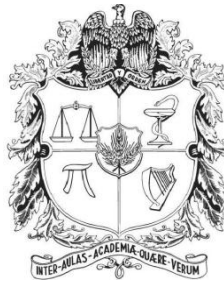


**Un Modelo de control inteligente para sistemas de
manufactura basado en los paradigmas Holónico y
Multi-Agente**

LUIS FERNANDO QUINTERO HENAO



Universidad Nacional de Colombia – Sede Medellín

Facultad de Minas

Escuela de Sistemas

Medellín

2009

Un Modelo de control inteligente para sistemas de manufactura basado en los paradigmas Holónico y Multi-Agente

LUIS FERNANDO QUINTERO HENAO

**Tesis de Maestría en Ingeniería de Sistemas
Línea de investigación en Inteligencia Artificial**

Director

DEMETRIO ARTURO OVALLE C. PhD.

Co Director

GERMAN ZAPATA MADRIGAL. PhD. (c)



Universidad Nacional de Colombia – Sede Medellín

Facultad de Minas

Escuela de Sistemas

Medellín

2009

AGRADECIMIENTOS

Al Director y Codirector de esta tesis, profesor Demetrio A. Ovalle y Germán Zapata

Dr. Edgar Chacón por el permanente acompañamiento.

Proyecto “Aplicabilidad del modelo de integración holónico en sistemas a eventos discretos con observabilidad limitada”

Proyecto “Arquitectura holónica de bajo costo para la supervisión, integración y control de sistemas de producción (Low cost Holonic Architecture)”

Grupo LOCHA – Grupo de Automática de la Universidad Nacional de Colombia – GAUNAL y Grupo TyT

RESUMEN

Los Sistemas de control Inteligente de Manufactura son organizaciones altamente distribuidas, que toman ventaja de las tecnologías de la información y las comunicaciones, de las técnicas de inteligencia artificial y de la teoría moderna de control, integrándolas en estos sistemas que compiten en un medio altamente dinámico y de economías ágiles. En estas circunstancias, el desafío se ha centrado en desarrollar sistemas de control inteligente con capacidades de autonomía, cooperación y de inteligencia, adaptación rápida a los cambios del entorno y más robustos contra la ocurrencia de disturbios. Para ello se han propuesto varias arquitecturas basadas en los paradigmas holónico (control holónico) y la inteligencia artificial distribuida mediante sistemas Multi-Agente.

Por definición, las unidades de producción holónicas son unidades autónomas que permiten el modelamiento de la información e infraestructura que compone el sistema de control inteligente de manufactura. El holón recurso definido como componente de la unidad de producción holónica, permite el modelamiento del comportamiento de las dinámicas existentes en los elementos que realizan una parte del proceso de manufactura. La misión y la ingeniería del holón representan las componentes funcionales del holón y representan el objetivo y el conocimiento de producción respectivamente.

En este trabajo se propone desarrollar un modelo de control inteligente que incluya las exigencias de modelamiento, diseño y análisis de los sistemas de control desde la teoría de sistemas dinámicos a eventos discretos, y atributos y requerimientos de representación tomados desde los paradigmas Holónico y Multi-Agente.

ABSTRACT

The intelligent control manufacturing systems are organizations highly distributed. They take advantage from information technologies and communications, from intelligent artificial techniques and modern theory control, integrate them in these systems that compete among a highly dynamic medium and agile economics. In these circumstances, the challenge has centered in developing intelligent control systems with features of autonomy, cooperation and intelligence, fast adaptation facing the environment changes, having robust parameters against the disturbs occurrence. Several architectures have been presented, based in holonic paradigm (holonic control) and intelligent artificial distributed through Multi-Agent systems.

By definition, the holonic production units are autonomous elements that permit to model information an infrastructure composing the intelligent control manufacturing system. The resource holon defined as a component of the holonic production unit, permit the modeling of the existing dynamics in the elements to do a part of manufacturing process. The mission and engineering of the holon represent the functional components and the objective and knowledge of production respectively.

This work propose to developing a intelligent control model including the modeling exigencies, design and analysis from control systems and their dynamic theory to discrete events, with the attributes a requirements took from holonic paradigm and Multi-agent.

TABLA DE CONTENIDO

CAPÍTULO 1	6
INTRODUCCIÓN	6
1.1 Introducción.....	6
1.2 Definición del Problema	8
1.3 Preguntas de Investigación.....	8
1.4 Objetivos	9
Objetivo General.....	9
Objetivos Específicos.....	9
1.5 Metodología.....	10
1.6 Aportes	12
CAPÍTULO 2	13
MARCO TEÓRICO.....	13
2.1 Sistemas a Eventos Discretos	13
2.2 Control de Sistemas a Eventos Discretos (DES)	15
2.3 Control de sistema de Manufactura	16
2.4 Control de Sistemas de Manufactura Basado en Agentes.....	18
2.5 Control de Manufactura basados en sistemas Holónicos.....	23
CAPÍTULO 3	26
REVISIÓN DEL ESTADO DEL ARTE	26
CAPÍTULO 4	32
MODELO PROPUESTO DE CONTROL INTELIGENTE DE SISTEMAS DE MANUFACTURA.....	32
3.1 Control Inteligente	32
3.2 Descripción de los Componentes Principales de Sistema de Control Inteligente .	34
3.2.1. Holón Recurso	34
3.2.2 Misión del Holón	36
3.2.3 Ingeniería del Holón.....	37
3.2.4 Actores y sus Roles	38
3.3 Marco Ontológico de la Arquitectura de Control Inteligente y Módulos Misión e Ingeniería del Holón.....	39
3.3.1 Ontología de Control Inteligente	39
3.3.2 Ontología del Módulo Misión	43
3.3.3 Ontología del Módulo Ingeniería.....	45
3.4 Supervisión y Comportamiento Dinámico	46
3.4.1 Modelos en redes de Petri del holón recurso	46
3.4.2 Modelos en redes de Petri de la misión del recurso	51
3.5 Autonomía y Negociación de la UPH.....	53
3.5.1 Autonomía de la UPH	53

3.5.2	Negociación en la UPH	63
3.6	Esquema conceptual.....	73
3.6.1	Modelo del dominio	73
3.6.2	Diagramas de secuencia	74
3.7	Enfoque de Desarrollo de un Sistema de Control Inteligente de Manufactura Basado en Unidades de Producción Holónica (UPH).....	77
3.7.1	Requisitos del sistema	77
3.7.2	Análisis.....	77
3.7.3	Diseño	78
3.7.4	Implementación	78
CAPÍTULO 5	80
VALIDACIÓN Y VERIFICACIÓN DEL MODELO DE CONTROL INTELIGENTE	...	80
4.1	Validación y Verificación Mediante CPN Tools.....	80
4.1.1	Análisis de Simulaciones.....	80
4.1.2	Esquema del Grafo de Alcanzabilidad	81
CAPÍTULO 6	87
CASO DE ESTUDIO.....		87
5.1	Definición de recursos	87
5.2	Definición de trabajos	88
5.3	Diagramas de proceso (OTIDA).....	93
5.4	Supuestos.....	94
5.5	Modelamiento del caso de estudio.....	95
5.5.1	Módulos del Aplicativo Desarrollado	95
5.5.2	Arquitectura del Laboratorio	99
5.5.3	Aplicación y Funcionalidades de la Herramienta	100
CAPÍTULO 7	104
PRUEBAS DEL SISTEMA DE PRODUCCIÓN HOLÓNICA	104
6.6	Comportamiento del sistema de producción holónico.....	104
6.6.1	Creación y Manufactura de órdenes de producción	104
6.6.2	Existencia de falla en el sistema holónico	110
CONCLUSIONES Y TRABAJOS FUTUROS	114
REFERENCIAS BIBLIOGRÁFICAS	116
ANEXO 1	126
CONTROL DE SISTEMAS A EVENTOS DISCRETOS.....		126
1.1	Modelado De Sistemas A Eventos Discretos	126
1.2	Formalismos de Modelamiento de Sistemas a Eventos Discretos	129
1.3	Redes de Petri	131
1.3.1	Propiedades de las Redes de Petri.....	136

1.3.2	Métodos de análisis de propiedades	137
ANEXO 2	139
	PROPIEDADES DINÁMICAS DE LAS REDES DE PETRI	139
2.1	Definiciones Formales de Propiedades	139
2.1.1.	Grafo de Alcanzabilidad.....	139
2.1.2.	Componentes Fuertemente Conectados	140
2.1.3.	Propiedades de Alcanzabilidad.....	141
2.1.4.	Propiedades de Acotación	141
2.1.5.	Propiedades de Reversibilidad.....	142
2.1.6.	Propiedades de Vivacidad	142
2.1.7.	Propiedades de Imparcialidad.....	143
ANEXO 3	146
	REPORTES DE VALIDACIÓN Y VERIFICACIÓN DE MODELOS	146
3.1	Estructura del reporte.....	146
3.2	Misión del recurso	150

CAPÍTULO 1

INTRODUCCIÓN

1.1 Introducción

Un sistema de control de manufactura, para un proceso de producción, se compone de elementos software junto con las diferentes entidades físicas del entorno de fabricación: recursos, productos, pedidos de clientes, operaciones de coordinación, etc. El elemento software y la entidad física, acoplados mediante un apropiado mecanismo de interacción, representan a los holones en este proceso. Cada uno de estos holones, una vez creados, se asume que son capaces de cierto grado de razonamiento local, con capacidad de decisión, y con la habilidad para comunicarse de manera interactiva con otros holones. El término holón es propuesto por Arthur Koestler en el libro “The Gost in the machine” (Koestler, 1967), en este un holón es una parte identificable de un sistema que tiene una identidad única, sin embargo, formado por partes subordinadas y a su vez es parte de un todo mayor.

La estrategia de control holónico realiza una reagrupación de los elementos distribuidos necesarios para realizar las actividades de manufactura, identificando los tipos de holones o agrupaciones que debe realizarse. Es necesario además, establecer las responsabilidades, la estructura de comportamiento y de interacción entre agrupaciones u holones.

Los holones realizan un control descentralizado para transportar, transformar, almacenar y/o validar información y objetos físicos. El control holónico implementado en el sistema de manufactura o proceso de control, permite mejorar el nivel de reacción a disturbios (comportamiento heterárquico para la reprogramación de operaciones y reacción a fallas en los recursos de producción) e implementar políticas claras de optimización del proceso (comportamiento jerárquico en estado estable del sistema). En consideración de los características holónicas mencionadas, los holones deben ser capaces de identificar cooperación (Leitao, 2004), (Bongaerts, 1998), (Van Brussel, 1998) para comprometerse en coordinación o negociación, y finalmente ejecutar la operación acordada. Por tanto, es necesario establecer los atributos de los sistemas distribuidos como negociación y cooperación (Sousa and Ramos, 1999), (Gou et al, 1998), a la vez que el sistema mantiene un comportamiento reactivo y proactivo (Chokshi and McFarlane, 2002), (Fletcher, 2000).

Los sistemas Multi-Agente (SMA) ayudan al modelamiento del control holónico en sistemas de manufactura, en el sentido que este no modela de manera explícita o son incapaces de modelar el comportamiento autónomo, y cooperativo (negociación).

La tecnología SMA a través de sus características de distribución, estrategias de optimización de desempeño de tareas, comportamiento autónomo y capacidad de cooperación presentan gran acoplamiento para el modelamiento de atributos de sistemas de manufactura holónica (HMS), y en consideración de estos la mayoría de los desarrollos en HMS tiene su fase de implementación en SMA. En consecuencia, los sistemas Multi-Agente son la tecnología para lograr la cooperación

entre los diferentes sistemas autónomos que componen un sistema de manufactura bajo el paradigma holónico.

Considerando además que los modelos de sistemas de manufactura que se han tratado anteriormente (Devia, 2004), (Chacón, 2003), (Giret, 2005), (Leitao, 2004), (Wyns, 1999) están relacionados sobre el modelamiento de sistemas de control de manufactura y considerando que la composición de las dinámicas presentes en un sistema de manufactura provienen de la existencia de estados discretos en su evolución dinámica, se hace necesaria la inclusión de atributos de diseño y modelamiento desde la teoría de control de sistemas a eventos discretos (Cassandras y Lafortune, 1999). La inclusión de estos atributos permite asegurar que el sistema bajo control se comporte de una manera adecuada.

Diferentes formalismos, metodologías y herramientas desde la teoría de control discreto han emergido y permiten validar el desempeño de los sistemas a eventos discretos (DES). En este campo los modelos lógicos se utilizan para describir características cualitativas de estos sistemas (Caillaud et al, 2002), (Sugiyama, 2006), las más representativas para evaluar son las propiedades que garantizan que el sistema está libre de bloqueos (liveness property), las que prueban que el sistema es acotado y estable (boundedness property, safeness property), las que permiten verificar que el sistema alcanza los estados deseables (reachability) y las que demuestran que el sistema tienen políticas claras de solución de conflictos (fairness property).

Por tanto, esta propuesta busca un Modelo de Control Inteligente para sistemas de manufactura basado en los paradigmas Holónico, Multi-Agente y desde el enfoque de los sistemas a eventos discretos.

En síntesis, el modelo de control inteligente que aquí se propone busca lograr un aporte al modelamiento de sistemas de manufactura mediante holones y sistemas Multi-Agente considerado el enfoque de los sistemas a eventos discretos, siendo este el enfoque que permite validar y verificar las dinámicas existentes en el comportamiento del sistema de manufactura modelado.

Lo que continua de este capítulo realiza una contextualización del trabajo de investigación realizado. El capítulo tres presenta las bases teóricas en relación a los sistemas a eventos discretos y los sistemas de Control de Manufactura basados en sistemas Multi-Agente y Holónico. El capítulo 4 muestra los componentes del control inteligente y la caracterización y modelamiento holónico de sistemas de manufactura. Al finalizar se presenta un enfoque de desarrollo de un sistema de control inteligente de manufactura basado en unidades de producción holónica. La validación y verificación de los modelos en redes de Petri obtenidos del holón recurso y la misión del holón, son analizadas en su comportamiento y estructura mediante los reportes y el grafo de alcanzabilidad calculado en el software CPN Tools. Finalmente en el capítulo 6, se ilustra el caso de estudio que permite validar el modelo de control inteligente y se presenta los resultados obtenidos desde el aplicativo desarrollado, que integra los atributos del control de sistemas a eventos discretos y los paradigmas Multi-Agente y holónico.

1.2 Definición del Problema

El Modelamiento del Control para Sistemas de Manufactura (Sistemas a Eventos Discretos - DES) presenta muchas limitaciones actualmente con respecto a la flexibilidad, autonomía y cooperación entre los elementos del sistema, propias de la adopción de arquitecturas de integración jerárquicas.

Además, las arquitecturas exploradas en el estado del arte no contemplan los requerimientos de diseño exigidos por la teoría de control.

Tampoco se incluyen herramientas para validar y verificar que el sistema de control sea robusto y cumpla con los requerimientos exigidos para un controlador en un Sistema de Manufactura.

A través de esta investigación se pretende a través de enfoques novedosos tales como los paradigmas Holónico y Multi-Agente y la perspectiva de los Sistemas a Eventos Discretos afrontar las limitaciones enunciadas anteriormente.

1.3 Preguntas de Investigación

A continuación se presentan las preguntas de investigación que surgen a partir de las limitaciones encontradas en el planteamiento del problema.

1. ¿Cómo se puede modelar a través de DES (Sistemas a Eventos Discretos) los atributos holónicos (autonomía, flexibilidad y cooperación) de los sistemas de manufactura?
2. ¿Se puede validar y verificar que el sistema de control para un Sistema de Manufactura sea robusto y cumpla con los requerimientos exigidos para un controlador?
3. ¿Cómo probar que el control integrado al Sistema de Manufactura Holónico cumple con los siguientes atributos: Autonomía, Cooperación, Proactividad y Reactividad?
4. ¿Son suficientes los DES para representar la dinámica y la inteligencia de un HMS?
5. ¿Cómo probar las propiedades de no-bloqueo y estabilidad en un sistema holónico?

1.4 Objetivos

Objetivo General

Proponer un modelo de control inteligente para sistemas de manufactura basado en los paradigmas holónico, Multi-Agente y modelamiento lógico de sistemas a eventos discretos.

Objetivos Específicos

1. Determinar los requerimientos exigidos desde la teoría de control para la representación de un sistema de manufactura holónico controlado, considerando los atributos de autonomía, cooperación, reactividad y proactividad.
2. Diseñar un modelo de control holónico de manufactura basado en unidades de producción y sistemas Multi-Agente para proveer flexibilidad, autonomía, cooperación, reactividad y proactividad en los sistemas de manufactura.
3. Proponer un método de síntesis y modelamiento de control holónico de DES considerando las exigencias de diseño a partir de la teoría de control y el enfoque de los sistemas Multi-Agente.
4. Implementar técnicas formales de validación y verificación de las dinámicas holónicas para comprobar los atributos holónicos en un sistema de manufactura diseñado desde DES.
5. Implementar un prototipo que valide el modelo propuesto y realizar pruebas de desempeño cualitativo y cuantitativo mediante indicadores definidos por la teoría de control.

Nota:

El objetivo específico No.1 busca responder las preguntas de investigación No.2, mediante el cual se identifican las necesidades desde la teoría de control determinación de un sistema de manufactura controlado y su modelamiento. Mediante el Objetivo específico No.2, las preguntas de investigación No.1 y No.3 serán trabajadas para buscar su respuesta. El objetivo específico No. 3 examina la solución de las preguntas de investigación No.4 y No.5. La pregunta de investigación No.6 encuentra solución en los objetivos específicos No. 5 y No.6 que son la validación e implementación de los modelos y el método obtenido.

1.5 Metodología

A continuación se presenta la metodología de trabajo para el desarrollo de la investigación a través de 4 fases:

Fase 1.

Especificación de requerimientos

La primera fase pretende realizar la caracterización de los paradigmas utilizados en la representación del control de sistemas de manufactura inteligente, identificando los elementos constitutivos, funciones de los elementos, requerimientos, atributos y tipologías existentes.

Además es necesario identificar los requerimientos y exigencias de síntesis de sistemas de control implementables en sistemas a eventos discretos, esto considerando que se ha establecido que las dinámicas de un sistema de manufactura son netamente discretas y definidas por la existencia de eventos en el sistema.

En esta fase es necesario considerar el modelamiento de los atributos holónicos de cooperación, autonomía, reactividad y proactividad mediante herramientas de sistemas a eventos discretos.

Fase 2.

Análisis y diseño del modelo

Esta fase utiliza la caracterización obtenida de los paradigmas implementados en el diseño de control de sistemas de manufactura holónica. De manera específica se considera la arquitectura de control basada en unidades de producción, la cual ya considera técnicas de diseño clásicas, a estas unidades es necesario definirles atributos de inteligencia distribuida y características de interacción entre UPH's que conforman el modelo de control inteligente, permitiendo atributos de autonomía, cooperación, reactividad y proactividad definidos desde el enfoque de los sistemas Multi-Agente.

El comportamiento dinámico del modelo de control holónico que se establezca deberá estar diseñado desde la perspectiva de los Sistemas Multi-Agente y los modelos lógicos establecidos desde la teoría de control para la representación de atributos de control en sistemas a eventos discretos.

Fase 3.

Propuesta Metodológica

La fase de aproximación metodológica permitirá construir un método de diseño y modelamiento de sistemas de control holónico basado en unidades de producción bajo la perspectiva de sistemas Multi-agente y la utilización de modelos lógicos de la teoría de control.

Una vez definido el método deberá definirse una notación de modelado y el alcance de su aplicación e implementación.

Fase 4.

Implementación y validación del prototipo

Mediante la utilización de técnicas formales, es decir, técnicas que tengan un soporte matemático, una sintaxis y semántica bien definida, y los atributos de desempeño definidos desde los sistemas a eventos discretos se realizarán las pruebas del modelo de control obtenido, estas pruebas pueden ser: Identificación de secuencias de eventos prohibidas o estados prohibidos, evaluación de existencia de bloqueos, entre otras.

Además se realizara una validación cuantitativa y cualitativa de este mediante un prototipo de pruebas que deberá ser seleccionado. Cabe señalar que en esta fase es un factor de consideración la selección de la plataforma informática para la implementación de los sistemas Multi-Agente mediante el cual se realizará la implementación del modelo de control holónico obtenido.

1.6 Aportes

- El modelamiento mediante un formalismo del holón recurso y la misión del holón, permiten representar la agregación y especialización en la estructura recursiva de la arquitectura de control inteligente, al mismo tiempo que permite verificar y validar su comportamiento, con el objetivo de garantizar el correcto desempeño del sistema.
- El modelo de control inteligente incorpora atributos de los sistemas Multi-Agente y Holónicos, a la vez que considera las características de representación de un sistema a eventos discretos de sistemas de manufactura, caracterizados por ser sistemas dinámicos complejos.
- En este trabajo se realiza una extensión del protocolo Contract Net (CNP) definido en la Inteligencia Artificial Distribuida (DAI) con el objetivo de cumplir las exigencias de los sistemas de manufactura holónica en la asignación distribuida de trabajos, y caracterizando la forma en la que los elementos cooperan ante una perturbación en el sistema.

CAPÍTULO 2

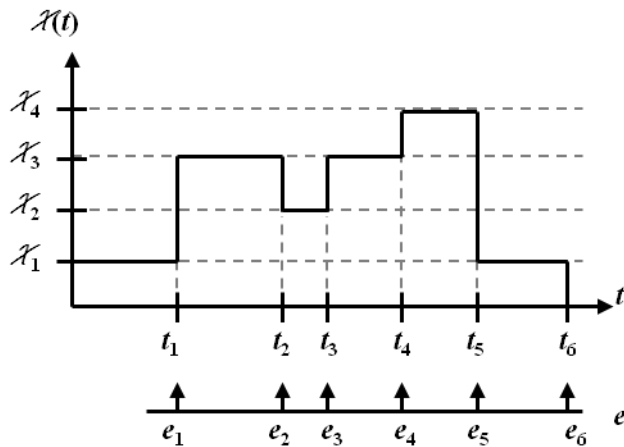
MARCO TEÓRICO

2.1 Sistemas a Eventos Discretos

Considerando que la teoría de control clásica se ha identificado desde antes para tratar con el control de sistemas netamente continuos, fue necesario definir una técnica para el control de sistemas discretos. Ramadge and Wonham (Ramadge and Wonham, 1989) definieron la propuesta para el control de estos sistemas, mediante los Sistemas a eventos discretos (DES).

Un sistema a eventos discretos (DES) es un sistema cuyo espacio de estados es finito, numerable y la evolución de estados depende de una ocurrencia asincrónica de eventos discretos (Cassandras y Lafortune, 2008). En estos sistemas un evento es la representación de un cambio instantáneo en algún punto del sistema y en un tiempo definido por la existencia del evento. Por tanto, un evento puede caracterizarse por un valor y un instante en el que ocurre. El valor puede ser un número, un vector, una palabra o, en general, un elemento cualquiera de un conjunto determinado por $X(t)$, este determina la trayectoria definida por una secuencia de eventos para todo valor de tiempo. En cada evento, la trayectoria toma el valor correspondiente (variable de estado) al del evento en cuestión. La Figura 1 muestra una trayectoria de eventos que toma valores X_1 en el tiempo t_1 , X_2 en el tiempo t_2 , entre otros.

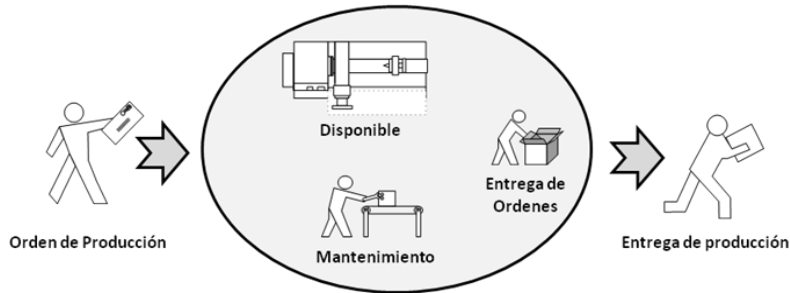
Figura 1. Ejemplo de trayectoria de un DES



Considere un proyecto de automatización para el control de un sistema de producción. En este el ingeniero de planta requiere reconocer los estados del sistema de producción: Disponible, en Falla y en Operación.

El ingeniero entrega los eventos que permiten un cambio de estado en el sistema de producción: liberación de una orden de producción hacia el sistema de producción, El sistema no tiene órdenes de producción, los recursos del sistema pueden fallar en un momento dado y por tanto deberá realizarse una actividad de mantenimiento.

Figura 2. Sistema de Producción



Los eventos que definen la dinámica del sistema son:

- e_1 : Liberar una orden de producción hacia el sistema de producción.
- e_2 : Entregar todas las órdenes de producción
- e_3 : Reportar falla en algún recurso de producción
- e_4 : Recibir informe de reparación del recurso de producción

Entonces $E = \{\text{Liberar, Entregar, Reportar, Recibir}\}$, que es el conjunto de eventos que pueden cambiar el valor de la variable de estado del sistema de producción.

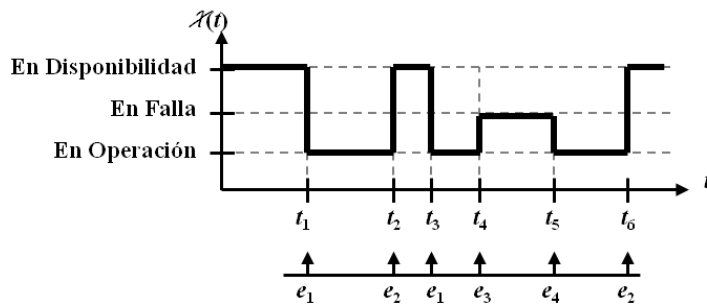
El espacio de estados definido por la ocurrencia de estos evento está definido por $\mathbf{X}(t)$:

$$\mathbf{X}(t) = \begin{cases} \text{En Operación} \\ \text{En Falla} \\ \text{En Disponibilidad} \end{cases}$$

Donde $\mathbf{X} = \{\text{En operación, En Falla, En Disponibilidad}\}$, que es la representación de estados que puede alcanzar el sistema de producción.

La Figura 3 ilustra una trayectoria representativa del sistema de producción.

Figura 3. Ejemplo de posible trayectoria del sistema de producción



De acuerdo al ejemplo presentado, es posible identificar que los DES tienen la capacidad de representación de eventos asíncronos de los sistemas a eventos discretos, que permite considerar en el modelamiento y simulación los efectos aleatorios presentes en la mayoría de los procesos reales.

En consecuencia los DES son investigados con el propósito de identificar los modelos analíticos más adecuados para el control y la propia síntesis de control de sistemas (Capkovic, 2000).

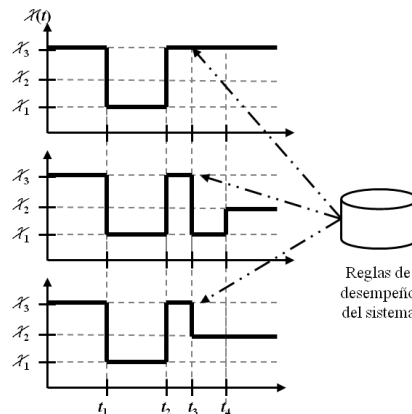
2.2 Control de Sistemas a Eventos Discretos (DES)

Cuando se ha definido un DES mediante un modelo, uno de los interrogantes de mayor interés es la forma de influir en su comportamiento que se realiza sin restricciones. La forma en cómo se influye este comportamiento es lo que se denomina el control de Sistemas a Eventos Discretos (DES).

Los sistemas a eventos discretos permiten el control automático de un sistema con respecto a una especificación describiendo una secuencia deseada de eventos. (Capkovic and Capkovic, 2001), (Capkovic, 2004).

La síntesis de control de DES consiste en hallar una secuencia adecuada de eventos discretos controlables (respetando los espontáneos), esto debe lograrse conjugando el modelo del sistema y la base de conocimiento (esencialmente conocimiento heurístico que representa las reglas de desempeño del sistema). Las posibilidades de control son generadas a través del modelo DES en cualquier paso del procedimiento de síntesis de control. Estas posibilidades de control son probadas con respecto a la existencia de condiciones. Cuando existen varias posibilidades de control satisfaciendo la existencia de condiciones, la posibilidad más apropiada es elegida por medio de la base de conocimiento, como se ilustra en la Figura 4.

Figura 4. Control de DES a partir de una base de conocimiento



Para el caso de la síntesis de control de un DES, Ramadge and Wonham (Ramadge and Wonham, 1989) definen los eventos controlables y no controlables. Los eventos controlables son eventos que pueden ser desactivados para evitar que ocurran o activados. Los eventos no controlables y no controlables son denotados por $\Sigma\chi$ y $\Sigma\upsilon\chi$ discretos (Cassandras y Lafortune, 2008), donde el comportamiento del sistema viene definido por:

$$\Sigma = \Sigma\chi \cup \Sigma\upsilon\chi, \Sigma\chi \cap \Sigma\upsilon\chi = \phi$$

La restricción del comportamiento completo del sistema es llevado a cabo restringiendo los eventos controlables en el momento en que sea necesario. Este método definido es llamado Control Supervisorio.

En el anexo X se presenta el modelado de sistemas a eventos discretos y los requisitos para lograr y una representación de un sistema dinámico. Además se presentan los formalismos de modelamiento de DES y en especial las Redes de Petri (PN) que son una herramienta matemática y gráfica que permite modelar, simular y controlar DES (Silva, 1985), (Guasch et al., 2002). Esta sección incluye las propiedades y los métodos de análisis de las PN.

2.3 Control de sistema de Manufactura

Los sistemas de manufactura involucran actividades de producción de bienes mediante el uso de recursos y conocimientos de manufactura. Adicionalmente, los sistemas de control de manufactura agrupan las actividades directamente responsables de manejar la transformación de la orden de producción planeada en un conjunto de salidas (productos) (Zapata, 2008A). De acuerdo a (Leitao, 2004) un sistema de control de manufactura es un factor clave en sistemas de producción automatizados, en las etapas de bajo y alto nivel.

En el caso del control de Bajo-Nivel, fácilmente se tienen dispositivos tales como robots industriales y máquinas NC (Numerical control), que requieren técnicas de control que regulen su comportamiento acorde a un objetivo específico, esto es que definan los movimientos. De otro lado el control de Alto-Nivel es concerniente a coordinar las actividades de los recursos de manufactura que permita producir los productos deseados. En ese caso la agenda que es el concepto de control de sistemas de manufactura presentado por (Quintero, 2008), define las actividades que un recurso debe llevar a cabo en un periodo de tiempo. En ese sentido un sistema de control de manufactura de alto nivel está definido por algoritmos que deciden que producir, como producirlo, cuando finaliza la producción y cuando los recursos están disponible u ocupados, cuando los trabajos son liberados en la planta, que trabajos son liberados, rutas de producción, y secuencias operación/trabajo.

De las principales funciones de un sistema de control de manufactura se encuentran las siguientes (Leitao, 2004):

- Funciones relacionadas al proceso (de acuerdo al sistema de manufactura se tendrán rutas de producción, tratamiento de órdenes de producción, entre otras.
- Planeación y Programación de producción.

La planeación de producción es el proceso de seleccionar y secuenciar actividades de procesamiento individual a fin de lograr uno o más objetivos para satisfacer los requerimientos del cliente.

La programación de la producción es un proceso para asignar apropiadamente los recursos necesarios para la fabricación de tareas y para identificar la secuencia y el calendario para cumplir los parámetros de estas tareas.

- Ejecución del plan (despacho, monitoreo y reacción a disturbios).
- Manejo de es estados anormales), manejo de bloqueos, entre otros.

En sistemas de control de manufactura al hablar de flexibilidad de un modelo de control de manufactura indica que un sistema debe ser capaz de adaptarse a varias circunstancias sin ver afectado de manera significativa su desempeño. Un cierto grado de flexibilidad (Causado por acciones) es característico de los sistemas de inteligencia artificial distribuida - Sistemas Multi-Agente que son flexibles por definición (Mönch and Stehli, 2006).

Adicionalmente, el desempeño de un sistema de manufactura cambia con el tiempo, basado en el comportamiento del cliente, la gama de productos, cuellos de botella como consecuencia de falla en las máquinas, cambios de prioridad en los trabajos, llegada de nuevas órdenes, entre otros. Por lo tanto estrategias de control de estas deben requerimientos específicos de la estructura del control del sistema de manufactura.

Los DES permiten modelar apropiadamente el comportamiento dinámico y estocástico de los sistemas de manufactura. Una vez definido el modelo de control, es posible realizar una evaluación del desempeño mediante las propiedades definidas por el formalismo

De otra parte, tradicionalmente los sistemas de control de manufactura han sido implementados usando técnicas de control centralizado y jerárquico, presentando buenas respuestas en términos de salidas debido a sus capacidades de optimización.

Los beneficios de un sistema jerárquico son los siguientes (Frayret, 2004):

- Unidades de control coordinan objetivos
- Implementación del sistema facilitada
- Complejidad limitada
- Tiempo de respuesta mejorado

Debilidades:

- Inadecuado para responder ante eventos inesperados
- Pobre capacidad de extender el sistema y realizar modificaciones imprevistas, lo que implica dificultad de reconfiguración del sistema.
- Pobre confiabilidad (parálisis de los niveles debajo de un punto de falla)
- Pobre tolerancia a las fallas.
- No soporta eficientemente la distribución y descentralización de funciones y entidades
- Dificulta la escalabilidad y reusabilidad del modelo de control.

En la actualidad el desempeño de los sistemas de manufactura debe considerar la flexibilidad y la agilidad del sistema de control. Las técnicas de control heterarquías introducen buena respuesta a los requerimientos de flexibilidad y agilidad, pero degradan la optimización de la producción.

En éstos, el funcionamiento global del sistema se acuerda por homologación de todas (o parte de) las unidades que componen el sistema, existiendo la posibilidad de comunicar cada una de estas directamente.

Es posible identificar las siguientes ventajas para una heterárquia (Frayret, 2004):

- flexibilidad y reactividad a los disturbios
- Son facilitadas las extensiones del sistema y las modificaciones imprevistas
- Complejidad reducida
- Tolerancia a fallos

Como se aprecia, muchas de las falencias de los sistemas jerárquicos son solucionadas al distribuir la responsabilidad en una heterárquia. Sin embargo, aún los sistemas de control heterárquicos presentan falencias:

- Falta de previsibilidad
- Pobre habilidad para definir cargas óptimas
- Falta de soluciones analíticas
- Posibilidad de deadlock (callejón sin salida)

Los cambios actuales hacia los nuevos modelos de control de manufactura son el desarrollo de sistema de control que combinen las técnicas heterárquica y jerárquica, satisfaciendo los requisitos impuestos por los entornos actuales, tal caso es implementado por los sistema de control de manufactura holónico que serán tratados en secciones más adelante.

2.4 Control de Sistemas de Manufactura Basado en Agentes

Lo nuevos requerimientos para los sistemas de control actual son las unidades de sistemas distribuidas y descentralizadas, conocimiento y habilidades, mecanismos de comunicación para la integración de sistemas distribuidos, rápida adaptación en el control y la reacción a disturbios (falla y cambios de organización) e interacción entre dispositivos físicos. Desde el punto de vista de sistemas Multi-Agente los “Sistemas de manufactura inteligente” consisten en una red integrada de recursos distribuidos, conocimiento de control y procesamiento de materiales.

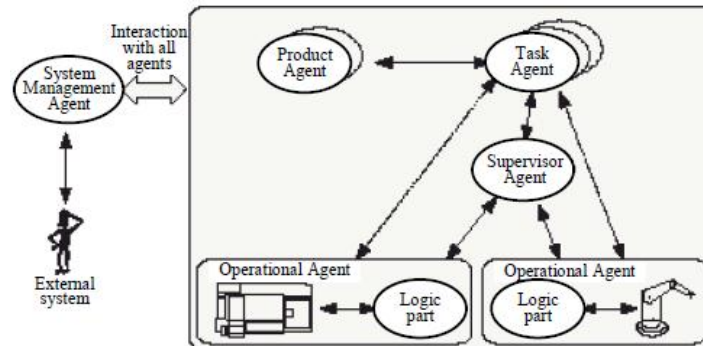
En la automatización y manufactura un agente puede ser entendido como un objeto de software con

un código de programación independiente que controla objetos tales como máquinas CNC, robots, dispositivos PLC's y sensores (Mačerauskas and Teresius, 2004), y como objetos lógicos un agente puede representar programas y ordenes de producción.

Los sistemas Multi-Agente han sido reconocidos como un promisorio paradigma para la próxima generación de sistemas de manufactura (Shen et al, 2006), como un enfoque importante para desarrollar sistemas de manufactura inteligente distribuida. En resumen, la comunidad científica ha logrado aplicar tecnología de agentes para la integración de sistemas de manufactura, planeación y programación de producción, y manufactura holónica como una implementación metodológica (Deen, 2003), entre otras aplicaciones específicas. En este sentido es posible identificar desarrollos en sistemas de control basados en agentes que son autónomos, organizados y reactivos (Deen, 2003). Algunas arquitecturas reconocidas son presentadas a continuación:

La arquitectura presentada por (Mačerauskas and Teresius, 2004) es específicamente determinada por los agentes operacionales que controlan las interacciones con otros recursos físicos, tales como máquinas CNC, robots, operarios, dispositivos PLC y sensores. El agente operacional se caracteriza por estar constituido por dos componentes: una parte de lógica que controla y una parte física representada por los recursos de manufactura. El agente supervisor coordina y supervisa otros agentes, acorde a la estructura organizacional y puede representar un controlador de celda. El agente de producto representa los recursos disponibles en el sistema y se ocupa del conocimiento e información asociada al producto, como el plan de proceso. El agente de tarea controla la ejecución de una tarea a fin de producir un producto y contiene la información dinámica sobre las órdenes de producción. El agente de gestión del sistema administra el sistema, supervisa y registra los agentes pertenecientes al sistema. La Figura 5 representa la arquitectura planteada en (Mačerauskas and Teresius, 2004).

Figura 5. Clases de Agentes (Mačerauskas and Teresius, 2004)



(Lim and Zhang, 2003) implementa un sistema Multi-Agente en el que el proceso de planificación y producción de programación están integrados. Esta estrategia es considerada como paso preliminar para lograr un sistema de manufactura ágil. La arquitectura del sistema Multi-Agente para el control de un sistema de manufactura definido por (Lim and Zhang, 2003) en cuatro clases de agentes resource manager agent (RMA), product manager agent (PMA), task manager agent (TMA), and execution manager agent (EMA). Cada agente administrador contiene un conjunto de

procedimientos para interactuar e intercambiar datos e información que permitan la utilización óptima de recursos y alternativas de reconfiguración del sistema de manufactura.

Giret propone en (Giret, 2005) un agente abstracto como artefacto de modelamiento para entidades autónomas con estructuras recursivas. Este agente facilita la traducción de la arquitectura holónica a un sistema Multi-Agente en su implementación.

MetaMorphic es una arquitectura Multi-Agente desarrollada en el proyecto MetaMorph (Maturana, 1999) para sistemas de manufactura inteligente. Su principal característica es la reconfigurabilidad, es decir, su capacidad de cambiar su estructura de control, para que se adapte dinámicamente a llegada de nuevas tareas y la evolución del entorno. Los tipos de holones primarios o básicos son los siguientes: holones de producto, holones de modelo de producto, holones de recurso. Un holón de producto por una parte consta de un componente físico, el producto en sí desde el inicio hasta el final; y por otra, almacena información acerca del estado del proceso de los componentes del producto durante la fabricación.

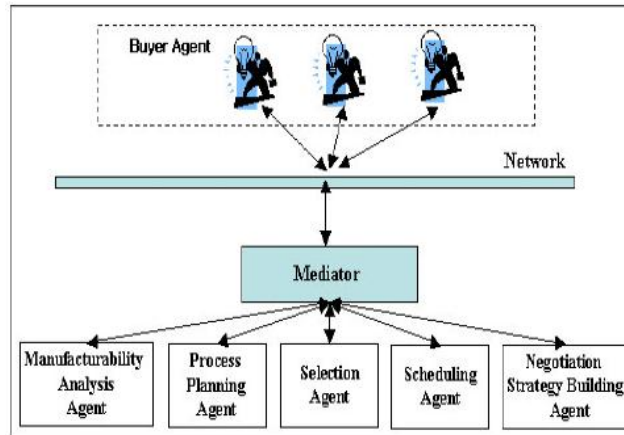
Un holón de modelo de producto almacena información sobre configuración, diseño, plan de proceso, materiales, calidad, etc., sobre el ciclo de vida del producto. Los holones recurso son utilizados para representar dispositivos y operaciones de fabricación.

MetaMorph II (Shen et al, 2000) es una extensión de MetaMorph I en él los agentes forman una federación basada en mediadores que tienen dos papeles: permitir que los agentes se encuentren entre sí y coordinarlos. Aquí, los agentes pueden ser usados para representar recursos industriales (máquinas, instrumentos etc.) y partes, encapsular sistemas de software existentes, funcionar como coordinadores de sistema/subsistema (mediadores), y realizar una o varias funciones de cadena de suministro.

PABADIS (Based Product Oriented Manufacturing Systems for Re-Configurable Enterprises) (Diep et al, 2003), extiende la aplicabilidad de la inteligencia distribuida a los niveles de control. La arquitectura PABADIS es construida con base a una colección de agentes autónomos que manejan las órdenes de trabajo con absoluta autonomía durante la programación y la ejecución del programa de producción (Klostermeyer, and Klemm, 2003) y (Ferrarini et al, 2006).

MANESS (Rim Choi , 2003) es una arquitectura Mediador desarrollada para responder a los cambios que suceden en la operación del sistema de manufactura a través de la cooperación de cada agente. Esta tiene simultáneamente la propiedad de una arquitectura de agentes autónomos ya que los agentes de MANESS tienen su propia función y resuelven un problema mediante la cooperación con otros, por tanto MANESS es una arquitectura híbrida de agentes mediador y autónomos. Concretamente el sistema consiste de un Mediador con la función de negociación, intercambio de mensajes y coordinación entre agentes, agentes de análisis de manufacturabilidad, agente de planeación de procesos, agente de programación, agente selección agente de estrategia negociación. Estos agentes desempeñan funciones relacionadas al control de un sistema de manufactura, y entregan la información necesaria al mediador para la negociación.

