuo [2]. El comportamiento dinámico de esta unidad de producción puede describirse por medio de un Sistema de Eventos Discretos (DES). La composición de varias unidades de producción implica un acoplamiento de DES, a los que se les sintetiza un supervisor que mantenga el comportamiento global dentro de unas metas deseadas. Blanc [1] aplica el paradigma PROSA en el monitoreo y planificación de un sistema de manufactura discreto, en el que el proceso productivo gira alrededor de la dinámica de las órdenes de producción. Aunque estos trabajos describen la supervisión, el control y la dinámica del estado del producto, no describen de la misma manera el estado de producción, el cual recibe especial énfasis en la propuesta descrita en este documento.

Para describir la evolución dinámica se tiene como fundamentación teórica el aporte de los sistemas de eventos discretos, bien por medio de autómatas [11] o bien por medio de redes de Petri [6] o, incluso, una combinación de ambos. El modelado conceptual de la unidad de producción se lleva a cabo con el uso de un lenguaje de modelado, como el UML propuesto por Booch, Rumbaugh y Jacobson [8]. Otros autores como Fishwick [4] establecen que el modelado conceptual orientado a objetos es un paso para hacer proyectos de simulación de la dinámica del sistema industrial, objeto de estudio. Otros van más lejos y proponen sistemas que controlan procesos productivos industriales.

Como base para obtener una descripción se utiliza el concepto de morfología, que es un método multidimensional de clasificación. Como ejemplo se tiene la tabla periódica en la que se describe cada elemento con sus propiedades, estructura y forma de relacionarse con su entorno. La idea es obtener una morfología que ayude a caracterizar el estado de un proceso industrial que tiene insumos, método de producción, producto final, equipamiento y servicios, entre otros.

### 3. MODELADO CONCEPTUAL DEL COMPORTAMIENTO DE UN SISTEMA SUPERVISADO

Por comportamiento de un sistema se concibe la manera o el modo en el que actúa y reacciona cualquier actividad que se registre y pruebe. Dentro del contexto de la automatización, se refiere a la descripción de lo que hace para llevar a cabo un proceso de producción y cómo cambian sus atributos de interés en el tiempo. Un modo para capturar el comportamiento del sistema es por medio de los casos de uso. Estos diagramas UML describen al sistema y sus relaciones con su entorno. En la Fig. 3 se muestra, para cada unidad de producción, un diagrama de caso de uso general, en el que se han identificado los actores y casos de uso que intervienen en la supervisión. Por actor se entiende que es todo lo que interactúa con el sistema, bien sea una máquina o una persona. Por caso de uso se establece que es una secuencia de actividades ejecutadas por el sistema y que produce valores medibles para un actor particular. El objetivo de este diagrama es presentar la funcionalidad del sistema supervisado y como base del modelado conceptual estructural y dinámico de la unidad de producción.



Fig. 3. Caso de uso general

Como se puede observar en la Fig. 3, se han identificado cuatro actores posibles dentro del holón de producción: sensor, detector de eventos, supervisor y actuador. Estos cuatro actores son roles que desempeña el sistema de automatización industrial en un momento determinado, y se consideran como sistemas autónomos, gestionados por unidades autónomas denominados agentes. La manera en que estos actores utilizan las funcionalidades del sistema proporciona los casos de uso identificados en la figura y que se relacionan en la siguiente lista:

- Actualizar valores. Incluye el caso de uso para medir variables del proceso físico. El actor detector de eventos solicita al sensor que mida las magnitudes de las variables para controlar y registra los valores medidos.
- Detectar eventos. Ocurre cuando el actor supervisor le solicita al actor detector de eventos si ha ocurrido alguno, lo que desencadena un flujo de eventos que incluye el caso de uso comunicar eventos cuando el detector da cuenta al supervisor de la ocurrencia de algún evento.
- Actualizar estado. Lo desencadena el actor supervisor cuando se ha detectado algún evento. La imagen se actualiza y el supervisor comunica el nuevo estado al detector de eventos, para dar cuenta del cambio de condiciones de operación. Para ello se incluye el caso de uso comunicar estado.
- Generar acción de control. Es un caso de uso que inicia el supervisor cuando el sistema se encuentra en un estado determinado y desea llevarlo a otro estado o iniciar una secuencia de eventos que impidan que el sistema llegue a un estado no deseado. Por medio de este caso de uso el actor supervisor le envía comandos al actor actuador. El caso de uso actualizar control está incluido en la generación de acción de control.
- Ejecutar acción de control. Es un caso de uso que utiliza el actor actuador cuando ya tiene un comando para ejecutar que le ha indicado previamente el actor supervisor.

Para describir las funciones de la unidad de producción se han establecido unos flujos de eventos para los casos de uso principales, de manera que se explique la interacción entre actores y casos de uso y detallar los objetos que intervienen en esta interacción y tener una idea de la estructura computacional que va a implementar una unidad de producción holónica. Para ello se usarán los diagramas de secuencia que proporciona el lenguaje de modelado UML [8].

#### 3.1 Flujo de eventos en la detección de eventos

En la Fig. 4 se observa el flujo de eventos básico para la medición de variables, en el que se muestra que el puerto de un sensor constantemente (iterativamente) está midiendo variables físicas asociadas con el proceso industrial por medio de algún dispositivo físico. Este puerto comunica al objeto sensor, que, a su vez, envía valores de la lectura al detector de eventos, que

asume los roles del actor sensor, que es un dispositivo electrónico con capacidad de comunicación. Es posible hallar flujos de eventos alternos, como por ejemplo cuando el sensor no funciona correctamente; en este caso, no se registra la lectura sino una condición de falla que desencadena otros eventos. El detector obtiene el valor de las variables continuas y aplica los algoritmos de detección de eventos para determinar si han ocurrido cambios en las condiciones de operación. Si han ocurrido cambios informa al supervisor sobre la naturaleza del evento y éste se encarga de actualizar la imagen que representa el proceso que se está controlando. Cuando se actualiza la imagen se informa al actor detector para que las próximas ejecuciones del algoritmo de detección de eventos se hagan con base en este nuevo estado. El supervisor, cuando recibe la notificación de ocurrencia de un evento, ha actualizado el estado del proceso y solicita el estado en que se encuentra el control, con el fin de ejecutar un algoritmo que le permita derivar una acción que se pueda ejercer sobre el control y sobre el proceso. Si ha derivado esta acción, la convierte en un conjunto de comandos que envía al actuador para su ejecución. El actuador ejecuta estos comandos y notifica al supervisor de su ejecución, para que este último actualice los estados del control. Escenarios alternos: el actuador no logra ejecutar los comandos, el algoritmo de control no produce acciones ejecutables, o no se detectaron eventos. Dos objetos muy importantes que son la base de este artículo se encuentran en este diagrama de secuencia: el estado del proceso y el estado del control.

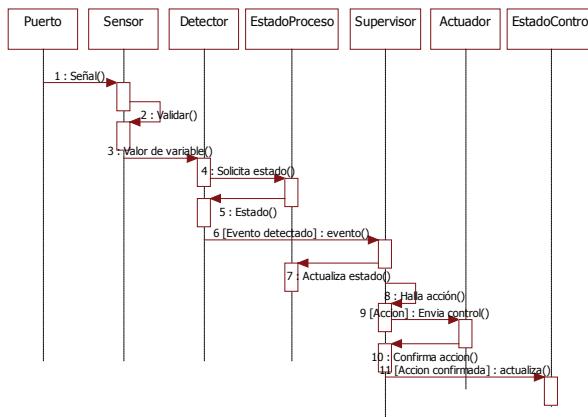


Fig. 4. Diagrama de secuencia de la medición de variables

Varios casos alternos pueden ocurrir cuando no se detectan eventos. En este caso, el detector le informa al supervisor que no han ocurrido eventos. También se puede dar el caso de que no haya acciones candidatas

por parte del supervisor, o no existen mediciones confiables (falla en sensores) de manera que no se registre el valor de una variable determinada.

### 3.2 Observación del proceso por parte de un holón agregado

En la Fig. 5 se aprecia el diagrama de secuencia que describe este flujo de eventos. El flujo principal de eventos es el siguiente: uno de los holones básicos envía un evento al objeto observador del holón agregado, que actualiza su estado abstracto y confronta el estado actual con el evento recibido. Un coordinador determina si hay acciones candidatas para mantener la trayectoria de estados, y para esto genera una acción que comunica al holón básico (o a otros holones básicos) y una vez recibe confirmación de esta acción actualiza el estado del control.

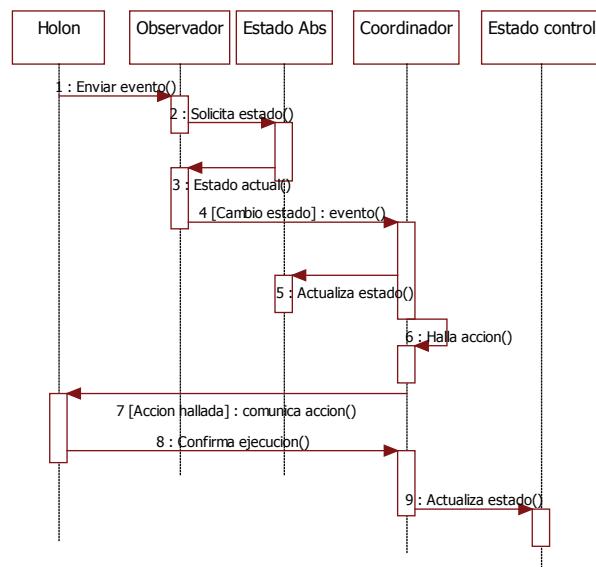


Fig. 5. Diagrama de secuencia de la observación del proceso por el holón agregado

### 4. MODELANDO LA CONFIGURACIÓN ESTÁTICA DE LA UNIDAD DE PRODUCCIÓN AUTÓNOMA

Para modelar la estructura computacional de una unidad de producción se recurre a los diagramas de clases [8], en los que se representan los objetos abstractos que manejan información relacionada con el funcionamiento de esta unidad. Para entender mejor estos diagramas se incluye una figura para mostrar los aspectos presentes en estos diagramas. Esto se muestra en la Fig. 6. El dibujo general de las clases representa al modelo de la estructura estática de un sistema, y las relaciones entre ellas. Los rectángulos

representan a las clases, que son plantillas de objetos, y las líneas representan las relaciones de agregación (componente/parte), especialización (herencia) y asociación. Esto ayudará a comprender mejor los diagramas subsiguientes, en los que se señalan las clases que intervienen dentro de una unidad de producción y sus relaciones.