Figura 6. Arquitectura MANESS



Adicionalmente, la tecnología Multi-Agente es adecuada para entornos de manufactura distribuida, considerando los requerimientos de las aplicaciones desde la automatización y la manufactura como modularidad, descentralización e intercambiabilidad.

Los beneficios de la tecnología Multi-Agente son (Mačerauskas and Teresius, 2004), (Lim and Zhang, 2003):

- **Autonomía:** Una agente opera sin la directa intervención de influencia externa y tiene algún grado de control sobre su comportamiento.
- **Cooperación:** Los agentes interactúan con otros agentes, con objeto de lograr una meta común. Esto es, los agentes combinan sus capacidades dentro de grupos de colaboración para adaptarse y responder a diversos eventos y metas.
- **Reactividad:** los agentes perciben su entorno y responde rápidamente a los cambios que ocurren en él.
- **Proactividad:** Los agentes no actúan simplemente en respuesta a su entorno.
- **Adaptación y descentralización:** Los agentes pueden organizarse en una estructura descentralizada, y fácilmente puede reorganizasen dentro de diferentes estructuras organizacionales.
- **Habilidad Social:** Los agents interactúan con otros agents a través de lenguajes de comunicación de agentes, y tienen la habilidad para participar en actividades sociales con objeto de lograr sus metas colectivas.
- **Estructura de agentes:** estructura interna de los agentes y el nivel de auto-contenido.

- **Formación de grupos:** Definición de un protocolo de comunicación que permita un lenguaje común de intercambio para la cooperación. La formación de grupos para adaptar y responder a diversos eventos y misiones.
- **Reconfigurabilidad:** sistemas abiertos (adición, eliminado y sustitución de máquinas)
- **Escalabilidad:** Adecuación potencial para ampliar el nivel de la empresa.
- **Optimización global y local:** cómo lograr el óptimo global con agentes persiguiendo sus propios objetivos. Definición de un radio óptimo para el comportamiento jerárquico y heterárquico en una situación dada.

Aparte de las mencionadas características se han identificado otras cualidades, las cuales pueden ser atribuidas a los agentes en menor o mayor grado (Durán, 2005), (Maturana, 2003):

- **Racionalidad:** Un agente es capaz de razonar en base a determinados datos percibidos con el objeto de optimizar un resultado que de solución a un problema.
- **Adaptabilidad:** Propiedad de los agentes para aprender bajo determinadas situaciones y modificar su propio comportamiento en base ha dicho aprendizaje.
- **Movilidad:** La capacidad de un agente para moverse en una red.
- **Veracidad:** Todo agente debe entregar información fidedigna y no falsificarla deliberadamente.
- **Benevolencia:** La capacidad para ir en auxilio de otro agente (siempre que esto no juegue en contra de sus objetivos).
- **Tolerancia a fallas:** Agentes inteligentes poseen la capacidad de detectar fallos de los equipos y aislar las fallas para evitar su propagación.
- **Proactivo:** Los agentes periódicamente o asíncronamente proponen estrategias para controlar el sistema.

En general, las mejoras de la utilización de tecnología Multi-Agentes en automatización y sistemas de manufactura son la adaptación rápida a la reconfiguración de sistema (por ejemplo adición o eliminación de los recursos, diferentes estructuras de organización, entre otras.), la reutilización de código para otras aplicaciones de control, aumento de la flexibilidad y adaptación de la aplicación de control y desarrollo de software más optimizado y modular (Mačerauskas and Teresius, 2004). Esto implica que un SMA cumple con lo siguiente:

- No existe un sistema de control centralizado.
- Los datos se encuentran distribuidos.
- La computación (el procesamiento) de los datos es asíncrona.

- La asignación de tareas es consecuencia de la interacción dinámica de los agentes; entre todos se ponen de acuerdo en qué tareas se realizarán y quiénes las realizarán (Durán, 2005).

En (Mönch and Stehli, 2006) se presenta algunas características claves para decidir si los agentes son un marco viable para el control de sistemas de manufactura:

- **Representación de jerarquías de agentes:** las jerarquías de agentes son necesarias para modelar sociedades de Multi-Agentes para el control de sistemas de manufactura.
- **Representación de reglas de decisión:** la capacidad para representar a las reglas de decisión es esencial para permitir que el agente decida cuál es la siguiente acción basada en diferentes criterios, sus propios conocimientos y las circunstancias del entorno.
- **Modelamiento de restricciones del proceso:** el control de manufactura toma en cuenta las restricciones de los procesos controlados. Por lo tanto, se espera una forma fácil de modelar estas restricciones.
- **Soportar simulación para sistemas a eventos discretos:** la simulación de sistemas a eventos discretos para evaluar el desempeño del esquema de control de manufactura desarrollados con el marco existente. Una herramienta de simulación es necesaria para la emulación del comportamiento dinámico y Estocástico del sistema de manufactura a ser controlado.

2.5 Control de Manufactura basados en sistemas Holónicos

Los Sistemas Inteligentes de Manufactura y específicamente la manufactura holónica, son organizaciones altamente distribuidas, donde la inteligencia se distribuye sobre las entidades individuales llamadas holones, los cuales son entidades autónomas, cooperativas e inteligentes. En (Fischer et al, 2003), (Colombo et al, 2006), (Mondel and Tiwary, 2002) se presenta la definición de holón según Arthur Koestler: un holón es una estructura auto-similar o fractal que es estable y coherente y que consiste en varios holones como subestructuras, donde ninguno de estos componentes puede ser entendido sin los subcomponentes o sin la súper componente de que forma parte.

El concepto de holón desde el punto de vista de su aplicación en sistemas de manufactura, es posible que sea formado a partir de la definiciones encontradas en (Mondel and Tiwary, 2002), (Chacón, 2003), (Van Brussel, 1998), (Kotak et al, 2003) y en la que se encuentra la definición del consorcio HMS: holón es un bloque de construcción de un sistema de manufactura que está en capacidad de controlarse a sí mismo, considerando que son capaces con cierto grado de razonamiento local, con capacidad de decisión, comportamiento autónomo y con la habilidad para comunicarse de manera interactiva con otros holones. Esta funcionalidad es utilizada para transformar, transportar, almacenar y/o validar información de objetos físicos. El holón está conformado por una parte de procesamiento de información (componente de software) y una de procesamiento físico opcional (componente de hardware).

Los holones realizan un control descentralizado llamado control holónico, mediante una arquitectura holárquica implementada en el sistema de control de manufactura. Esta permite mejorar el nivel de reacción a disturbios e implementar políticas claras de optimización del proceso.

De acuerdo a (Kotak et al, 2003) una holarquía se presenta en un sistema de holones que pueden cooperar para lograr una meta u objetivo. La holarquía define reglas básicas para la cooperación de los holones y por tanto los límites de su autonomía.

El control holónico usualmente se describe como un concepto que combina las mejores características del control jerárquico y heterárquico. Acorde a van Brussel (Van Brussel, 1998), el control holónico difiere del control tradicional en los siguientes aspectos:

- Los holones permiten múltiples jerarquías
- Los holones forman jerarquías temporales
- Los holones no dependen de otros holones para obtener sus trabajos (autonomía).
- Los holones actúan cooperativamente en situaciones de perturbación (Cooperación)

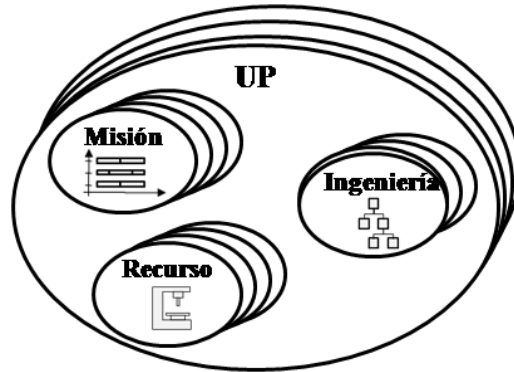
Estos atributos de acuerdo a (Kotak et al, 2003) hacen que los holones sean un paradigma adecuado para materializar la concurrencia y resolver los problemas de incremento de la dinámica en sistemas de manufactura.

Van Brussel introdujo una arquitectura holónica llamada PROSA (Van Brussel, 1998), que incluye tres tipos de holones básicos: holón orden, holón producto y holones recursos. Estas entidades básicas son estructuradas usando conceptos orientados a objetos tales como agregación y especialización. Adicional a los tipos de holones básicos definidos, la arquitectura PROSA también prevé holones Staff para ayudar a los holones básicos a realizar su trabajo.

La arquitectura ADACOR (Leitao, 2004) define un modelo similar a PROSA, con holón supervisor que introduce coordinación, formación de grupos y optimización global dentro de un control descentralizado. El objetivo primario es incrementar la agilidad y reconfigurabilidad de los sistemas de producción.

La Unidad Producción (UP) (Chacón et al, 2004), (Chacón, 2002) es una propuesta para alcanzar la construcción de sistemas de automatización integrados de control con base en una concepción holónica, mediante la cual ésta se concibe como la composición de un conjunto recursos, misiones o metas y métodos de producción o ingeniería, que son organizados y configurados de tal manera que permitan realizar los procesos de transformación en la cadena de valor, con el objetivo de obtener los productos exigidos. La Figura 7 presenta el esquema de una unidad de producción.

Figura 7. Unidad de producción holónica.



De acuerdo a (Chacón et al, 2004), y (Farahvash. and Boucher, 2004), las aplicaciones de manufactura holónica presentan características para las cuales los SMA se adaptan mejor al momento de la implementación, estas son: modularidad, descentralización, modificación, funciones complejas y los atributos básicos de comportamiento autónomo, y cooperativo (negociación). Además, trabajos de diseño de arquitecturas de integración holárquicas tienen su fase de implementación mediante el control inteligente de SMA (Leitao, 2004), (Vasko, 2000), (Giret, 2005), (Wyns, 1999), (Deen, 2003), (Marik, 2003).

De acuerdo a (Farahvash and Boucher, 2004) es razonable ampliar la capacidad de representación de estos holones con las funcionalidades de agentes inteligentes de software, considerando que los equipos o recursos de manufactura presentan comportamientos reactivos, proactivos y escalables, que son el fundamento para la cooperación y formación dinámica de grupos autónomos de holones (Kotak et al, 2003).

La arquitectura que se presenta en este trabajo para el control de sistemas de producción holónicos se basa en Unidades de Producción Holónicas (UPH). Estas son unidades fundamentales de representación y modelamiento de sistemas de producción que se caracterizan por un comportamiento autónomo, distribuido y orientados por el objetivo para obtener el producto requerido, con características de modularidad, agregación y descentralización de su comportamiento.

CAPÍTULO 3

REVISIÓN DEL ESTADO DEL ARTE

A continuación se presentan los trabajos más representativos en el área de control inteligente basados en Holones y sistemas Multi-Agente. Para ello se presenta una breve descripción señalando fortalezas y debilidades

➤ **PROSA: (Arquitectura Producto-Recursos-Orden-Staff-Arquitectura para sistemas de manufactura holónica). (Wyns, 1999)**

PROSA consiste principalmente en una arquitectura inter-holónica, que identifica los tipos de holones, sus responsabilidades, y la estructura en la cual interactúan. La arquitectura propuesta se basa en el concepto de agentes cooperativos y autónomos, llamado los holones (Brucker et al, 1998).

La arquitectura básica consiste de tres tipos de holones básicos orden de trabajo, producto, y recursos. El holón Staff es considerado en esta arquitectura como un elemento que incorpora conocimiento experto a los holones básicos.

Estos se estructuran utilizando conceptos orientados a objetos como agregación y especialización. Cada uno de los holones básicos es responsable, respectivamente, de un aspecto del control en la manufactura:

Un holón de recurso contiene una parte física (un recurso de producción dentro de un sistema de fabricación) y una parte de procesamiento de la información que controla el recurso. Ofrece capacidad y funcionalidad de producción a los demás holones. Un holón de producto mantiene el conocimiento de proceso y de producto para asegurar la fabricación correcta del producto. Actúa como un servidor de información para los demás holones del HMS. Un holón orden de trabajo representa una tarea en un sistema de fabricación. Es responsable de realizar el trabajo asignado de manera correcta y a tiempo. Gestiona los productos físicos que se están produciendo, el modelo de estado de los productos, y toda la información logística de procesamiento relacionada con la tarea (Valckenaers et al, 1994).

Fortalezas	Debilidades
Desarrolla un benchmark con una planta prototipo en el que se evidencia el impacto en el desempeño del sistema bajo esta arquitectura de control.	No se tiene herramientas para validar y verificar que el sistema de control sea robusto y cumpla con los requerimientos de un controlador para un Sistema de Manufactura Holónico.
El sistema de control es basado en un sistema de agentes los cuales son los encargados de hacer cumplir el objetivo de control.	
Es una arquitectura de referencia genérica para diseñar sistemas de control de producción	

➔ **ADACOR: Arquitectura Holónica Adaptativa para el Control de Sistemas de Manufactura. (Leitao, 2004).**

ADACOR es una de las arquitecturas holónicas mayormente referenciada que define un sistema de holones autónomos e inteligentes que apuntan mejorar el funcionamiento del sistema de control de producción caracterizados por la ocurrencia frecuente de disturbios inesperados.

La arquitectura se basa en un conjunto de entidades autónomas, inteligentes y cooperativas (holones), para representar los componentes de la fábrica. Estos componentes distribuidos pueden ser tanto recursos físicos (máquinas de control numérico, robots, controladores programables, etc.) como entidades lógicas (productos, órdenes, etc.)

ADACOR define los holones de manufactura en las clases de holones tarea, producto, operacional y supervisor.

Cada producto es representado por un holón de producto que contiene todo el conocimiento relacionado con él, y es responsable del proceso de planificación. Cada orden de manufactura se representa por un holón de tarea, que es responsable del control y supervisión de la ejecución de la orden. Incluye la descomposición de la orden, plan de asignación de recursos y la ejecución de

este plan. Los holones operacionales representan a los recursos físicos de manufactura, tales como operarios, robots y máquinas de control numérico. Gestionan el comportamiento de estos recursos de acuerdo a los objetivos, restricciones y habilidades y tratando en todo momento de optimizar su agenda. Los holones de producto, tarea y operacionales son bastante similares a los holones de producto, orden de trabajo y de recurso, presentados en la arquitectura de referencia PROSA. El holón supervisor de ADACOR representa al holón staff de PROSA, realiza tareas de coordinación y optimización global, coordinando a varios holones operacionales y supervisores (Leitao y Restivo, 2006).

Fortalezas	Debilidades
Implementa herramientas para validar, verificar y evaluar las especificaciones de estructura y comportamiento del sistema de control.	No evalúa el comportamiento de los agentes para la representación eficaz de las características de ambientes HMS.
Implementa funciones de control supervisorio.	No realiza un benchmark contra otras arquitecturas para validar el desempeño del sistema.
Implementación de herramientas de alto nivel para la representación de la dinámica de los holones.	No se especifica cual es la relación que existe entre la estructura estática del sistema y los distintos modelos de comportamiento de los holones

➤ **ANEMONA: Una metodología Multi-Agente para sistemas holónicos de fabricación. (Giret, 2005).**

Presenta una metodología para el desarrollo de Sistemas Holónicos de Manufactura en la que muestra como la tecnología Multi-Agente es apropiada para el desarrollo de HMS debido a la similitud de ambos enfoques, y muestra como la tecnología Multi-Agente ha sido utilizada como herramienta de implementación en aplicaciones del área de los HMS (Giret et al, 2005).

La propuesta se basa en la noción de Agente Abstracto y en los requisitos de modelado para HMS. Propone una guía de modelado clara y específica para el dominio HMS.

La definición de Agente Abstracto pretende emparejar la noción de agente a la de holón, "Un Agente Abstracto es un sistema informático con entidad única, situado en algún entorno, y que

como un todo, percibe el entorno (entradas sensibles de su entorno). A partir de esas percepciones determina y ejecuta acciones de forma autónoma y flexible - reactivo y pro-activo – que le permiten alcanzar sus objetivos y cambiar su entorno. Desde un punto de vista estructural, un Agente Abstracto puede ser un ente atómico; o ser un Sistema Multi-Agente (con entidad única) constituido por Agentes Abstractos no necesariamente homogéneos”.

Fortalezas	Debilidades
Definición de una lista de requisitos de modelado para Sistemas Holónicos de Fabricación.	Por ser una metodología netamente de modelado de HMS mediante agentes, no es posible a través de ella evaluar las especificaciones propias de los sistemas de control holónico.
Establecen etapas de desarrollo contando con metodologías específicas para HMS basadas en principios de la ingeniería del software.	
Presenta y desarrolla un caso de estudio de una empresa del sector cerámico.	
Estudio comparativo del grado de adecuación de metodologías de desarrollo de tres áreas: HMS, Modelado de Empresas y Metodologías SMA	

➤ **(Chacón et al, 2003). Unidad Holónica de Producción.**

Unidad Holónica de Producción es una propuesta para alcanzar la construcción de sistemas de automatización integrados de control con base en una concepción Holónica de las Unidades de Producción (UP), mediante la cual la UP se concibe como la composición de un conjunto de unidades elementales o recursos que son organizados y configurados de tal manera que permitan realizar los procesos de transformación en la cadena de valor, con el objetivo de obtener los productos exigidos. La UP toma sus propias decisiones (autonomía) respecto al cumplimiento de su objetivo, pero está obligada a informar su estado en el cumplimiento de una meta o si ésta no se puede cumplir debido a una falla o errores en su comportamiento.

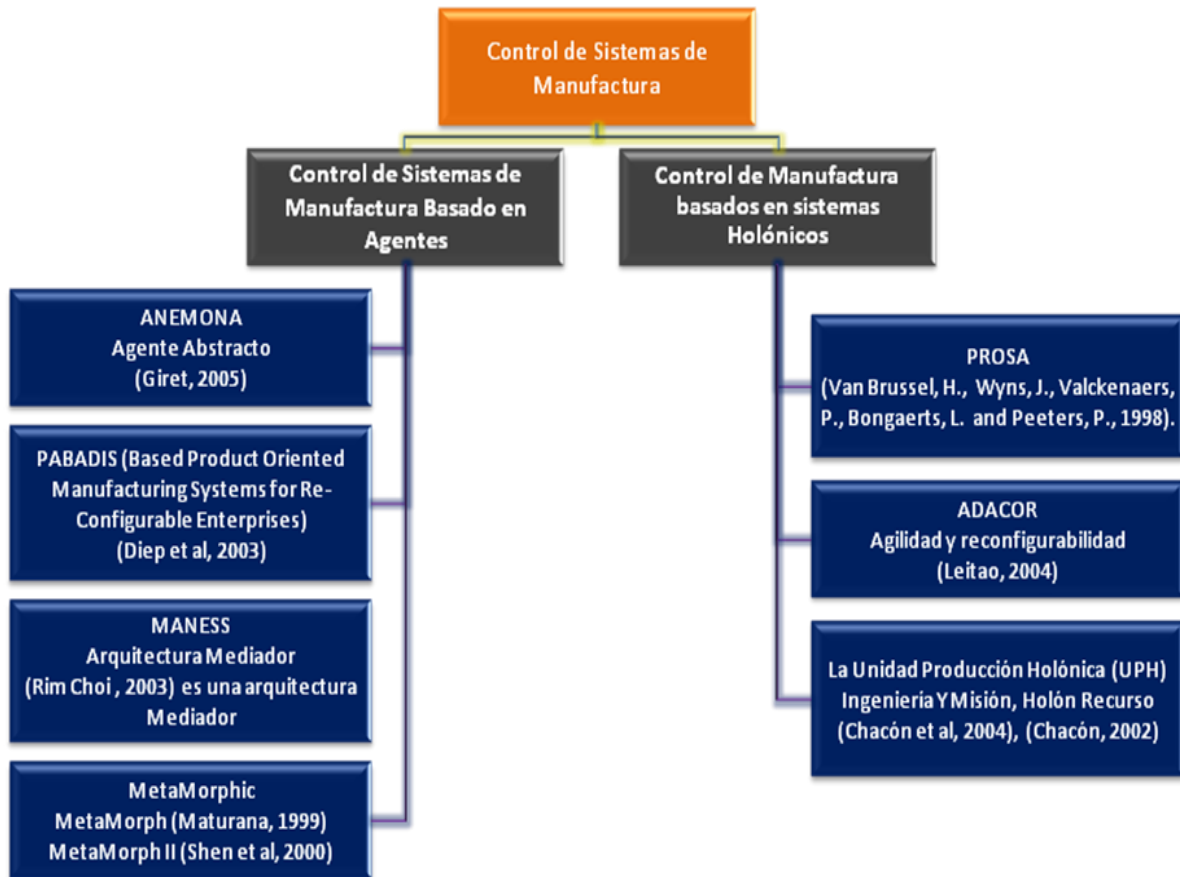
El enfoque de Unidad Holónica de Producción divide a la UP en holones, estos son **Misión, Recursos y Método**. En las dinámicas del holón misión se especifica el producto a ser completado, con sus calidades y cantidades. El método de producción, está asociado al uso de los recursos y materiales con el fin de cumplir una misión propia de la actividad productiva de un sistema de manufactura. Los recursos que conforman el conjunto de instalaciones, equipos, personal especializado, servicios e insumos, necesarios para ejecutar un método de producción conforman el holón Recurso (Devia, 2004).

Dentro de los elementos a ser representados mediante una unidad de producción están: elementos físicos (Infraestructura, Productos, configuración de la Unidad de Producción objetos de información (misión: producto a ser obtenido, cantidad, calidad, estado de la unidad de producción: avance de la misión, estado de los recurso y estado del proceso).

Fortalezas	Debilidades
Consideración de técnicas de control clásica en el diseño de la estructura de supervisión de las unidades de producción.	No tienen una metodología de implementación de la arquitectura.
	No se establecen etapas de desarrollo. No se especifica cómo se traduce cada modelo de UP en alguna plataforma de implementación.
La arquitectura tiene la posibilidad de ser extendida a los sistemas de producción continúa.	No validan el comportamiento de la unidad de producción mediante herramientas formales.
	No implementan un método para la especificación de sistemas holónicos de control, ni para la composición de dinámicas.

Otros trabajos relacionados con el control de sistema de manufactura se presentan en la Figura 8.

Figura 8. Trabajos de referencia en Control de Sistema de Manufactura



CAPÍTULO 4

MODELO PROPUESTO DE CONTROL INTELIGENTE DE SISTEMAS DE MANUFACTURA

3.1 Control Inteligente

El control automático se entiende como un proceso en que las máquinas pueden ser operadas y controladas automáticamente acorde a programas definidos o procedimientos, se caracteriza por ser un control sin interferencia humana. De acuerdo (Cai, 1997) el control automático de sistemas ha encontrado múltiples dificultades proveniente de la incertidumbre en los modelos, alto grado de no linealidad, perturbaciones, cambios abruptos en la dinámica, toma de decisiones de forma jerárquica y distribuida, múltiples escalas de tiempo, patrones complejos de información, gran cantidad de datos, requisitos de desempeño, entre otros. Esto ha dificultado el análisis y diseño de modelos de control, que las hipótesis planteadas en el modelo no coincidan en la práctica y pérdida del desempeño de los sistemas de control.

En consecuencia Cai (Cai, 1997) presenta que el grado en que un sistema de control trata con éxito las anteriores dificultades, depende del nivel de inteligencia en el sistema de control y por tanto el control automático implementado hasta ahora, debe resolverlo usando técnicas de inteligencia artificial en los sistemas de control, o aplicar metodologías híbridas que consideren el control automático tradicional y las técnicas de control inteligente. Esto permite obtener sistemas de control que consideran múltiples variables, control adaptativo a fallas, control estocástico y no lineal, sistemas de control con parámetros distribuidos, sistemas de control con variables y eventos discretos, procesamiento de información y estructura de comunicaciones para el procesamiento distribuido de información y toma de decisiones. Por tanto, el control inteligente es el proceso que controla unas máquinas inteligentes autónomas para que logre sus metas autónomamente sin interacción humana. Un sistema de control inteligente debe estar provisto de inteligencia y capacidades de programación y ejecución de tareas y puede ser obtenido por la intersección de múltiples disciplinas como la cibernética, inteligencia artificial, informática, investigación de operaciones y el control automático (Galán et al, 2000).

Entre las técnicas de Inteligencia Artificial usadas en Control Inteligente destacan: Sistemas Expertos (la base de conocimientos de control se obtiene de un experto humano, Lógica Borrosa (trata con la imprecisión de los sistemas), Algoritmos Genéticos (el sistema de control evoluciona a partir de la tasa de fallos y aciertos).

Adicionalmente, en (Velazco, 2007) trata que el control inteligente se caracteriza por:

- Sistema que tiene la habilidad para actuar de forma apropiada en un entorno incierto
- Inteligencia=Proceso de análisis, organización y conversión de datos en información estructurada (conocimiento).

- Sustitución a la mente humana en la toma de decisiones, planificación y aprendizaje.
- Utiliza de forma combinada técnicas de Inteligencia Artificial, Investigación Operativa y Control.
- Capacidad del sistema de asemejar el comportamiento de alguno de sus elementos a alguna de las cualidades cognoscitivas (Galán et al, 2000) del comportamiento humano, como el aprendizaje, el razonamiento simbólico, la planificación o la adaptación a un medio cambiante.

La estructura de control inteligente de sistemas de manufactura se caracteriza por soportar la adaptación ágil a diferentes estructuras organizacionales y la auto-organización, que permita la reorganización de los Elementos constitutivos de la arquitectura-Holones-Agentes dentro de diferentes estructuras organizacionales (Mačerauskas and Teresius, 2004). En adición, estos cambios de configuración pueden ocurrir como consecuencia de sistemas con múltiples modos de funcionamiento, debido a cambios en la estrategia de producción, o debido a la aparición de una gran diversidad de posibles perturbaciones en el sistema (Galán et al, 2000). La flexibilidad de un sistema viene dada entonces por la capacidad de tomar decisiones y responder con rapidez a las exigencias siempre cambiantes de los clientes y a las condiciones de la planta.

Algunos de las necesidades específicas para el control inteligente en sistema de manufactura se detallan a continuación:

- Las técnicas de control inteligente permite controlar sistemas distribuidos
- El objetivo general es aumentar el grado de autonomía del controlador, a fin de tolerar una gama más amplia de variaciones de parámetros de planta y disturbios.
- Capacidad de negociación por cambios de operación en el sistema de manufactura.
- Aplicar la técnica de control inteligente en los procesos de producción, permite imitar la experiencia humana en la construcción de modelos basados en conocimiento y generalizados, que permitan implementar un razonamiento, decisión y control automático.

De forma complementaria (Davis, 1999) presenta que una arquitectura de control efectiva deberá poseer las siguientes propiedades:

- Permite al usuario especificar las tareas de alto nivel, que sean recursivamente descompuestas en tareas más detalladas que deben ser ejecutadas.
- La consideración de la planeación y problemas en varias escalas de tiempo y en varios niveles de detalle. Estos es, la planificación y control son de naturaleza multi-resolucional.
- Permite comportamientos funcionales complejos para ser descompuestos en sub-funciones más manejables.
- El objetivo de control inteligente es integrar la planificación y control a fin de permitir que un sistema planifique y ejecute su respuesta de una manera on-line.

- La planeación del sistema de control debe definir con qué recursos de procesamiento se realizarán las tareas y a continuación programar su ejecución.

Conjuntamente, cuando las tareas de un sistema de control están siendo consideradas, la respuesta del sistema debe tratarse mediante sistemas a eventos discretos, de modo que sea posible describir la secuencia de acciones que el sistema de control inteligente debe implementar para estabilizar el sistema u orientarlo a la consecución del objetivo mediante el supervisor del sistema, que implementa la intervención sobre las variables controlables. En consecuencia el comportamiento del sistema de control está definido por los estados y eventos Discretos que permiten diseñar y analizar de una manera más completa un sistema de manufactura integrado, además de la obtención mediante PN de un modelo del comportamiento de los elementos que componen la arquitectura de control.

En esta tesis se presenta una propuesta de control inteligente basada en técnicas de inteligencia artificial y control automático de sistemas de manufactura. Los elementos de la arquitectura son definidos a partir de los sistemas de manufactura holónica basada en Unidades de Producción con atributos adquiridos desde los sistemas inteligentes distribuidos Multi-Agente, de aplicación en control de procesos de manufactura complejos. A pesar de que el control inteligente se caracteriza por la ausencia de la intervención humana, es reconocido que los sistemas de manufactura a nivel local son sistemas de islas de automatización y por lo tanto ciertas situaciones en el sistema de control inteligente deben considerar el modelamiento de operarios. El control inteligente se afecta entonces por interacción de los sistemas a eventos discretos, sistemas Multi-Agente y Sistemas de control Holónico como se ilustra en la Figura 9.

3.2 Descripción de los Componentes Principales de Sistema de Control Inteligente

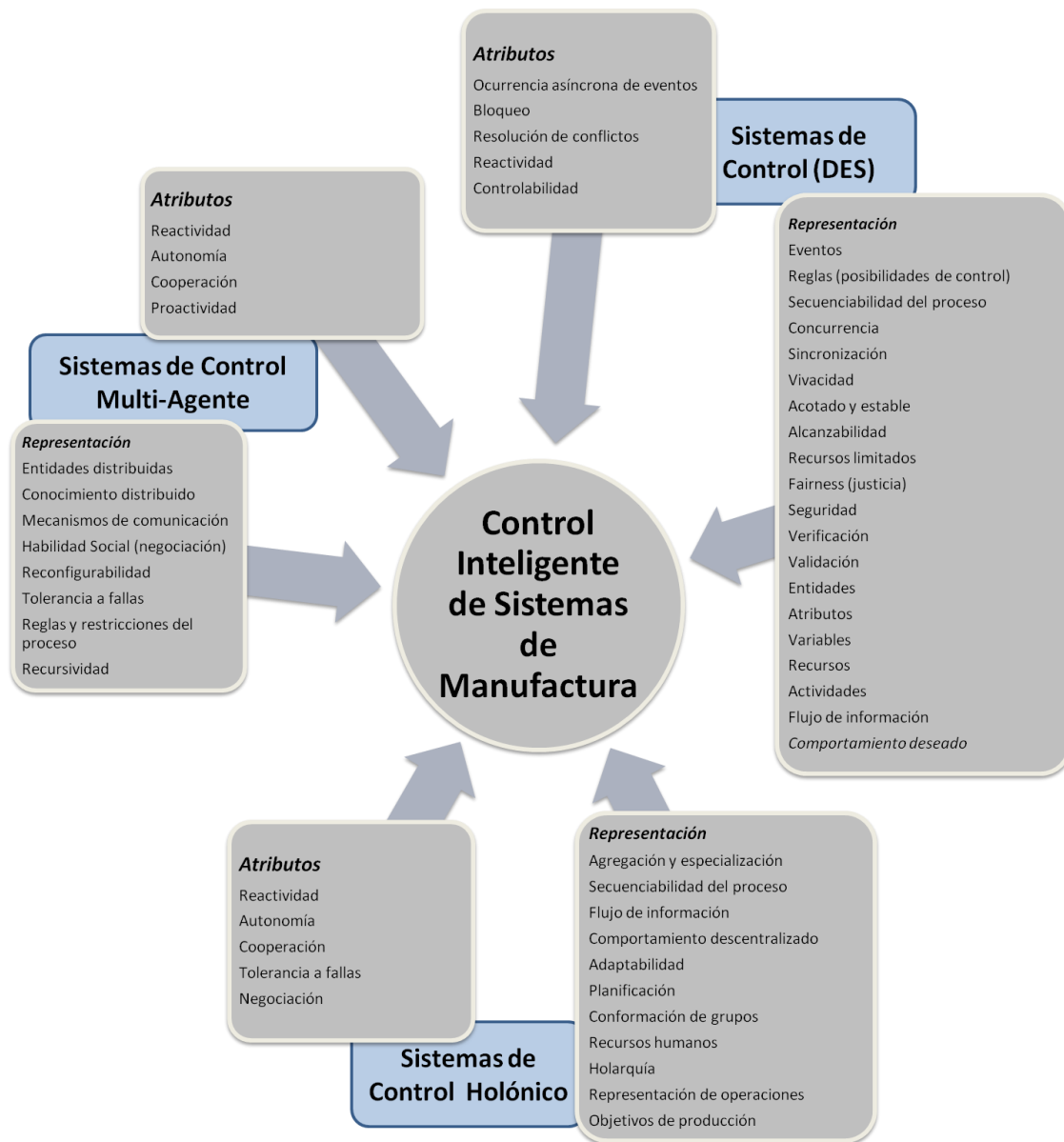
La arquitectura de control inteligente para la que se propone su modelamiento, se define a partir de la unidad Holónica de Producción, esta es una propuesta para alcanzar la construcción de sistemas de automatización integrados de control inteligente, mediante la cual la UP se concibe como la composición de un conjunto de unidades elementales o recursos que son organizados y configurados de tal manera que permitan realizar los procesos de transformación en el sistema de manufactura. Adicionalmente, es necesario definir otros elementos básicos como la misión e ingeniería del holón que representa el objetivo que el sistema de control debe lograr y el conocimiento necesario para lograrlo respectivamente. Los elementos básicos de la arquitectura de control inteligente basados en UP se presentan a continuación.

3.2.1. Holón Recurso

El holón recurso se refiere a la infraestructura que posee una unidad de producción, las cuales pueden ser fábricas, máquinas e incluso personas. Todas ellas se destacan por tener una componente física y una componente inteligente, la cual toma decisiones en base a su meta, su estado actual, su

agenda y sus relaciones con su entorno. Si el elemento como tal no tiene inteligencia, entonces lo clasificamos como un simple dispositivo o equipo, aclarando que estos elementos sin inteligencia si pertenecen a un holón recurso. Una componente importante que contiene el holón recurso es el inventario que en nuestra teoría será llamado almacén, en el cual se guardan materias primas y piezas terminadas. Las tres funciones principales con las que debe cumplir un holón recurso son de transformación, almacenaje y transporte, las cuales van a estar supervisadas por una unidad llamada supervisor del holón recurso, encargado de vigilar que todo este organizado, verificar el avance de la misión, cooperar con otros holones recurso en caso de desvíos en su objetivo, sincronizar información con el supervisor de la unidad de producción y readaptarse a condiciones adversas.

Figura 9. Necesidades para la representación y atributos de sistemas de control inteligente de sistemas de Manufactura



En el inicio de los procesos de operación de cualquier holón recurso, se recibe en primera instancia una orden de trabajo generada por un cliente y recibida por el supervisor del holón recurso para ser verificada en el almacén, el cual entregara información sobre si la orden está en inventario o si existe un porcentaje de producción de esa orden. En caso de no tener la pieza completa en el almacén, se procede con la negociación de la meta, consultando la disponibilidad de todos los recursos que pertenecen a ella, preparando una propuesta según la disponibilidad de trabajo por cada holón recurso de la UPH, sin importar que tan poco óptima pudiera ser, para ser enviada al supervisor de la UPH. Luego, desde un nivel superior, se concede esa orden de trabajo a la mejor propuesta y entonces este recurso programa estas nuevas tareas en su agenda para después ejecutarlas. Cuando se finaliza esa tarea, se procede a una actividad de envío a otras entidades o de almacenamiento en el almacén.

El anterior escenario es ideal dado que se desarrolla en operación normal del sistema. A continuación se describe el caso contrario, cuando el sistema es afectado por una perturbación. En un primer caso, las tareas que se están ejecutando pueden estar tardando más de lo previsto, por tanto, el supervisor del holón recurso puede ordenar una actividad de cooperación con otros holones recurso que tengan la misma competencia para apoyar el holón recurso afectado por la perturbación y terminar en el tiempo justo con la orden. La otra opción consiste en empezar a retrasar el resto de operaciones de ese holón, en este punto el sistema de manufactura se ejecuta en modo de operación degradada, este estado se mantiene hasta que otros holones recurso puedan colaborar con el holón recurso bajo perturbación y se establezca otra vez una operación normal. Otro caso es cuando definitivamente un recurso entra en falla y no puede ejecutar ninguna tarea, por lo que el supervisor del recurso ordena una reprogramación de las tareas en otros recursos.

Un requisito primordial que debe ser manejado en este holón es el manejo de fallas y el mantenimiento, que permite tener reacción a eventos inesperados evitando la suspensión completa del proceso. Existe el mantenimiento programado y el correctivo. El primero es realizado periódicamente o ser implementado por un humano a una hora determinada, reservando un tiempo en la agenda de cada recurso para verificar a fondo la total funcionalidad del recurso, mientras que el correctivo, ocurre cuando un recurso entra en falla, por lo que se debe arreglar inmediatamente.

Una de las fortalezas del holón recurso se encuentra en su capacidad de operación. Cuando existe una falla o algunas actividades tienen mal desempeño, se puede iniciar un trabajo colaborativo entre holones y ayudar a soportar trabajos que algún recurso no puede, es decir, el trabajo que se libera de un recurso en particular debe ser asimilado por otros recursos que tengan la misma competencia.

3.2.2 Misión del Holón

La misión es un módulo de un holón recurso que puede describirse como el encargado de organizar y establecer adecuadamente la producción de la empresa, por medio del supervisor de la misión. Cada vez que entra una nueva orden de producción, se crea una nueva misión en el sistema

holónico, el cual tiene información sobre los plazos de ejecución de estas órdenes, el avance de la producción y la relación dada entre órdenes de producción y órdenes de trabajo.

En el proceso, inicialmente se verifica que las órdenes de producción requeridas no estén en recursos de almacenaje, es decir inventario, de ser así, se verifica si allí se tiene el 100% del pedido para ser entregado inmediatamente, o en otro caso producir el porcentaje del producto solicitado. Luego de comprobar la no existencia del producto, se solicita a la ingeniería la descomposición de las órdenes de producción en órdenes de trabajo para el enviarlas al holón recurso, para que ellos confirmen la existencia de materia prima y su disponibilidad para realizar estas tareas de fabricación. Cuando el holón recurso presenta las propuestas de ejecución de las órdenes de trabajo, el supervisor de la misión evalúa cual es la mejor opción y entonces reserva ese recurso y lo empieza a vigilar en su agenda para un buen manejo de toda la unidad de producción, teniendo en cuenta que a veces, ciertos procesos pueden ser cancelados o reprogramados. Aquí es donde el control y la visión que se tiene en la negociación con los holones recurso e ingeniería son fundamentales, por esto, el supervisor de la misión observa el comportamiento y avance de la producción, la cual puede estar en condiciones normales, de retraso, de imposibilidad o de finalización.

El supervisor de la misión toma las decisiones más complicadas porque debe analizar que sería lo más óptimo para la empresa. En algunos escenarios por ejemplo, el supervisor de la misión podría encontrar que ciertos recursos empiezan a atrasarse con sus tareas, luego tiene que empezar a analizar cómo podría solucionar la situación porque ya no estaría en las condiciones normales de operación, por lo que empieza a solicitar propuestas de holones recursos que contengan las mismas competencias y decidir si la UPH puede operar en condiciones subóptimas.

3.2.3 Ingeniería del Holón

La ingeniería es el elemento que tiene menor número de actividades a realizar pero que no dejan de ser importantes, las cuales están muy ligadas a los otros holones, pero en general, éste contiene la mayor parte de información de los productos fabricados en la unidad de producción, reglas, indicaciones y requisitos para que todo funcione adecuadamente.

Entre esta información se encuentra cuantas ordenes de trabajo representa cada producto, la materia prima que requiere, cómo y en qué orden se deben crear las piezas por medio del diagrama de precedencias, la forma de acoplar las mismas por medio del diagrama de ensamble, y por último las especificaciones de calidad con la que debe cumplir cada producto. Esta información es requerida por los otros holones en el transcurso del proceso y enviada adecuadamente a estos.

Adicionalmente, dentro de la arquitectura de control inteligente es posible identificar actores y roles que se consideran en este trabajo como los que más se adaptan a cualquier empresa de manufactura. Esto considerando que los múltiples *roles de proceso* interactúa o colaboran intercambiando *productos de trabajo* y activando la ejecución de determinadas actividades del sistema de manufactura, dado que el rol es un elemento central de modelado adecuado para representar

colaboraciones entre elementos dentro de la arquitectura de control. Además de describir las propiedades estructurales y de comportamiento de los objetos.

3.2.4 Actores y sus Roles

Negociador: Puede ser un módulo de software o un humano. Sus principales funciones son recibir una misión, hacer evaluar su factibilidad y decidir si aceptar o rechazar la misma. Se caracteriza por comunicación con el exterior y elementos internos, tener fuerza de cálculo, tener un protocolo de negociación con otros elementos de la UPH, enviar información al planificador y recibir las conclusiones de éste. Podría localizarse dentro del elemento de supervisión de la UPH.

Planificador: Es un módulo de software porque necesariamente utiliza herramientas matemáticas y de control para entregar conclusiones al negociador. Sus funciones más destacadas son evaluar la factibilidad de una posible misión y entregar esa factibilidad al negociador. La evaluación se hace mediante un método de producción ya conocido, tomando en cuenta la capacidad y disponibilidad de recursos en ese momento. Si la misión es aceptada, reserva los recursos implicados para llevar a cabo esa misión. Podría localizarse dentro del elemento de supervisión de la UPH.

Programador: Puede ser un módulo de software o un humano. A partir del plan de producción, el programador asigna los recursos para cada una de las actividades en un instante de tiempo, es decir, establece el cronograma de las tareas a ejecutar en la agenda. Podría localizarse dentro del elemento de supervisión de la UPH.

Supervisor de la misión: Puede ser un módulo de software o un humano. Se encarga principalmente de verificar la misión, tomando decisiones respecto al proceso real (existencia de anomalías), ajusta la ejecución de las actividades, parametriza los controladores, actualiza la imagen del proceso y compone el vector de estados a partir de la información del supervisor del recurso.