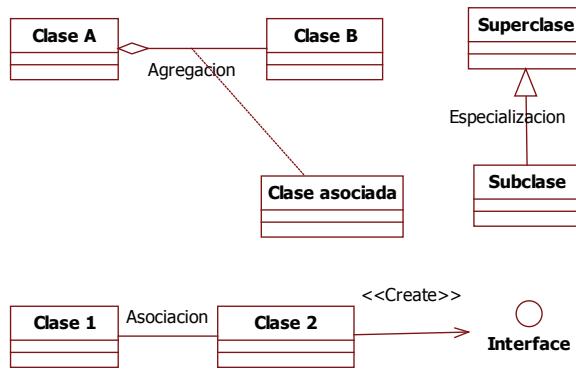


Fig. 6. Esquemas para interpretar un diagrama de clases

Una descripción basada en el lenguaje unificado de modelado (UML) para el estado del proceso se propone en esta sección y en las subsiguientes, para establecer un diseño que lleve a una posterior implementación del sistema holónico. La Unidad de Producción Autónoma (UPA de aquí en adelante) es un tipo particular de Holón, donde la información está relacionada con la disponibilidad de los recursos, objetivos de producción, métodos de producción y la dinámica del proceso continuo. Se asume que la UPA está basada en la arquitectura holónica de referencia PROSA [10], y el diagrama de clases que relaciona esta arquitectura con la propuesta de supervisión de la UPA se muestra en la Fig. 7. Se aprecia que el punto de partida son los holones Recurso, Producto y Orden, los cuales son especializaciones de la clase holón. Una UPA se compone de estos tres holones básicos, pero cuando se refiere a la supervisión de un holón básico se concentra en la información que generan las relaciones entre estos holones. Por ejemplo, a partir de la asociación entre holón Recurso y Producto se tiene la clase Tareas, que mantiene la información sobre el estado de los recursos, las tareas que hay que desarrollar en un recurso para obtener un producto, como se ejecuta un proceso en un recurso y las capacidades del recurso, entre otras. La clase *MetodoProduccion* surge de la asociación entre holones Recurso y Orden y

maneja información sobre cómo obtener un producto, dados ciertos recursos y las secuencias que se deben ejecutar. La clase *EstadoProceso* resulta de la relación entre Recurso y Orden, y contiene la información asociada con el progreso de la ejecución de los diversos procesos, como iniciar, reservar recursos, monitorear, interrumpir un proceso y reanudarlo, entre otras. El mecanismo de supervisión de la UPA está relacionada con estas tres clases, y a continuación se describe su estructura y comportamiento, así como su aplicación a sistemas de producción continuos.

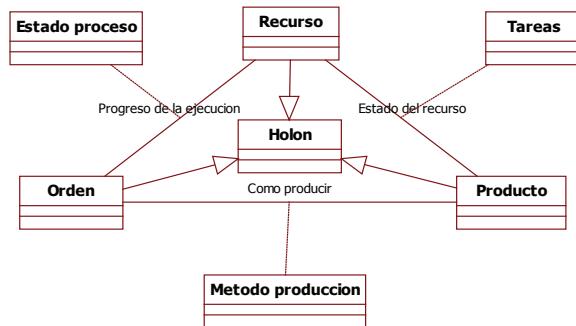


Fig. 7. Diagrama de clases para la Unidad de Producción dentro de la arquitectura PROSA.

#### 4.1 Unidad de Producción y su relación con la configuración de la planta

La clase *Unidad de producción* se compone de otras clases que manejan información sobre el estado de la unidad de producción, los métodos de producción, los recursos, el supervisor, asociado con una misión de producción. En la Fig. 8, adaptada de [2], se puede apreciar la relación que tienen y se establece que la misión está asociada con la clase *Supervisión*, cuya función es mantener el estado de la unidad de producción dentro de los estados deseados. Estas clases son: *Recursos*, que maneja la información asociada con los recursos para desarrollar el proceso productivo, cuya información se describe en la clase *Método de producción*. La información asociada con el estado de la unidad de producción se encuentra en la clase *Estado unidad de producción*, y el sub-sistema de control para mantenerlo dentro del comportamiento deseado se proyecta dentro de la clase *Supervisor*. De la interacción resultante entre el *Supervisor*, los *Recursos* y el *Método de producción* se obtiene la clase *Configuración*, que describe cómo está organizada la planta en un instante deter-

minado para ejecutar una tarea. Además, se advierte que la clase *Unidad de producción* contiene otras unidades de producción y un estado propio que se debe monitorear.

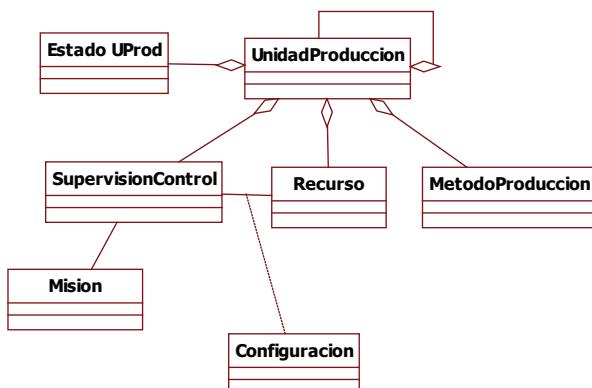


Fig.8. Diagrama de clases para la Unidad de producción.

#### 4.2 Supervisión y control del Estado de la unidad de producción

Como se planteó anteriormente, la Unidad de producción debe monitorearse para que exhiba una trayectoria de variables deseadas. Las teorías de control, supervisión y detección de eventos son útiles para llevar a cabo la función de monitoreo y, por lo tanto, deben considerarse en la implementación del sistema de información que registra el comportamiento de la Unidad de producción. Para supervisar el comportamiento de esta unidad de producción, es necesario conocer su estado interno y para ello se consideran las clases *Estado unidad de producción* y *Supervisión y Control* en una asociación que permite relacionar los mecanismos de control con los estados en los que se encuentran los recursos y la productividad. El estado del supervisor también es un componente de la unidad de producción. Unas clases que son indispensables para controlar la evolución dinámica de la unidad de producción son las que implementan al supervisor, detector de eventos y actuador, las cuales interactúan con el estado global de la unidad de producción y con el estado de las actividades de control, y son parte de la clase *Configuración y control*, que actualiza la configuración de la planta.

- La clase *Recurso* tiene la información sobre los recursos que se van a utilizar.
- La clase *Configuración* maneja las configuraciones para lograr la meta de producción, dadas las condiciones de los recursos existentes y el método de producción seleccionado.
- La clase *Orden* maneja información sobre lo que se va a producir en cuanto a flujo y calidad del producto.
- La clase *Estado unidad de producción* describe las condiciones en las que se encuentra la unidad de producción en un momento determinado. Se compone de la clase *Estatus Recurso*, que describe en qué estado se encuentra el recurso, y la clase *Estatus del proceso*, que resulta de la relación entre los recursos y la orden de producción.
- La clase *Supervisión y Control* controla la ejecución del proceso productivo por parte de la unidad de producción, procura cumplir el objetivo de producción con las restricciones impuestas por los recursos y el método de producción, y determina una configuración que debe aplicar la clase *Actuador*, que es uno de sus componentes. La clase *Detector* obtiene el estado de la unidad de producción, y por medio de la medición de variables asociadas con el proceso indica a la clase *Supervisor* si ocurrió un evento que haga necesario cambiar su imagen. Otro componente es la clase *Estatus del Control* que describe las medidas que debe tomar el supervisor ante la ocurrencia de un evento determinado, y el estado actual en el que se encuentran las medidas del control supervisorio.
- La clase *DES* describe a un sistema de eventos discretos y establece un marco común para describir los estados de los recursos y del control.

Conocer el estado de la Unidad de producción facilita la supervisión de su comportamiento, siempre y cuando exista un supervisor acoplado a esta Unidad de producción que ejecute ciertas acciones. Este supervisor es un sistema de eventos discretos, y tiene un estado interno, que consideraremos como el estado del control.

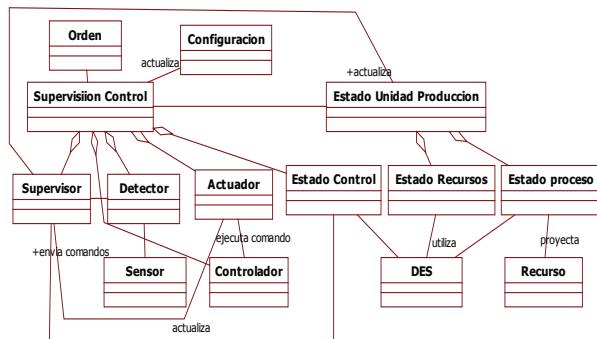


Fig.9. Diagrama de clases para representar la configuración estática de una unidad de producción.

Las relaciones antes mencionadas se observan en la Fig. 9, que muestra que las clases que representan al estado del proceso y al estado del control heredan características de la clase *DES*, diseñada para representar sistemas a eventos discretos de diferente naturaleza y sus componentes más importantes son los Autómatas de estados finitos, y las redes de Petri. Se profundiza en esta sección sobre lo que se considera como estado de los recursos.

#### 4.3 El Estado de los recursos

Como Recursos se considera el equipamiento para ejecutar las operaciones productivas, la energía, los servicios de transporte, por ejemplo, y el recurso humano para ejecutar las actividades de producción. Un atributo en común de estos recursos es la unidad de medida porque establece cómo se cuantifica la cantidad o disponibilidad de un recurso. Implican restricciones sobre el proceso de producción y se proyectan a una configuración definida para llevar a cabo un método de producción determinado. Conocer el Estado de la UPA permite que el supervisor controle esta unidad y, a su vez, mantenga actualizada su imagen. Una breve descripción operacional tiene la siguiente forma: la clase *estado Recursos* recibe información de los estados de los recursos que están operando por medio de la clase supervisor, y algunos recursos interactúan con la clase detector de eventos a través del sensor que informa al supervisor sobre la ocurrencia de eventos que cambien este estado, con el fin de que ejecute acciones para mantenerlo con el comportamiento deseado. Otros atributos y operaciones de esta clase conforman otras clases, cuyo diagrama UML estático se ve en la Fig. 10.