Supervisor del recurso: Puede ser un módulo de software o un humano. En general, sigue los procesos particulares realizados en cada recurso ya que conoce todas las propiedades y restricciones operativas del recurso, envía información al supervisor de la misión y se responsabiliza del monitoreo de la condición del recurso. Es la parte donde se encuentra la inteligencia de todos los recursos, el cual puede contener otros actores como negociador, planificador y programador.

Planta: Es el conjunto de dispositivos físicos donde se desarrolla el proceso y el cual es monitoreado por el resto de actores. Es considerado el proceso real.

Recurso: El recurso puede ser o no inteligente, si posee inteligencia es considerado una UPH dentro de otra UPH y en ese caso tiene todos los actores descritos anteriormente. Por lo tanto es capaz de comunicarse, de negociar, ejecutar tareas y en general, cumplir con todas las propiedades de un sistema holónico. Aquellos recursos que no son inteligentes están contenidos dentro de uno que si posea esa inteligencia.

Operario humano: Este rol se refiere a las personas que tendrán una interacción directa con el sistema informático y que se encargara de supervisar, configurar y colaborar en la toma de decisiones del sistema de la unidad de producción.

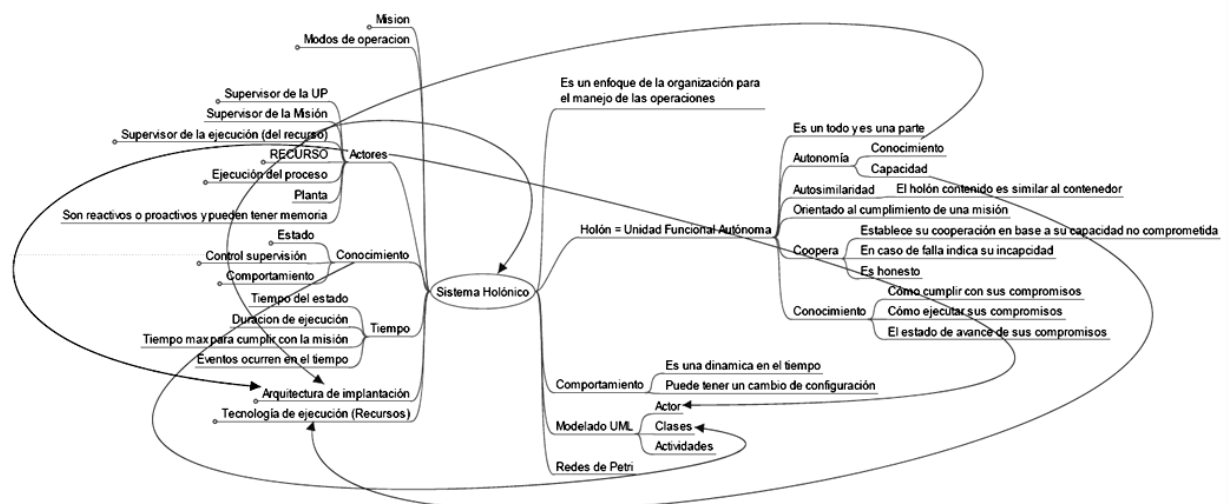
3.3 Marco Ontológico de la Arquitectura de Control Inteligente y Módulos Misión e Ingeniería del Holón

Esta sección presenta una agrupación de conceptos alrededor del control inteligente de sistemas de manufactura, con el objetivo de apoyar la definición de las características de estas arquitecturas presentadas en la sección anterior. Adicionalmente, se presenta una conceptualización alrededor del módulo Misión e Ingeniería de holón.

3.3.1 Ontología de Control Inteligente

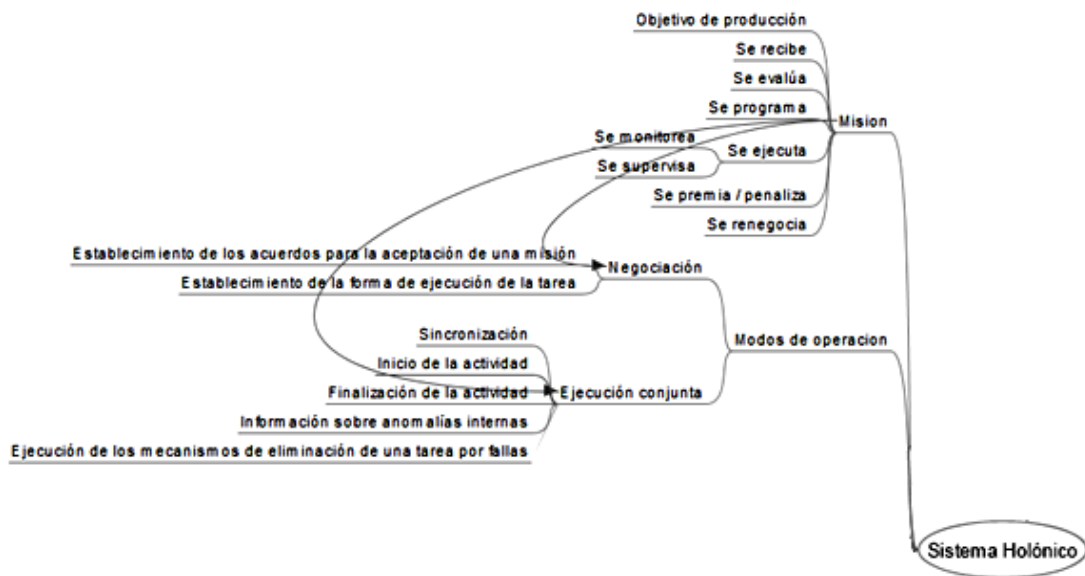
La construcción de la ontología permite identificar todos los elementos involucrados en el desarrollo de la arquitectura de control inteligente de manufactura. En la Figura 10, se puede observar en el lado izquierdo todos los elementos encontrados en los sistemas holónicos de producción y que en su mayoría son tangibles y asociables a módulos en una empresa de manufactura. Algunas relaciones se destacan tales como la adaptación de los actores en una arquitectura de implantación y el concepto de holón recurso como un sistema holónicos a su vez. Por otra parte, en el lado derecho se constituye el significado de estos sistemas con sus correspondientes características, así mismo se visualiza las herramientas que permiten definirlo como UML y otras que utiliza en si el paradigma, como las redes de Petri.

Figura 10. Ontología de un sistema holónico



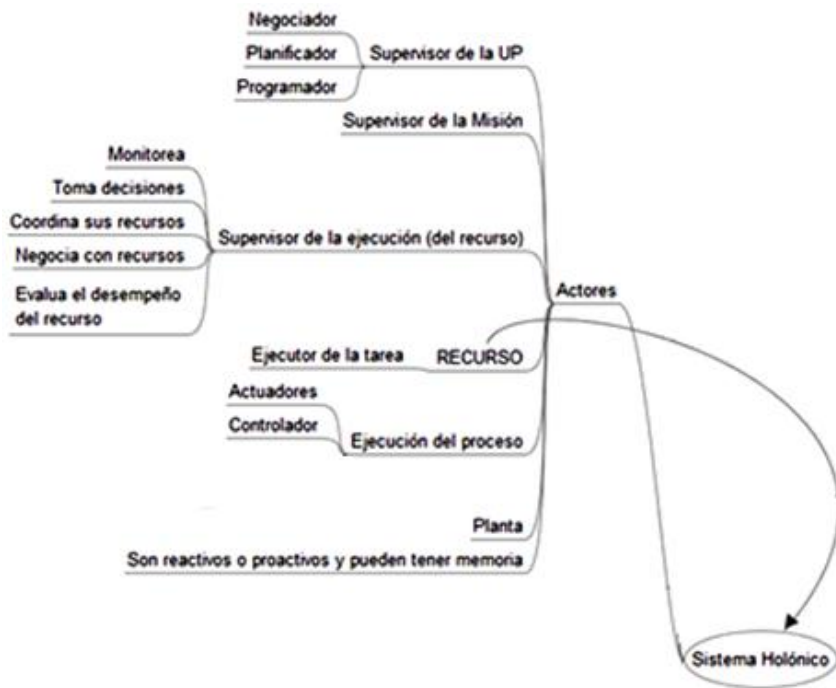
Los elementos de la ontología pueden resumirse en la misión de producción que debe cumplir la fábrica, el modo de ejecución, los actores implicados, el conocimiento que debe manejar la unidad de producción en cualquier aspecto, el manejo del tiempo, la arquitectura de implantación y la tecnología que utiliza la UPH. A continuación se ilustran los despliegues de estos elementos. En primera instancia, la misión puede describirse como el objetivo de producción de la UPH, el cual se recibe, se evalúa y se programa, y luego de esto se ejecuta y se monitorea. El modo de operación se refiere a la forma de operar las cuales pueden ser negociación, cooperación, ejecución conjunta, y resolución de fallas por medio de mantenimiento entre otras. La Figura 11 enseña cómo se establece la forma de ejecución y los mecanismos ágiles de readaptación en caso de existir anomalías.

Figura 11. Despliegue Ontología parte A



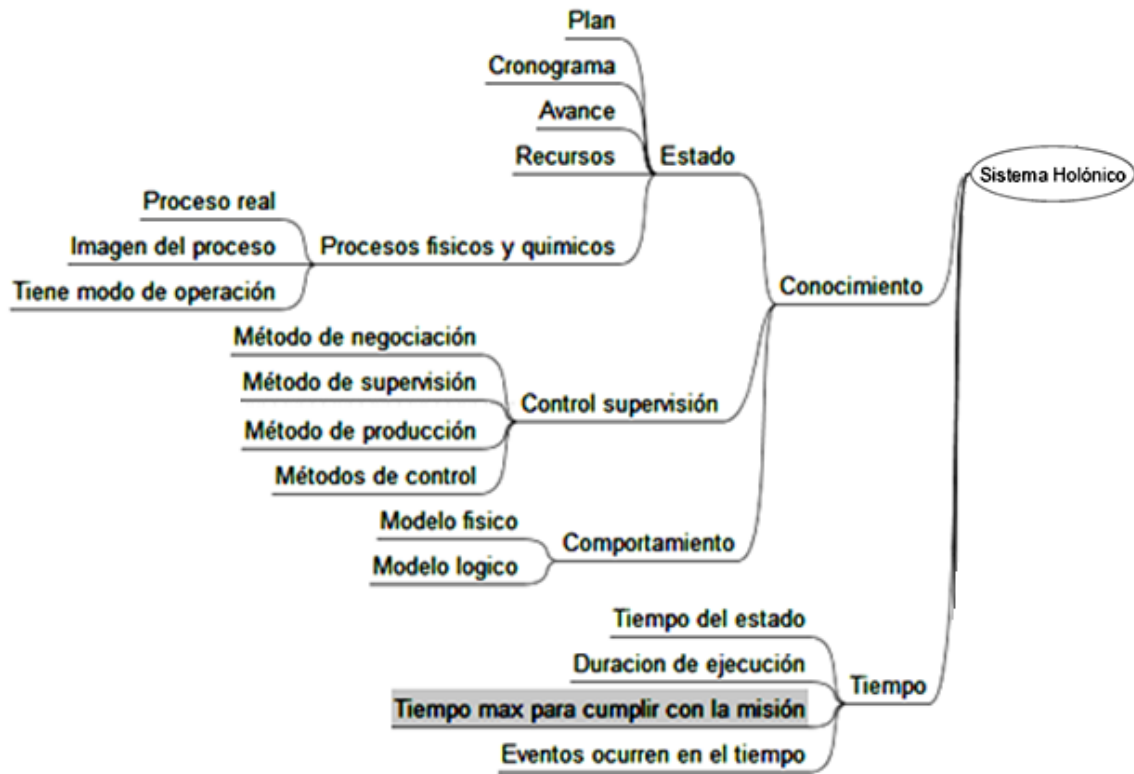
En la Figura 12 se presentan los actores principales implicados en los sistemas holónicos. Estos son los que se quieren plasmar en el modelos control inteligente de manufactura holónico, más no corresponde totalmente con los actores que actualmente es posible encontrar en los sistemas de producción real.

Figura 12. Despliegue Ontología parte B



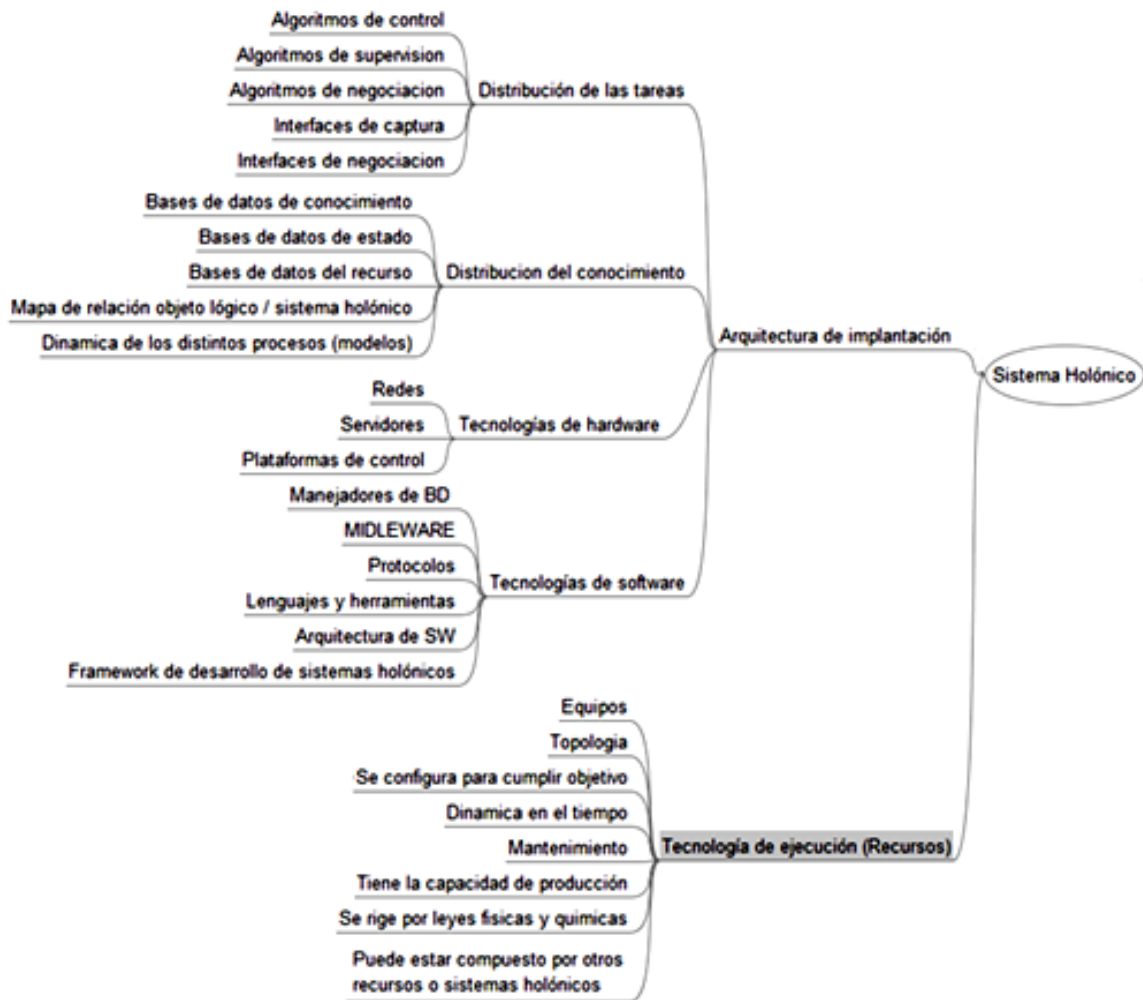
Otro elemento fundamental en una arquitectura de control es el conocimiento, en donde se encuentra la forma de controlar todos los elementos y con esta información dotar a los supervisores, además de tener las nociones de comportamiento tanto a nivel físico como lógico, esto finalmente desencadena en un elemento llamado estado, el cual indica los planes y cronogramas que la UPH tiene en tiempo presente., y este a su vez, permite medir la misión de la UPH. La Figura 13 muestra estos elementos.

Figura 13. Despliegue Ontología parte C



Por último, el sistema holónico debe contener una arquitectura que permita un acercamiento a todas las descripciones hechas hasta el momento para la comunicación de los elementos tangibles en la fábrica. Esta arquitectura, Figura 14, está dividida en una distribución de tareas, distribución del conocimiento, tecnologías de hardware y de software. Aquí también es reconocida la tecnología propia de las máquinas y recursos que cada empresa en particular implementa, lo que indica que se debe estar abierto a muchas tecnologías para ser más flexibles, teniendo en cuenta aspectos como topologías, leyes físicas y químicas que manejan y dinámicas en el tiempo entre muchas más.

Figura 14. Despliegue Ontología parte D



3.3.2 Ontología del Módulo Misión

La misión del holón logra el cumplimiento de la misión representada en un producto u orden de producción a ser manufacturada, esta es supervisada mediante la identificación de estados que representan el conocimiento en el progreso de producción de la orden en el interior de la UPH. La figura 12 representa los elementos que se considera describen los elementos que deben estar presentes en el módulo de misión de cualquier holón.

Figura 15. Agrupación de conceptos sobre el módulo Misión de una UPH



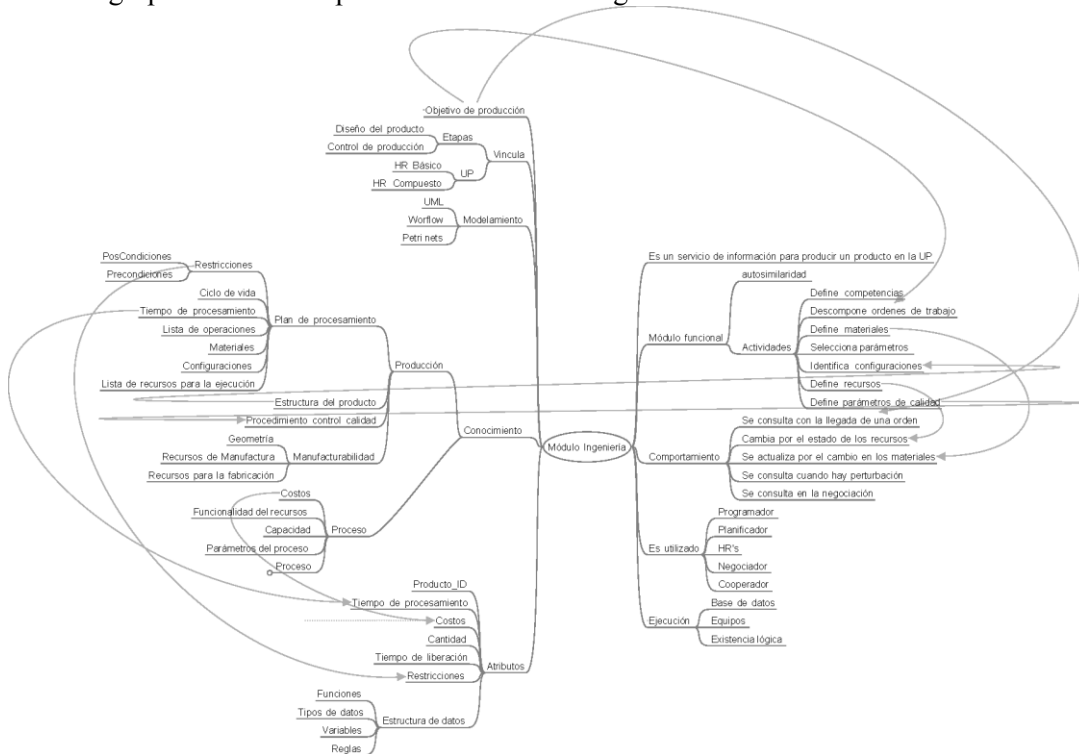
3.3.3 Ontología del Módulo Ingeniería

El módulo de ingeniería se realiza mediante la identificación de todos los Productos y Sub-Productos manufacturados en la fábrica. Para una caracterización completa de un producto es necesaria la descripción de la estructura del producto (es decir, relaciones con otros subproductos) y la descripción del plan de procesamiento (Process plan).

Este módulo sirve a los demás holones en el sistema dándoles información sobre las funcionalidades y un uso eficiente de recursos. Esto implica un plan de procesamiento (plan process) que especifica una velocidad óptima de corte, prever operaciones alternativas que pueden ser utilizadas en caso de un recurso no funcione correctamente y minimizar la necesidad de asistencia humana durante el procesamiento.

La manufacturabilidad de un producto es la probabilidad de que pueda ser producido acorde con un conjunto disponible de maquinaria, herramientas y procesos. La optimización de la manufacturabilidad no incluye la generación de planes de procesos detallados o estimación de costos. De hecho, es una fase inmediatamente anterior al proceso de planeación. Esta aproximación es una novedosa forma de optimizar la calidad del diseño del producto antes de ser enviado al sistema de planeación de los procesos, evitando el desperdicio de recursos involucrados en la planeación de procesos de diseños que no son manufacturables. La figura 14 agrupa los demás conceptos restantes alrededor de la concepción de lo que representa el módulo ingeniería dentro de la dinámica de control de una UPH.

Figura 16. Agrupación de conceptos sobre el módulo Ingeniería de una UPH



3.4 Supervisión y Comportamiento Dinámico

Con objeto de modelar la estructura y el comportamiento de un sistema de control de manufactura que permita facilitar el entendimiento de los elementos de la arquitectura, además de tener un soporte de una metodología formal de modelamiento. Los modelos Las redes de Petri a diferencia de los modelos UML son una herramienta que permite el modelamiento del comportamiento dinámico y la validación formal de sus especificaciones definidas.

3.4.1 Modelos en redes de Petri del holón recurso

Los holones recurso de una UPH son quienes en últimas son los responsables de llevar a cabo la tarea del cumplimiento de la misión del sistema de manufactura, usando una infraestructura física, materiales, y servicios que deben estar disponibles con objeto de completar las actividades de producción que desempeña la UPH. Complementariamente, un Holón Recurso es una abstracción de medios de producción como una fábrica, una celda de manufactura, las máquinas, las fresas, los CNC, los operarios, hasta llegar a holones recurso más básico que dependen de la toma de decisiones de holones recurso superiores (Torrealba, 2005). En general, un holón recurso contiene una parte física, denominada recursos de producción del proceso y una parte del procesamiento de información que controla los recursos.

El criterio para catalogar un recurso como un holón es considerando su estructura interna, es decir evaluando las siguientes capacidades: programación de trabajo (metas), método de trabajo (ingeniería), supervisión y monitoreo, manipulación y/o transformación de información. Los recursos que no cumplan con algunas de estas especificaciones pueden ser considerados como dispositivos o equipos de la infraestructura del proceso, que a su vez dependen de la inteligencia de los holones recurso con los que se relacionan de acuerdo al proceso de manufactura.

La red de Petri que modela los aspectos de comportamiento del holón recurso se presenta en la Figura 17 y las

Tabla 1 y Tabla 2 que describen la nomenclatura utilizada. Este modelo especifica los distintos estados discretos por los que debe pasar y las precondiciones y poscondiciones de cada uno de ellos. Concretamente, la figura 19 modela los estados básicos que se consideran como necesarios para representar la dinámica del holón recurso en operación normal y bajo perturbación durante el ciclo de vida operacional, este modelo representa el punto de partida para la representación de sistemas de control más complejos.

En general, las dinámicas del HR son:

- Dinámicas relativas a los HR básicos: disponible, no disponible, en mantenimiento, reparado, modo degradado.
- Dinámicas relativas a los Recursos Compuestos: disponible, no disponible, en falla, en mantenimiento programado, en mantenimiento, liberación, asignación, reservado, disponible degradado (Zapata, 2008B).

Tabla 2. Eventos del holón recurso

Evento	Descripción
σ_r	Reservar recurso
σ_d	Establecer recurso en disponible
σ_{ss}	Arrancar el recurso
σ_0	Inicio
σ_n	Establecer recurso en operación normal
σ_δ	Establecer recurso en operación degrada
$\sigma_{d\delta}$	Establecer recurso en operación disponible degradada
σ_f	Establecer recurso en modo de falla
σ_f	Realizar mantenimiento por perturbación
$\sigma_{\mu p}$	Realizar mantenimiento programado
$\sigma_{f\mu p}$	Finalizar mantenimiento programado
σ_s	Arrancar el recurso
$\sigma_p + \sigma_{et}$	Parada o secuencia terminada

En un holón recurso de una UPH, la región de comportamiento u operación normal es aquella en la que el sistema debe operar cumpliendo su función. Para esto, el modelo permite establecer si el recurso está disponible, no disponible (por falla o mantenimiento) o reservado para una operación. Cuando se lanza la producción, el recurso pasa a un estado de operación. El recurso se libera cuando termina su participación en la orden de producción para la cual fue requerido.

Si el desempeño del sistema se desvía de sus criterios de desempeño especificados, el sistema se clasifica como exhibiendo un comportamiento “degradado”. Este comportamiento es causado por variaciones paramétricas durante el ciclo de vida operacional.

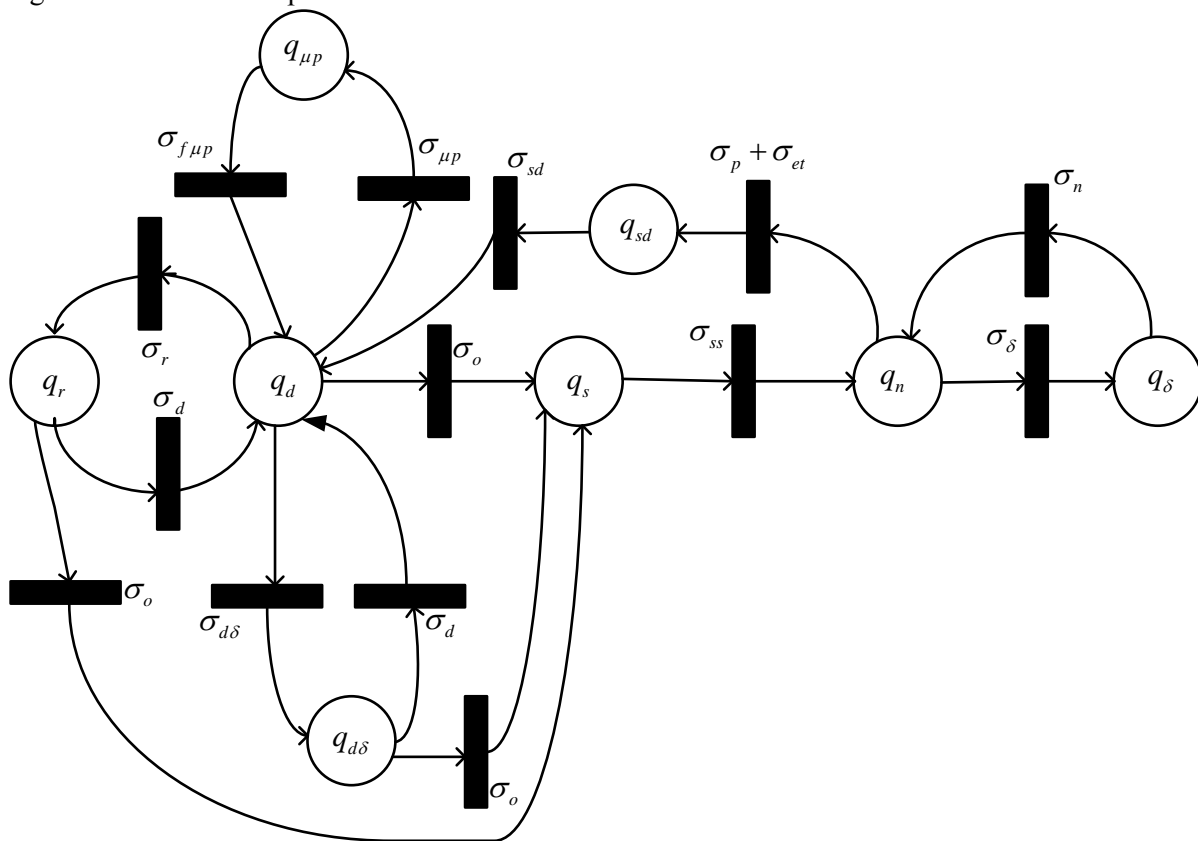
La región de comportamiento degradado corresponde a una región de funcionamiento a la que el sistema se desplaza después de la aparición de un fallo no lo suficientemente grave para el sistema. En esta situación y en caso que la situación permanezca, el holón recurso reacciona y frente al mismo deberá activar las acciones de recuperación preestablecidas de cara a evitar una mayor degradación que desplace al sistema hacia la región de comportamiento inadmisibles e incluso de peligro, considerando que en este estado el HR es más vulnerable a una falla, (Zapata, 2008B).

Existen diferentes factores en la operación del proceso que pueden llevar a un funcionamiento degradado del mismo. El más grave es que, como resultado, se produzca la rotura de la herramienta de corte en un CNC. Pero otros factores, sin llegar a ese extremo, pueden llevar a situaciones de operación degradada en el HR (como, por ejemplo, desgaste progresivo de la herramienta, cambio del material de la pieza, entre otras). El estado de operación degradado involucra que el holón recurso ponga en riesgo el cumplimiento de la misión y la calidad de los productos terminados.

La reserva del holón recurso se realiza desde su módulo de inteligencia, que se representará mediante el mecanismo de autonomía y que se basa en la agenda del recurso. Esta reserva del HR consiste en evitar que la agenda del recurso sea comprometida en otros procesos de negociación que se lleven a cabo en la UPH. Conjuntamente, la reserva puede consistir en la definición de un recurso respalda los recursos de la UPH ante una perturbación en el sistema.

El estado del HR de disponible degradado se define para los HR compuestos, esto es, HR que se componen a partir de otros HR de forma recursiva. Si uno de los HR de una UPH está en falla y sus recursos están en estado disponible, se dice que el estado de la UPH es disponible degradado, esto se da considerando que la UPH no se encuentra disponible con su capacidad nominal. Estos estados de operación son modelados en la Figura 18.

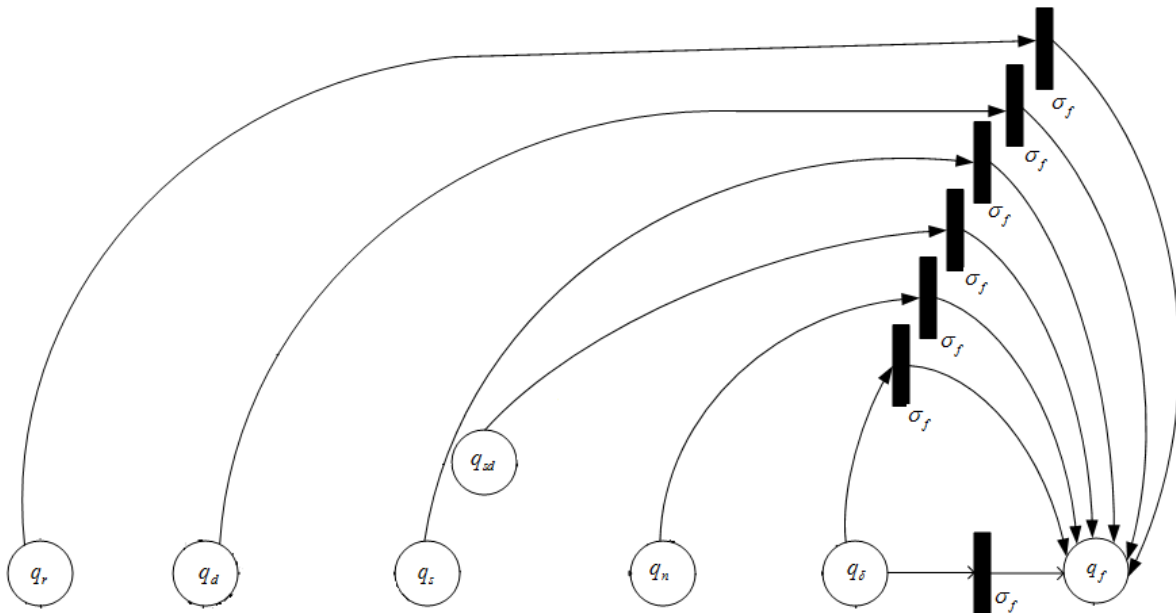
Figura 18. Estados de Operación del Holón recurso



La consideración del estado de falla en el sistema de control de un sistema de manufactura permite al HR reaccionar con rapidez a los disturbios. La falla del recurso causa su indisponibilidad para ejecutar y negociar la producción órdenes de trabajo, además de recortar la capacidad productiva de la UPH. El tiempo de restablecimiento del recurso depende de la valoración que realice el módulo mantenimiento y de su disponibilidad para llevar a cabo la operación de mantenimiento. Mientras esto sucede el mecanismo de negociación realiza la reasignación de los trabajos que se encontraban programados en la agenda del recurso. El modelamiento del estado de falla se realiza en la Figura

19. En este sentido, los HR tendrá la capacidad de negociar autónomamente con otros HR disponibles en la UPH, describiendo un comportamiento holárquica del sistema de control de manufactura.

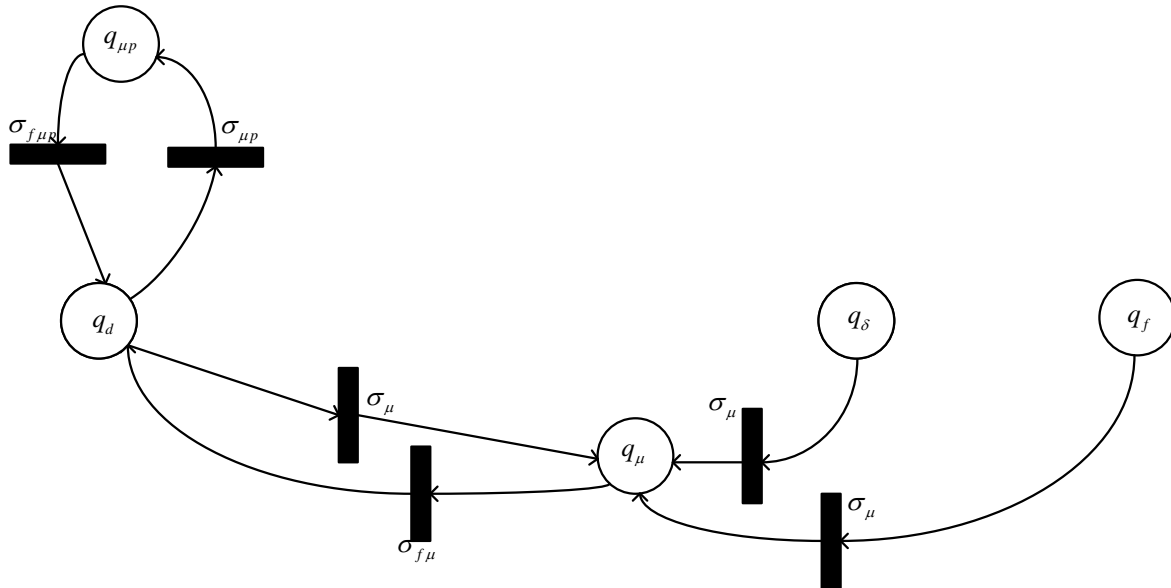
Figura 19. Estado de falla del Holón Recurso



A partir de la presencia de una falla o un estado degradado en el holón recurso se toma la decisión de enfrentar las situaciones de falla llevando al recurso a un estado de mantenimiento mediante un evento controlable del sistema. En el mantenimiento se implementan las políticas de diagnóstico y mantenimiento que para el recurso se encuentran definidas en el módulo de ingeniería de la UPH.

El mantenimiento preventivo permite minimizar la ocurrencia de fallas y de esa manera evitar la necesidad de implementar rutinas de mantenimiento correctivo, que implica parar los recursos de la UPH y discutir su capacidad nominal de producción.

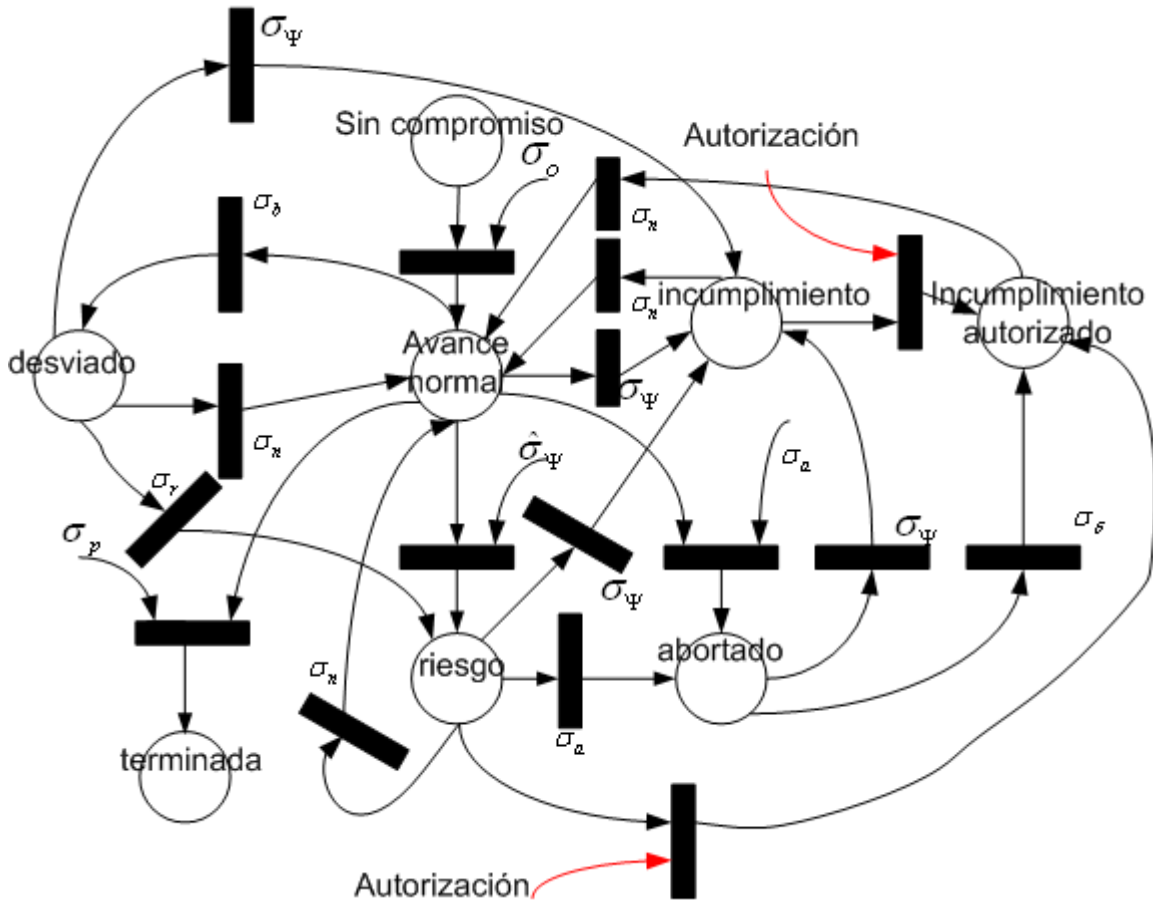
Figura 20. Estados de Mantenimiento del holón recurso



3.4.2 Modelos en redes de Petri de la misión del recurso

La misión de unidad de producción holónica en un sistema de control inteligente de manufactura es obtener n ítems de un producto en un intervalo esperado de tiempo. En la UPH se tiene una misión o producto por cada orden de trabajo, la cual puede estar en cualquier de los siguientes estados que son definidos a partir del progreso de la producción de una orden de trabajo: sin compromiso, incumplimiento, avance normal, desviado, en riesgo y abortado. El modelo que representa el comportamiento dinámico del módulo misión se ilustra en la Figura 21, este modelo puede ser invocado recursivamente, implicando que una orden de producción pueda ser descompuesta en otras órdenes de producción con subproductos claramente identificables.

Figura 21. Red de Petri de la misión del recurso



Los estados de la PN de la Misión del holón permiten monitorear y supervisar la ejecución de los planes, dado que la supervisión es esencial para responder rápidamente a las perturbaciones que se presenten y que pongan en riesgo el cumplimiento del objetivo. La supervisión incluye la detección de eventos no controlables, requeridos para tomar acciones por parte del operador, tales como la detección de desviaciones (Zapata et al, 2008D), que lleven a situaciones de riesgo y por tanto a un posible incumplimiento o la necesidad de abortar una misión. Dada una situación de incumplimiento es posible permitir un estado de incumplimiento autorizado, definidas por los costos de cambio de recurso para la ejecución de la orden o simplemente por preacuerdo con el cliente para entregar la orden en tiempo posterior. Los eventos del modelo de la Misión son definidos en la Tabla 3.

Tabla 3. Eventos de la Misión del holón

Evento	Descripción
σ_ψ	Misión desviada
$\hat{\sigma}_\psi$	Estimación de desvío de la misión
σ_θ	Inicio de la misión
σ_p	Fin de la misión
σ_n	Misión en curso normal
σ_a	Misión abortada
σ_b	Desajuste (Breakdown)
σ_σ	Desvío autorizado de la misión
σ_r	Misión en riesgo

3.5 Autonomía y Negociación de la UPH

De acuerdo a Montilva (Montilva et al, 2008) cada holón es autónomo en sus decisiones y coopera con otros para alcanzar los objetivos del sistema del cual forma parte. La presente sección contiene una caracterización de la autonomía de las unidades de producción holónica y el mecanismo de negociación como componente de cooperación propuesto en el control inteligente de sistemas de manufactura.

3.5.1 Autonomía de la UPH

De acuerdo a (Chacón et al, 2004) las unidades de producción son consideradas autónomas, ya que son capaces de tomar sus propias decisiones por medio de sus dispositivos de control local, con el fin de mantener el sistema seguro y de trabajo de acuerdo a las especificaciones recibidas del supervisor. Adicionalmente, se presenta que la implementación de la capa de decisión e información de una unidad de producción holónica (UPH) se puede lograr mediante el uso de agentes inteligentes de software, que son capaces de controlar el sistema interno, tomando decisiones sobre la mejor manera de asegurar las metas de producción y el intercambio de información con el entorno donde la UPH está localizada.