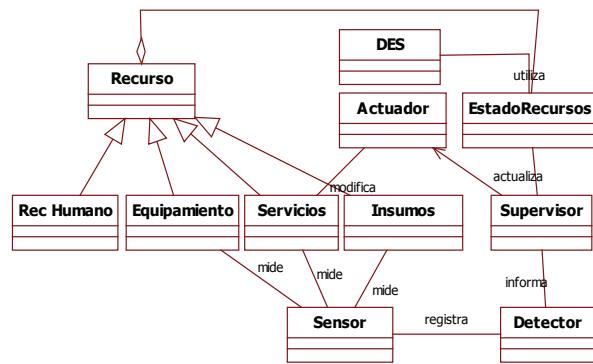


Fig.10. El Estado de los Recursos.

- Clase *RecHumanos* describe el estado en el que se encuentra el recurso humano asociado con lo productivo. Tiene los atributos *Identificador*, y *Rol asignado*.
- Clase *Equipos* maneja la información concerniente al equipamiento involucrado en la producción. Sus principales atributos son fabricante, fecha de compra, fecha mantenimiento, número de partes, ubicación, que describen el equipamiento, la capacidad que maneja este equipo (potencia eléctrica, peso máximo, potencia de carga, volumen) y el valor asociado con esta capacidad. Las operaciones que maneja son: la lectura sobre los puertos de comunicación, que es una subclase que sirve para relacionar el estado del equipamiento con la clase Sensor, que, además, ejecuta las funciones de medir variables que se desea controlar y las guarda en una estructura de datos. Además de las variables y los valores para medir, es importante conocer el período de medición del sensor y un mecanismo para variarlo, como *setPeriod()*, por ejemplo.
- Clase *Servicios* maneja información concerniente con los servicios utilizados para llevar a cabo el proceso productivo. Dentro de sus atributos están la capacidad nominal, la confiabilidad del servicio, el costo por utilizar una unidad de servicio y la equivalencia entre la unidad de servicio y el trabajo realizado. Esto permitirá ejecutar las operaciones de calcular el costo de uso del servicio. En general, cada equipamiento está asociado con el uso de por lo menos un servicio, con el consiguiente costo asociado por hacer funcionar un equipo determinado.

- Clase *Insumos* maneja información sobre el material que fluye sobre los otros recursos y se mide en términos de cantidad, masa, volumen o volumen por unidad de tiempo.

#### 4.4 Eventos en la operación de los Recursos

En la Fig. 11 se muestra una secuencia de eventos sobre cómo opera la supervisión de un recurso en particular. Un detector de eventos recibe datos periódicamente de un controlador, que, a su vez, los obtiene del recurso. Cuando este recurso es supervisado el detector de eventos indica al supervisor la ocurrencia de un evento. El supervisor produce una acción que ejecuta el actuador y modifica la configuración del recurso, y luego actualiza el nuevo estado del recurso. Dentro de un sistema holónico, los holones recurso y los de orden interactúan y producen el proceso, cuyo estado se está supervisando. De manera que esta dinámica no está aislada del contexto de producción. Esta secuencia de eventos se representa por medio de un sistema de eventos discretos, cuya dinámica se describe con el uso de algunas estructuras matemáticas y formales.

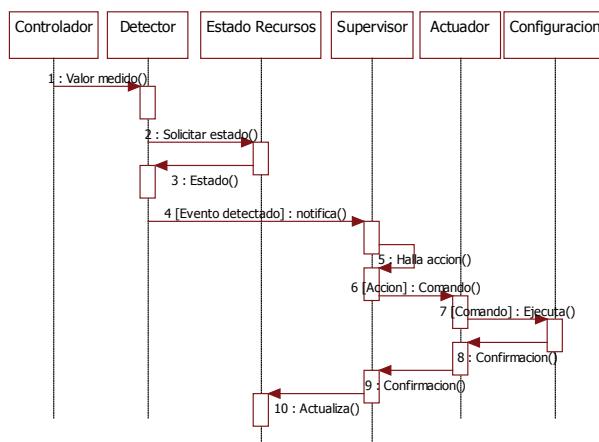


Fig.11. Secuencia de actividades dentro de un Holón Recurso.

#### 4.5 Estado del proceso

Una orden de producción resulta de la relación entre los holones Orden y Recurso. El primero aplica un método de producción que requiere que reserve algunos recursos para ejecutarlo. La clase *EstadoProceso* indica en qué etapa se encuentra el método de producción y proporciona una medida del producto obtenido, bien en términos de unidades ab-

solutas o en términos de flujo por unidad de tiempo. El comportamiento dinámico de esta clase lo describimos por medio de la clase DES. Un método que se aplica es el de *cambiar()*, en el que se registra la evolución sobre cómo se ejecuta la orden de producción. La Fig. 12 ilustra cómo se relacionan las clases antes mencionadas con las clases que ejecutan la supervisión y el control de la unidad de producción: el supervisor, detector de eventos, sensor y actuador. Para monitorear el sensor periódicamente mide las variables del proceso productivo y las notifica al detector de eventos, que establece la ocurrencia de algún evento que informará al supervisor. Este supervisor genera una orden de producción que, en el caso de los sistemas continuos, implica aplicar nuevas leyes de control. Esta orden de producción cancela la orden que se ejecutaba hasta ahora. El supervisor cambia la configuración del sistema por medio del actuador. La información asociada con las actividades cuya ejecución se monitorea se maneja por medio de la clase actividades, donde a cada tarea individual se le asocia un tiempo de duración (en caso de que sea predeterminada), precondiciones para ejecutar la tarea y cuáles tareas son habilitadas luego de su ejecución, es decir, las postcondiciones.