El concepto de autonomía ha sido definido desde los conceptos de la teoría de control, los sistemas Multi-Agente, holones y unidades de producción. En (Johansson et al, 2004) por autonomía se indica que cada módulo del sistema tiene su propia funcionalidad y que es independiente de otros módulos. Para (Antsaklis, 1994) la autonomía no es parte del diseño del sistema de control, es una propiedad que emerge del comportamiento estratégico de los sistemas de control inteligente, que proporciona a cada elemento la capacidad de maximizar la posibilidad de presentar y ejecutar sus

habilidades. Por tanto un sistema de control está diseñado de manera que pueda lograr autónomamente sus metas, entre tanto que sus componentes, objetivos de control, modelo de la planta y leyes de control no están completamente definidas, ya sea porque no son conocidos en el tiempo de diseño o porque cambian inesperadamente. Esto posibilita al sistema adaptarse a la incertidumbre y a la operación durante períodos largos de tiempo sin intervención humana. En (Kotak et al, 2003) la autonomía es definida mediante agentes autónomos que se comportan autónomamente con respecto a sus acciones y percepción de su dominio, posibilitando su operación en entornos caracterizados por la incertidumbre e imprevisibilidad sin la directa intervención de humanos u otros agentes.

Una definición más clásica de autonomía es encontrada en (Colombo, 2006), (Luder, 2005), (Wyns, 1999), (Leitao, 2004), definiéndola como la capacidad de una entidad de crear, controlar y monitorear (supervisar) la ejecución de sus propios planes y/o estrategias de comportamiento. En (Villa, 2002) la autonomía representa el grado o la manera de cuantificar la inteligencia dada al holón.

Ante la necesidad de representar el comportamiento de las dinámicas que definen los elementos de autonomía y que fueron expuestos anteriormente, las Redes de Petri (PN) (Silva, 1985), (Kurt, 1990) constituyen una metodología formal de modelamiento que permite la especificación y análisis (Murata, 1989) de sistemas a eventos discretos (Gharieb, 2004), necesaria para definir de manera formal el comportamiento dinámico de un sistema de manufactura (Zhou and DiCesare, 1990). Las PN son un sistema experto (Zhang, 2001) basado en modelo que permite representar el comportamiento asíncrono y concurrente propio de estos sistemas.

Agenda como Motor de Autonomía del Holón Recurso

El HR permite el modelamiento del comportamiento y de las dinámicas existentes en los elementos que realizan una parte del proceso de producción. En la arquitectura holónica basada en unidades de producción se tiene un HR por cada recurso existente en el sistema de manufactura. El HR tiene el conocimiento y los procedimientos para programar, analizar, supervisar y monitorear órdenes de trabajo. La base de estas funcionalidades son la agenda o programa del HR.

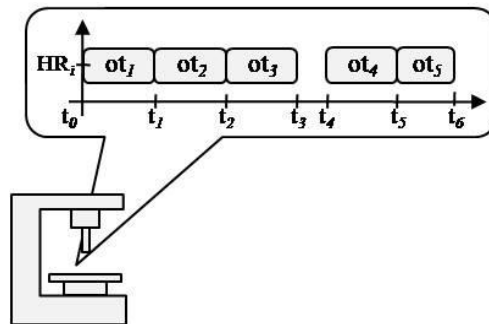
La *Agenda* representa la secuencia de órdenes de trabajo programadas por el HR en función del tiempo. Matellán Y Borrajo definen en (Matellán and Borrajo, 2001) la agenda como el programa predefinido que cada agente compone de manera oportunística para lograr un comportamiento inteligente. La agenda bajo esta consideración es utilizada como lista de pendientes, que llegan a requerir o no acciones predefinidas sobre el sistema de manufactura. En (Quintero, 2008) la agenda debe contener las actividades en ejecución y es encargada de la reprogramación dentro de la ventana de tiempo comprendida por la agenda.

Formalmente la Agenda se representa a través de un secuencia finita ordenada de n órdenes de trabajo, $Agenda = \{OT_1, \dots, OT_n\}$, donde n es el número de órdenes de trabajo (OT) asignadas al HR. Además cada OT está dada por $OT = (TI, TF, TE, PRE, C, Op, REF)$, donde TI y TF son el tiempo

planeado de inicio y finalización, TE, PRE y C en orden contienen el tiempo definido de entrega, las restricciones de proceso y la cantidad de ítems a elaborar. Op y REF definen la operación requerida por la orden de trabajo y la referencia de la orden de producción respectivamente.

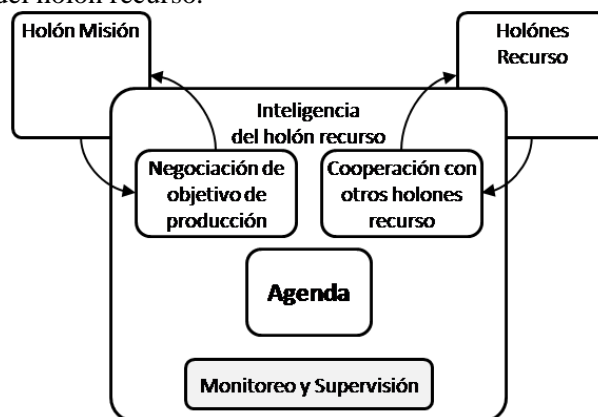
La herramienta utilizada para la representación de la agenda es el diagrama de Gantt, la cual en su eje horizontal representa el horizonte de tiempo ocupado o tráfico de órdenes de trabajo en el recurso. El eje vertical muestra el centro de trabajo o recurso que está siendo programado. Una muestra del diagrama de Gantt que representa la agenda de un recurso se presenta en la Figura 22.

Figura 22. Agenda del holón recurso.



La agenda en su interior tiene un registro de capacidad, historial y de trazabilidad del HR. Estas funcionalidades le proporcionan al recurso la capacidad de tener un comportamiento reactivo y de entrar en negociación o cooperación con otros holones en caso de fallas, perturbaciones por ingreso de nuevas órdenes o identificación de operación degradada. Por tanto, la agenda es el centro de la inteligencia del holón, considerando que con base en ésta el holón podrá determinar en cualquier momento qué operación está realizando, qué operación es sucesora a la actual, cuál es el estado de ejecución de la tarea o evaluar el grado de adelanto o atraso con respecto al plazo de entrega. La agenda provee al holón la capacidad de definir un nivel de negociación y cooperación con otros holones, permitiendo que la negociación, la cooperación, la supervisión del proceso y la misma agenda, se ajusten en una estructura lógica para el HR, que le permita el manejo de información del proceso. Estas unidades se muestran en la Figura 23.

Figura 23. Inteligencia del holón recurso.



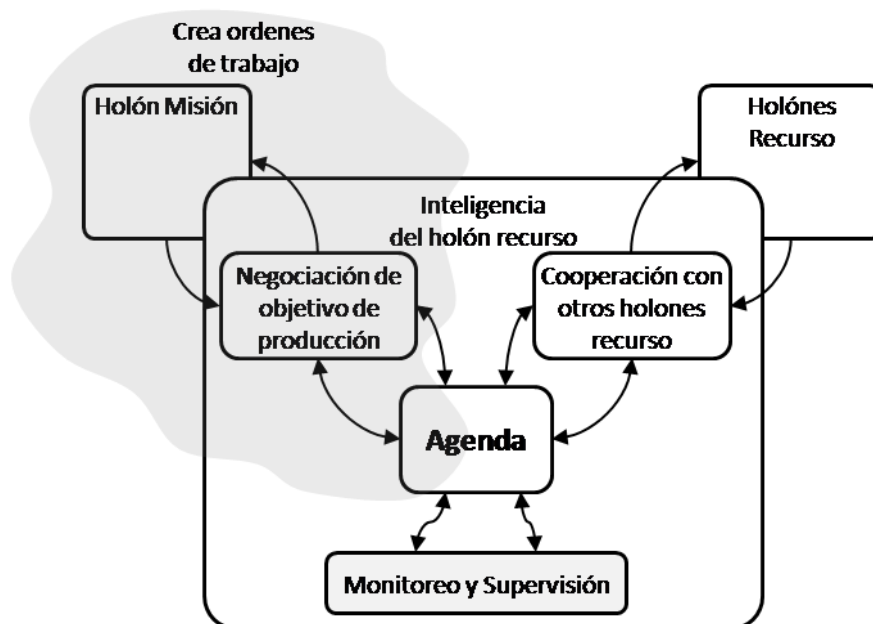
La capacidad del HR de tomar decisiones de acuerdo a la evaluación local de su estado con base en la agenda de producción, le confiere el carácter de unidad autónoma.

En (Zhang and Lesser, 2002) la agenda es utilizada como método de negociación e índice clave de flexibilidad, que le permite al agente razonar en las interacciones de negociación con otros agentes. De igual manera en (Park et al, 2003) la agenda es la base de la negociación para la cooperación con otros agentes, donde el agente interactúa considerando la máxima fecha de entrega asignada a la orden de trabajo.

Caracterización de los Elementos de Autonomía del Holón Recurso

Como se estableció en (Colombo, 2006), (Luder, 2005), (Wyns, 1999), (Leitao, 2004), los elementos que definen la autonomía en un sistema de manufactura son: la capacidad para crear, controlar y monitorear sus planes de trabajo. De acuerdo con esos elementos, en este trabajo se propone la creación de la agenda como espacio en que el holón crea sus estrategias y planes de trabajo (los negocia con el holón misión), y contiene la información para el seguimiento del objetivo de producción, su monitoreo y control. La Figura 24 ilustra la relación de los componentes agenda, negociación, cooperación y monitoreo.

Figura 24. Relación componentes lógicos de la inteligencia del holón recurso.

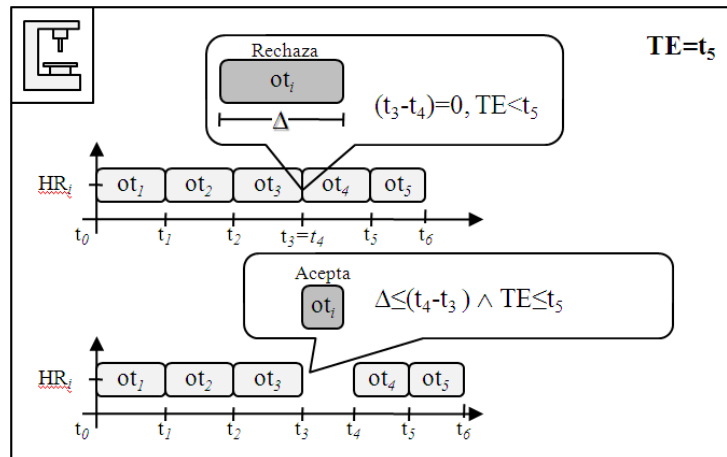


El comportamiento autónomo del HR definido por el concepto de autonomía antes mencionado, se compone específicamente de: evaluación de solicitudes de trabajo, régimen de operación normal, manejo de disturbios (reactividad), programación reactiva y seguimiento de avances de órdenes de trabajo los cuales se presentan a continuación.

Evaluación de Solicitudes

Cuando llega una orden de producción a la UPH se genera un módulo misión (HM) que la descompone en órdenes de producción. Estas órdenes se convierten en propuestas para los holones recurso que tienen competencias para ejecutarlas. El HR de manera autónoma realiza una evaluación con base en la agenda para determinar su capacidad de producir la orden propuesta, en la que se considera conjuntamente el tiempo de entrega (TE) definido para la orden de trabajo. Esta evaluación se ilustra en la Figura 25.

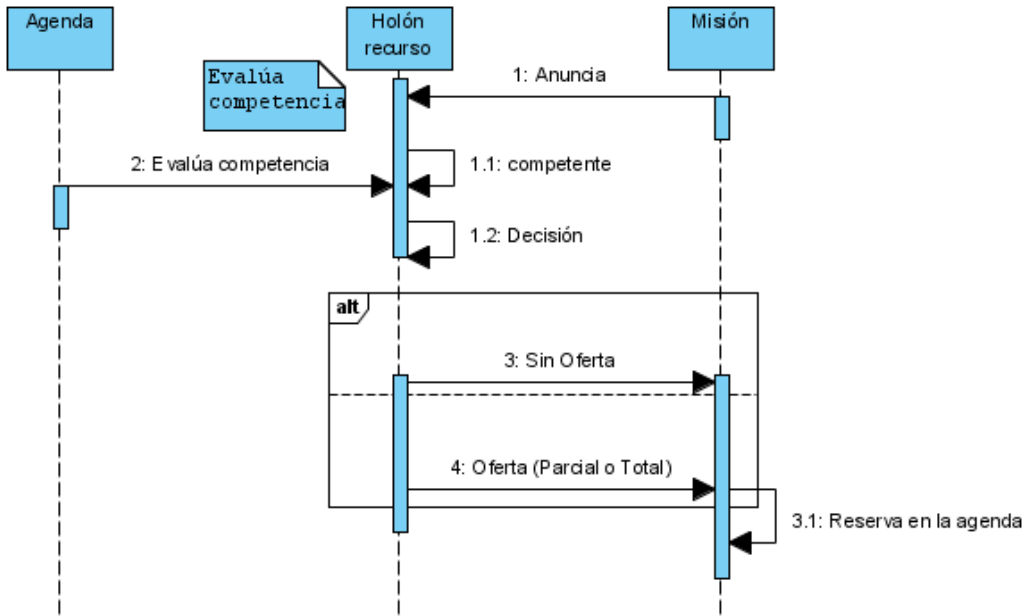
Figura 25. Evaluación de solicitud de manufactura de órdenes de trabajo.



En el caso de que el HR acepte producir la orden de trabajo o proponga una alternativa de producir una fracción de esta, se genera una reserva con el mismo formato de la orden, perteneciente al resultado de esa evaluación, es decir, el HR memoriza el resultado para tener capacidad de respuesta ante la posibilidad de que el HM acepte producir la orden de trabajo en ese HR.

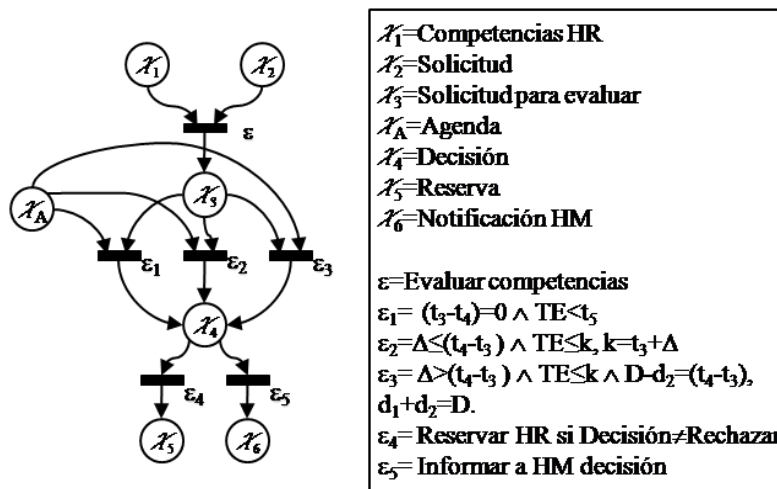
El diagrama de interacción para la evaluación de solicitud de forma local en el holón recurso, que tiene las competencias para manufacturar la orden de trabajo que llega como solicitud, se muestra en la Figura 26.

Figura 26. Diagrama de interacción para evaluación de solicitudes de órdenes de trabajo.



La Figura 27 representa el comportamiento mediante Redes de Petri del HR para la evaluación de una solicitud.

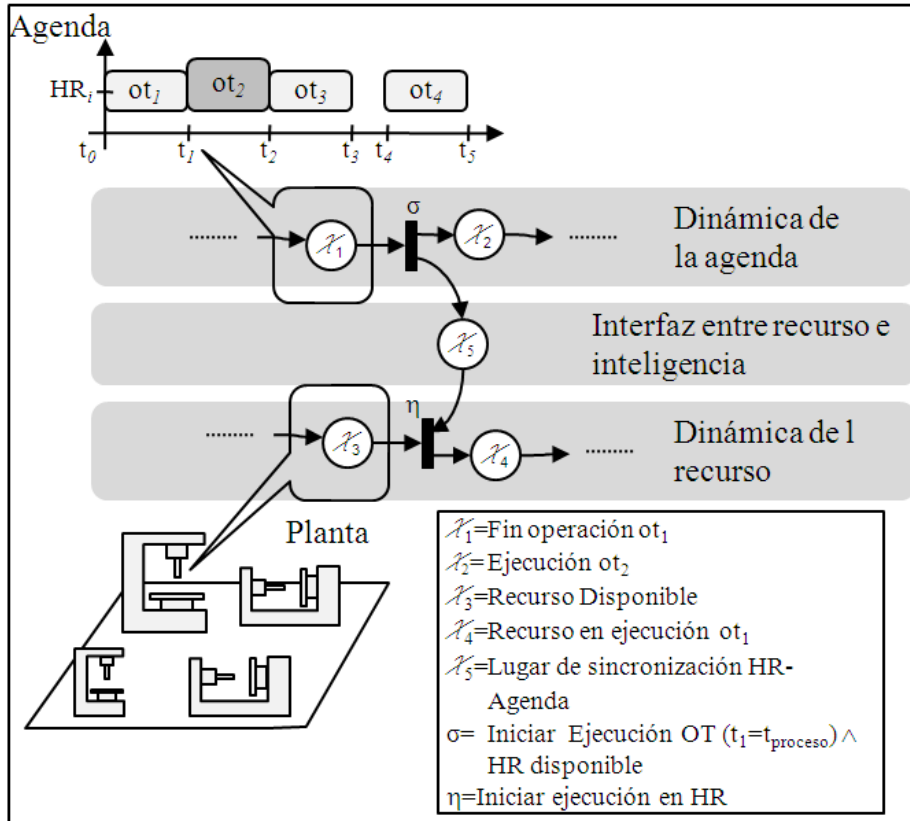
Figura 27. Representación mediante Redes de Petri de las dinámicas para evaluación de solicitud.



Régimen de Operación Normal

Considerando que las órdenes de trabajo constituyen la agenda del HR y que tiene asignados tiempos de inicio de ejecución, y que el supervisor conoce el estado real de disponibilidad del HR, son estos dos los que definen de manera conjunta el momento en que debe iniciarse la ejecución por parte del HR. La Figura 28 ilustra el régimen de operación normal.

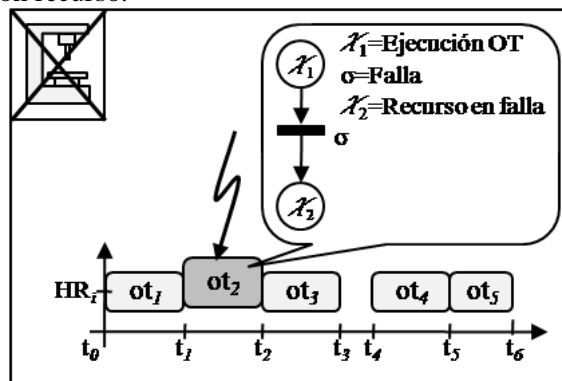
Figura 28. Inicio de ejecución en régimen de operación normal.



Manejo de Disturbios

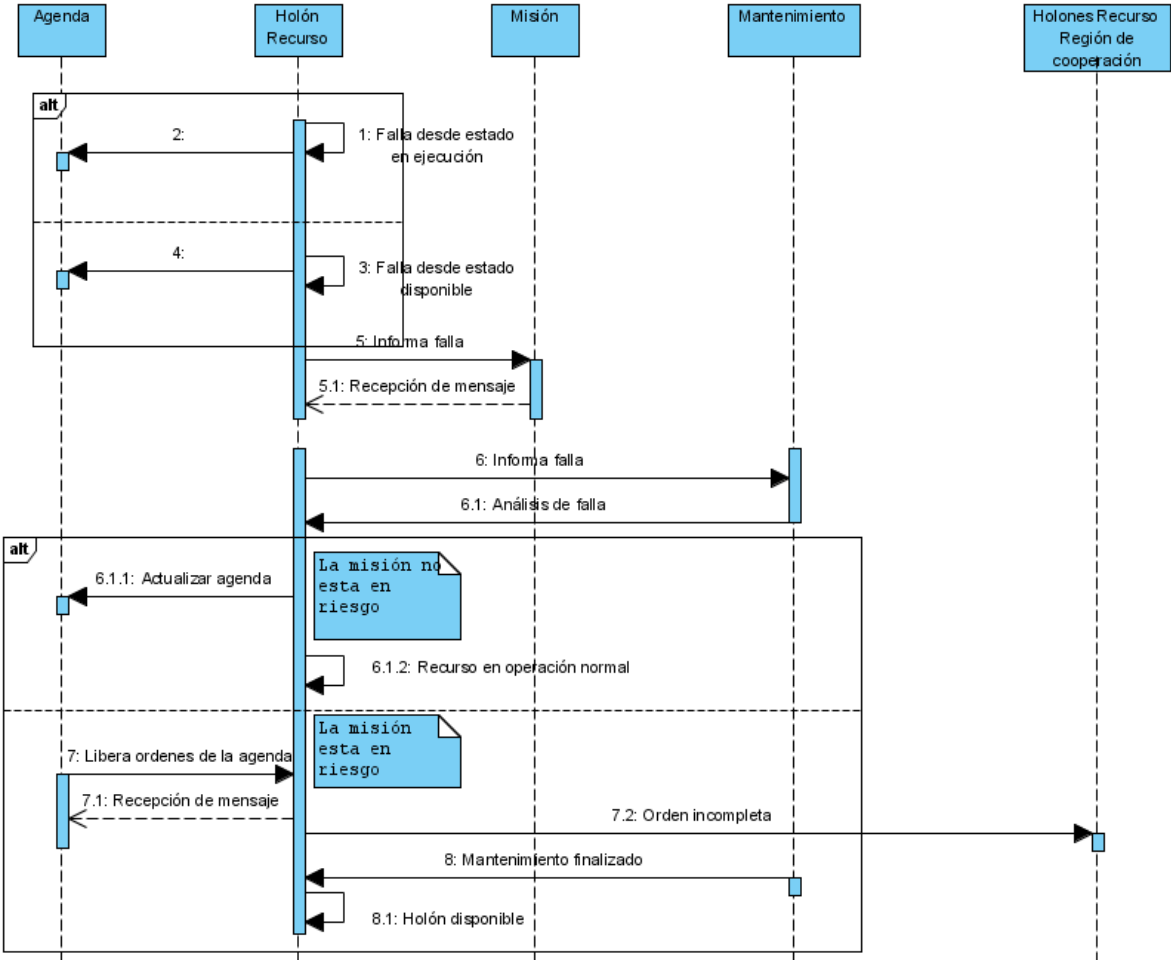
Cuando una falla ocurre en un recurso del sistema de manufactura como se ilustra en la Figura 29, el sistema de control debe tomar decisiones con flexibilidad y robustez con base en la agenda de producción del recurso, de manera que el holón pueda superar la falla sin afectar el desempeño global de la UPH. El HR debe entrar en cooperación con el holón mantenimiento, que a su vez es un HR, para determinar el impacto de la falla en la ejecución y cumplimiento de la misión.

Figura 29. Falla en el holón recurso.



En el caso en que el recurso determine que la misión está en riesgo, el HR de manera autónoma debe entrar en cooperación con otro holón que tenga las competencias y se encuentre en la región de cooperación, para que sea cumplida la orden de trabajo antes de su tiempo de entrega determinado. La Figura 30 ilustra las interacciones existentes en el HR en existencia de falla.

Figura 30. Diagrama de interacción de existencia de falla en el HR.



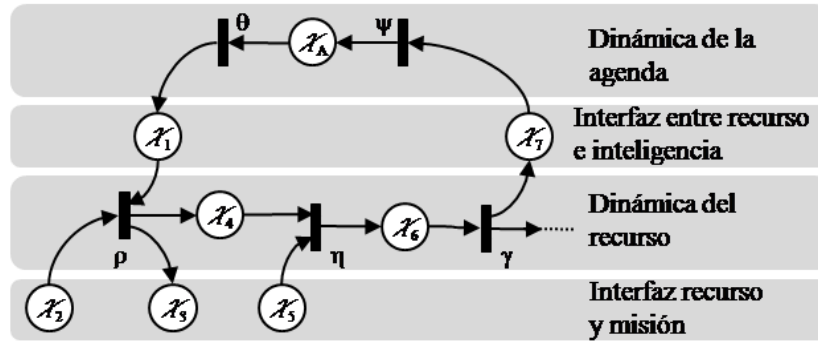
Programación Reactiva (Rescheduling)

Considerando que los sistemas de manufactura son sistemas altamente dinámicos con un entorno de incertidumbre alrededor de su comportamiento, y por tanto un evento inesperado puede ocurrir en cualquier momento, la programación reactiva se convierte en una herramienta de actualización de la agenda del HR en línea con el proceso. Específicamente, se abarca la programación reactiva generada por perturbaciones (fallas).

En la sección de "evaluación de solicitudes" se presentaron las interacciones producidas por las solicitudes enviadas por el HM de la UPHH. Allí se genera una reserva en el holón recurso de la orden de trabajo evaluada. Cuando se realiza la reserva, el HR adicionalmente envía información al

HM referente a la propuesta, con el objetivo de que el HM la evalué. Luego de que el HM analiza la propuesta de los HR y acepte la mejor opción, la reserva del HR seleccionado se convierte en una orden de trabajo, debiendo actualizarse la agenda de este HR, la Figura 31 representa el comportamiento del HR producto de la programación reactiva.

Figura 31. Dinámica de la reprogramación de la agenda del holón recurso mediante Redes de Petri.



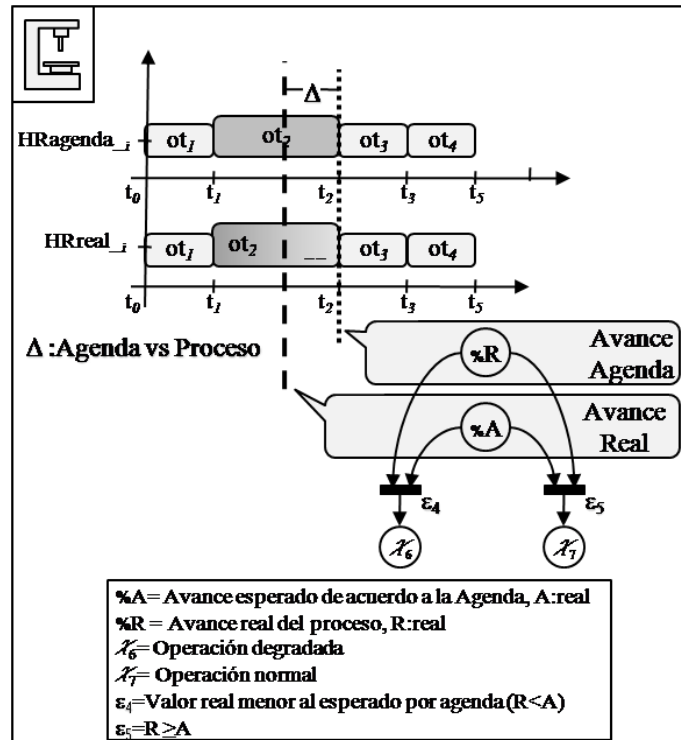
- | |
|---|
| <p>X_A=Agenda
 X_1=Lugar de sincronización HR-Agenda
 X_2=Lugar de sincronización HR-Misión con solicitud de trabajo
 X_3=Propuesta de producción de ot_i
 X_4=Reserva para ot_i
 X_5=Lugar de sincronización HR-Misión con decisión de aceptación para ejecución de ot_i
 X_6=Reserva levantada
 X_7=Lugar de sincronización HR-Misión con solicitud de trabajo
 ρ=Evaluar solicitud
 η=Recibir propuesta
 γ=Levantar reserva y ordenar programación
 ψ=Realizar programación
 θ=Enviar estado agenda</p> |
|---|

Seguimiento de Avances de Órdenes de Trabajo

El HR realiza una supervisión y monitoreo de la evolución de ejecución de las órdenes de trabajo pertenecientes a su agenda. Por tanto realiza permanentemente la consulta de su agenda para la determinación del estado del HR, obteniendo el avance de la misión en cuanto a sus órdenes de trabajo.

La UPH puede comportarse de manera proactiva a través del aprendizaje del desempeño del HR, que le permita por medio del monitoreo del grado de ejecución de las órdenes de trabajo, identificar un estado de operación degradada o un posible incumplimiento de la misión. En consecuencia el HR realiza una comparación cuantitativa de la agenda que es la ejecución esperada, y el desempeño del proceso que es la ejecución real, tal como se ilustra en la Figura 32.

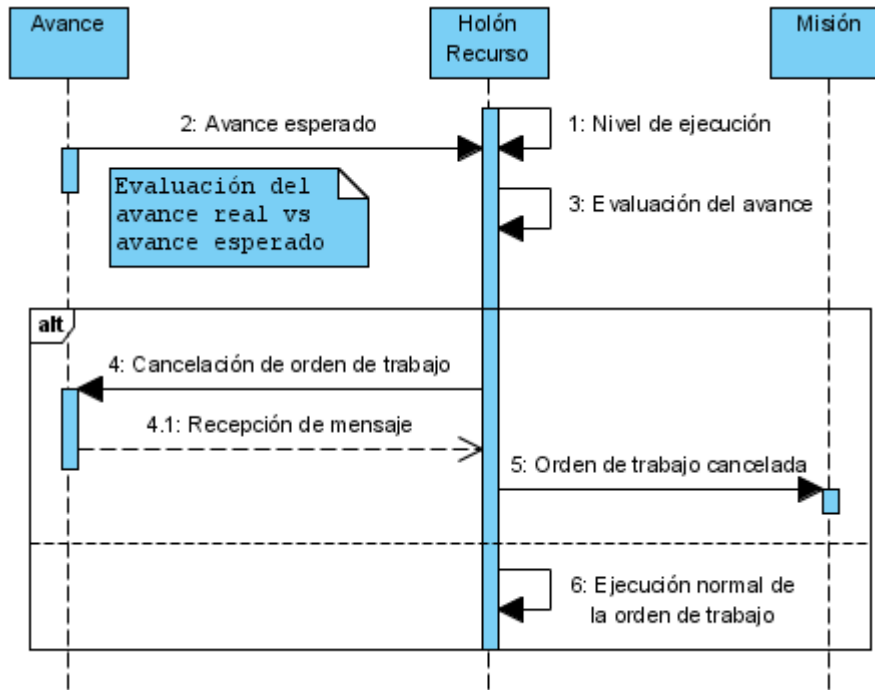
Figura 32. Seguimiento de avance de órdenes de trabajo considerando la agenda y el desempeño del proceso.



La red de Petri de la Figura 32 incorpora la capacidad de representación de tipos de datos mediante la utilización de variables, para el caso %R y %A. Considerando que el estado del proceso determinado por el avance de la misión es identificado para ser un valor real y por tanto un dato asociado a la marca que complementa la definición de estado de una red de Petri.

Cuando el HR determina que está en riesgo la misión, cancela la continuación de la orden de trabajo y la presenta a los holones recurso que están dentro de la región de cooperación, con el objeto de evaluar que holón puede finalizar la orden de trabajo en riesgo, tal como presenta en la Figura 33.

Figura 33. Diagrama de interacción para seguimiento de ejecución de las órdenes de producción



3.5.2 Negociación en la UPH

Ramos (Sousa, 2007), (Almeida, 2007), (Sousa, 1999) y (Silva and Ramos, 1999) trabaja con base en la *Agenda* de los recursos, que representa la herramienta del holón para negociar con el holón tarea los objetivos de producción que este contratará. El recurso analiza su *Agenda* considerando los intervalos de tiempo donde puede ejecutar la operación requerida. Esta *Agenda* es modificada por la dinámica del sistema de manufactura: nuevas órdenes, fallas y retrasos en los recursos.

Según (Sousa, 2007) los sistemas de manufactura holónica (HMS) requieren de un mecanismo de coordinación y colaboración robusto para asignar recursos disponibles para lograr las metas de producción, definidas por los planes de producción generados por las órdenes que llegan al sistema.

De acuerdo con (Salazar, 2003) la negociación es uno de los conceptos mayormente utilizados en el campo de la Inteligencia Artificial Distribuida (DAI) donde algunos de los objetivos generales de los agentes dentro del campo de la negociación son: la modificación de los planes de agentes si no se alcanzan resultados adecuados y la identificación de situaciones en las que las interacciones entre agentes son posibles, con la intención de alcanzar decisiones consensuadas. Adicionalmente en (Antsaklis, 1994) se sustenta que en la DAI es necesario coordinar las tareas pues cada agente tiene sus propios objetivos.

El método más utilizado en la negociación para la asignación de tareas en sistemas distribuidos es el CNP (Contract Net Protocol) (Kotak, 2003). El CNP es un protocolo de alto nivel para lograr una cooperación eficaz y robusta y fue introducido por Davis y Smith en 1980 en (Jarvis, 2001).

En (Knabe, 2002) se propone el Holonic Contract Net With Confirmation Protocol (HCNCP) que evita los problemas de realizar superposición de compromisos y la imposibilidad de generar soluciones óptimas a través de la adición de una etapa de confirmación y una segunda propuesta como posible respuesta a una solicitud. En (Knabe, 2002), (Zafeer Alibai B.A.Sc., 2003) el CNP tiene como objetivo la asignación y reasignación de tareas en un sistema Multi-Agente como característica clave de la negociación automática en un sistema de manufactura distribuida a través del uso de contratos.

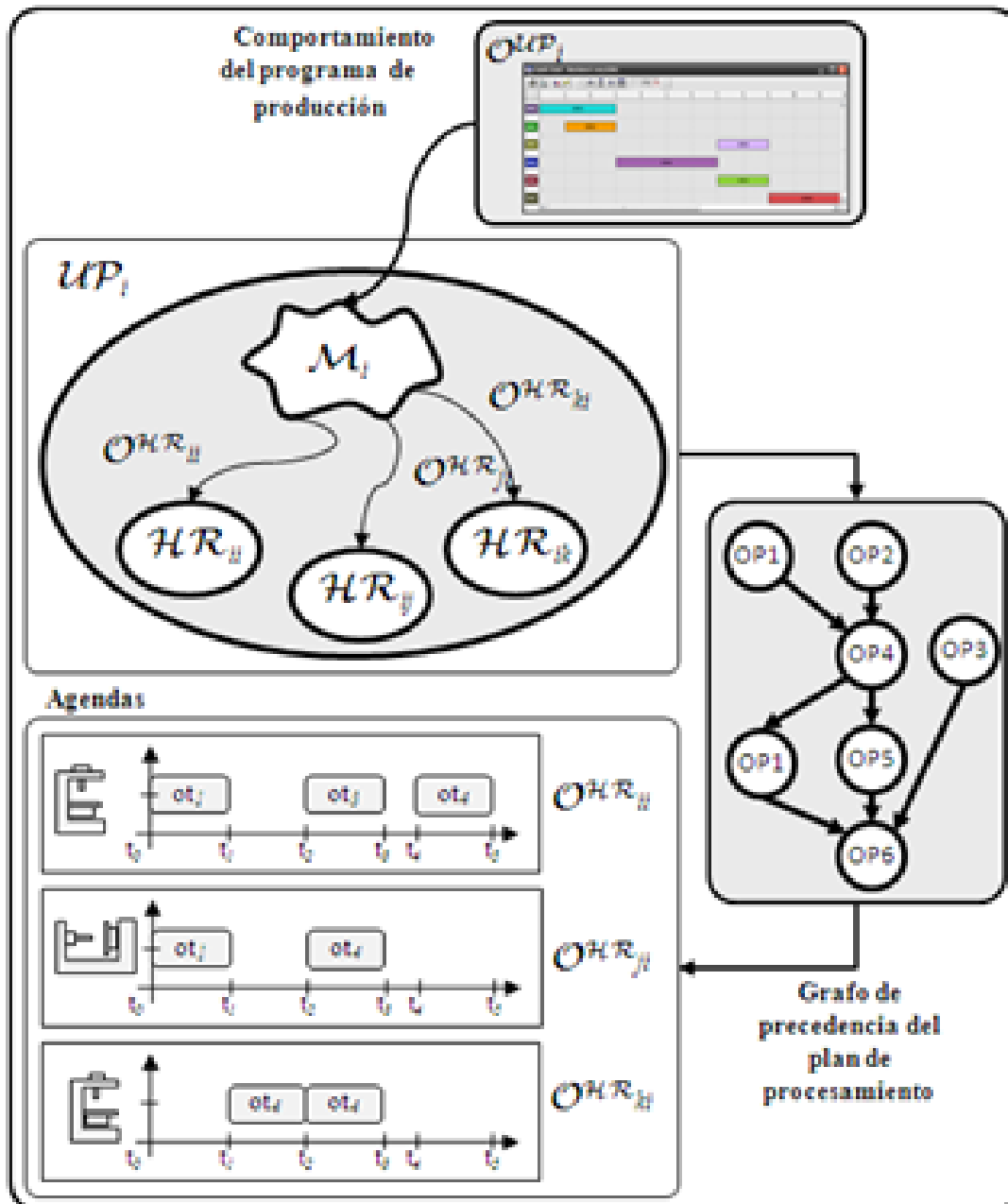
En (Farahvash and Boucher, 2004) se justifica la utilización del CNP, considerando que es un protocolo que aplica un mínimo de mensajes que lo hace muy eficiente para la asignación de tareas entre agentes. Agrega que la utilización de dichos protocolos en la elaboración de sistemas de agentes asegura que mientras siga estos protocolos, no llegará a una conversación sin sentido.

La *Agenda* de un Sistema de Manufactura Holónico basado en Unidades de Producción (HMS_UPH) es obtenida a través de la implementación de un mecanismo de negociación que le permite lograr acuerdos en relación a la asignación de objetivos de producción a los holones recurso de la UPH.

En (Zapata et al, 2008C), (Zapata et al, 2009) se presenta la negociación entre la misión y los holones recursos o entre holones recursos según sea el caso, del objetivo de producción permitiendo su descomposición en objetivos más simples que facilita su distribución hacia los holones recurso de la UPH. En este caso, una orden de producción recibida es descompuesta de acuerdo a criterios de manufacturabilidad dentro de un conjunto de tareas. Después de descompuesta la misión en operaciones u objetivos de producción, estos deben ser asignados a los HR apropiados de la UPH, como se ilustra en la Figura 34.

En (Salazar, 2003) se define que la negociación es una discusión en que las partes interesadas intercambian información con el objetivo de alcanzar un acuerdo y que por tanto viene definida por tres aspectos importantes: (a) el flujo de información es bidireccional, (b) cada negociador evalúa la información desde su propia perspectiva y (c) la decisión final se toma de mutuo acuerdo.

Figura 34. Descomposición del Plan de producción en *Agendas* para los recursos



El CNP se desarrolló como mecanismo de negociación para sistemas distribuidos inteligentes, área en la que ha encontrado su mayor potencial de aplicación. Muestra de ello es el trabajo presentado en (Gharieb, 2004) en el que se menciona que las conversaciones entre agentes frecuentemente entran en patrones típicos, donde ciertas secuencias de mensajes son esperadas en puntos específicos de la conversación, y son estos patrones de intercambio de mensajes lo que se conoce como protocolos de interacción (PI).

De (Zafeer Alibai B.A.Sc., 2003), el CNP propuesto por Smith and David en 1980, facilita la distribución de subtareas entre varios agentes y cada agente toma en el sistema un rol de *Iniciador* o *Participante* en la negociación para la asignación de la tarea.

Como se presenta en (Faúndez and Cristian, 2005), CNP es uno de los protocolos de interacción entre agentes incorporado en la amplia biblioteca de protocolos FIPA (Knabe, 2002). FIPA incorpora el CNP original con una menor modificación. Existen algunas modificaciones adicionales del CNP, donde Algunos investigadores modificaron este protocolo para encontrar algunos requerimientos especiales de su aplicación. Algunas de las extensiones definidas en el estándar son: FIPA Propose Interaction Protocol, FIPA Request Interaction Protocol, FIPA Request When Interaction Protocol, FIPA Contract Net Interaction Protocol, entre otras (Karageorgos, 2003) y (Gharieb, 2004).

Flujo de Información para la Negociación en una Unidad de Producción

En el CNP, los holones juegan dos roles: *Iniciador* y *Participante*. El *Iniciador* es el administrador y los *Participantes* los Contratistas del objetivo de producción. Las entradas del CNP son órdenes de producción, información técnica del proceso y estado actual de la planta. Las órdenes de producción representan los pedidos que algún cliente emite a la planta para que sean elaborados. Las órdenes de producción pueden contener diferentes datos pero es bastante común que estas se compongan de la siguiente información: ítem (producto o componente) a manufacturar, la cantidad que hay que producir, plan de producción y el plazo de entrega.

⟨Producto, No.Partes, Fecha_Limite, Fecha_limite⟩

Una vez la orden de producción es descompuesta en objetivos de producción, entra en operación el CNP para asignar estos objetivos en los HR de la UPH.

El mecanismo CNP está compuesto por cinco etapas para establecer un contrato entre un *Iniciador* y uno o más *Participantes*, tal como se ilustra en la Figura 35 y que se describen a continuación:

Solicitud de ofertas

El *Iniciador* anuncia una tarea a todos los posibles *Participantes* mediante mensajes tipo Multicast, como se presenta en la Figura 36.

El *Iniciador* en un HMS_UPH está caracterizado de acuerdo al generador del llamado a propuestas, esto es, si la negociación es disparada por una nueva orden, el *Iniciador* es el Modulo Misión creado para representar el objetivo de producción, de otra forma si la negociación se presenta por perturbación en el sistema, el *Iniciador* será un HR de la UPH. Entonces, el anuncio de solicitud de oferta contiene la siguiente descripción de la tarea.

⟨*Iniciador*, *Participantes*, Fecha_Limite⟩

Figura 35. Diagrama de secuencia del CNP

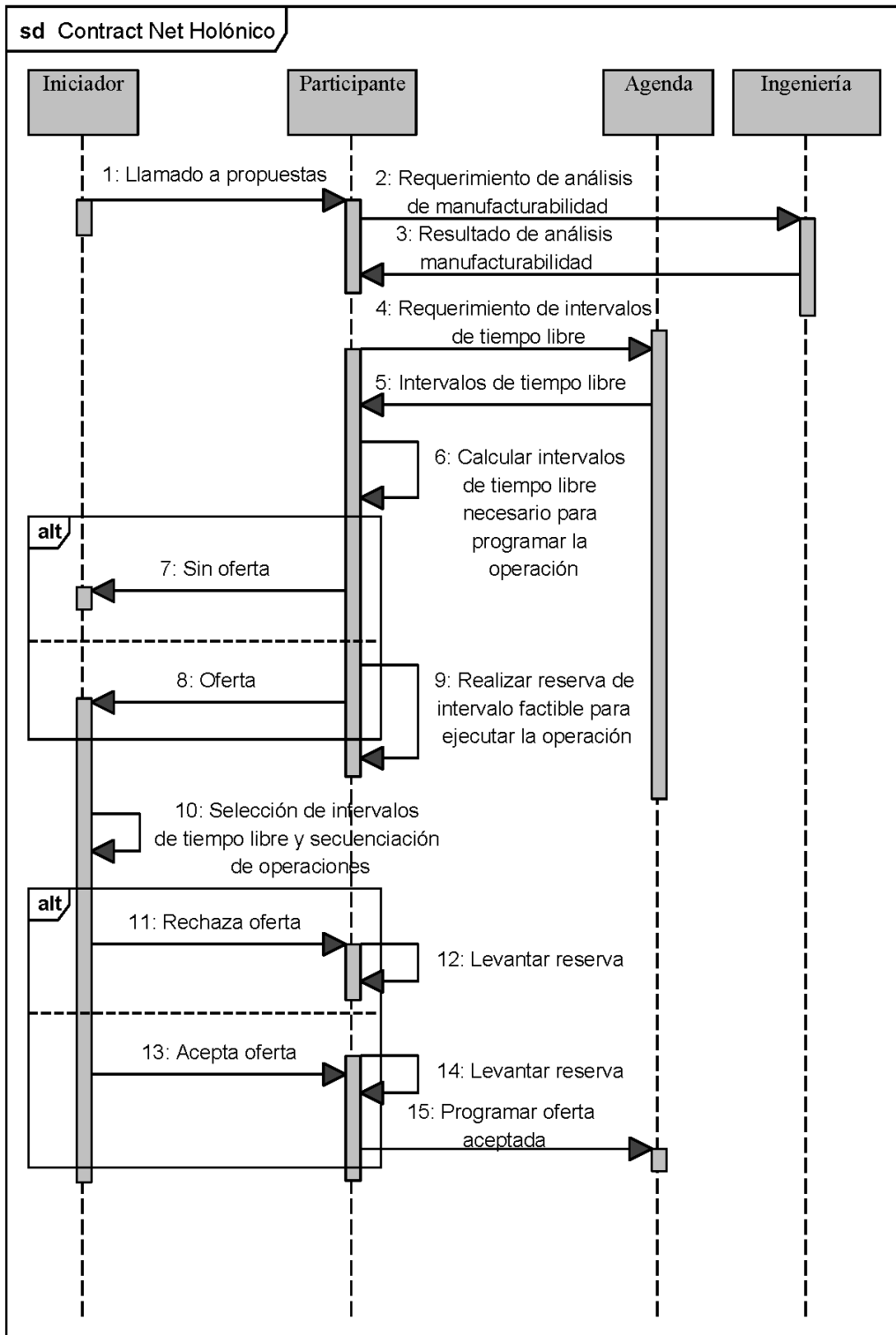
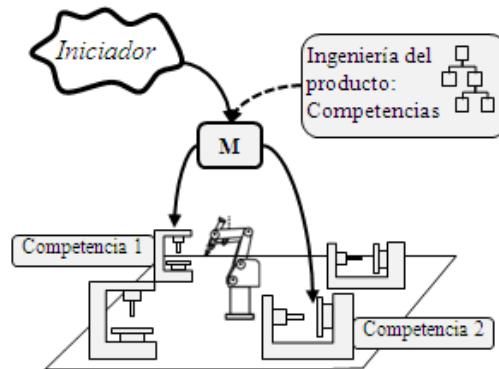


Figura 36. Solicitud de ofertas a los HR de la UPH.



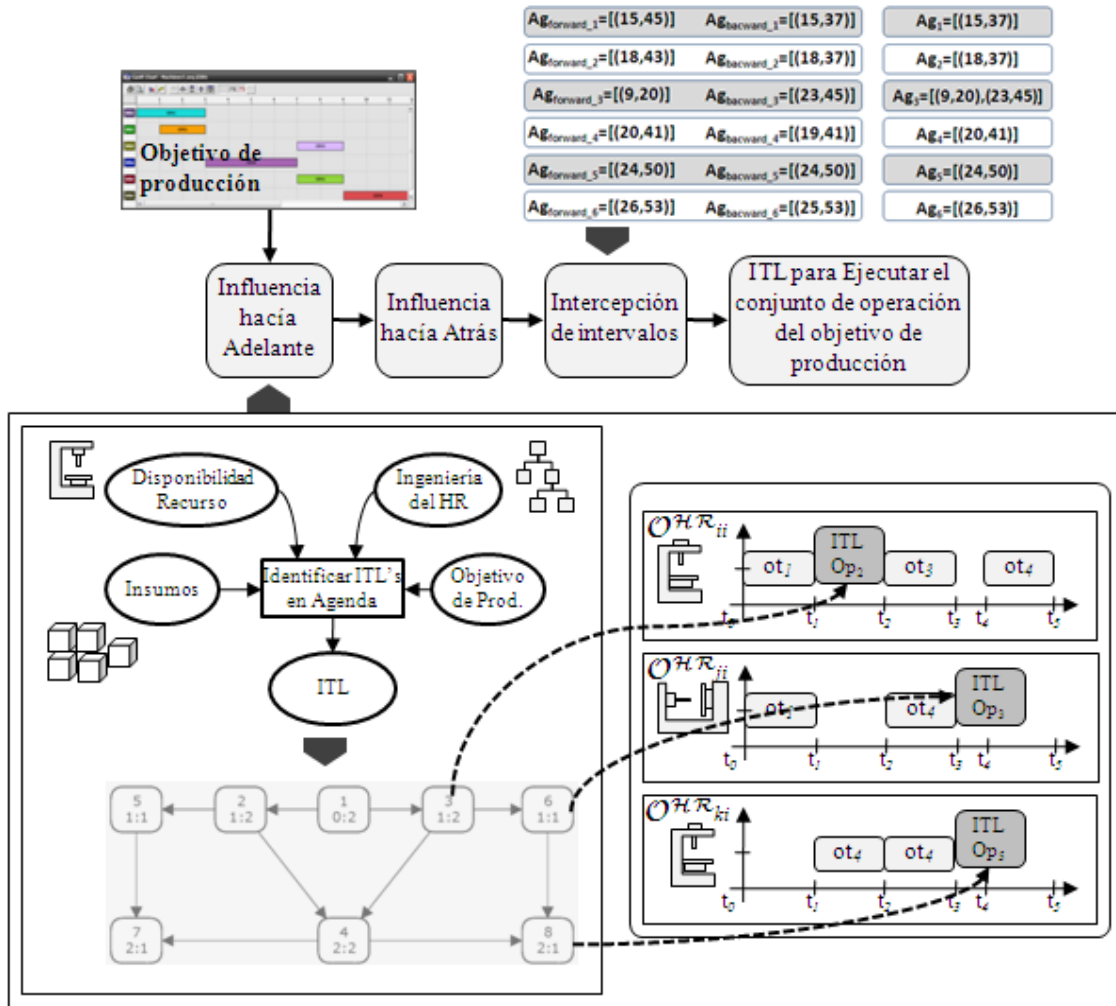
Análisis de la solicitud propuesta

En respuesta a un anuncio de solicitud de propuesta, los *Participantes* pueden evaluar su interés mediante procedimientos de evaluación de la tarea consistente en la consulta al módulo ingeniería acerca de la manufacturabilidad del producto, para determinar a partir de la concepción del diseño del producto los costos, especificaciones funcionales, disponibilidad de materiales, funcionalidad de recursos y su plan de procesamiento.

De acuerdo a este análisis de manufacturabilidad, el HR evalúa si tiene la competencia y a continuación, ese *Participante* trabajará en la construcción de una oferta para comprometerse a realizar la tarea.

En esta etapa, se implementa un método basado en los trabajos de Ramos (Sousa, 2007), (Almeida, 2007) y (Sousa, 1999) para identificar los intervalos de tiempo libre (ITL) en la *Agendas* de los HR de la UPH que tienen la respectiva competencia para ejecutar el conjunto de operaciones del objetivo de producción. Este involucra dos fases para que los holones recurso realicen su oferta: la fase de influencia hacia adelante y la fase de influencia hacia atrás, tal como se ilustra en la **Figura 37**.

Figura 37. Análisis de la solicitud de propuesta



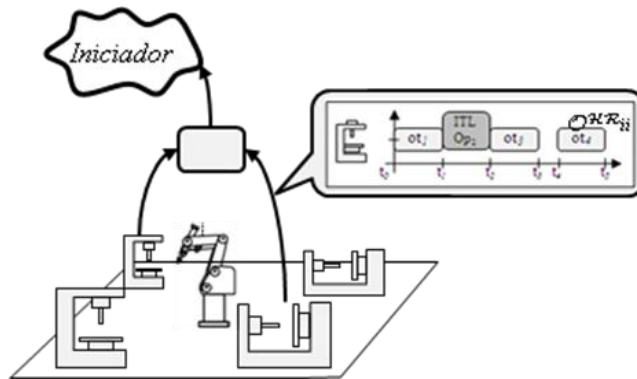
Las operaciones de Influencia hacia Adelante e Influencia hacia Atrás se realizan para ajustar el inicio y el final de los intervalos de tiempo libre en las *Agendas* de los recursos, acorde con la *Agenda* de los HR predecesores y sucesores de acuerdo al plan de procesamiento. Estos intervalos de tiempo libre son los ofertados por los HR para ejecutar el trabajo propuesto. Con el objetivo de facilitar el análisis, el método de Ramos es implementado aquí a través de los grafos de alcanzabilidad. En el grafo de alcanzabilidad cada nodo representa una lista de intervalos libres identificados en la *Agenda* de un HR para ejecutar la operación, y cada arco representa un evento de cambio *Agenda* a ser analizada.

Presentación de ofertas

Cuando cada HR calcula la *Agenda* final de acuerdo al método de Ramos, estos ofertan al *Iniciador*, como se muestra en la Figura 38, los intervalos de tiempo libre donde las operaciones pueden ser programadas. El formato de la oferta se muestra a continuación:

$\langle \text{Iniciador}, \text{Participante}, \text{Operación}, \text{Intervalo_de_tiempo_Libre} \rangle$

Figura 38. Presentación de ofertas

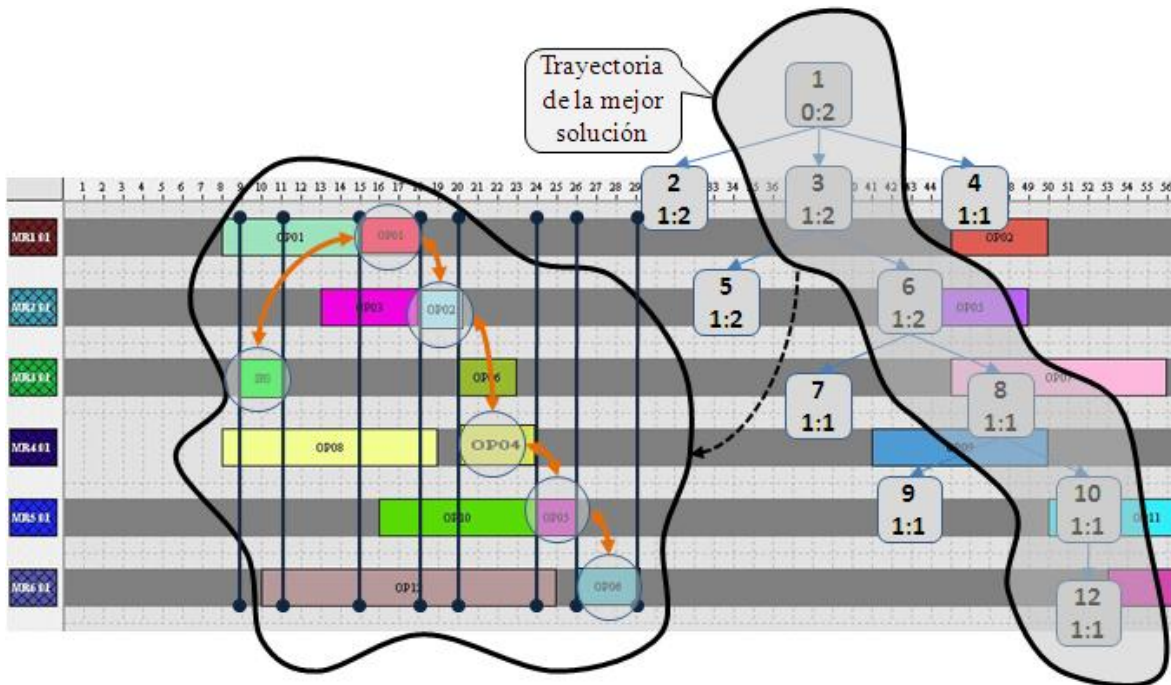


Análisis de ofertas

En este paso el *Iniciador* concede el contrato al mejor postor. La operación se selecciona entre todas aquellas que no están programadas y que según las relaciones de precedencia se pueden programar.

El análisis se realiza nuevamente a partir del modelo en Redes de Petri de Alto nivel de la planta del sistema de manufactura. El marcaje inicial del modelo consiste en los intervalos de tiempo libre ofertados en la fase de presentación de ofertas. A partir del modelo de la planta se obtiene el grafo de alcanzabilidad y utilizando la búsqueda a lo ancho mediante la regla heurística de la operación con fecha de entrega más próxima (*Due Date*), se realiza la identificación de la mejor asignación de tareas o la trayectoria de la mejor solución, como se ilustra en la Figura 39.

Figura 39. Análisis de las ofertas



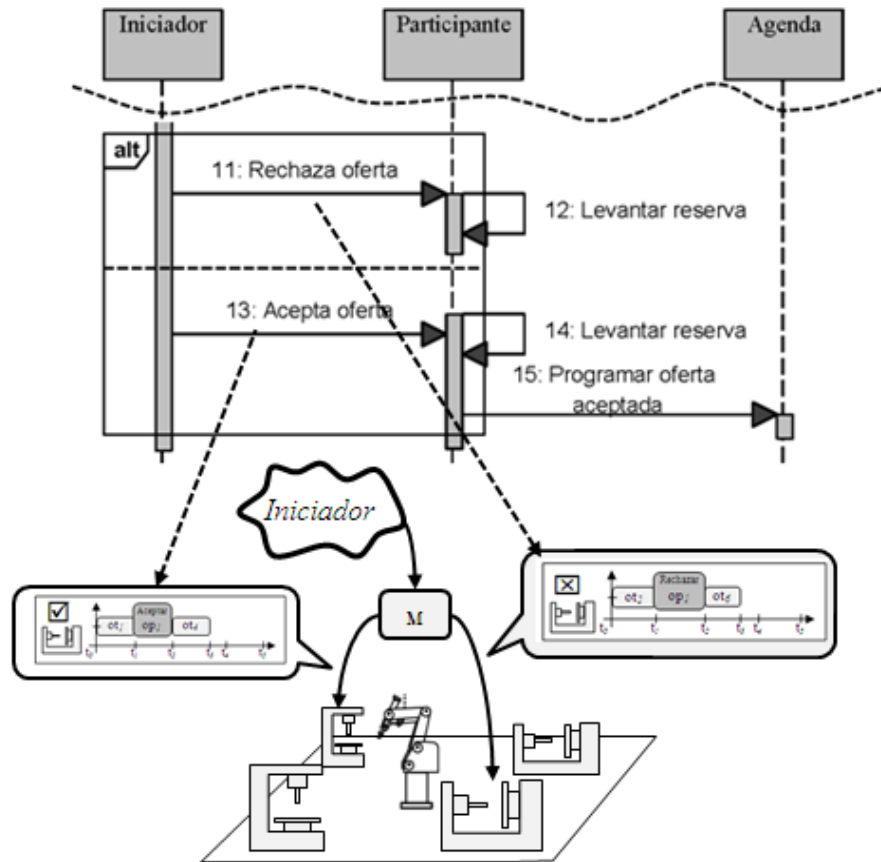
De (Domínguez, 1995), aunque las reglas de prioridad no proporcionan soluciones óptimas al problema de la secuenciación, si pueden facilitar soluciones aceptables desde el punto de vista del objetivo elegido.

Una vez obtenido el mejor camino en el grafo de alcanzabilidad, el *Iniciador* está en capacidad de adjudicar mediante contratos las operaciones a los HR elegidos.

Contratación

Una vez se determinan los momentos de comienzo y finalización de las operaciones que debe realizar cada HR de acuerdo a las ofertas presentadas, el paso a seguir es el envío de los mensajes de aceptación y rechazo de las ofertas recibidas de cada *Participante* de la UPH, como se ilustra en la Figura 40.

Figura 40. Fase de Contratación en el CNP



Cuando un HR recibe la notificación de aceptación, este añadirá este nuevo acuerdo a su programa de operaciones u *Agenda*. En caso contrario aceptará la decisión de rechazo y levantará la reserva realizada en la fase de obtención de la oferta. El mensaje enviado tiene la siguiente estructura:

$\langle \text{Iniciador}, \text{Participante}, \text{Operación}, \text{Intervalo_de_tiempo_Libre_Seleccionado} \rangle$

En (Zafeer Alibai B.A.Sc., 2003) se identifican algunas ventajas del CNP comparadas con otras estrategias de coordinación, que implementadas para el caso de sistemas de manufactura basada en unidades de producción, serían:

1. Las tareas son asignadas dinámicamente (Adjudicación de contratos).
2. Las tareas son equilibradas entre todos los HR. Los HR que ya cuentan con contrato no tienen que pujar por otras nuevas en la misma secuencia de negociación. Si un HR ya está utilizando todos sus recursos, no podrá pujar por nuevos contratos hasta que los actuales se han completado.
4. Una estrategia fiable para aplicaciones distribuidas, que pueden recuperarse de perturbaciones en la UPH.

3.6 Esquema conceptual

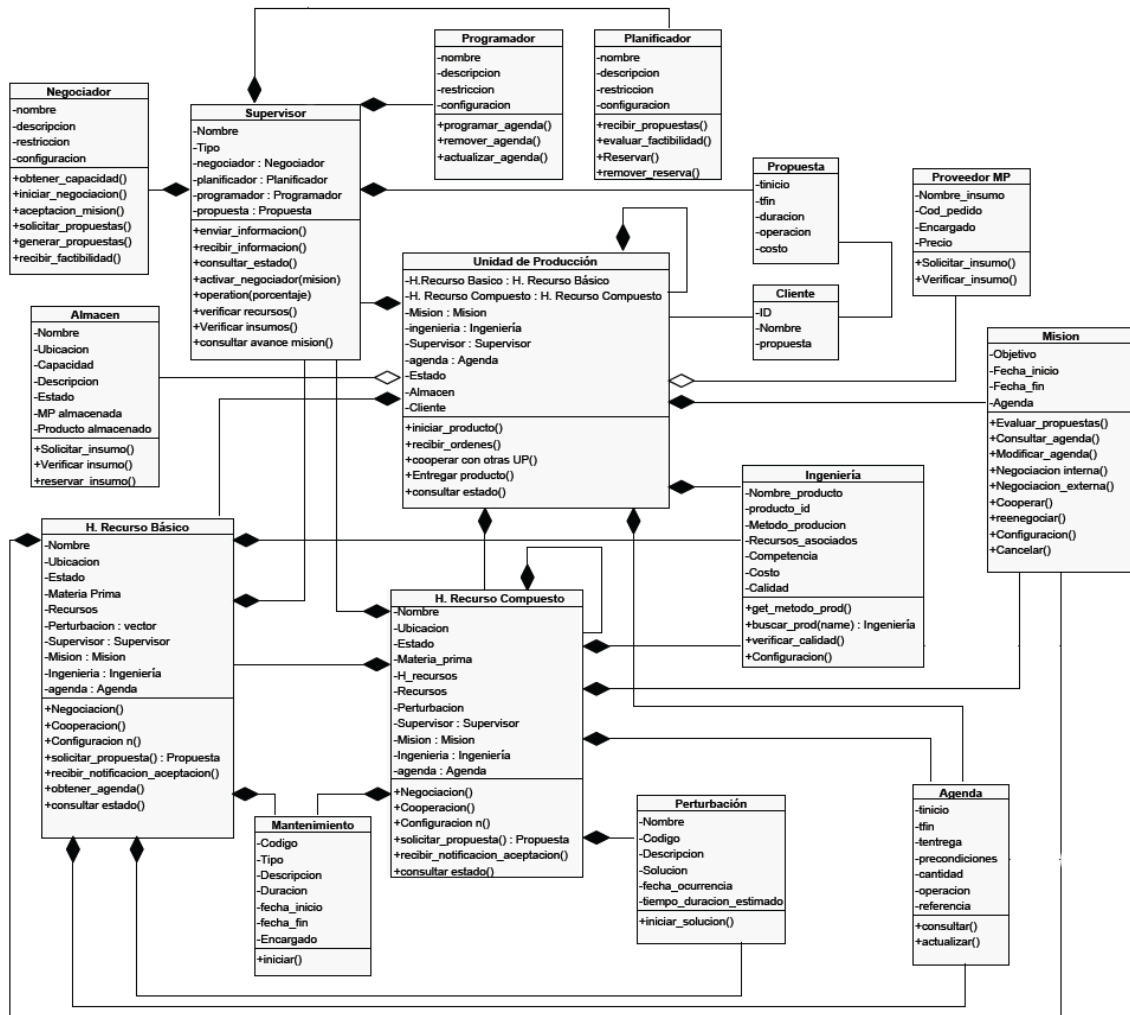
De acuerdo a lo presentado en (Fundación universitaria Los Libertadores, 2008), los esquemas conceptuales son Organizadores Gráficos que caracterizan, jerarquizan y relacionan información, recurriendo a descripciones detalladas y breves explicaciones. En consecuencia, con el esquema conceptual se pretende especificar:

- Los objetos del sistema.
- Los aspectos estructurales de los objetos del sistema que garantizan el comportamiento propuesto.
- Los aspectos dinámicos de los objetos del sistema que garantizan el comportamiento propuesto.
- Las restricciones sobre los objetos del sistema que son necesarias para garantizar el comportamiento propuesto.

3.6.1 Modelo del dominio

Con el fin de introducirnos más a la terminología del área representada en la ontología, se modela los conceptos principales del sistema holónico, en la cual se referencian las clases conceptuales significativas del dominio del problema y sus relaciones. Aunque se utilice la notación de un diagrama de clases, se aclara que este no es el diagrama de clases a implementar en la etapa de construcción, solo se busca mostrar lo que se desea plasmar, la unidad de producción holónica como unidad fundamental de la holarquía y las entidades e información que constituyen y permiten integrar estas unidades. Este modelo se observa en la Figura 41.

Figura 41. Modelo del dominio para un sistema holónico



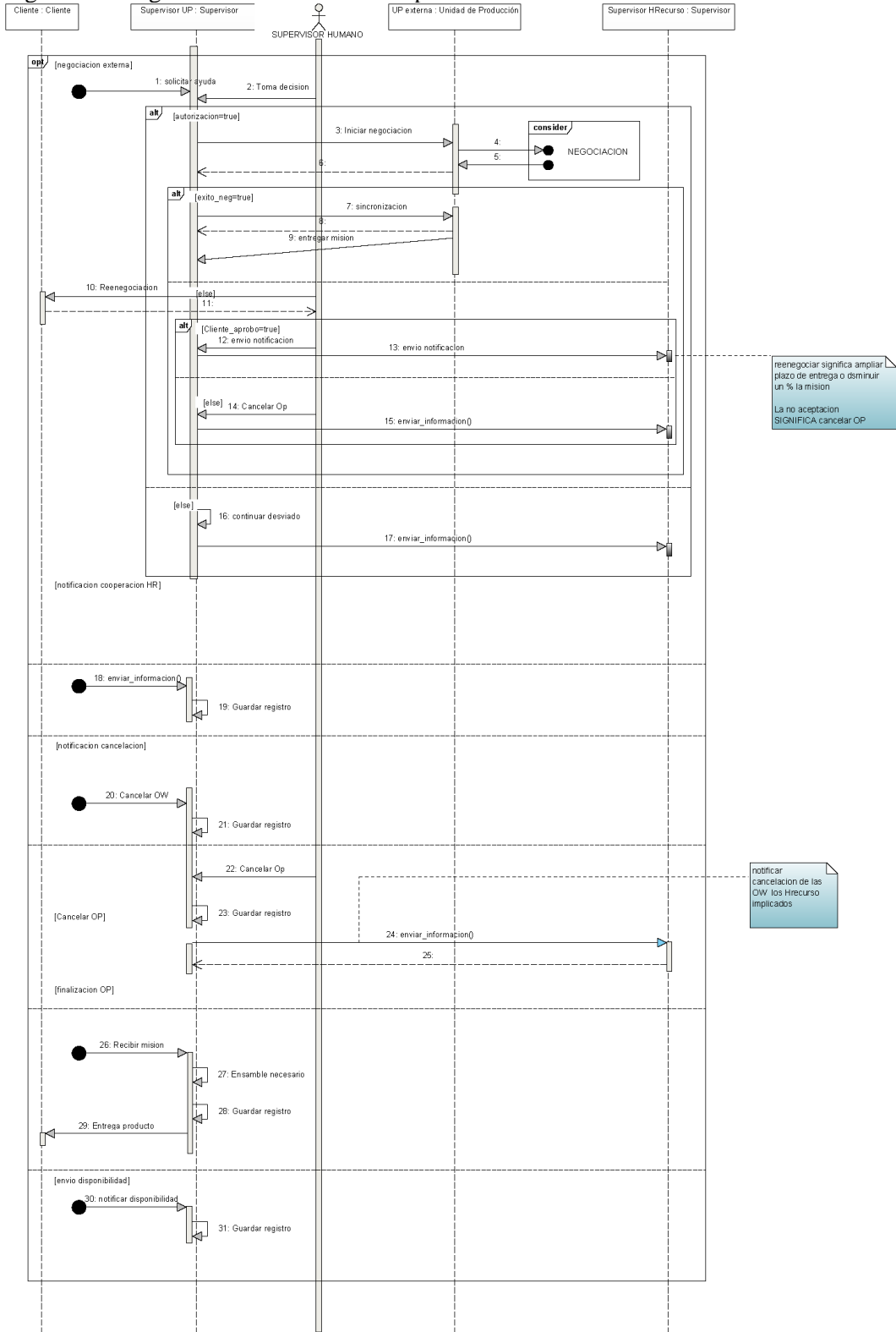
3.6.2 Diagramas de secuencia

Un diagrama de secuencia es un diagrama de interacción que muestra los objetos como líneas de vida a lo largo de la página, con interacciones representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino. (OMG, 2009), (Lin et al, 2004). Los diagramas de secuencia son apropiados para mostrar la comunicación entre objetos y qué mensajes disparan esas comunicaciones.

Los diagramas de secuencia contienen básicamente líneas de vida representadas con un rectángulo que contiene el nombre del objeto o actor de la arquitectura holónica y una línea larga punteada debajo de este. También se encuentran las ocurrencias de ejecución que son pequeños rectángulos que se añaden a la línea de vida para expresar actividad en esa porción de tiempo. Por último, los mensajes muestran el tipo de información enviada con su respectiva dirección. Estos diagramas también utilizan otras convenciones llamadas *fragmentos combinados* los cuales son muy útiles para detallar los modelos. La

Figura 42 la operación de la UPH mediante diagramas de secuencia, igualmente la Figura 43 utiliza UML para representar las secuencias de operación en el holón recurso.

Figura 42. Diagrama de secuencia de la operación de la UPH



3.7 Enfoque de Desarrollo de un Sistema de Control Inteligente de Manufactura Basado en Unidades de Producción Holónica (UPH)

Para el diseño de un sistema de control inteligente de manufactura es relevante la definición de un enfoque que indique los pasos a seguir para su desarrollo. Para esto se toma como base la metodología de desarrollo para sistemas holónicos de manufactura (HMS) definido por Giret en (Giret, 2005), y que permite identificar e implementar holones. Este método de desarrollo es una metodología Multi-Agente para el desarrollo de Sistemas Holónicos de Manufactura. Esta propuesta se basa en los requisitos de modelado para Sistemas Holónicos de Fabricación y en la noción de Agente Abstracto para el modelado de entidades autónomas con estructuras recursivas.

En consecuencia, las fases definidas para el desarrollo de un sistema de control inteligente están dadas por la definición de los requisitos del sistema de control, análisis, diseño e implementación de holones. Específicamente una breve descripción del objetivo y posibles herramientas a utilizar se describen a continuación:

3.7.1 Requisitos del sistema

En esta fase se tiene como objetivo realizar una contextualización del sistema candidato a establecer un sistema de control inteligente. Se propone que es necesario realizar una identificación del sistema de control, identificación de los procesos del sistema de manufactura que deben ser controlados y las especificaciones de las condiciones de operación del sistema de manufactura. La contextualización acerca del sistema puede complementarse con la identificación de los actores y roles del sistema de manufactura y la definición de un marco ontológico que represente la estructura, y clasificación taxonómica de los conceptos y objetos que definen el dominio del sistema de manufactura.

Esta fase puede ser desarrollada utilizando diagramas IDEFO, diagramas de actividades y caos de uso de UML. Adicionalmente, la contextualización del sistema puede realizarse mediante lenguaje natural y realizar una agrupación de conceptos mediante mapas mentales. El comportamiento del sistema puede ser representado con Diagramas o IDEFO o preferiblemente mediante PN. Las PN pueden ser implementadas en cada una de las fases de diseño, considerando sus atributos ya definidos para el modelamiento de sistema de control inteligente.

3.7.2 Análisis

El análisis se realiza con el objetivo de identificar holones y módulos que representan los objetos del sistema de control inteligente. Una propuesta de los holones y módulos del sistema puede ser obtenida a través de los roles identificados en la fase de requisitos del sistema. Tanto los holones como los módulos definen los modelos en los que se identifican y especifican los objetos que implementan una determinada holarquía. Estos modelos permiten representar tanto la estructura como el comportamiento del sistema de control inteligente. Para describir el comportamiento debe incluirse los modelos de coordinación y supervisión, considerando que el sistema de control inteligente se caracteriza por el control de sistemas distribuidos, reactivos, autónomos y

cooperativos. Los aspectos de autonomía y cooperación permiten el cumplimiento de objetivos y la reacción a los disturbios que se presenten en el sistema.

Considerando que los sistemas de control inteligente son sistemas autocontenidos, esta fase se repite para cada nivel de abstracción hasta que todo holón y módulo se defina completamente. La descomposición física (la más obvia): los holones recurso son utilizados para representar las entidades del mundo físico, tales como trabajadores, máquinas y herramientas.

En la descomposición funcional los módulos como agrupación de componentes e información son utilizados para encapsular funcionalidades tales como órdenes de trabajo, planificación, scheduling e información de productos.

Las PN en el análisis permiten trabajar de manera natural la agregación, descomposición y recursividad características de la arquitectura de control. Los diagramas de casos de uso aquí permiten identificar los objetos del sistema de control inteligente y los dominios de cooperación. Complementariamente los diagramas de actividades permiten representar el conjunto de tareas que desarrolla cada entidad y su secuencia de ejecución.

3.7.3 Diseño

En la fase diseño se realiza la composición de los modelos para formar la holarquía que permite el control y supervisión del sistema de manufactura. Esta etapa se caracteriza por una refinación de los modelos de análisis y por incluir mayores detalles de la implementación.

En esta fase es necesario garantizar que el objetivo o misión del holón corresponda a la agrupación de logros de un conjunto de tareas y el conocimiento suficiente para lograrlo. Aquí es necesario dejar claridad de las interacciones a llevarse a cabo entre entidades de la arquitectura.

Además de las PN, la utilización de los diagramas de secuencia es necesaria para definir el comportamiento e interacción de las componentes de la holarquía.

3.7.4 Implementación

En la fase de implementación de holones, el programador debe implementar el HMS siguiendo la Arquitectura del Sistema definida en la fase de diseño. En esta fase se debe definir e implementar las distintas plataformas que permitan programar en la Arquitectura del Sistema, es decir, implementar los distintos holones y módulos identificados en la Arquitectura del Sistema.

De acuerdo a (Gómez, 2003) existen plataformas de desarrollo que dan soluciones parciales al modelado de comportamiento y a la coordinación de agentes. En (Pérez, 2007) se precisa que el rango de estas soluciones considera proporcionar servicios básicos como gestión de agentes, librerías de algoritmos, localización de agentes o movilidad.

A la hora de desarrollar un SMA para el control inteligente se pueden considerar dos filosofías de trabajo, una basada en la utilización de un lenguaje de especificación de agentes y la otra a través del desarrollo de un sistema de software a partir de una plataforma que sirve de base a la construcción del SMA. En este trabajo se consideró y se identificó que es fundamental iniciar el trabajo con una plataforma, considerando que la tendencia actual en el desarrollo de SMA es la utilización de plataformas de desarrollo y las dificultades de los lenguajes que involucran un nivel considerable de conocimiento para programar un desarrollo en agentes.

Las plataformas de desarrollo más extendidas es JADE (Bellifemine et al, 2008) y Grasshopper (Breugst y Magedanz, 1998), que es la implementación oficial del estándar FIPA (FIPA, 2008), y soporta todos los servicios básicos de infraestructura especificados en FIPA (comunicaciones, movilidad, gestión de agentes y localización de agentes), a los que añade algunas utilidades gráficas para facilitar la administración de las plataformas y la depuración de los mensajes intercambiados por agentes en tiempo de ejecución.

El caso de estudio presentado en este trabajo se realiza sobre la plataforma JAVA, esta realiza la negociación de la misión del holón mediante el llamado de una librería de JADE que implementa el protocolo Contract Net.

CAPÍTULO 5

VALIDACIÓN Y VERIFICACIÓN DEL MODELO DE CONTROL INTELIGENTE

La validación y verificación permite evaluar la capacidad del modelo de representar el comportamiento controlado con que debe operar el sistema real, es decir, evaluar si el modelo se comporta como debería. Estas propiedades son independientes de las funciones específicas que realiza el sistema (ausencia de bloqueos, conjunto de estados finito, ausencia de conflictos y exclusión mutua), por tanto, la validación y verificación da idea también de la forma en que el modelo caracteriza la lógica del sistema y permite identificar errores.

Entonces, mediante un análisis de verificación y validación se determina si los modelos diseñados cumplen con las propiedades y especificaciones definidas sobre su comportamiento y estructura, es decir, cumple con las propiedades descritas en el CAPITULO 1. Los métodos de análisis pueden clasificarse principalmente en tres grupos (Orozco et al, 2004): Árbol de alcanzabilidad, análisis mediante Ecuaciones Matriciales y Técnicas de Reducción y Síntesis. En esta capítulo se utiliza el árbol de alcanzabilidad como herramienta de validación y verificación de los modelos realizados mediante redes de Petri. Adicionalmente, la herramienta permite aprovechar el cálculo mediante el cual se generó el árbol de alcanzabilidad para obtener un reporte que lo complementa (CPN Group, 2009).

4.1 Validación y Verificación Mediante CPN Tools.

En la validación y verificación de los modelos se utiliza el software CPN Tools que es una herramienta para la edición, simulación y análisis de modelos en redes de Petri. El análisis de los modelos en esta herramienta puede ser realizado mediante la simulación de los modelos (Método Informal) o la obtención del grafo de alcanzabilidad (Método formal).

El grafo de alcanzabilidad es un grafo dirigido que posee un nodo para cada estado posible del modelo y arcos que van de cada uno de estos estados a sus posibles estados subsecuentes. CPN Tools soporta generación, análisis y trazado del espacio de estados de un modelo.

4.1.1 Análisis de Simulaciones

Se basa en la observación de la forma en que evoluciona el modelo. Esta técnica consiste básicamente en ejecutar la simulación y estudiar la forma en que se habilitan y ocurren las transiciones, así como los marcajes en un paso determinado. Esta forma de analizar el modelo permite al usuario visualizar gráficamente en qué estado se encuentra el proceso, cuáles acciones no se están haciendo adecuadamente y cuáles son las partes del modelo que presentan bloqueos, con base en lo cual el usuario puede intervenir inmediatamente el modelo y hacer las correcciones necesarias.

4.1.2 Esquema del Grafo de Alcanzabilidad

Como se explicó en el Capítulo 1, el Grafo de Alcanzabilidad de una red de Petri es usado para estudiar las propiedades de Alcanzabilidad, Acotación, Reversibilidad, Vivacidad y Imparcialidad (Imparcialidad). CPN Tools tiene la capacidad de generar de manera automática el Grafo de Alcanzabilidad de una PN y le permite al usuario dos opciones de análisis: la primera se basa en la observación del grafo mismo y la segunda consiste en analizar los resultados entregados por el generador de espacio de estados del programa.

Interpretación del Esquema del Grafo de Alcanzabilidad - Holón Recurso

La Figura 44 muestra el Grafo de Alcanzabilidad completo del holón recurso que se ilustra en la Figura 17. Adicionalmente en la Figura 45 se identifica que cada nodo y cada arco tiene asociado un cuadro de texto al cual se le ha denominado *Descriptor de Nodo* o *Descriptor de Arco* según sea el caso, adicionalmente los nodos han sido dotados con tres números: El valor superior indica el número de nodo (o estado) en el que se encuentra el proceso, el valor inferior izquierdo muestra el número de nodos (o estados) antecesores a dicho estado y finalmente el valor inferior derecho detalla el número de nodos (o estados) sucesores.

Figura 44. Grafo de alcanzabilidad del modelo en PN del holón Recurso

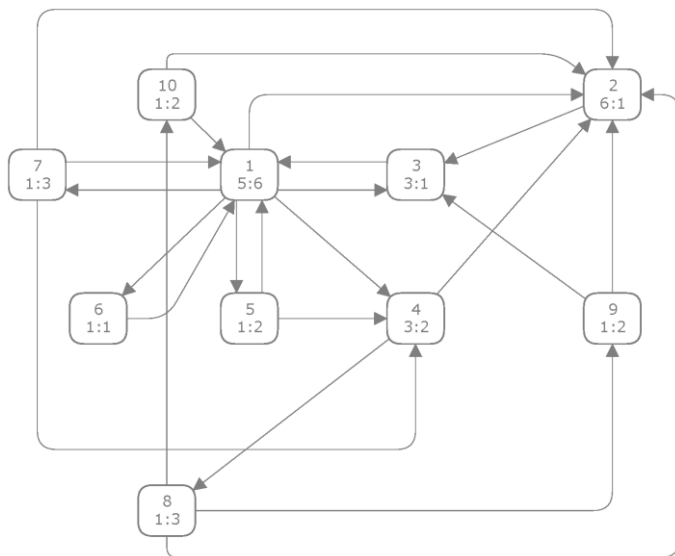
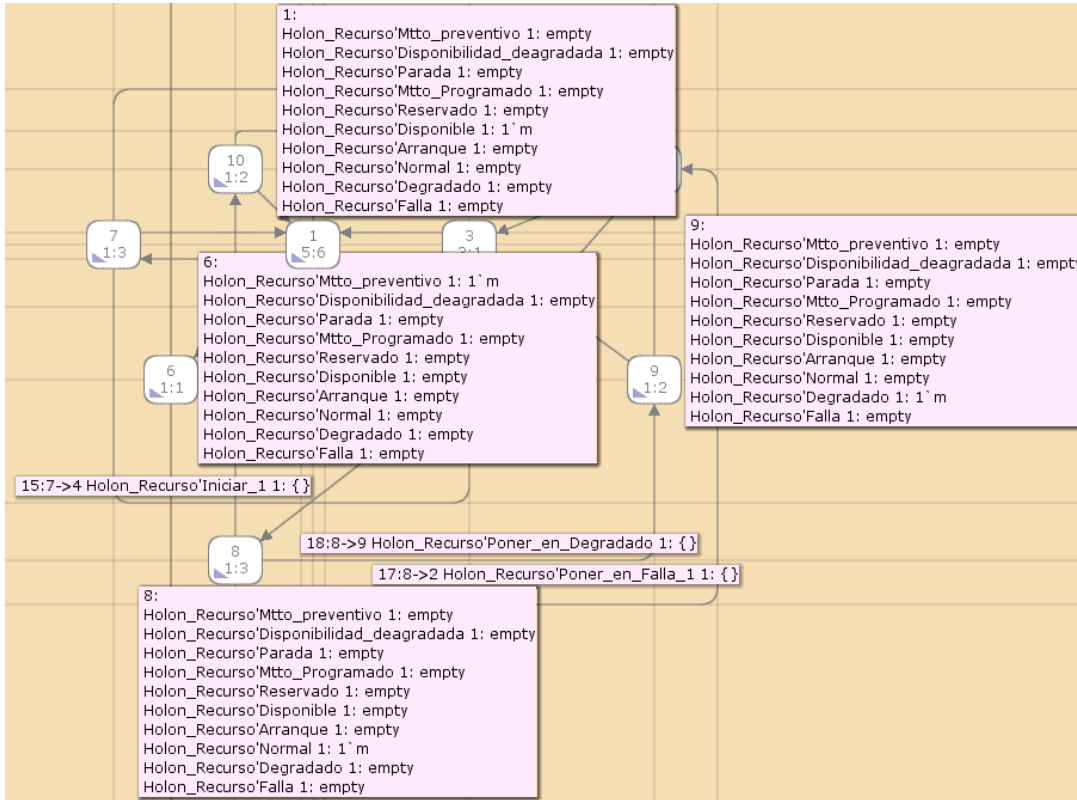


Figura 45. Regiones del grafo de alcanzabilidad del modelo en PN del holón Recurso



Descriptores de Nodo: Muestran el estado, marcaje en cada uno de los lugares (junto con la página donde estos se encuentran) y los valores de las variables (si existen) en dicho estado.