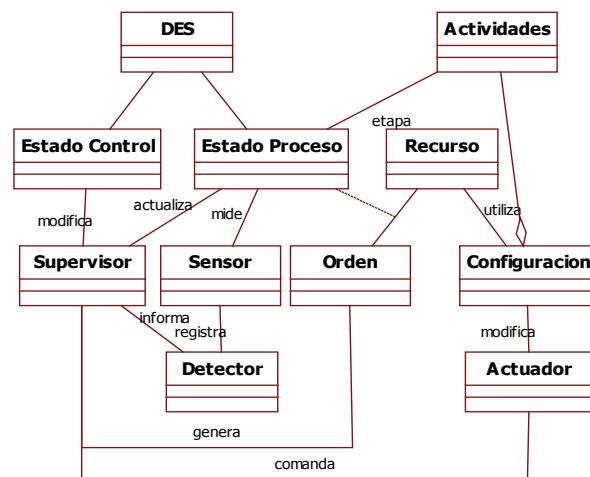


Fig.12. Clases utilizadas para describir el Estado del proceso.

#### 4.6 Dinámica del Control supervisorio y los holones

Si se considera la ejecución del proceso de producción como un sistema dinámico, se pueden obtener aportes de la teoría de Control supervisorio propuesta por Wonham y Ramadge. Esta teoría establece

que la dinámica de un sistema evoluciona de acuerdo con la ocurrencia de eventos y matemáticamente se describe de la siguiente forma:

$$\Phi_D : X_D \times \Sigma \rightarrow X_D$$

$X_D$  es el conjunto de estados de naturaleza discreta en que puede encontrarse el sistema en un momento dado, mientras que  $\Sigma$  es el conjunto de eventos que ocurren y ocasionan que el sistema cambie de un estado a otro. Este conjunto  $\Sigma$  está conformado por eventos controlables y no controlables, ambos disyuntos entre sí, es decir, un evento no puede ser controlable y no controlable a la vez. Los eventos controlables son aquellos que pueden generarse/inhibirse por un supervisor, mientras que los eventos no controlables dependen de la evolución del sistema dinámico, y generalmente están asociados con cambios de las variables de estado del sistema cuando pasan de una condición de operación a otra. Si se denota a los eventos controlables como  $\Sigma_c$  y a los eventos no controlables como  $\Sigma_{nc}$  se expresan estas relaciones como  $\Sigma = \Sigma_c \cup \Sigma_{nc}$ , y  $\Sigma_c \cap \Sigma_{nc} = \emptyset$ . La función  $\Phi_D$  matemáticamente corresponde a una función de transición, pero que en el campo de los sistemas de producción se hace corresponder con la aplicación de una ley de control, que se asocia con una configuración de la planta, particular para un estado determinado.

La configuración de la planta resulta de la relación entre el Holón orden, (objetivo de producción para un producto dado). La información de los recursos humanos, equipos, insumos y del avance del objetivo. El estado de cada uno de los componentes está enmarcado dentro del conjunto de modelos que describen el comportamiento de cada componente del Holón. El comportamiento global resulta de la composición de la dinámica de los componentes particulares, y se describe como la composición de un DES, que se muestra en la siguiente ecuación:

$$\Phi_k(\Phi_{k-1}(\dots \Phi_1(\Phi_0(X_R, X_P, t_0, t_1)) \dots))$$

$X_R$  y  $X_P$  son los estados de los recursos y del proceso de producción, respectivamente. La dinámica del control supervisorio determina cuál es la función  $\Phi_i$  para aplicar en el intervalo  $(t_{i-1}, t_i)$ , hasta que la evolución del sistema indique que el supervisor deba aplicar otra configuración que corresponda a la función  $\Phi_j$ , o que el sistema dinámico evolucione hasta llegar a ese estado. La aplicación de una configura-

ción la asociamos con la generación de una orden de producción por parte del supervisor, es decir, cada vez que sea necesario el supervisor interactúa con los diversos holones de la unidad de producción, y la secuencia de esta interacción se ilustra en la Fig. 13.

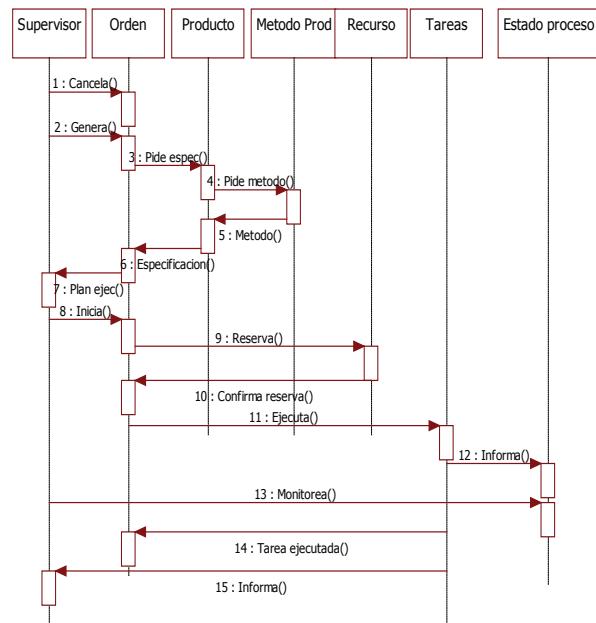


Fig.13. Secuencia de interacciones entre Supervisor, holones y clases asociadas.