Descriptores de Arco: Muestran el número del arco, el cambio de estado y la transición (junto con la página donde esta se encuentra) por la cual se dio dicho cambio.

Del Grafo de Alcanzabilidad se pueden analizar ciertas características de las PN, para llevar a cabo este tipo de análisis se puede tener en cuenta lo siguiente:

Acotación: El número de marcas en cada uno de los lugares puede ser observado en los descriptores de nodo, si este número es finito, la red es *acotada*, si es igual a 1 la red es 1-Acotada es segura.

Reversibilidad: Los arcos indican los caminos que pueden seguirse para ir de un estado a otro, si existe algún camino mediante el cual es posible siempre regresar a un estado, dicho estado es *reversible*. La *Ciclicidad* de una PN es un caso particular de la propiedad de *Reversibilidad* en el cual el marcaje (o estado) inicial siempre puede ser alcanzado desde cualquier otro marcaje (o estado).

Vivacidad: Los estados terminales indican estados de bloqueo en el sistema, un estado Terminal es un nodo en el cual no hay arcos de salida. Una red *No Viva* posee al menos un estado Terminal.

Una característica adicional que puede observarse en el Grafo de Alcanzabilidad es la Conflictividad.

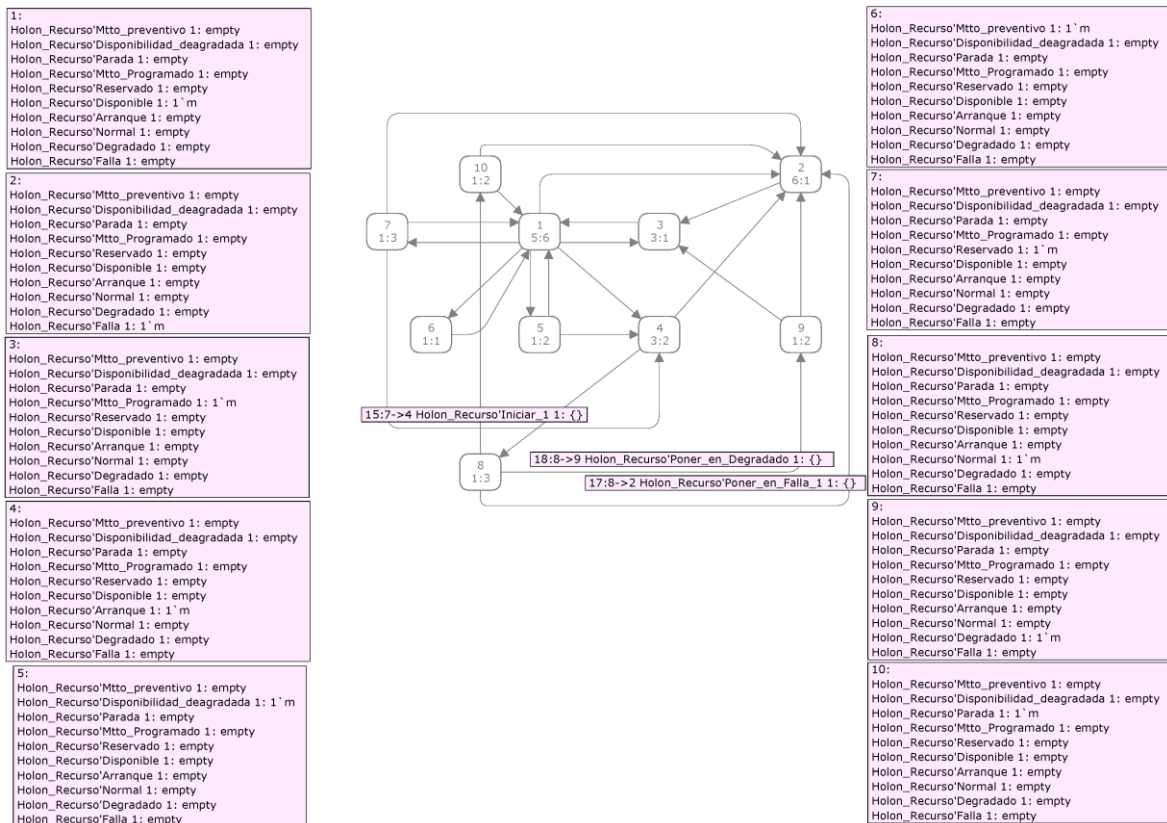
Conflictividad: Un estado que posee más de un sucesor indica un conflicto en la red.

De la Figura 46 es posible identificar que los estados 1, 4, 5, 7, 8, 9 y 10 que corresponden a los estados Disponible, Arranque, Disponibilidad Degradada, Reservado, degradado y Parada respectivamente son estados que están en conflicto considerando que tiene más de un sucesor. En este caso el modelo del holón recurso debe garantizar que las transiciones que generan el conflicto tengan reglas claras para resolverlo. Esto es, en el estado de Disponibilidad el supervisor del holón debe entregar el estado actual del sistema y de acuerdo a este el holón debe regular su comportamiento mediante una ley de control que le indique que transición debe disparar.

Adicionalmente, valorando el marcaje de los lugares en cada nodo descriptor, se identifica que la PN del holón recurso es acotada y por tanto segura.

La PN es reversible considerando que siempre es posible encontrar una trayectoria que lleve a un estado determinado, es decir, cualquier estado es alcanzable desde cualquier otro marcado, conjuntamente, esto permite definir que el modelo PN es controlable. La vivacidad del modelo se garantiza dado que no tiene nodos terminales.

Figura 46. Descriptores de nodo del grafo de alcanzabilidad del holón recurso.



Como pudo identificarse, la anterior técnica de análisis no permite determinar otra propiedad igualmente importante como lo es la ***Imparcialidad***. Esta razón lleva a generar el reporte textual que puede obtenerse a través de la opción de Grafo de Alcanzabilidad.

Reporte Estándar desde el Grafo de Alcanzabilidad

Ahora se genera el Reporte Estándar como una extensión de la utilidad del grafo de Alcanzabilidad a nivel gráfico de CPN Tools, este reporte detalla las características y propiedades más importantes del modelo de acuerdo con las propiedades definidas en el capítulo 1.

El reporte obtenido para el modelo PN del holón recurso se encuentra en el ANEXO 3 y puede concluirse lo siguiente:

- Se validan las propiedades de vivacidad, acotabilidad y reversibilidad (Home properties).
- Como se mencionó anteriormente mediante el análisis del árbol de alcanzabilidad, la PN del holón recurso, es una red que no es libre de conflictos, pero que la mayoría de las transiciones de este modelo lo resuelven de manera autónoma y justa, esto la transiciones en conflicto pueden habilitarse varias veces pero no siempre ocurrirán.
- Las transiciones “Holon_Recurso'Sin_Falla”, “Holon_Recurso'Fin_Mtto_Preventivo” y “Holon_Recurso'Mtto” son transiciones objetivas, esto es, son transiciones que en un conflicto no siempre se habilitan y por lo tanto no pueden ocurrir. Pero son transiciones con retardado en su ocurrencia, pero en algún momento se habilitarán para su disparo. Ej. La transición “Holon_Recurso'Sin_Falla” depende de la existencia de una falla en el sistema y que esta sea reparada, y considerando que las fallas del sistema son periódicas no deterministas, esta transición solo se habilita en ciertos momentos de la operación del holón recurso.
- La propiedad de imparcialidad, permite en un sistema de control inteligente que es de naturaleza concurrente y distribuida, evaluar la existencia situaciones en las que se debe tomar la decisión de optar por una alternativa en el flujo de ejecución en detrimento de otras. Esto es permite mostrar como un componente de la arquitectura puede satisfacer de forma autónoma una situación de bloqueo del sistema y cooperar para que el modelo sea un modelo de comportamiento vivo con ausencia de bloqueo.

Interpretación del Esquema del Grafo de Alcanzabilidad – Misión del Recurso

Figura 47. Grafo de alcanzabilidad del modelo en PN de la misión del recurso

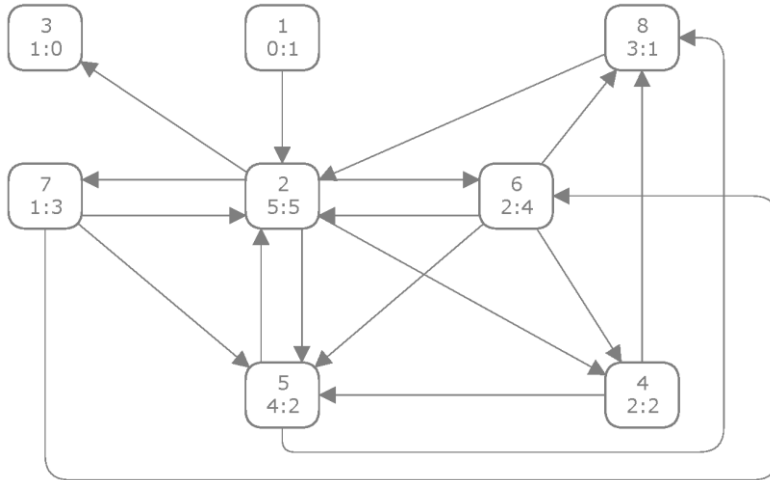
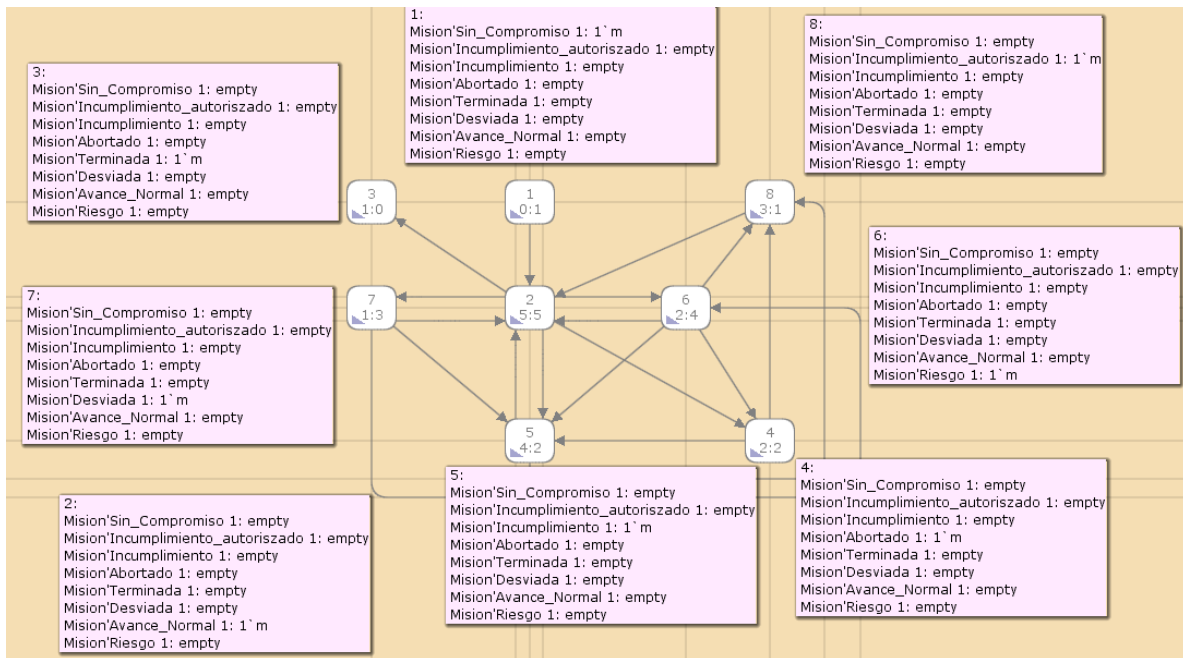


Figura 48. Descriptores de nodo del grafo de alcanzabilidad de la misión del recurso



De la Figura 47 se identifica que en el nodo 3 el lugar “Mision'Terminada” representa un estado terminal de la PN de la misión del recurso, esto es lógico considerando que este estado representa la terminación de una misión, y como se mencionó, una misión se crea con la llegada de una orden de producción y se destruye cuando la cumple.

Adicionalmente, valorando el marcaje de los lugares en cada nodo descriptor, se identifica que la PN del holón recurso es acotada y por tanto segura.

De la Figura 47 se visualiza que siempre un estado puede ser alcanzado desde cualquier otro marcaje, a excepción de los estados que se ilustran en los nodos descriptores 1 y 3, relacionados con la creación y destrucción de la misión del holón. Conjuntamente de la evaluación de los nodos descriptores se infiere que la PN es acotada y segura, dado que ningún estado sobre pasa la cantidad de una marca en cada lugar de la PN de la misión del holón.

Adicionalmente se visualiza la existencia de conflictos en algunos estados del grafo de alcanzabilidad. Esto queda mayormente descrito en el reporte que se presenta en el ANEXO 3 para la misión del holón. En este reporte se identifica nuevamente que el nodo 3 representa un nodo terminal. Simultáneamente, se lee que las transiciones del modelo PN resuelven de forma justa los conflictos existentes en la operación del modelo. La transición “Mision'sn_2”, resuelve el conflicto de manera objetiva, esto es debido a que es una transición que se presenta luego de un autorización del sistema, esto es, para que la transición se habilite depende de la existencia de una autorización como información de entrada en una transición previa en la dinámica del modelo.

La dinámica de la misión del modelo nota disminuida su autonomía por la existencia de autorizaciones como información de entrada que permite que el sistema de control evolucione en su operación sobre el sistema de manufactura. De otra parte, la cooperación del modelo no se ve afectada considerando que la mayoría de las transiciones son justas y objetivas, lo que demuestra la cooperación de los sistemas, para que el comportamiento del este no sea de inanición, demostrando la existencia de paso de mensajes entre objetos de la arquitectura de control.

CAPÍTULO 6

CASO DE ESTUDIO

El caso de estudio que permite validar el modelo de control inteligente, está constituido recursos físicos reales ubicados en el laboratorio de Mecatrónica de la Universidad Nacional de Colombia-Facultad de Minas y sus demás componentes son definidos de forma académica, tales como sus piezas, rutas de acuerdo a las competencia definidas para los recursos, competencias de los recursos, entre otras. Este sistema consta de 7 recursos:

- 2 tornos (Torno Sena y Torno Unal)
- 1 fresadora
- 4 operarios

El sistema de manufactura tiene la capacidad de producir 3 tipos de piezas mecanizadas (J_1 , J_2 y J_3), que requieren de una operación de ensamble con otra pieza que se adquiere externamente. Los recursos, trabajos y la información de producción se presentan a continuación

5.1 Definición de recursos

El sistema de manufactura propuesta está constituido por los siguientes recursos:

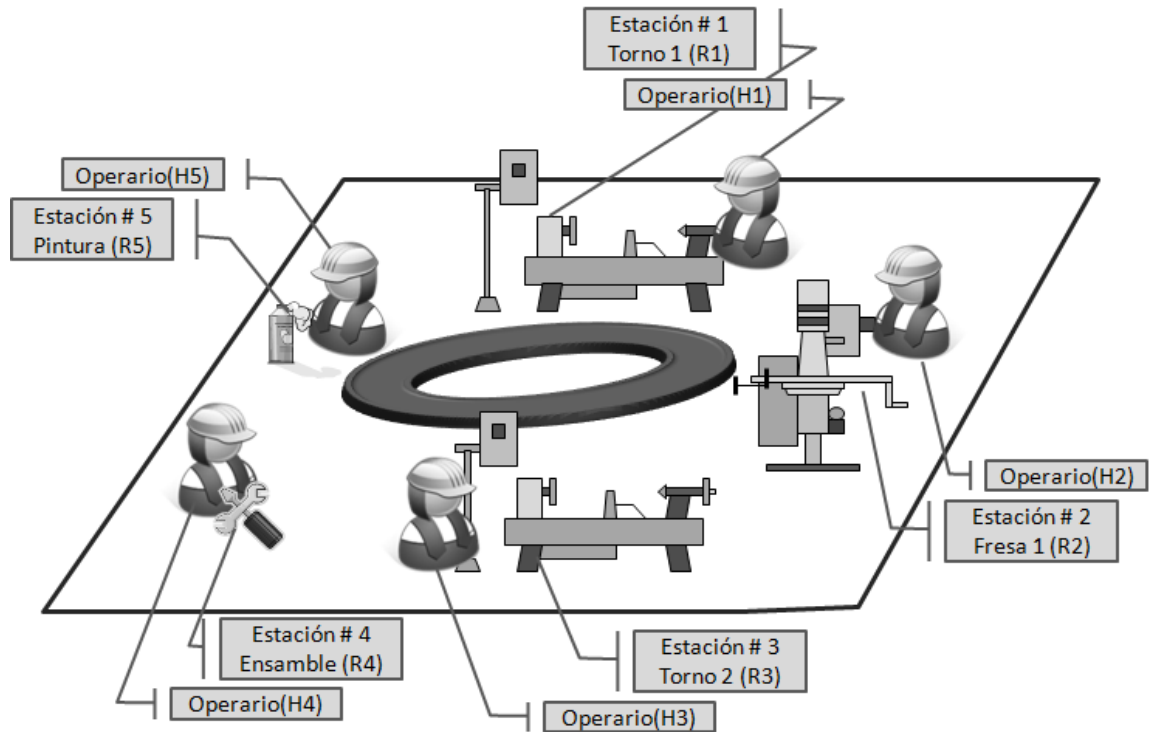
- R_1 → Torno (Referencia 1-“Torno Unal”)
- R_2 → Fresadora
- R_3 → Torno (Referencia 2-“Torno Unal”)

Con el objetivo de realizar un caso de estudio más cercano a la realidad de los sistemas de manufactura local, se definen recursos tipo operario, de este modo es posible validar el paradigma holónico en sistemas de manufactura caracterizados por presencia de islas de automatización.

- R_4 → Puesto-Operario Ensamble
- R_5 → Puesto-Operario Pintura

A partir de los recursos definidos se obtiene la distribución de planta que se representa en la Figura 49. El caso de estudio se plantea entonces con tres recursos tipo Máquina y cuatro recursos tipo Operario.

Figura 49. Distribución del sistema de manufactura para el caso de estudio



5.2 Definición de trabajos

De acuerdo a las capacidades y especificaciones de los recursos definidos para el sistema de manufactura, estos permiten proponer para la producción los siguientes trabajos:

5.2.1 J_1 Piñón + Buje (comprado externamente)

Figura 50. Trabajo J_1



Este trabajo (producto), está conformado por dos partes, una placa circular de un diámetro de aproximadamente 6 cm, y un ancho o espesor de 0.5 cm (insumo 1) y por una cilindro de un diámetro de aproximadamente 2 cm y un ancho de 2.5 cm (insumo 2).

Luego de realizar las operaciones de maquinado, la operación de ensamble se puede realizar a presión, de esta forma se unen las dos piezas que componen le trabajo j_1 .

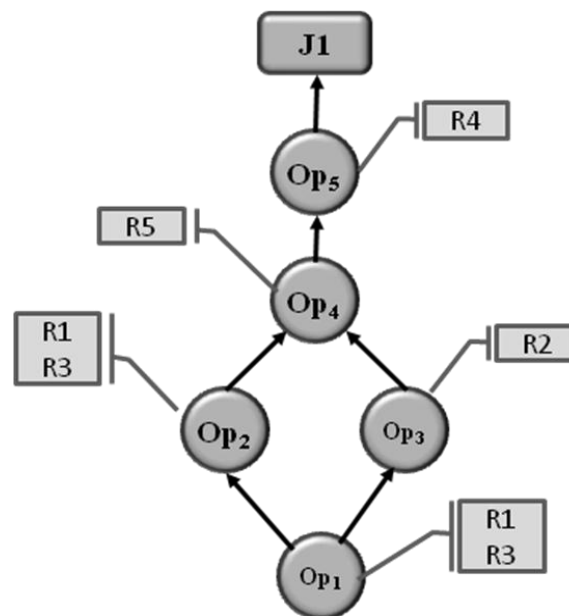
➤ **Propuesta de producción para el trabajo 1 (J_1)**

Tabla 4. Descripción del trabajo J_1

Operación	Nomenclatura	Descripción
1	Op_1	Corte insumo 1 (obtener piñón) → Tornos
2	Op_2	Torneado insumo 1 (obtener buje) → Tornos
3	Op_3	Perforación 2 → fresa
4	Op_4	Pintura → Puesto de pintura
5	Op_5	Ensamble → Puesto de ensamble

➤ **Grafo de precedencia para el trabajo 1 (J_1) - Plan Process**

Figura 51. Trabajo J_1



5.2.2 J2 Tapón Roscado + Tuerca (Comprado externamente)

Figura 52. Trabajo J₂



Este trabajo (producto), está conformado por dos partes, un cilindro de un diámetro aproximado de 3 cm y un largo de aproximadamente 4 cm (insumo 1) y una tuerca (insumo 2) que se va a comprar ya lista para ensamblar.

Luego realizar las operaciones de maquinado el proceso de ensamblado se puede realizar enroscando la tuerca en el tapón roscado que se obtiene de las operaciones definidas para este trabajo, de esta forma se unen los dos piezas.

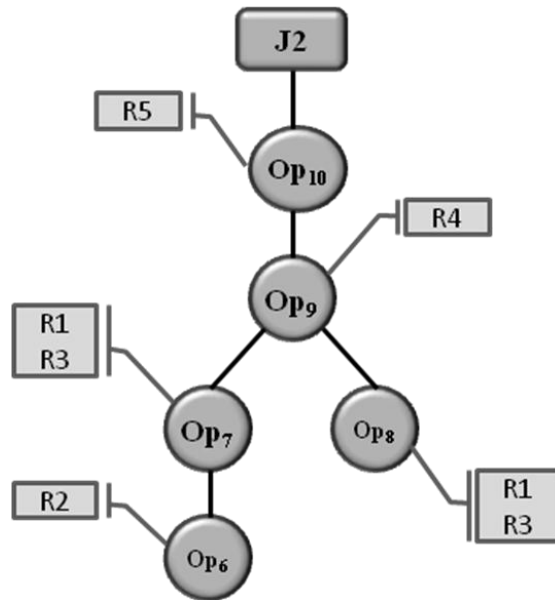
⇒ Propuesta de producción para el trabajo 2 (J2)

Tabla 5. Descripción del trabajo J₂

Operación	Nomenclatura	Descripción
1	Op ₆	Perforación insumo1 → fresa
2	Op ₇	Disminución de diámetro insumo1 → tornos
3	Op ₈	Roscar superficie exterior → tornos
4	Op ₉	Ensamblar → Puesto de ensamble
4	Op ₁₀	Pintura → Puesto de pintura

⇒ Grafo de precedencia para el trabajo 2 (J2) - Plan Process

Figura 53. Trabajo J₂



5.2.3 J2 Barra Tensora (parte superior se compra externamente)

Figura 54. Trabajo J₃



Este trabajo (producto), está conformado por dos partes, un cilindro de un diámetro aproximado de 2 cm y un largo de aproximadamente 6 cm (insumo 1) y una barra de un diámetro aproximado de 1 cm y de un largo de aproximadamente 15 cm (insumo 2)

Luego de que se le realizan las operaciones de maquinado la operación ensamble se realiza a presión, de esta forma se unen las dos piezas obtenidas con las operaciones de maquinado que se definen a continuación:

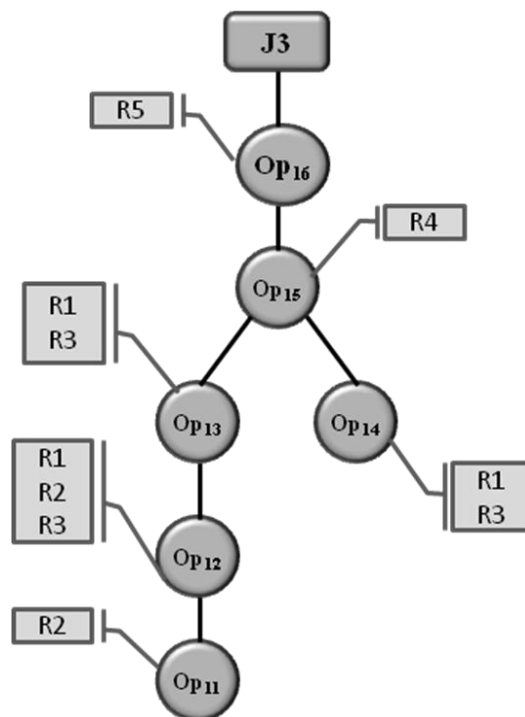
➤ **Propuesta de producción para el trabajo J3**

Tabla 6. Descripción del trabajo J₃

Operación	Nomenclatura	Descripción
1	Op ₁₁	Perforación insumo1 → fresa
2	Op ₁₂	Refrentado insumo1 → tornos y fresa
3	Op ₁₃	Devastar y cortar insumo1 → tornos
4	Op ₁₄	Devastar y cortar insumo2 → tornos
5	Op ₁₅	Ensamble → puesto de ensamble
6	Op ₁₆	Pintura → puesto de pintura

➤ **Grafo de precedencia para el trabajo 3 (J3) - Plan Process**

Figura 55. Trabajo J₃



5.3 Diagramas de proceso (OTIDA)

El Diagrama OTIDA (Operación – Transporte – Inspección – Demora - Almacenamiento) muestra la representación gráfica de la sucesión de hechos o fases que se presentan al aplicar el método o procedimiento de trabajo; indica las diversas actividades a que da lugar un trabajo o un producto en el sistema de manufactura (Heizer, 2004). La representación gráfica de la sucesión de actividades que resultan al aplicar el método o procedimiento de trabajo para el caso de estudio se representan en la Figura 56,

Figura 57 y Figura 58.

Estas indican las diversas actividades a que da lugar un trabajo o un producto en el sistema de manufactura, el recurso con capacidad de realizar cada trabajo y sus respectivos tiempos por unidad.

Figura 56. Diagrama de flujo del trabajo J₁

Descripción		RECURSOS	TIEMPO UNIDAD (min.)	O	T	I	D	A
TRABAJO	OPERACIONES							
J1	OP1	R1	5	X				
		R3	5	X				
	OP2	R1	19	X				
		R3	17	X				
	OP3	R2	1	X				
	OP4	R5	2	X				
	OP5	R4	2	X				

Figura 57. Diagrama de flujo del trabajo J₂

Descripción		RECURSOS	TIEMPO UNIDAD (min.)	O	T	I	D	A
TRABAJO	OPERACIONES							
J2	OP6	R2	1	X				
	OP7	R1	25	X				
		R3	22	X				
	OP8	R1	15	X				
		R3	13	X				
	OP9	R4	2	X				
OP10	R5	2	X					

Figura 58. Diagrama de flujo del trabajo J₃

Descripción		RECURSOS	TIEMPO UNIDAD (min.)	O	T	I	D	A
TRABAJO	OPERACIONES							
J2	OP11	R2	1	X				
	OP12	R1	3	X				
		R2	2	X				
		R3	3	X				
	OP13	R1	12	X				
		R3	11	X				
	OP14	R1	20	X				
		R3	18	X				
	OP15	R4	2	X				
	OP16	R5	2	X				

5.4 Supuestos

Para el modelamiento del sistema de producción se parte de los siguientes supuestos:

- Los tiempos de procesamiento de las operaciones son determinísticos.
- Los tiempos de alistamiento (setup times) son conocidos, y deben incluirse a los tiempo de tiempo procesamiento de acuerdo al recurso, como se presenta en la Tabla 7:

Tabla 7. Tiempos de preparación de los recursos

Recurso	Tiempo de preparación (Minutos)
Torno UNAL (R1)	4
Fresadora (R2)	2
Torno SENA (R3)	4
Puesto de Ensamble	2
Puesto de Pintura	2

- Una máquina puede ser utilizada para hacer operaciones diferentes.
- Una operación sólo puede ser realizada en una máquina.
- Cada máquina puede llevar a cabo sólo una operación a la vez.

- Las operaciones de un trabajo se realizan en diferentes máquinas en una secuencia que se debe respetar. Las máquinas no tienen fallas y siempre están disponibles durante el periodo de programación.
- No existen restricciones de precedencia entre trabajos. Solo se consideran restricciones de precedencia entre las operaciones de un mismo trabajo.
- Cada trabajo consiste de n operaciones distintas, una operación en cada máquina (no recirculación).
- Restricción de precedencia: una actividad puede realizarse solamente después de que un conjunto de actividades que la preceden hayan sido terminadas. Las estructuras de precedencia pueden ser representadas mediante gráficos de árbol y cadenas. (Pinedo 2001).
- Restricción de operarios: se cuenta con un número restringido de operarios para la realización de las operaciones.
- No existe recirculación en el job shop. Es decir, que un trabajo no puede visitar una máquina en más de una ocasión.
- No se consideran tiempos de transporte para los trabajos entre una y otra máquina.
- Todos los trabajos tienen la misma prioridad dentro del sistema.

5.5 Modelamiento del caso de estudio

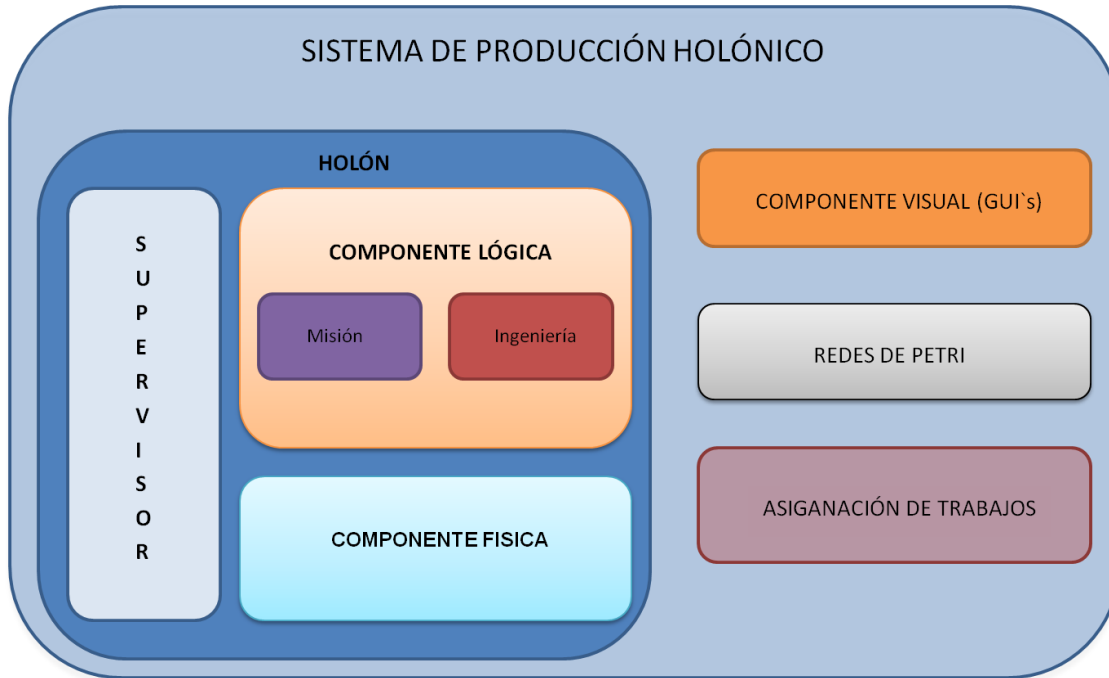
Basados en los atributos del paradigma holónico y Multi-Agente se desarrolla un aplicativo que los integre y permita al usuario de esta simular un sistema de control inteligente de sistema de manufactura holónico basado en UP.

Este desarrollo es un prototipo inicial dentro de una solución perteneciente a un proyecto de investigación en sistemas holónico de manufactura que realiza el grupo de Automática de la Universidad Nacional de Colombia GAUNAL.

5.5.1 Módulos del Aplicativo Desarrollado

Los módulos del aplicativo que permite tratar el control inteligente de sistema de manufactura holónico, integra principalmente los atributos de autonomía, reactividad, proactividad y cooperación. Para lograr esto, el aplicativo se ha diseñado hasta ahora a partir de la definición de cuatro módulos, entre ellos el más relevante es el módulo holón. Los módulos se ilustran en la Figura 59.

Figura 59. Módulos del aplicativo de control inteligente de sistemas de manufactura basados en UP



MÓDULO HOLÓN: Posee toda la dinámica investigada sobre el comportamiento del holón. Esta se caracteriza por ser una componente que puede ser manifestada un número considerable de veces, es decir, este módulo existe como tantos holones recurso tenga el sistema de manufactura que se trata.

- *Componente lógica:* Se destaca la manera en cómo debe exteriorizar atributos como la cooperación y la reactividad. En la primera, la cual puede ser una consecuencia lógica de la segunda, en una perturbación que afecte a largo plazo la producción de las misiones, este debe cooperar con otros holones que posean sus mismas competencias para no fallarle a toda la UPH, esto se realiza emulando el protocolo de Contract Net. Cuando esta cooperación no es exitosa, se informa al supervisor de la unidad de producción sobre el evento no deseado para que este tome decisiones posiblemente de reconfiguración. Por otro lado, tenemos 2 componentes, misión encargada de mantener actualizada las tareas a realizar en ese recurso e ingeniería, que se refiere al conocimiento y las competencias propias de ese recurso con parámetros como las operaciones que puede realizar, su costo, su duración. Cabe resaltar que el módulo Ingeniería definido en la arquitectura de control inteligente esta embebido en la componente física que se refiere a los movimientos en ejes x, y z, taladrado y así sucesivamente según la máquina.

- *Componente física:* Se refiere a la parte de conectividad entre las decisiones que puede tomar un holón y que se envían, en nuestro caso a PLC`s. Estos tienen un programa especial de operación según sus competencias pero lo importante es que estos también poseen una parte de supervisión para detectar fallas y enviarlas al software. A nivel de programación, se utilizó una librería de uso libre llamada OpenSCADA para la conectividad con dispositivos de este tipo.
- *Supervisor:* Este elemento es fundamental para la aplicación porque es allí donde se expresa la inteligencia y las reglas para la toma de decisiones frente a todos los eventos y perturbaciones. A grandes rasgos podemos mencionar reglas como cuando iniciar una misión en la máquina, con que unidades cooperar en caso de perturbaciones, como reconfigurar las misiones de varias máquinas y actualizar los estados tanto del recurso y de cada misión. Los estados que pueden ser aplicados a un holón estándar y que fueron utilizados se presentan en la Tabla 8.

Tabla 8. Estados del holón recurso

ESTADOS MISION	ESTADOS RECURSO
Sin compromiso	Disponible
Comprometida	Arranque
Avance normal	Normal
En riesgo	Degradado
Incumplida	Falla
Completada	Mantenimiento
Cancelada	

MÓDULO REDES DE PETRI: Está orientado a la unificación de las redes de Petri y el aplicativo. Contiene diferentes clases que toman archivos con extensión .cpn y logra extraer un espacio de estados reducido, considerando que en el modelo de redes de Petri que aquí se considera aún tiene lugares y eventos solo de interés en el modelamiento.

La agregación de estas redes al sistema, está encaminado a la exploración de todas las posibles rutas para realizar una pieza, es por ello que las redes modeladas se refieren a la realización de trabajos y esto se convierte en el primer paso para ejecutar la aplicación. Esto se podría describir como la ingeniería de la UP, el conocimiento de lo que puede hacer basado en operaciones y recursos. Luego, todo esto se convierte en un vector de posibles rutas para hacer un trabajo en particular que se almacena para luego ser llamado cuando entra una orden.

MÓDULO DE ASIGNACIÓN: El aplicativo dispone de un módulo flexible que permite remover o añadir los algoritmos más útiles para la organización, en este caso de estudio solo se utilizó SPT (Shortest Path Processing).

Esta unidad entra en funcionamiento cuando ingresa una orden de producción, que trae información acerca de las cantidades de cada trabajo a producir y la fecha de vencimiento, esta es una información clave considerando que el algoritmo heurístico de programación la requiere para una adecuada asignación.

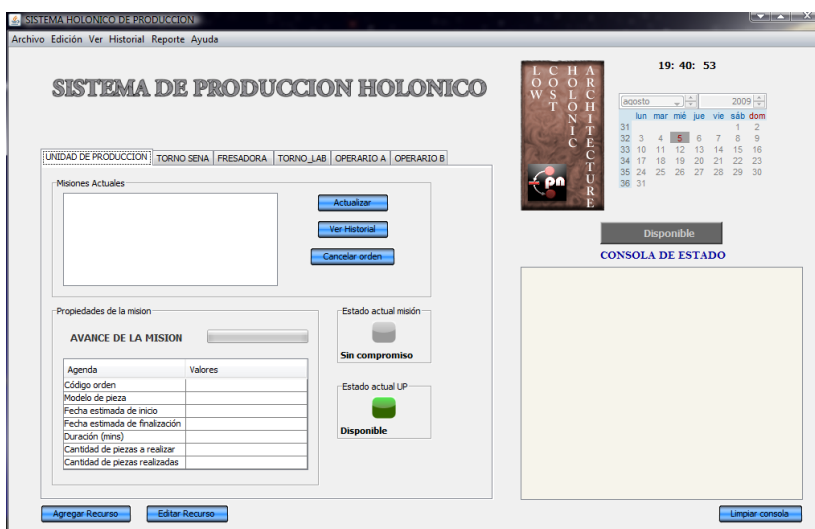
Si la orden de producción referencia un trabajo que el sistema de manufactura está en capacidad de fabricar, la red de Petri de este producto se encuentra almacenada en el módulo ingeniería. En consecuencia, a través de la PN se obtiene el vector de rutas de cada trabajo para su fabricación y se determina cual es la programación con la mejor solución en los recursos involucrados, teniendo en cuenta las agendas de los recursos actualizadas en cada vector.

Finalmente, la asignación de los trabajos de la orden de producción negociada son incluidos en las agendas de los recursos de la UPH. El supervisor de la UPH realiza el monitoreo de estos trabajos a través de la dinámica del modelos en PN del holón recurso. Puede existir un hecho particular cuando al explorar y planificar las distintas rutas no se logre cumplir con la fecha de vencimiento, en este caso se advertirá sobre la situación para que el cliente encargado de dar esta información reasigne, ampliando la fecha o descartando la orden totalmente.

MÓDULO VISUAL (GUI's): Este módulo comprende todas las interfaces gráficas que permite la integración de la dinámica del control inteligente y la información entregada a las personas que manejan el aplicativo. El aplicativo tiene una ventana principal que se muestra en la

Figura 60, la cual consta de varias pestañas principales las cuales representan los recursos, aclarando que la primera pestaña es la unidad de producción: es importante resaltar que cada recurso tiene la misma estructura, un panel de misiones, un panel de estados y un panel de información detallada de la misión. La interfaz principal contiene una consola de estado donde se muestra la distinta información entre recursos, tal como negociación y supervisión. En la parte inferior se encuentra un botón añadir recurso el cual despliega como se configurara este holón, con sus respectivas competencias; y un botón editar recurso para modificar la información ya asociada a este.

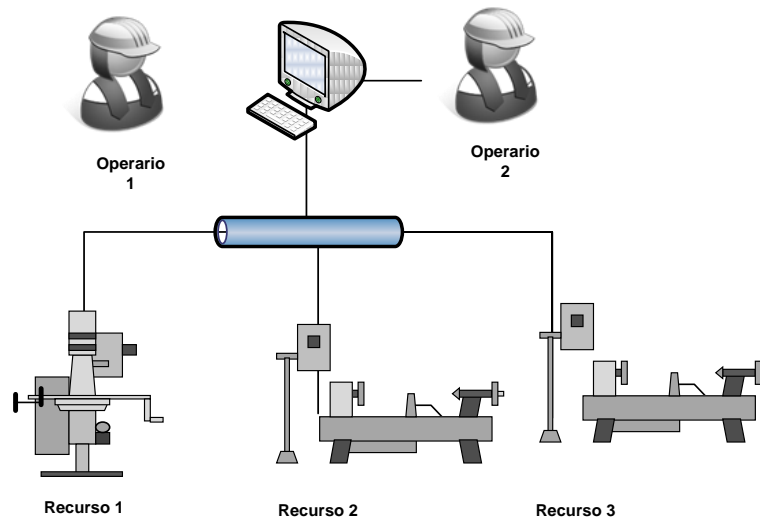
Figura 60. Interfaz Gráfica principal



5.5.2 Arquitectura del Laboratorio

La implementación real con las máquinas y los operarios se manifiestan en la Figura 61 donde existe una comunicación serial entre el computador donde se encuentra el aplicativo y los tornos y la fresadora. Como los operarios recurso también se controlan con la estructura similar a las máquinas, estos estarán atentos a sus agendas en la pantalla del PC. La conexión entre máquinas y el software también ha sido probado sobre una red LAN, con las mismas bondades que la conexión serial, esto para futuros trabajos empresariales donde los recursos estén sobre una red Ethernet. A continuación se explicara las funcionalidades básicas del software construyendo piezas reales en el laboratorio.

Figura 61. Arquitectura de la planta prototipo



5.5.3 Aplicación y Funcionalidades de la Herramienta

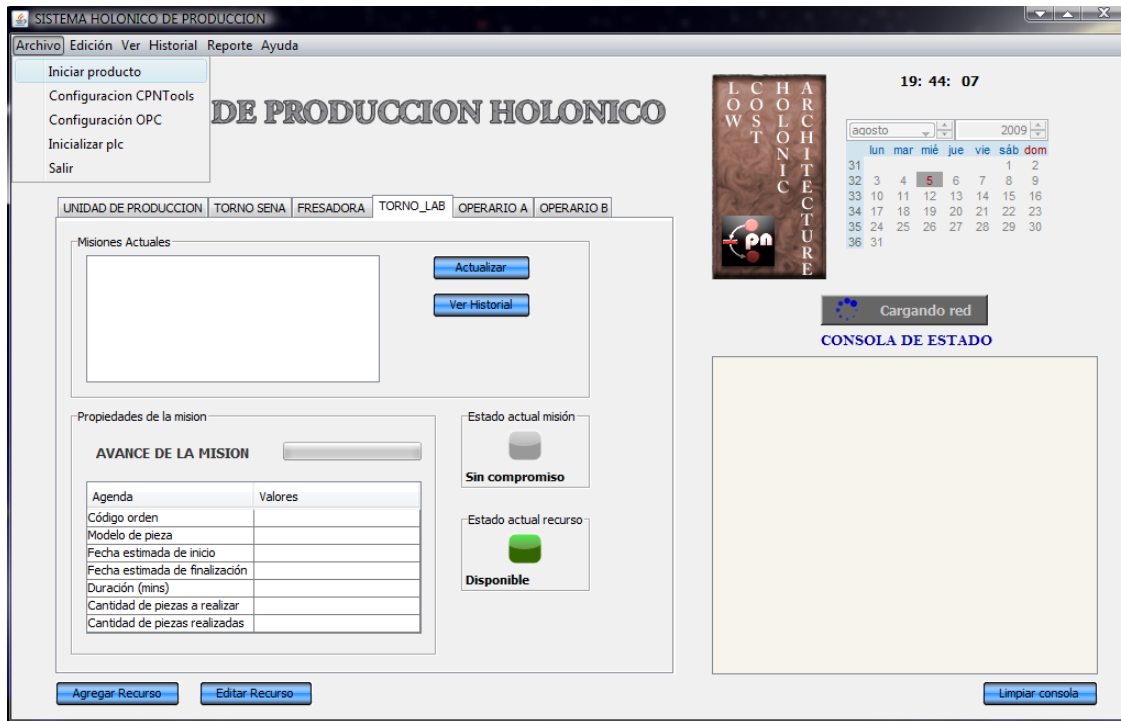
A continuación se explicara las distintas opciones que posee la barra del menú del aplicativo, donde se resalta la propiedad de recursividad que existe entre las distintas pestañas y sin importar la magnitud del recurso este se caracteriza por unas misiones, unos estados y unas competencias.

Menú Archivo: En el menú que se ilustra en la Figura 62 se encuentran las configuraciones iniciales que deben realizarse para el correcto funcionamiento del aplicativo las cuales son:

- **Iniciar producto:** Haciendo clic en este ítem se procede a definir una orden de producción, sin embargo, es necesario primero agregar las redes de Petri en el módulo ingeniería.
- **Configuración CPN Tools:** En el apartado de módulo de redes de Petri descrito anteriormente se hablaba de un etiquetado especial pues bien, esto abre dos campos de texto donde la primera se refiere al nombre de la pestaña que quiere ser analizada y el segundo campo es el nombre que las transiciones claves tienen en común, es decir que en el modelo CPN se escribe una palabra clave a cada transición que quiera ser clave para alcanzar un espacio de estados.
- **Configuración OPC:** Tal vez es el segundo paso que se lleve a cabo porque permite la conectividad con las diferentes máquinas, (PLC`s), la cual consiste de la autenticación con el host local, es una configuración de permisos que por defecto utiliza la librería de OpENSCADA para la comunicación con dispositivos. Por ende, si se encuentra en un ambiente Windows, este debe poseer un usuario y una contraseña con permisos de administrador.

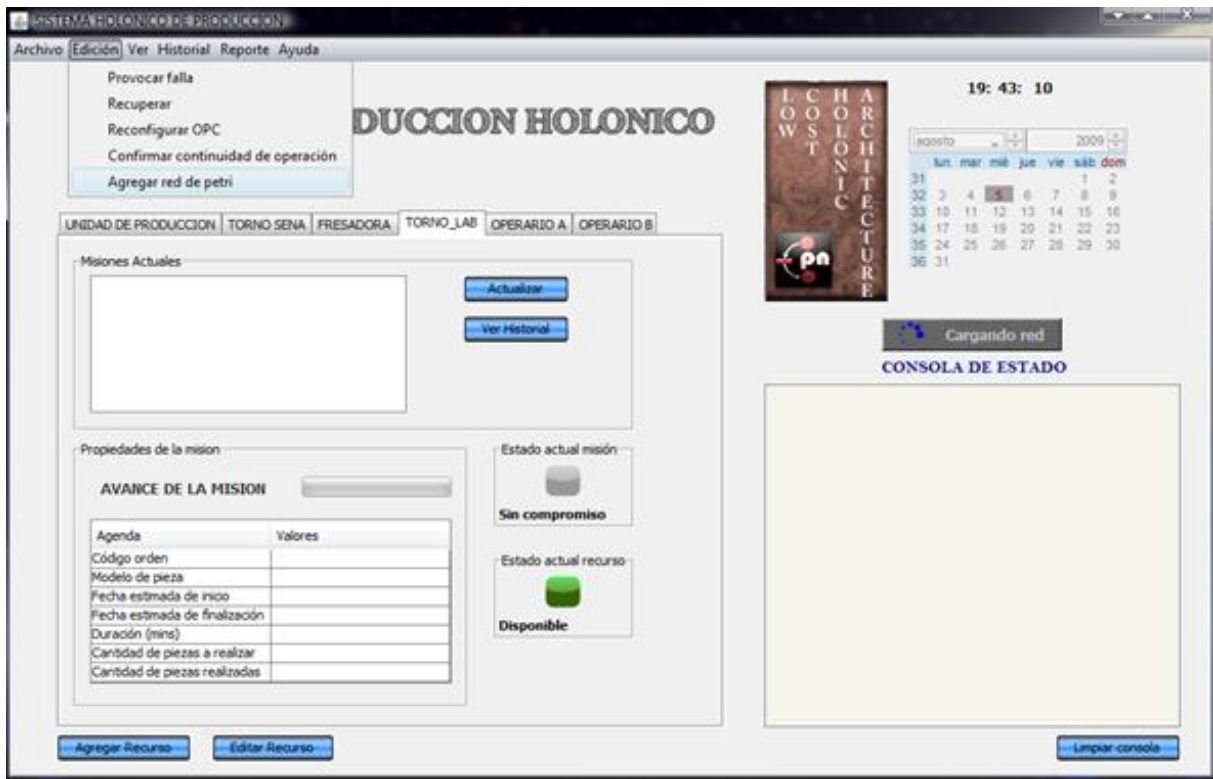
- **Iniciar PLC:** Este paso se puede ejecutar si previamente ya se ha configurado el PLC, el cual se orienta a inicializar las variables que utiliza a cero y evitar posibles envíos de información erróneos a los dispositivos.

Figura 62. Menú archivo



- **Menú Edición:** Se encarga de algunas funcionalidades cuando el sistema ya está en marcha para generar perturbaciones y añadir conocimiento a la UPH, los menús asociados se encuentran en Figura 63.
- **Provocar falla:** Este ítem despliega los recursos configurados actualmente para seleccionar de forma manual aquellos que quieren entrar en falla y ver la reactividad en los holones involucrados. Cabe mencionar que los recursos en estado de falla no participan en la negociación de nuevas misiones.
- **Recuperar:** Despliega los recursos que actualmente están en mantenimiento o falla para habilitarlos de nuevo a su comportamiento habitual.
- **Reconfigurar OPC:** Este elemento es utilizado para darle un reinicio a las variables del PLC, dado que algunas veces los dispositivos son sobrecargados por lo que se hace necesario esta utilidad.
- **Confirmar continuidad de operación:** Este elemento es utilizado cuando existe una perturbación y a pesar de que existe una advertencia de incumplimiento, el supervisor humano puede ayudar a tomar la decisión de que continúe con la operación, porque este ha analizado por ejemplo que tendría más costo una cancelación o una reconfiguración.
- **Agregar red de Petri:** Al dar clic, se despliega una ventana de selección de archivos para elegir un archivo .cpn, inmediatamente hace la evolución de la red y almacena el espacio de estados. Como se dijo previamente, es con esta acción que debería empezar la aplicación.

Figura 63. Menú edición



Menú Ver: Esta orientado a la amigabilidad del software para posicionarse en la vista del recurso seleccionado, esta opción solo mostrara los recursos configurados en el momento.

Menú Historial: Según los recursos actuales del aplicativo, cada uno puede contener un historial de las operaciones realizadas, tiempos de duración, costos asociados y perturbaciones acontecidas.

Menú Reporte: Esta ajustado para desplegar bien la información de la unidad de producción acerca de su desempeño y órdenes completadas y las fallas que han ocurrido en los distintos recursos. Esta información se puede ver tanto en la Figura 64 como en la Figura 65.

Figura 64. Menú reporte

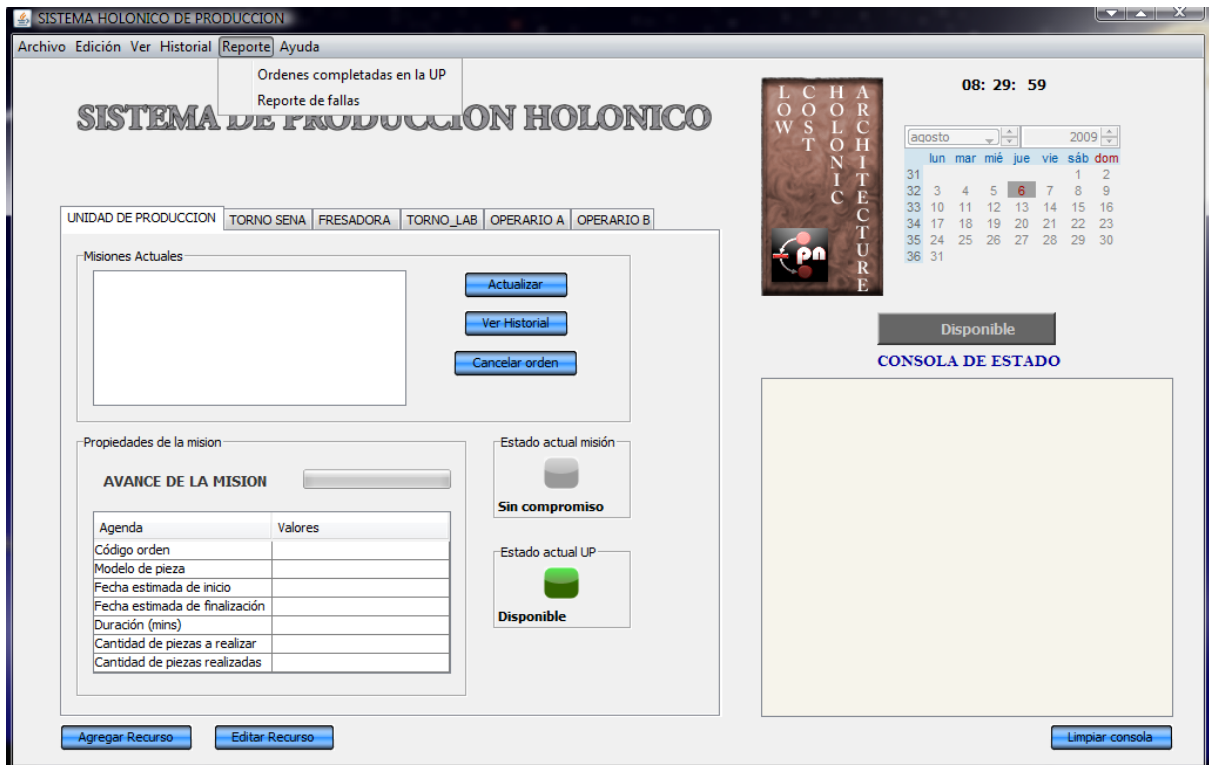


Figura 65. Reporte de falla

Recurso	Hora de falla	Cod misión	Tipo de error
TORNO_LAB	07/06/09 08:23	054	Desajuste de máquina
FRESA	05/07/09 12:25	120	Desconexión

CAPÍTULO 7

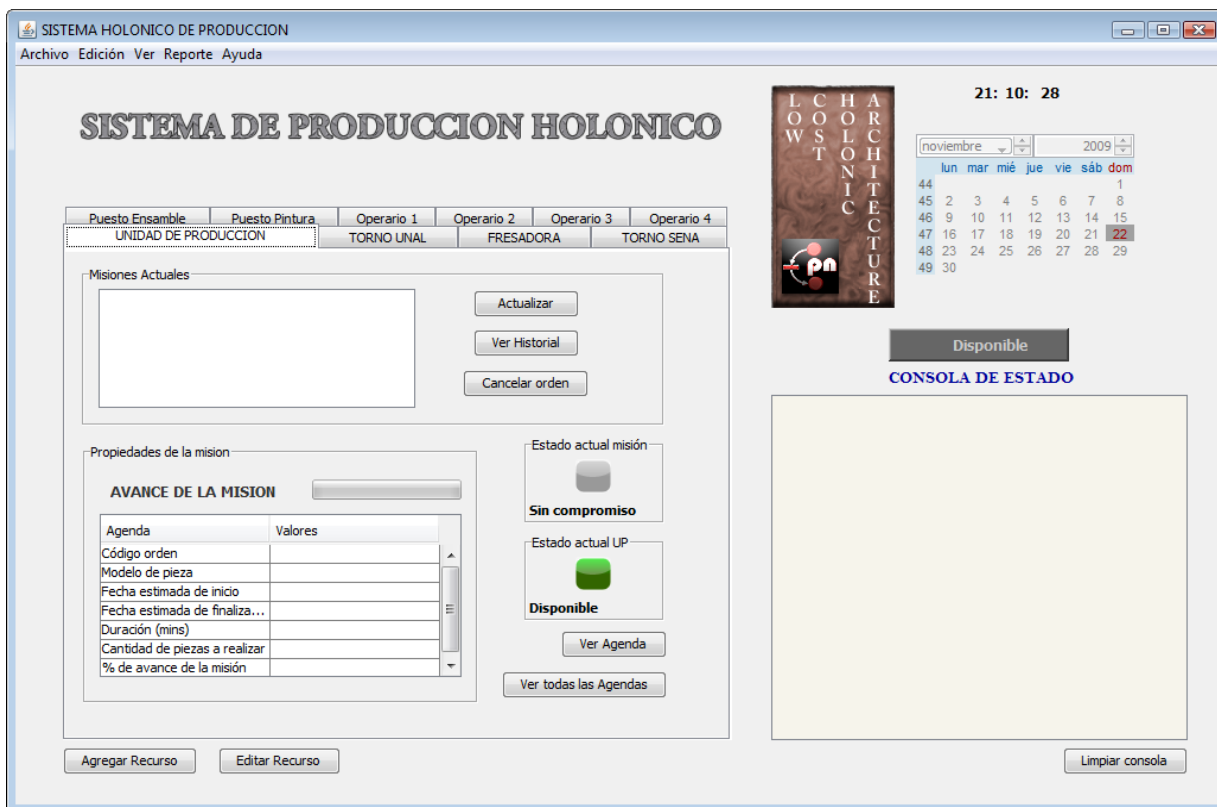
PRUEBAS DEL SISTEMA DE PRODUCCIÓN HOLÓNICA

6.6 Comportamiento del sistema de producción holónico

6.6.1 Creación y Manufactura de órdenes de producción

Una vez realizado los pasos previos y ajustado el software al caso de estudio descrito, es posible proceder con la creación de una orden de producción, en este momento la unidad de producción estará disponible tal como se ilustra en la Figura 66.

Figura 66. Sistema de producción holónico en estado disponible



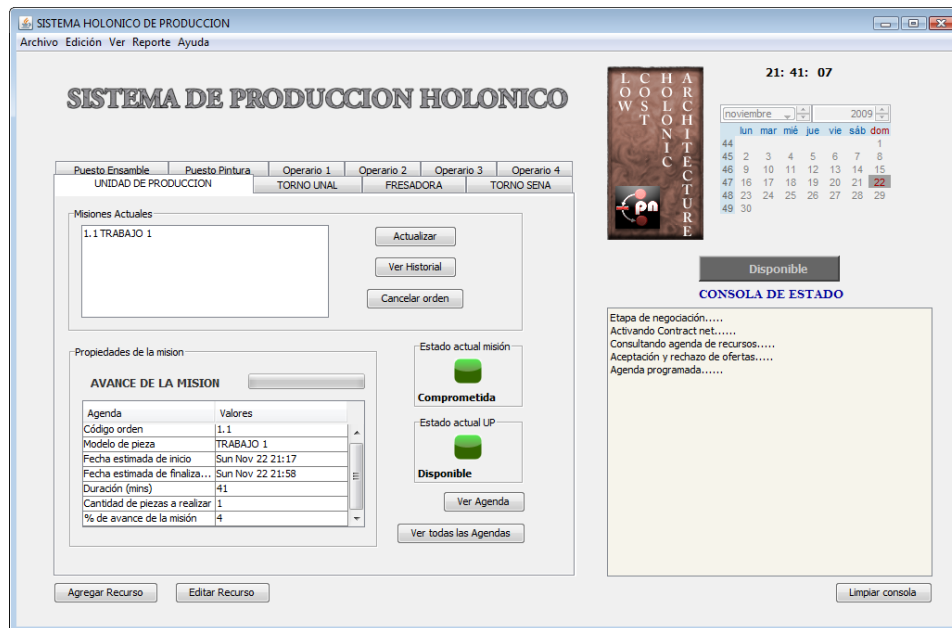
Partiendo del estado disponible del sistema de producción holónico, se habilita la opción de “Iniciar producto” y se ingresa la orden agregando la fecha de vencimiento de esta, tal como se ilustra en la Figura 67.

Figura 67. Creación de una orden



Al realizar esto, la próxima información desplegada en pantalla es la actualización de agendas de los recursos que participarán en la manufactura del trabajo, de lo contrario desplegará la advertencia de que no se pudo crear la orden, esto puede deberse a la no disponibilidad de agenda de los recursos para manufacturar la orden ingresada.

Figura 68. Agenda programa de la UP



La Figura 68 ilustra el estado de “Agenda programada” en la consola de estado del sistema de producción holónico, luego de implementar el protocolo Contract Net para la negociación de las ordenes de producción para el trabajo J1. El sistema de producción holónica presenta las misiones programadas en la margen superior izquierda, específicamente en el recuadro de “Misiones Actuales”.

La programación obtenida por el sistema se ilustra en la Figura 69 y adicionalmente la Figura 70 presenta el resumen de esta programación para la Unidad de Producción.

Figura 69. Diagrama de Gantt de la programación del trabajo J1 por el sistema de producción holónico

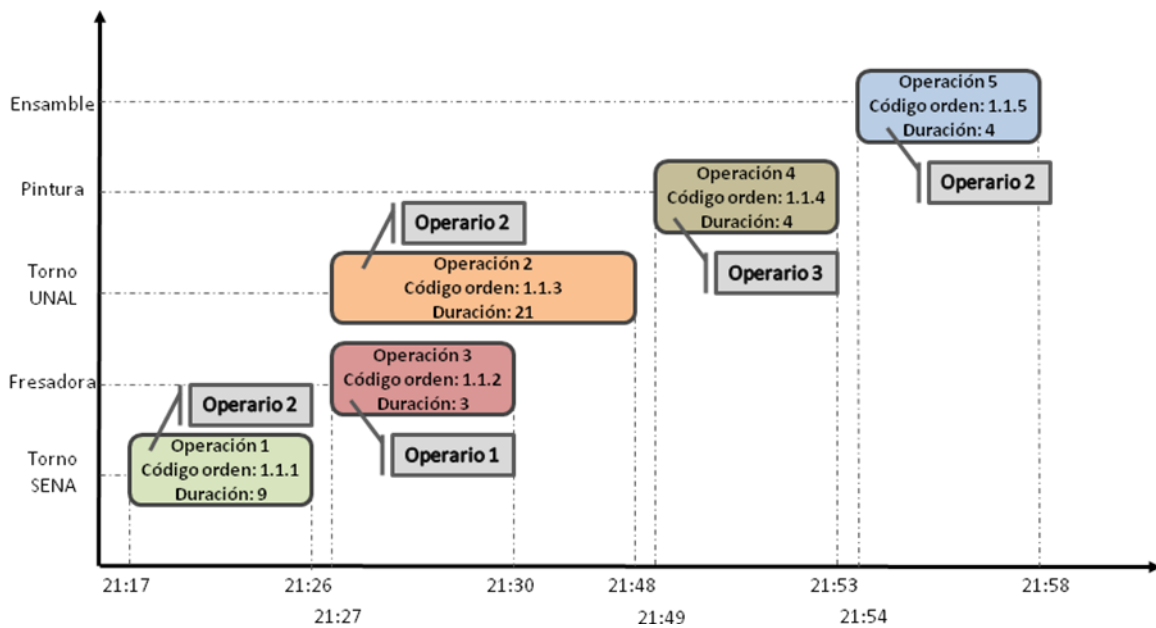


Figura 70. Información de la programación realizada por el sistema de producción holónico para el trabajo J1.

SISTEMA DE PRODUCCION HOLONICO

Archivo Edición Ver Reporte Ayuda

21: 41: 07

noviembre 2009

	lun	mar	mié	jue	vie	sáb	dom
44							1
45	2	3	4	5	6	7	8
46	9	10	11	12	13	14	15
47	16	17	18	19	20	21	22
48	23	24	25	26	27	28	29
49	30						

Disponibile

CONSOLA DE ESTADO

Etapa de negociación.....
 Activando Contract net.....
 Consultando agenda de recursos.....
 Aceptación y rechazo de ofertas.....
 Agenda programada.....

AVANCE DE LA MISION

Agenda	Valores
Código orden	1.1
Modelo de pieza	TRABAJO 1
Fecha estimada de inicio	Sun Nov 22 21:17
Fecha estimada de finaliza...	Sun Nov 22 21:58
Duración (mins)	41
Cantidad de piezas a realizar	1
% de avance de la misión	4

Estado actual misión: **Comprometida**

Estado actual UP: **Disponibile**

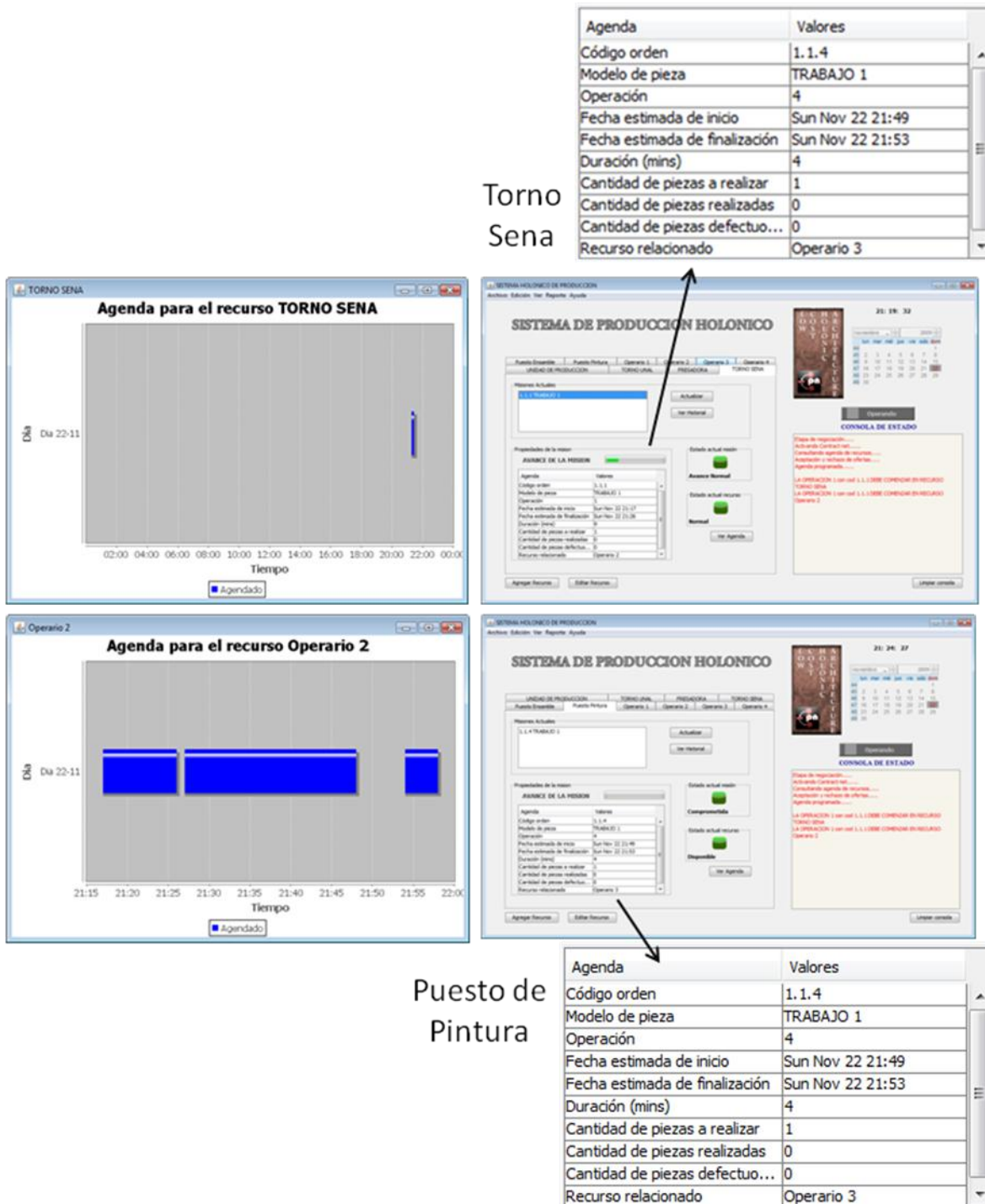
Ver Agenda

Ver todas las Agendas

Agregar Recurso Editar Recurso Limpiar consola

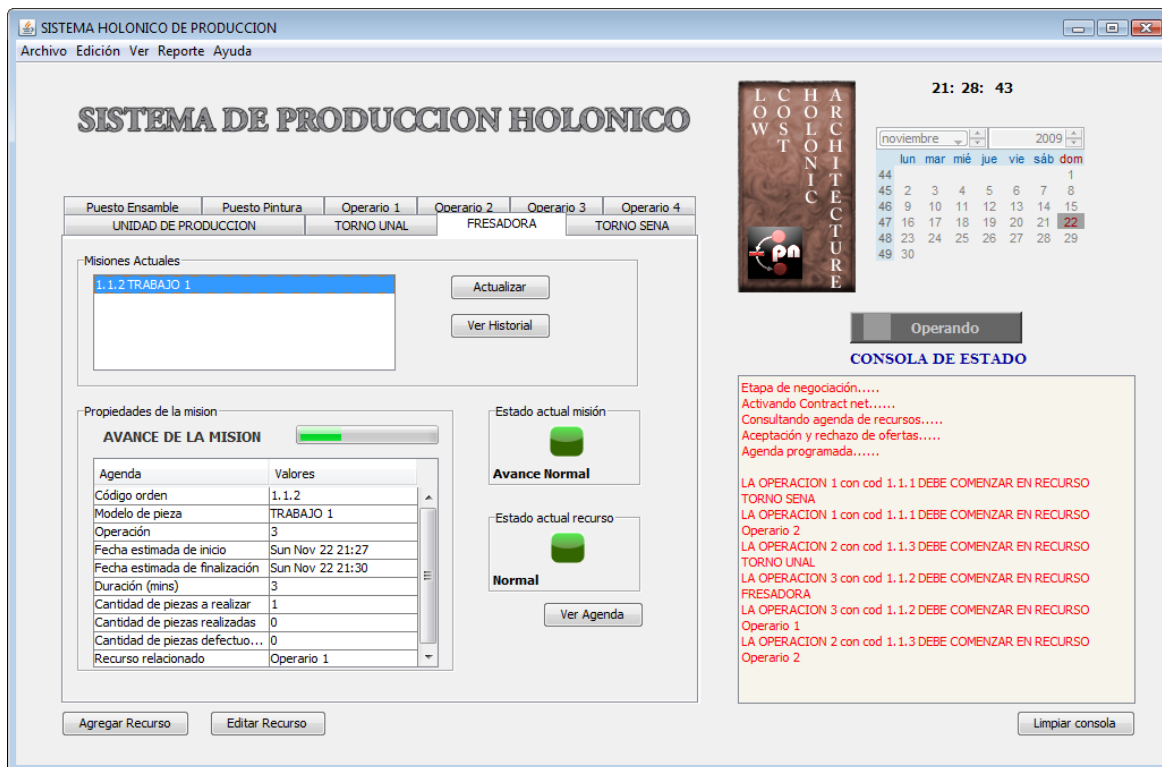
La Figura 71 muestra la programación de alguno de los recursos y las agendas ilustradas por el sistema de producción holónico.

Figura 71. Información de la programación de los holones de la UP.



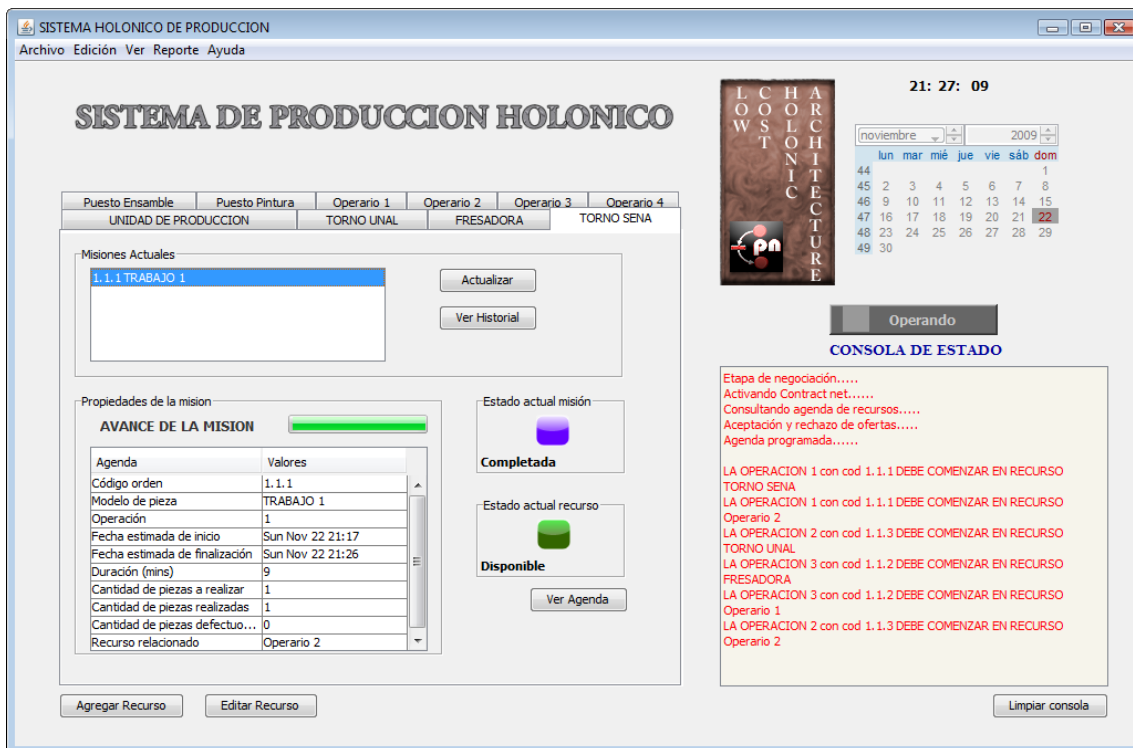
A medida que las órdenes de producción van iniciando a través del tiempo, el sistema de producción holónico va indicando esta situación en la consola de estado, tal como se ilustra en la Figura 72.

Figura 72. Ordenes de trabajo iniciadas en la unidad de producción



Una vez una orden de producción es completada por un holón de la UP, el sistema de producción holónico actualiza el “Estado actual de la misión”, tal como se ilustra en la Figura 73. Esto fado que el supervisor del sistema monitorea el inicio, desarrollo y finalización de cada operación. También, cualquier detalle realizado en la ejecución de los trabajos programados en el Sistema de Producción Holónico se presenta en la consola de estado como lo muestra la Figura 73.

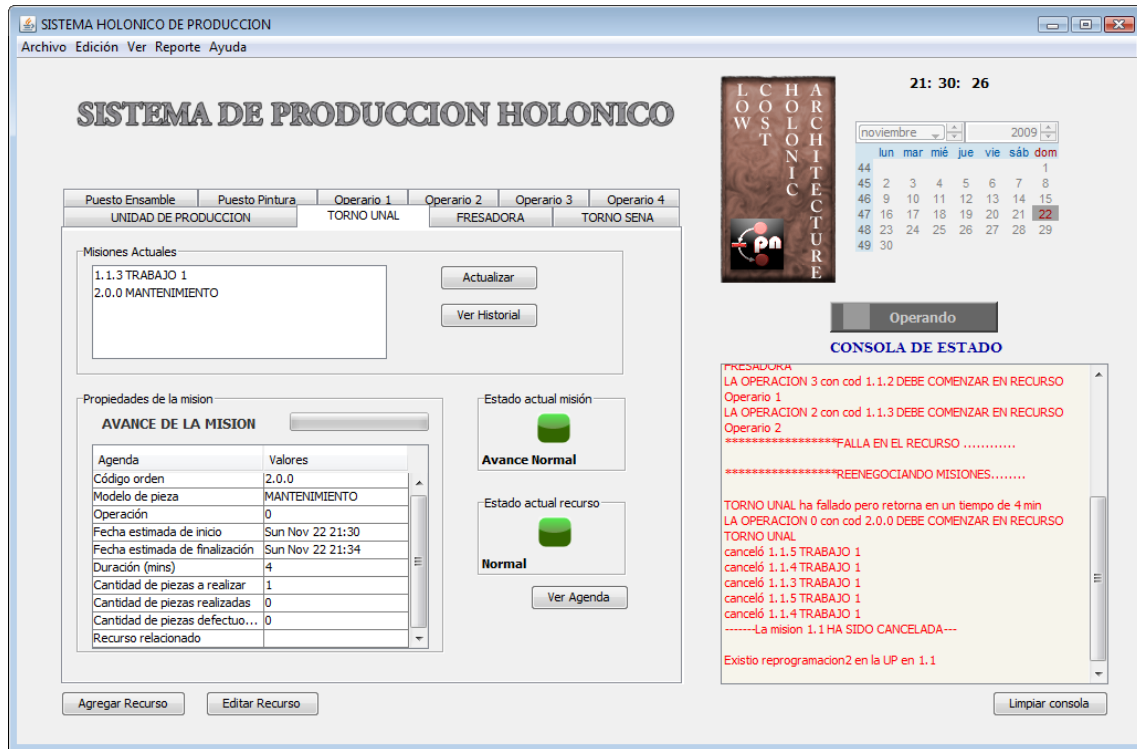
Figura 73. Orden de producción completada en el holón Torno SENA.



6.6.2 Existencia de falla en el sistema holónico

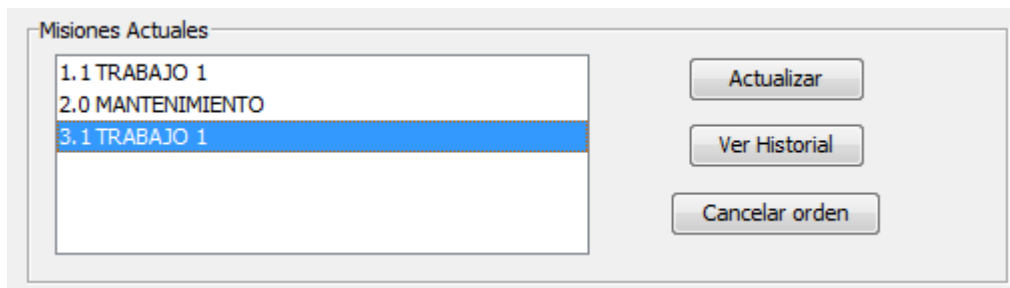
El último suceso importante es cuando el sistema automáticamente detecta una falla o de manera manual se procede a este tipo de perturbación, es entonces que la consola de estado muestra comentarios indicando falla en un recurso y la fase de renegociación de operaciones, como se muestra en la Figura 74. Al final, desaparecen las misiones comprometidas en el recurso en falla y se traspasan por negociación a otro que posea las mismas competencias, los estados tanto de misiones como del recurso físico se actualizan. Adicionalmente es necesario programar una operación de mantenimiento mediante la cual se realiza la reparación del recurso de falla, esto se ilustra en la Figura 74 en el marco de “Misiones Actuales”.

Figura 74. Falla en el recurso Torno



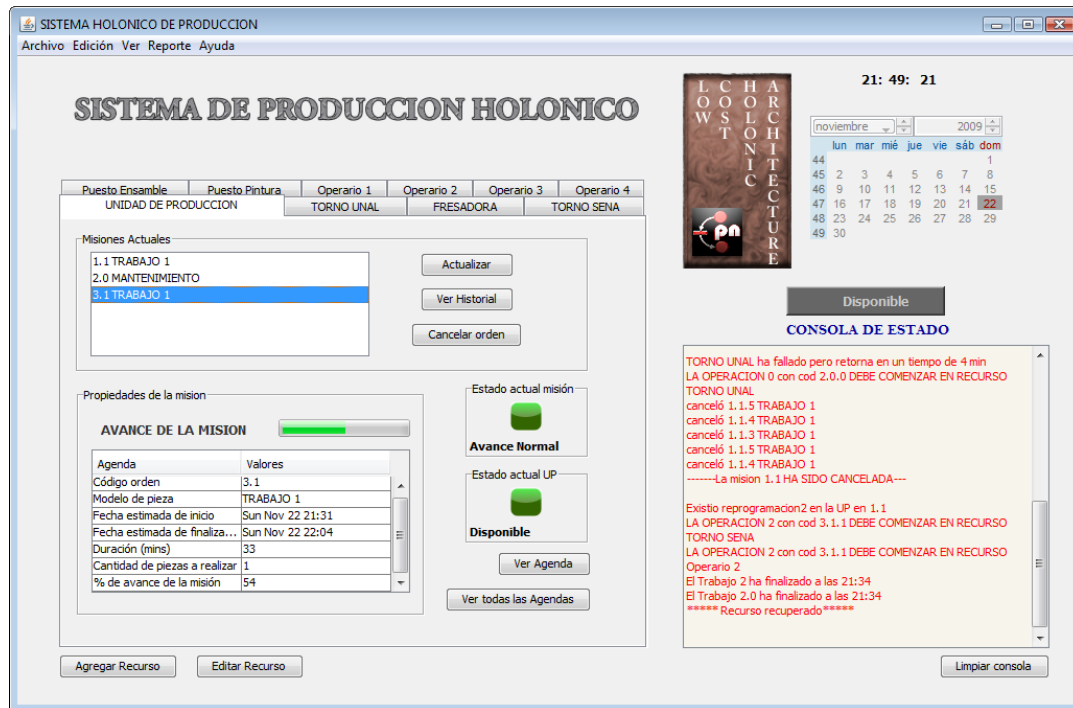
En el momento en el que el sistema de producción holónico realiza la reprogramación, se actualiza el marco de Misiones Actuales, en este se incluye el trabajo con las operaciones pendientes por manufacturar luego de la falla, tal como se ilustra en la Figura 75.

Figura 75. Misiones Actuales de la UP posterior a una perturbación de falla en la UP.



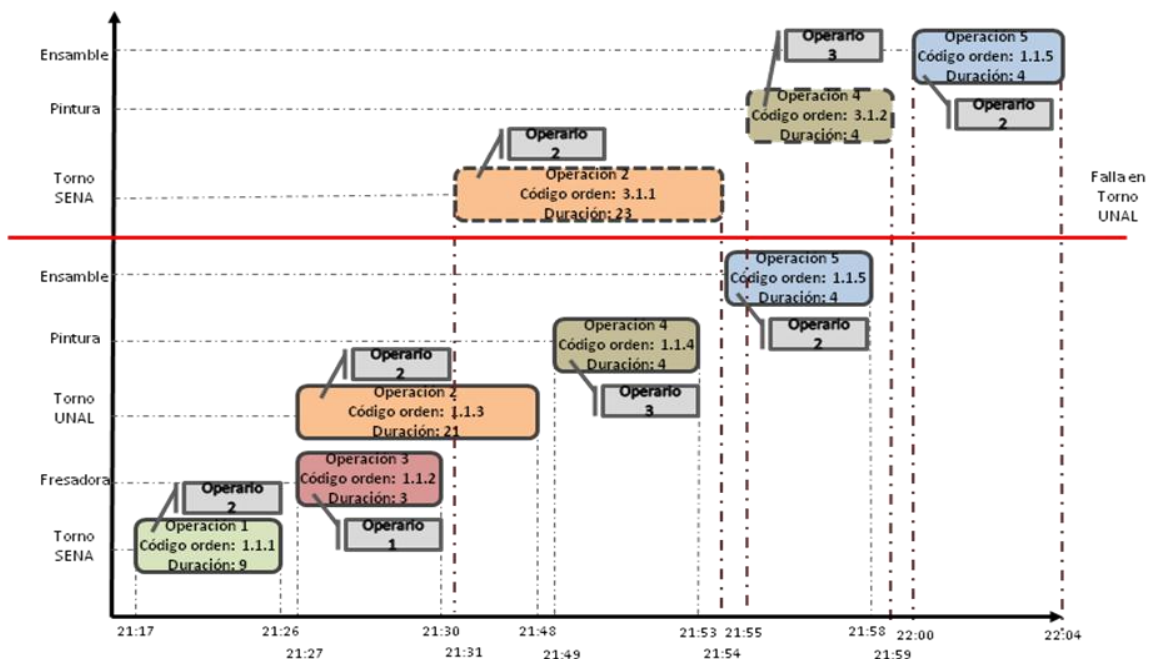
La unidad de producción se actualiza las propiedades de la misión, una vez el sistema de producción holónico ha realizado la renegociación, tal como se ilustra en la Figura 76, donde la nueva hora de terminación es a las 22:04 y la duración del trabajo con las operaciones restantes es 33 minutos.