Como se observa en la figura, el proceso inicia cuando el Supervisor cancela la orden actual y en su lugar crea una nueva orden (tal y como funciona en los sistemas continuos de un solo producto), para lo cual interactúa con el Holón Orden, que informa sobre las nuevas necesidades de producción. Este holón solicita al Holón Producto sobre las especificaciones del producto para producir, y el Holón Producto, a su vez, indica el método de producción que se aplicará. Con esta información, el Holón Orden informa al supervisor sobre cuál es su plan de ejecución. Una vez el Supervisor inicia la ejecución del proceso, el Holón Orden reserva los Recursos para ejecutar el método de producción seleccionado y empieza a ejecutar tarea por tarea. El Supervisor monitorea y actualiza el estado de su ejecución hasta que sea necesario generar otra orden.

## 5. LA DINÁMICA DE LOS SISTEMAS A EVENTOS DISCRETOS - DES

Para controlar una planta en la que ocurre un proceso industrial hay que llevar a cabo dos pasos: la

representación de su dinámica y la síntesis de un supervisor para procurar que exhiba una conducta deseada o, al menos, prevenir que ocurra una conducta no deseada. La dinámica del proceso industrial se representa por medio de un Sistema a Eventos Discretos (SDES), donde cada estado corresponde a un modo de operación, mientras que los eventos se consideran como el cambio de un estado a otro. Esta dinámica está muy relacionada con la evolución continua de las variables que rigen al proceso continuo, por lo que es necesario acoplar el SDES con el comportamiento de estas variables continuas. En todo proceso industrial las etapas y las condiciones de operación se expresan cualitativamente como estados, y el cambio de un estado a otro se denomina una transición. Si se etiqueta cada transición con un símbolo, el comportamiento de un proceso industrial se representa por medio de una secuencia de símbolos que representan las transiciones que hayan tenido lugar. Dentro de esta sección se muestra la síntesis del supervisor y la gestión del Holón.

Una manera de representar los Sistemas a Eventos Discretos de tal forma que permita su implementación dentro de una arquitectura computacional y de comunicaciones es por medio del diagrama estático UML de la Fig. 14. La clase principal, *SisEventosDiscretos*, está asociada con dos grandes clases: por un lado está la clase que describe a los Autómatas de estado finito, que se denomina *AEF*, y, por otro lado, está la clase que describe a los SDES como una red de Petri, que es la clase *PetriNet*. Cada una de estas clases está diseñada para cumplir con la especificación formal de un autómata y una red de Petri.

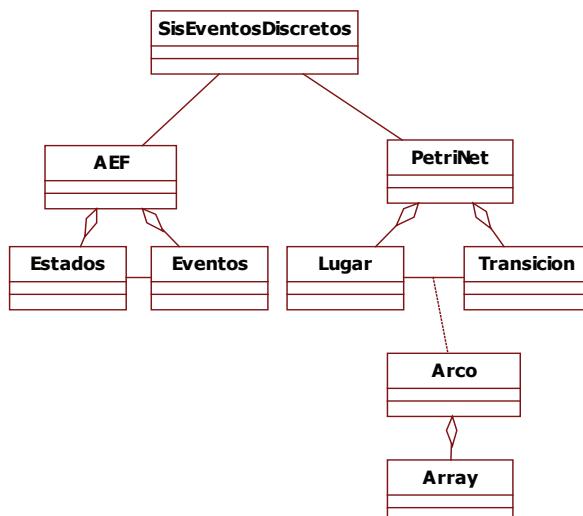


Fig.14. Diagrama de clases que implementa a un sistema de eventos discretos.

Antes de continuar con la especificación formal y estática de un AEF definido formalmente como un autómata, es necesario llevar a la formalización la definición de evento. Por evento se concibe el cambio de estado de un sistema y ocurre en un punto determinado en el tiempo. Generalmente, ese evento está asociado con alguna actividad que se despliega dentro del sistema y, si se denota cada evento por una letra de alfabeto, la evolución de un sistema crea un lenguaje, y así se obtiene una forma compacta de describir la dinámica de cualquier sistema. Una máquina *G* de estados finitos es una quíntupla que tiene la forma [9]:

$$G = (X, \Sigma, f, I, X_m)$$

Donde:

*X* es un conjunto finito de estados denotados por *x*, *x* es discreto.

*Σ* es un conjunto de eventos de entrada, denotado cada uno con una letra de un alfabeto.

*f* : *X* × *Σ* → 2<sup>*X*</sup> es la función de transición o evolución del autómata

*I* ⊂ *X* es el conjunto de estados iniciales

*X<sub>m</sub>* ⊂ *X* es el conjunto de estados deseados de *G* (estados marcados)

La clase *AEF* tiene como atributos el estado inicial, estado actual, y el evento entrante actual, por medio de los cuales implementa la operación *evolución()*, que actualiza a nuevos estados de acuerdo con el evento introducido. Los estados y eventos discretos conforman dos subclases del mismo nombre, y asociados con la clase principal *AEF*. Los eventos tienen otros atributos adicionales que permitirán su posterior control: unos indicadores de que sí son observables directamente y, a su vez, sí son controlables.

Otro enfoque para representar la dinámica discreta es el uso de Redes de Petri (o RdP, de aquí en adelante), ya que las especificaciones y la marcación de un sistema de producción se diseñan en la estructura de la Red de Petri. Las transiciones representan el comienzo o fin de una actividad, mientras que los lugares representan una operación o un estado del proceso. Los tokens, conocidos también como la marcación, son números enteros asignados a cada uno de los lugares y representan una disponibilidad o no disponibilidad para el status de un recurso o una cantidad para una operación.

Una Red de Petri es una quíntupla [7],  $PN = (P, T, F, W, M_0)$  donde:

$P = \{p_1, p_2, \dots, p_n\}$  es un conjunto finito de lugares  
 $T = \{t_1, t_2, \dots, t_m\}$  es un conjunto finito de transiciones  
 $F \subset (P \times T) \cup (T \times P)$  es un conjunto de arcos que relacionan lugares con transiciones y viceversa  
 $W : F \rightarrow \{1, 2, 3, \dots\}$  es una función de asignación de peso (ponderación)  
 $M_0 : P \rightarrow \{1, 2, 3, \dots\}$  es la marcación inicial  
 $P \cap T = \emptyset$  y  $P \cap F = \emptyset$

Para su implementación computacional, la clase *PetriNet* incorpora dos subclases: *Lugar* y *Transición* que reflejan al conjunto de lugares y de transiciones, respectivamente. Estas dos subclases están asociadas por medio de la clase *Arco*, la cual relaciona lugares con transiciones e implementa la función de asignación de pesos. Esta clase agrupa los pesos dentro de una matriz para obtener la matriz de incidencia, que utilizaremos para implementar las principales funciones de los lugares y las transiciones, a saber: el disparo de las transiciones (método *dispara()*) y la asignación de marcas a los lugares (función método *marca()*).

## 6. CONCLUSIONES

A partir de la confluencia de tres grandes áreas, como lo son las arquitecturas de referencia de automatización, la teoría de supervisión de procesos industriales y los métodos de modelado de sistemas, más experiencias obtenidas en la investigación sobre estas áreas, podemos llegar a la conclusión de que se puede hallar el estado en el que se encuentre para llevarlo a un esquema de persistencia de objetos que se implementa por medio de una base de datos y aplicaciones. Por medio de la aplicación de teoría de sistemas supervisores se puede sintetizar una supervisión de una unidad de producción compleja por medio de la agregación de supervisores de unidades de producción básicas.

Si se pueden implementar en forma computacional sistemas a eventos discretos, entonces se reutiliza esta implementación para representar estados de los recursos, del proceso productivo y del control, simplificar su diseño y permitir su posterior escalamiento para sistemas de producción de mayor complejidad. Por otro lado, la dinámica de este estado del proceso se valida matemáticamente y por medio de simulación, lo que permite implementarla en un sistema productivo real.

## REFERENCIAS

- [1] Blanc, P.: Pilotage par Approche Holonique d'un Système de Production de Vitres de Sécurité Feuilletées. Thèse de Doctorat. Université de Nantes, 2006.
- [2] Chacón, E., Besembel, I., Hennet, J.: Coordination and optimisation in oil and gas production complexes. Computers in Industry (bf 53) 17–37, 2003.
- [3] Chacón, E., Besembel, I., Narciso, F., Montilva, J., Colina, E. An Integration Architecture for the Automation of a Continuous Production Complex. ISA Transactions Vol. 41 No. 1 Jan. pp. 95-113. 2002.
- [4] Fishwick, P.: Simulation Model Design and Execution. Building digital worlds. Prentice Hall International Series in Industrial and Systems Engineering. 1995.
- [5] Larsen, M., Sorensen, C., Langer, G.: Development of a Production Meta Product State Model. Technical University of Denmark. 1999.
- [6] Moody, J., Antsaklis, P.: Supervisory Control of Discrete Event Systems using Petri nets. Kluwer Academic Publishers. 1998.
- [7] Murata, T.: Petri Nets, Properties, Analysis and Applications. Proceedings of the IEEE, vol 77, No. 4. 1989.
- [8] OMG.: The Unified Modeling Language. web: [www.omg.org](http://www.omg.org) 1998.
- [9] Ruiz, M.: Control y Supervisión de Sistemas Dinámicos de Eventos Discretos. Tesis de Licenciatura, Universidad de los Andes, Facultad de Ciencias. 2001.
- [10] Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P.: Reference architecture for holonic manufacturing systems. Computers in industry. 37 255–274. 1998.
- [11] Ramadge R, J. and Wonham W.: Supervisory control of a class of discrete event processes. SIAM J. Control and Optimization, 25(1), pp. 206 - 230, 1987.
- [12] Wyns, J.: Reference Architecture for Holonic Manufacturing Systems. Katholieke Universiteit Leuven, 1999.