Figura 76. Información de la misión actualizada posterior a una falla en el recurso.



La Figura 77 muestra la nueva programación del trabajo J1 una vez realizada la reprogramación de la UP por el sistema de producción holónico.

Figura 77, Diagrama de Gantt de la UP posterior a la falla en el holón Torno UNAL.



El sistema de producción holónico ha permitido realizar la validación del modelo de control inteligente, en cuanto a su capacidad de reactividad, autonomía y cooperación, tal como se ha presentado con el contenido de este capítulo. Este sistema permite validar además, la agilidad de la UP ante la existencia de perturbaciones, en cuanto a su capacidad de establecer un nuevo programa de producción, sin necesidad de que UP este comprometida en paros de producción. La posibilidad de edición de los recursos y la oportunidad de cargar los modelos en redes de Petri de los productos, hacen que el sistema de producción holónico sea escalable a cualquier tipo de producto acorde a las competencias de los recursos que se configuren para su producción en la UP.

CONCLUSIONES Y TRABAJOS FUTUROS

- Una arquitectura de control holónico basada en unidades de producción holónicas permite un desarrollo de un sistema de control inteligente de forma modular y autonomía operacional, dado que la UPH divide el proceso en unidades de producción que son unidades menos complejas y facilitan el estudio en cuanto a su autonomía, cooperación y reactividad.
- La agenda contiene las órdenes de trabajo en función del tiempo, evaluadas y aceptadas para ser procesadas en el holón recurso. Esta representa el registro de capacidad, historial y de trazabilidad del recurso. Por tanto, la agenda es el centro de la inteligencia y comportamiento del holón recurso. Esto, considerando que la agenda es un instrumento con base en el cual el holón toma decisiones.
- La utilidad razonable del CNP es definir los criterios de negociación en la reprogramación dinámica del sistema, permitiendo agilidad y reactividad en la programación y reprogramación del objetivo de producción ante perturbaciones del sistema.
- La mayoría de las arquitecturas de control de manufactura, tanto holónicas como Multi-Agente consideran todos los elementos relevantes de la arquitectura como holones o agentes, la arquitectura de control inteligente propone que la única componente que cumple con la definición de holón son los recursos inteligentes del sistema de manufactura. Los componentes como misión e ingeniería son considerados como módulos que agrupan información y almacenan un estado del comportamiento y estado de la arquitectura.
- El árbol de alcanzabilidad como método de análisis para las redes de Petri, ha sido utilizado en este trabajo para dos fines: la obtención de rutas para la asignación de trabajos en los holones recurso de la UPH y la validación y verificación de propiedades estructurales y de comportamiento del holón recurso y la misión del holón.
- En el desarrollo de la tesis de investigación, se presentaron dificultades en relación a la elección del método de reprogramación a ser implementado en el mecanismo de negociación, dado que los autores en este tema no presentaban en los artículos científicos, información completa que permitiera el entendimiento del algoritmo del método que se proponía.

Adicionalmente, fue dispendiosa la definición de los estados del holón recurso y la misión del holón, de tal manera que estos modelos consideraran los atributos de control inteligente tomados desde los paradigmas Holónicos y Multi-Agente, a la vez que debían considerarse requisitos desde el control de sistemas a eventos discretos.

- Una propuesta de trabajo futuro, deberá estar orientada a definir y caracterizar la capacidad de aprendizaje de los holones recurso a partir de la agenda de producción. Adicionalmente, el aplicativo “Sistemas de Producción Holónico” tiene posibilidades de mejoramiento de la presentación de la información de supervisión de los holones recurso y de la misión del holón. También, pueden realizarse trabajos alrededor de la presentación gráfica de la descomposición de las órdenes de trabajo en sus operaciones.

REFERENCIAS BIBLIOGRÁFICAS

- Almeida, a., ramos, c. And silva, s. (1999). Dynamic scheduling of manufacturing orders: a decision support system approach. Proceedings of the 1999 IEEE international symposium on assembly and task planning.
- Antsaklis, P. J. (1994). Defining Intelligent Control", Report of the Task Force on Intelligent Control, P.J Antsaklis, Chair, IEEE Control Systems Magazine, pp. 4-5 & 58-66.
- Antsaklis, P. J. (1994). Defining Intelligent Control. Report of the Task Force on Intelligent Control, P.J Antsaklis, Chair, IEEE Control Systems Magazine, pp. 4-5 & 58-66.
- Bellifemine, F., Caire, G., Poggi, A. and Rimassa, G. (2008). JADE: A software framework for developing multi-agent applications. Lessons learned. Information and Software Technology. Volume 50 , Issue 1-2. ISSN:0950-5849.
- Bongaerts, L. (1998). Integration of Scheduling and Control in Holonic Manufacturing Systems. Ph.D. Thesis PMA/K.U.Leuven
- Breugst, Markus; Magedanz, Thomas. On the Usage of Standard Mobile Agent Platforms in Telecommunication Environments. In: 5th Int. Conference on Intelligence in Services and Networks (IS&N), 5., 1998, Bélgica. Anais... Bélgica: Springer Verlag, 1998, p. 275-286.
- Cai, Z. (1997). Intelligent control: principles, techniques and applications. World Scientific, ISBN 9810225644. págs 450 páginas
- Caillaud, B., Lavagno, L. and Xie, X. (2002). Synthesis and Control of Discrete Event Systems. Kluwer Academic Publishers.
- Capkovic, F. (2000). Modelling and Control of Discrete Event Dynamic System. Basic Research in Computer Science. Brics Report Series. ISSN 0909-0878.
- Capkovic, F. and Capkovic, P. (2001). Intelligent Control Synthesis of Manufacturing Systems. In Monostori, L., Vancza, J., A., Ali, M. (Eds.): Engineering of Intelligent Systems. Springer Verlag, Berlin-Heidelberg-New York, Lecture Notes in Artificial Intelligence, Vol. 2070 of the Lecture Notes in Computer Sciences, pp. 767-776. ISBN 3-540-42219-6. ISSN 0302-974
- Capkovic, F. (2004). An Application of the DEDS Control Synthesis Method. In Bubak, M., van Albada, G.D., Sloot, P.M.A, Dongarra, J.J. (Eds.): Computational Science - ICCS 2004. Part III. Springer Verlag, Berlin-Heidelberg-New York, Lecture Notes in Computer Sciences, Vol. 3038 LNCS, pp. 528-536. ISBN 3-540-22116-6. ISSN 0302-9743
- Cassandras C.G. y Lafortune, S. (2008). Introduction to Discrete Event Systems. Springer. 2nd ed., 776 p., ISBN: 978-0-387-33332-8.

- Chacón, E. (2002). A Way to Implement Supervisors for Holonic Production Units". Proceedings IFAC'02. Barcelona.
- Chacon, E., Besembel, I. and Hennes, J.C. (2004). Coordination and Optimization in Oil & Gas Production Complexes". Computers in Industry, Vol. 53, Issue 1, pp. 17-37.
- Chacón, E., Besembel, I. and Jean Claude Hennes. (2003). Coordination and Optimization in Oil & Gas Production Complexes. Universidad de Los Andes. Mérida Venezuela and Laboratoire d'Analyse et d'Architecture des Systemes, Toulouse – France.
- Chen, Y. and Lin, F. (2000). Modeling of discrete event systems using finite state machines with parameters. Proceedings of the 2000 IEEE International Conference on Control Applications. Volume , Issue , 2000 Page(s):941 – 946
- Chokshi, N. N. and McFarlane, D. C. (2002). Rationales for Holonic Applications in Chemical Process Industries. Lecture Notes in Computer Science. Vol. 2322, pp. 51-86.
- Christensen, S., B. Jorgensen, J. and Kristensen, L. M. (2009). Design/CPN: A Computer Tool for Coloured Petri Nets. Documento en línea: <http://www.daimi.au.dk> Department of Computer Science University of Aarhus. Aarhus C, Denmark. pp 1 – 12.
- Colombo, A.W., Schoop, R. and Neubert, R. (2006). An agent-based intelligent control platform for industrial holonic manufacturing systems, Industrial Electronics, IEEE Transactions on Volume 53, Issue 1, pp. 322 – 337.
- CPN Group. (2009). <http://wiki.daimi.au.dk/cpntools/cpntools.wiki>. Department of Computer Science, University of Aarhus, Denmark. Consultada: 15 de noviembre de 2009.
- Davis, W.J. (1999). The distributed intelligent control of complex systems. Proceedings of the Second International Conference on Intelligent Processing and Manufacturing of Materials, 1999. IPMM '99. Volume: 1, On page(s): 615-621 vol.1. ISBN: 0-7803-5489-3.
- Deen, S.M. (2003). Agent-Based Manufacturing – Advances in the Holonic Approach, Springer-Verlag, Heidelberg, Germany, 2003.
- Devia, J.J. (2004). Propuesta de una metodología para la Automatización integrada de procesos de Producción continua. Tesis de maestría. Universidad de los Andes, Venezuela.
- Diep D., Massotte, P. and Meimouni, A. (2003). A Distributed Manufacturing Execution System Implemented with Agents: the PABADIS Model, INDIN 2003. Proceedings. IEEE International Conference on Industrial Informatics, 2003, pp. 301- 306, ISBN: 0-7803-8200-5.
- Domínguez Machuca, J.A.. (1995). Dirección de operaciones. Aspectos estratégicos en la producción y los servicios — Mc Graw Hill.

- Durán Faúndez, C. (2005). Control Distribuido Auto-organizado para un Sistema Flexible de Manufactura. Tesis presentada en conformidad con los requisitos para la obtención del título de Ingeniero Civil en Informática. UNIVERSIDAD DEL BÍO-BÍO.
- Farahvash, P. and Boucher, T.O. (2004). A multi-agent architecture for control of AGV systems”, Robotics and Computer-Integrated Manufacturing, Vol. 20, Issue 6, pp. 473-483. 13th International Conference on Flexible Automation and Intelligent Manufacturing.
- Farahvash, P., and Boucher. T.O. (2004). A multi-agent architecture for control of AGV systems, Robotics and Computer-Integrated Manufacturing, Vol. 20, Issue 6, December 2004, pp. 473-483. 13th International Conference on Flexible Automation and Intelligent Manufacturing.
- Faúndez, D. and Cristian R. (2005). Control Distribuido Auto-organizado para un Sistema Flexible de Manufactura. UNIVERSIDAD DEL BÍO-BÍO, DEPARTAMENTO DE SISTEMAS DE INFORMACIÓN. Tesis para la obtención del título de Ingeniero Civil en Informática.
- Ferrarini, L., Veber, C., Lüder, A. Peschke, J., Kalogeras, A., Gialelis, J., Rode, J., Wunsch, D. and Chapurlat, V. (2006). Control Architecture for Reconfigurable Manufacturing Systems: the PABADIS’PROMISE approach, ETFA '06. IEEE Conference on Emerging Technologies and Factory Automation, pp. 545-552, ISBN: 0-7803-9758-4.
- Fischer, K., Schillo, M. and Siekmann, J. (2003). “Holonc multiagent systems: The foundation for the organization of multiagent systems”, Proceedings of the First International Conference on Applications of Holonic and Multiagent Systems (Holo-MAS'03), Springer-Verlag.
- Fletcher, M. (2000). Holonic Manufacturing Systems: Some Scenarios and Issues. Odborný seminář INTELIGENTNÍ METODY PRO PRŮMYSLOVOU PRAXI: Diagnostika, plánování a řízení.
- Formella, A. (2006). Concurrencia y Distribución. Universidad de Vigo, Departamento de Informática. <http://www.ei.uvigo.es/~forme>
- Frayret, J., D'Amours, S. and Montreuil, B. (2004). Coordination and control in distributed and agent-based manufacturing systems. Taylor and Francis Ltd. Production Planning and Control, Volume 15, Number 1 , pp. 42-54.
- FIPA Foundation for Intelligent Physical Agents. (Consultada en febrero 2009). <http://www.fipa.org/>

- Fundación universitaria Los Libertadores. (2008). Herramientas maestras. Área de producción de material didáctico. www.ulibertadores.edu.co/virtual/Herramientas/esquemas_conceptuales.htm. Consultada: 15 de noviembre de 2009.
- Galán, R., Jiménez, A., Sanz, R. and Matía, F. (2000). Control Inteligente. Revista Iberoamericana de control inteligente. ISSN 1137-3601. España.
- Gharieb, W. (2004). Integrated control for discrete event systems. Electronic and Computer Engineering, 2004. ICEEC 04. 2004 International Conference on Electrical, pp. 377 – 382.
- Giret and V. Botti. (2004). Holons and Agents. Journal of intelligent manufacturing, Vol. 15 No. 5 pp. 645-659.
- Giret, Adriana. (2005). ANEMONA: Una metodología Multi-Agente para sistemas holónicos de fabricación. Tesis doctoral. Universidad Politécnica de Valencia. Departamento de Sistemas Informáticos y Computación. España.
- Giret. A. (2005). ANEMONA: Una metodología Multi-Agente para sistemas holónicos de fabricación, Tesis doctoral, Universidad Politécnica de Valencia, Departamento de Sistemas Informáticos y Computación. España.
- Gou, L., Luh, P., and Kyoya, Y. (1998). Holonic manufacturing scheduling: architecture, cooperation mechanism, and implementation. Computers in Industry, vol. 37, Issue 3, pp. 213-231.
- Guasch, A., Piera, M. A., Casanovas, J., Figueras, J. (2002), “Modelado y Simulación. Aplicación a Procesos Logísticos de Fabricación y Servicios”, Editions UPC.
- Heizer, R. (2004). PRINCIPIOS DE ADMINISTRACION DE OPERACIONES . Prentice Hall - Suave - pp. 704. 5 ed.
- Honderich, Ted. (2005). The Oxford Companion to Philosophy. Oxford University Press. ISBN 0-19-866132-0
- Huang Z.; Chandra V.; Jiang S.; Kumar R. Modeling Discrete Event Systems With Faults Using a Rules-based Modeling Formalism, Taylor and Francis Ltd Volume 9, Number 3, 2004 , pp. 233-254.
- Jarvis, D. , Jarvis, J. , McFarlane, D. , Lucas, A. and Ronnquist, R. (2001). Implementing a Multi-Agent systems approach to collaborative autonomous manufacturing operations", Aerospace Conference, 2001, IEEE Proceedings. Vol. 6, pp. 2803 – 2811.

- Johansson, B., Williams, E.J. and Alenljung T. (2004). Using autonomous modular material handling equipment for manufacturing flexibility, Winter Simulation Conference, Proceedings of the 36th conference on Winter simulation, pp. 1115 - 1121, ISBN:0-7803-8786-4.
- Karageorgos, A., Mehandjiev, N, Hammerle, A. and G. Weichhart, 2003. Agent-Based Optimisation of Logistics and Production Planning. Engineering Applications of Artificial Intelligence, Special Issue on Intelligent Manufacturing, 16, 4, (June 2003), p. 335-348
- Klostermeyer, A. and Klemm, E. (2003). PABADIS - An Agent Based Flexible Manufacturing Concept, INDIN 2003. Proceedings. IEEE International Conference on Industrial Informatics, On pp. 286- 293, ISBN: 0-7803-8200-5.
- Knabe, T., Schillo, M. and Fischer, K. (2002). Improvements to the FIPA contract net protocol for performance increase and cascading applications In International Workshop for Multi-Agent Interoperability at the German Conference on AI. <http://www.virtosphere.de/schillo/research/.../data/publications/workshops/2002/Knabe+.MAI2002.pdf>
- Koestler, A. (1967). The ghost in the machine. London: Arkana.
- Kotak, D., Wu, S., Fleetwood, M. and Tamoto, H. (2003). Agent-based holonic design and operations environment for distributed manufacturing, Computers in Industry Vol. 52, Issue 2, pp. 95-108.
- Kurt, J. (1990). Colored Petri Nets: A High-level Language for System Design and Analysis. Lecture Notes in Computer Science, Berlín, Alemania, vol. 483, pp 342-416.
- L, Vásquez, L. Llano, G. Zapata y L. Quintero. (2007). Diagnostico Automático de Fallas en un Sistema de Transporte de Energía. VII Congreso de la asociación colombiana de automática. Colombia.
- Leitao, Paulo. (2004). ADACOR: An Agile and Adaptive Holonic Architecture for Manufacturing Control. Facultad de ingeniería d la universidad de Porto. Tesis de doctorado en Electrotecnia e Ingeniería de la Computación. Portugal.
- Lin, C., Lin Y. and Jeng M.D. (2004). Design of intelligent manufacturing systems by using UML and Petri net. 2004 IEEE International Conference on Networking, Sensing and Control. Vol. 1, pp. 501 – 506.
- Lim, M.K. and Zhang, Z. (2003). A multi-agent based manufacturing control strategy for responsive manufacturing. Journal of Materials Processing Technology, Volume 139, Issues 1-3, Pages 379-384.

- Luder, A., Klostermeyer, A., Peschke, J., Bratoukhine, A. and Sauter, T. (2005). Distributed automation: PABADIS versus HMS, *IEEE Transactions on Industrial Informatics*, Vol. 1, Issue 1, pp. 31 – 38.
- Mačerauskas, V. and Teresius, V. (2004). Investigation of Multi-Agent Control System. *ELEKTRONIKA IR ELEKTROTECHNIKA*. 2004. Nr. 3. ISSN 1392 – 1215
- Marik, V., McFarlane, D. and Valckenaers, P. (2003). *Holonic and Multi-Agent Systems for Manufacturing*. Springer-Verlag, Heidelberg.
- Matellán, V. and Borrajo, D. (2001). ABC2 an Agenda Based Multi-Agent Model for Robots Control and Cooperation. *Journal of Intelligent and Robotic Systems* , Vol. 32 , pp. 93-114, ISSN:0921-0296..
- Maturana, F., Shen, W., and Norrie, D. 1999. Metamorph: An adaptive agent-based architecture for intelligent manufacturing. *Int. J. Prod. Res.* 37, 10, 2159--2174.
- Maturana, F., Tichý, P., šlechta, P. Staron, R. Discenzo, F. and Hall, K. (2003). A Highly Distributed Intelligent Multi-agent Architecture for Industrial Automation. *Lecture Notes in Computer Science*. Volume 2691/2003, ISSN 0302-9743 (Print) 1611-3349 (Online).
- Mönch, L. and Stehli, M. (2006). ManufAg: a multi-agent-system framework for production control of complex manufacturing systems. *Information Systems and e-Business Management*, Volume 4, Number 2, pp. 159-185.
- Mondel and M.K. Tiwari, (2002). "Application of an Autonomous Agent Network to Support the Architecture of a Holonic Manufacturing System", Publication *The International Journal of Advanced Manufacturing Technology* Editor Springer London, ISSN 0268-3768 (Print) 1433-3015 (Online), Vol. 20, No. 12.
- Montilva, J.; Rivero, D.; Barrios, J.; Martínez, A.; Besembel, I. y Sandia, B. Un modelo organizacional holónico para la gestión de programas de formación profesional. *Revista ESPACIOS*. Vol. 29 (3): 3-5. Venezuela. 2008.
- Murata, T. (1989). Petri Nets: Properties, Analysis and Applications", *IEEE Proc*, Vol. 77, pp. 541-580.
- Németh, E. and Hangos, K. M. (2004). Multi-scale Process Model Description by Generalized Hierarchical CPN Models. Technical Report of The Systems and Control Laboratory SCL-002/2004. Budapest MTA SZTAKI, Hungría.
- OMG, Object Management Group. (2009). UML Resource Page. <http://www.uml.org/>. Consultada: 17 de noviembre de 2009.
- Open SCADA Project. (2009). <http://www.oscada.org/>. Consultada: 16 de noviembre de 2009.

- Orozco, A., Alzate, A. and Holguín, G. (2004). DESARROLLO DE SOFTWARE PARA EL PROCESO DE SEÑALES UTILIZANDO REDES DE PETRI. *Scientia et Technica* Año X, No. 26, UTP. ISSN 0122-1701
- Park, Y. , Choi, H. and Kim, H. (2003). Automated Negotiation for Order Transaction of Injection Mold Manufacturer. *Proceedings of the 5th international conference on Electronic commerce*, pp. 488 – 497, ISBN:1-58113-788-5.
- Pérez, Y.S. (2007). Aplicación de metodologías ingenieras, ZEUS, MASINA al desarrollo de sistemas Multi-Agente, partiendo de SMA de subastas para la identificación de mejores prácticas. Trabajo presentado para optar por el título de Ingeniero de sistemas. Universidad De Pamplona, Facultad de Ingenierías y Arquitectura.
- Pinedo, M. (2001). *Scheduling: Theory, Algorithms, and Systems* (2nd Edition). Prentice Hall, 586 pages, ISBN-10: 0130281387, ISBN-13: 978-0130281388.
- Qinghua, C., Yongsan, W. and Hong, Z. (1993). Petri net tools for analysis and verification of task scheduling algorithm in real time distributed process control system. *Proceedings. Computer, Communication, Control and Power Engineering. 1993 IEEE Region 10 Conference on TENCON apos. Volume , Issue 0, 19-21 Oct 1993 Page(s):56 - 59 vol.4*
- Quintero, L., Zapata, G. y Chacón, E. (2008). Comportamiento autónomo del holón recurso basado en la Agenda de Producción. *Revista Avances en Sistemas e Informática*. Vol. 5 No. 1. ISSN 1909-0056.
- Ramadge P. J. and Wonham W. M., 1989. The control of discrete event systems. *Proceedings of IEEE*, 77(1), pp. 81 - 98.
- Rim Choi, H., Joo Park, B., Soo Kim, H., Sung Park, Y. and Jae Park, Y. (2003). Multi-Agent based negotiation support systems for order based manufacturers. *Proceedings of the 5th international conference on Electronic commerce*. Pages: 479 - 487, ISBN:1-58113-788-5
- Rus Mansilla, F. (2005). *Introducción a la programación Concurrente*. Departamento de Lenguajes y Ciencias de la Computación Universidad de Málaga.
- Salazar, C. (2003). *Agentes y Multiagentes Inteligentes: Conceptos, Arquitecturas y Aplicaciones*. Texto de apoyo para la asignatura Inteligencia Artificial. Universidad Mayor de San Simón, Facultad de Ciencias Y Tecnología, Departamento de Informática Y Sistemas.
- Shen, W. Hao, Q., Joong Yoon, H. and Norrie, D. (2006). Applications of agent-based systems in intelligent manufacturing: An updated review. Elsevier Ltd., *Advanced Engineering Informatics* Volume 20, Issue 4, Pages 415-431

- Silva, M. (1985). Las Redes de Petri en la Automática y la Informática. *Editorial AC*. Madrid, España.
- Silva, M.; Teruel, E.; Valette, R.; Pingaud, H. (1998). Petri Nets and Production Systems. Lecture Notes in Computer Science, Vol. 1492: Lectures on Petri Nets II: Applications. Springer-Verlag, ISBN: 3-540-65307-4.
- Silva, N. and Ramos, C. (1999). Infrastructures And Scheduling Method For Holonic Manufacturing Systems. Proceedings of the 1999 IEEE International Symposium on Assembly and Task Planning.
- Soto, L. (2008). Redes de Petri: Modelado e implementación de algoritmos para autómatas programables. *Tecnología en Marcha*, Vol. 21, N.º 4, pp. 102-125.
- Sousa, P. and Ramos, C. (1999). A distributed architecture and negotiation protocol for scheduling in manufacturing systems. *Computers in Industry* 38, pp. 103–113
- Sousa, p., Ramos, C. and Neves, J. (2007). Scheduling in holonic manufacturing systems. Process planning and scheduling for distributed manufacturing. ISBN 978-1-84628-751-0 (print) 978-1-84628-752-7 (online).
- Sugiyama, S. Ambalavanar Tharumarajah. (2006). Fundamental Behavior of Holonic System. IEEE International Conference on Service Operations and Logistics, and Informatics. pp.734-739. ISBN: 1-4244-0317-0.
- Torrealba, A. (2005). Desarrollo de un sistema de supervisión de sistemas de producción continua. Trabajo de grado presentado como requisito parcial para obtener el grado de Magíster Scientiae en Automatización e Instrumentación Universidad de Los Andes Mérida, Venezuela.
- Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L. and Peeters, P. (1998). Reference Architecture for Holonic Manufacturing Systems: PROSA”, *Computers in Industry*, Vol. 37, ISSN:0166-3615.
- Vasko, D., Maturana, F., Bowles, A. and Vandenberg, S. (2000). Autonomous Cooperative Factory Control. Lecture Notes In Computer Science; Vol. 1881, Proceedings of the Third Pacific Rim International Workshop on Multi-Agents: Design and Applications of Intelligent Agents, Pages: 156 – 169, ISBN:3-540-67911-1.
- Velazco L. (2007). Implementación de algoritmo de control fuzzy pi en un dispositivo de lógica programable. CEA 2007 Comité Español de Automática.
- Villa, A. (2002). Autonomy versus efficiency in Multi-Agent management of extended enterprises. *Publication Journal of Intelligent Manufacturing*, ISSN 0956-5515 (Print) 1572-8145 (Online).

- W. Shen, F. Maturana and D.H. Norrie, (2000). MetaMorph II: an agent-based architecture for distributed intelligent design and manufacturing. *Journal of Intelligent Manufacturing*, Volume 11, Number 3, June 2000, pp. 237-251(15).
- Wong, K.C.; Thistle, J.G.; Malhame, R.P.; Hoang, H.-H. (1998). Supervisory control of distributed systems: conflict resolution. *Decision and Proceedings of the 37th IEEE Conference on Control*, 1998. Volume 3, Issue , Page(s):3275 - 3280 vol.3
- Wyns , Jo. (1999). Reference architecture for holonic manufacturing systems -the key to support evolution and reconfiguration-. PhD dissertation. PMA/K.U.Leuven.
- Wyns, J. (1999). Reference architecture for holonic manufacturing systems -the key to support evolution and reconfiguration-PROSA, PhD dissertation. PMA/K.U.Leuven.
- Zafeer Alibai B.A.Sc. (2003). “What is Contract Net Interaction Protocol?”. IRMS Laboratory, SFU. July 25.
- Zapata M. G., Quintero H. L, Chacón R. E, Arboleda C. A. (2009). Modelamiento del Protocolo de Negociación de la Unidad de Producción Holónica. Congreso Regional de Eléctrica, Electrónica y Sistemas 2009. Colombia.
- Zapata, G. (2008A). Una Propuesta Para El Control De Sistemas De Producción Continua Desde La Teoría De Control Supervisorio De Sistemas A Eventos Discretos. Propuesta de doctorado en Ciencias aplicadas de la Universidad de los Andes – Mérida.
- Zapata, G. (2008B). Sistemas de Supervisión. Informe de avance en programa de doctorado "PROPUESTA PARA EL CONTROL DE SISTEMAS DE PRODUCCIÓN CONTINUA DESDE LA TEORÍA DE CONTROL SUPERVISORIO DE SISTEMAS A EVENTOS DISCRETOS".
- Zapata, G., Chacón, E. and Quintero, L. (2008c). LA AGENDA DE PRODUCCIÓN PARA LA SUPERVISIÓN DE SISTEMAS HOLÓNICOS. XIII Congreso Latinoamericano de Control Automático / VI Congreso Venezolano de Automatización y Control. Venezuela
- Zapata, G., Hoyos, B. and Quintero, L. (2008). Diseño del Automatismo para una Planta Piloto de Recubrimientos Electrolíticos. Parte I: Modelo Mediante Redes de Petri. *Revista Facultad de Ingeniería Universidad de Antioquia*. N.º 45 pp. 67-76.
- Zhang, Lin; B. Mitchell, L. Falzon, M. Davies, L. Kristensen and L. Billington. (2001). Model-based Operational Planning Using Coloured Petri Net”, 6th International Command and Control Research and Technology Symposium, pp. 1-15.

Zhang, X. and Lesser, V. (2002). Multi-Linked Negotiation in Multi-Agent Systems. Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 3, pp. 1207 – 1214, ISBN:1-58113-480-0.

Zhou, M.C. and DiCesare, F. (1990). A Petri net design method for automated manufacturing systems with shared resources. International Conference on Robotics and Automation, pp. 526 - 531 *vol.1*.

ANEXO 1

CONTROL DE SISTEMAS A EVENTOS DISCRETOS

1.1 Modelado De Sistemas A Eventos Discretos

Es clásico mencionar que un modelo representa un objeto artificial construido para representar de forma simplificada un aspecto problemático de la realidad dentro de un fenómeno o sistema real. Son muchos los sistemas o campos de aplicación en los que se desconocen las consecuencias de la ocurrencia de un cierto evento (como podría ser una perturbación), así como la influencia sobre el rendimiento global del proceso de una variación en la secuencia de eventos que se producirían como consecuencia de la aparición de un cierto evento situado en algún instante anterior en el tiempo. Esta situación en el modelamiento de Sistemas de Producción se torna compleja por la dificultad de formalizar las secuencias de actividades que puede desencadenar un determinado evento.

En un sistema de manufactura es habitual que existan fuertes y complejas interacciones entre sus variables y que éstas respondan a eventos discretos, algunos de ellos como la existencia de una falla o la llegada imprevista de una orden sin una causa o una temporalidad conocida, lo que dificulta el conocer cómo evolucionará el sistema en su conjunto. (Adaptación de ())

El modelamiento de sistemas a eventos discretos está generalmente asociado con algunos formalismos populares como Autómatas de Estado Finito, Redes de Petri, Grafos de Eventos (Event Graphs), Statecharts, entre otros. De acuerdo a (Chen and Lin, 2000) y (Huang et al, 2004) la teoría de control para sistemas a eventos discretos modelada como máquinas de estado finito, se ha desarrollado en relación de diversas situaciones fundamentales de control. Sin embargo, el modelado de máquinas de estados finito tiene la debilidad de la explosión de estados que hace inadecuado para muchas aplicaciones prácticas. En esta línea y según lo presentado en (Zapata, 2008) otro de los métodos ampliamente difundidos para modelar y controlar DES, son los basados en las denominadas Redes de Petri (PN) son una generalización de la teoría de autómatas. Las PN tienen una representación matemática sencilla empleando álgebra lineal y teoría de conjuntos, lo que las hace particularmente útiles para modelar y analizar sistemas a eventos discretos.

Las principales características de los sistemas a eventos discretos que deben ser consideradas por un formalismo que represente el control de un sistema a eventos discretos son:

- **Son asíncronos.** Porque algunos eventos pueden ocurrir en cualquier momento, sin ningún tipo de periodicidad ni de continuidad.
- **Están dirigidos por eventos.** Cuando ocurre un suceso cambia el estado del sistema.
- **Son secuenciales.** Porque puede haber eventos que guarden una cierta secuencia, tal que para que ocurra uno, antes debe de haber ocurrido el anterior.

- **Presentan sincronización.** Es el problema de retrasar la ejecución de un proceso hasta que se cumpla una determinada condición.
- **Presentan concurrencia.** Porque varios eventos pueden ocurrir al mismo tiempo. Aquél DES que contiene uno o más procesos que trabajan de forma conjunta para realizar una determinada tarea.
- **Pueden representar conflictos o exclusión mutua.** El conflicto se presenta cuando un recurso es compartido por varias entidades y se resuelve haciendo que no se puedan presentar al mismo tiempo dos solicitudes del recurso.
- **Pueden presentar parada por interbloqueo o deadlock.** Por ejemplo, el robot ha cogido una pieza de la máquina 1 y la máquina 2 requiere la pieza y no la recibe.

Antes de seleccionar un formalismo para la representación de control de DES, es necesario presentar los requerimientos impuestos desde la teoría de control para el modelamiento de DES. Conjuntamente se presentan las exigencias definidas hacia el modelamiento de control desde la complejidad de los sistemas representados. Por tanto, es necesario cumplir con los siguientes requerimientos para lograr una representación de un sistema dinámico:

- ***Reactividad:***

La reactividad consiste en mantener la operación productiva y/o en contribuir a reducir el tiempo de inactividad cuando se detecta un problema.

- ***Tratamiento de bloqueos.***

Cuando dos procesos P_0 y P_1 quieren tener acceso simultáneamente a dos recursos r_0 y r_1 , es posible que se produzca un bloqueo de ambos procesos (Formella, 2006).

1. los procesos tienen que compartir recursos con exclusión mutua
2. los recursos no permiten ser usados por más de un proceso al mismo tiempo

Ante las situaciones de bloqueo es posible uno de los siguientes comportamientos:

- ⇒ Detectar y actuar
- ⇒ Evitar
- ⇒ Prevenir

- ***Resolución de conflictos***

Los sistemas de control inteligentes deben combinar mecanismos de resolución de conflictos y estrategias de control para detectar, identificar y resolver conflictos lógicos causados por la posibilidad de eventos simultáneos, que es propio de sistemas no

determinísticos como los sistemas de manufactura. En (Wong et al, 1998) se definen prioridades independientes de la dinámica del sistema como esquema de resolución de conflictos entre controladores concurrentes.

➤ ***Vivacidad del sistema de control (Sistema libre de bloqueos)***

Una propiedad de viveza (liveness property) es aquella que garantiza que el programa entrará eventualmente en un estado válido (Rus Mansilla, 2005), (Formella, 2006).

Ejemplos de propiedades de vivacidad:

- ⇒ Ningún proceso se muere por inanición, es decir, el sistema resuelve justamente los conflictos. Esto es, si un proceso pide un recurso, lo consigue en algún momento.
- ⇒ Los procesos no se bloquean mutuamente
- ⇒ No se termina un proceso desde fuera sin razón
- ⇒ Un proceso no queda dormido o suspendido
- ⇒ la conexión entre procesos es fiable

➤ ***El control es acotado y estable (Seguro):***

- ⇒ Una propiedad de seguridad (safety property) es aquella que garantiza que el programa nunca entra en un estado no válido. Indica que no está pasando nada malo en el programa, es decir, el programa no ejecuta instrucciones que no debe hacer.
- ⇒ Un estado no es válido cuando algunas variables de estado tienen un valor incorrecto

➤ ***Alcanzabilidad***

Cuando un sistema cuenta con esta característica, entonces mediante un controlador se puede llevar este sistema desde un estado inicial hasta otro estado cualquiera, en un tiempo finito

➤ ***Controlabilidad***

Un sistema es controlable si se puede llevar desde cualquier punto al origen en tiempo finito. De acuerdo a (Formella, 2006) la controlabilidad no implica alcanzabilidad

Estas propiedades permiten validar el comportamiento del procedimiento de control de un sistema.

1.2 Formalismos de Modelamiento de Sistemas a Eventos Discretos

Mediante paradigma o formalismo se referencia a un conjunto de conceptos, leyes y medios que sirven para definir un conjunto de modelos. Hay varios paradigmas para especificar formalmente sistemas a eventos discretos y la mayoría de ellos tienen existencia conceptual independiente de los lenguajes de simulación que pueden usarse para llevar a cabo las simulaciones.

En el estudio de los sistemas a eventos discretos (DES) se han desarrollado soluciones teóricas acompañadas de métodos y algoritmos que implementadas a través de soluciones informáticas, han permitido el control, automatización e integración de estos sistemas.

Los formalismos mejoran la comprensión de los sistemas, permiten identificar parámetros clave e influencias, conducen a razonamiento más eficiente, ayudan en la aplicación, entre otros. Por otra parte, algunos métodos formales facilitan el diálogo entre las distintas personas implicadas en el diseño y operación, especialmente cuando se proporcionan representaciones gráficas e intuitivas. Adicionalmente, el uso de herramientas formales de modelamiento es una necesidad del medio industrial, debido a las propiedades que ofrecen, tales como el uso del mismo modelo para el análisis de las propiedades de comportamiento (propiedades de alcanzabilidad, seguridad y vivacidad) y la evaluación de desempeño mediante herramientas matemáticas (Zapata et al, 2008).

Algunos de los formalismos que han sido utilizados para representar el control de un sistema mediante sistemas a eventos discretos son: Las Redes de Petri, El GRAFCET, El formalismo DEVS y las máquinas de estado finito. De acuerdo a (Guasch et al, 2002) deben respetar las siguientes propiedades:

- El formalismo debe ser independiente de los constructores y herramientas que ofrecen los entornos de simulación.
- El modelo formalizado debe poder ser analizado para determinar relaciones entre componentes y evaluar alternativas que permitan la simplificación del modelo.
- El formalismo debe permitir una fácil transformación a las representaciones soportadas por los entornos de simulación.
- Adecuación para tratar con sistemas reales.
- Facilidad de uso e implementación.

Por tanto, los sistemas a eventos discretos definen requerimientos hacia los formalismos, estos provienen desde la necesidad definida por la dinámica natural y los requerimientos de representación de control de los DES:

- Concurrencia de operaciones.

- ⇒ Sincronización y operaciones
- ⇒ Secuencialidad
- ⇒ Recursos limitados
- ⇒ Representación de conflictos
- ⇒ Seguridad
- ⇒ Vivacidad
- ⇒ Fairness (Imparcialidad)

Cuando los procesos compiten por acceso a recursos compartidos, este se puede definir por los conceptos de Imparcialidad. Normalmente se quiere que todos los procesos manifiesten algún progreso en su trabajo. Sin embargo, eso no es necesario en programas concurrentes; se puede vivir bien con algunos procesos “muertos”, mientras no involucre otros problemas para el controlador. Siempre existe la posibilidad que el trabajo asignado a un proceso está hecho por otro proceso dejando el primero en espera infinita.

La Imparcialidad (“fairness”) es la garantía de que todo proceso tiene la posibilidad de evolucionar, independientemente de lo que hagan los restantes procesos.

En programas concurrentes es posible que un proceso nunca llegue a hacer nada si el planificador o el control de los recursos compartidos respectivamente no permiten que el proceso pueda cumplir con sus pedidos. Es decir, el proceso está sometido a una espera infinita, o en otras palabras, sufre una inanición. Para superar esto en el sistema debe definirse una regla de prioridad que le permita a los procesos acceder con Imparcialidad al recurso compartido.

“el comportamiento es “equitativo”.

⇒ **Verificación**

La finalidad de la verificación es comprobar que no se han cometido errores al traducir el modelo, bien usando un entorno de modelado o mediante un lenguaje de simulación o de programación.

Entre otros, pueden usarse los siguientes procedimientos para verificar el modelo:

- ⇒ Verificación manual de la lógica. Consiste en ejecutar la simulación durante un periodo de tiempo corto y comprobar manualmente los resultados obtenidos.
- ⇒ Árbol de alcanzabilidad
- ⇒

➤ *Validación*

La validación consiste en comprobar que el modelo supone una aproximación adecuada de la realidad para los objetivos particulares del estudio de simulación.

Puede considerarse que la validación del modelo tiene tres vertientes diferentes.

Consiste en determinar:

- ⇒ Si el modelo representa adecuadamente al sistema real (comprobación de la estructura del modelo).
- ⇒ Si los datos generados de la simulación del modelo reproducen de forma adecuada el comportamiento del sistema real (comprobación del comportamiento del modelo).

1.3 Redes de Petri

Las Redes de Petri (PN) son una herramienta matemática y gráfica que permite modelar, simular y controlar DES (Silva, 1985), (Guasch et al., 2002). Las PN representan una herramienta de modelado independiente de cualquier tecnología, clara, fácil de utilizar y no ambigua, que comprende los conceptos básicos de receptividad y sensibilidad, por los cuales es posible obtener descripciones de los sistemas a modelar con un mínimo de información conocida y suficiente para sintetizarlos. Además las PN son ideales como metodología de modelado para capturar las relaciones causales y de precedencia entre eventos y situaciones, es decir, facilitan la representación de evoluciones simultáneas, las cuales son algo clave a la hora de modelar y simular el comportamiento de un sistema de manufactura. Como se muestra en (Zapata, 2008), La Figura 78 muestra un DES que evoluciona entre los estados x_1 , x_2 , x_3 y x_4 por la ocurrencia de los eventos α , β , λ y σ . Esta figura representa la relación de la trayectoria definida por un sistema a eventos discreto y el modelo PN que constituye esta trayectoria.

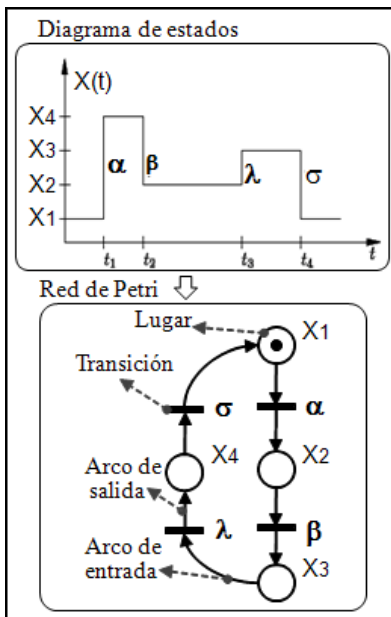


Figura 78. Representación de un DES mediante PN

La especificación de un modelo orientado a eventos discretos usando el formalismo de las PN permite obtener información del sistema, tanto si se analiza el comportamiento de la red como si se estudia su estructura.

Específicamente en sistemas de manufactura, la formalización con PN permite representar de forma muy natural clientes o peticiones, recursos y procesos como marcas situadas en los distintos nodos lugar. Adicionalmente (Silva et al, 1998) expresan que las PN son un formalismo que provee un marco o paradigma de trabajo apropiado para el diseño y la representación de la operación de sistema de producción.

En resumen, las PN obtienen su utilidad fundamentalmente porque permiten:

- Representan de forma explícita los estados y eventos del modelo.
- Los fenómenos de concurrencia, sincronismo y dependencia causal se representan de forma natural.
- El conjunto de recursos restringidos se representa de forma explícita en el modelo.
- Hay muy pocas reglas, lo que facilita su aprendizaje.
- Su representación gráfica es muy intuitiva.
- Su semántica es precisa y sin ambigüedades.
- Es independiente de la herramienta de simulación que se emplee.

- Capacidad para representar de forma natural concurrencia, la causalidad, la sincronización, recursos compartidos, los conflictos, comportamientos no deseados del sistema (Situaciones de bloqueo), tal como se ilustra en la Figura 79.

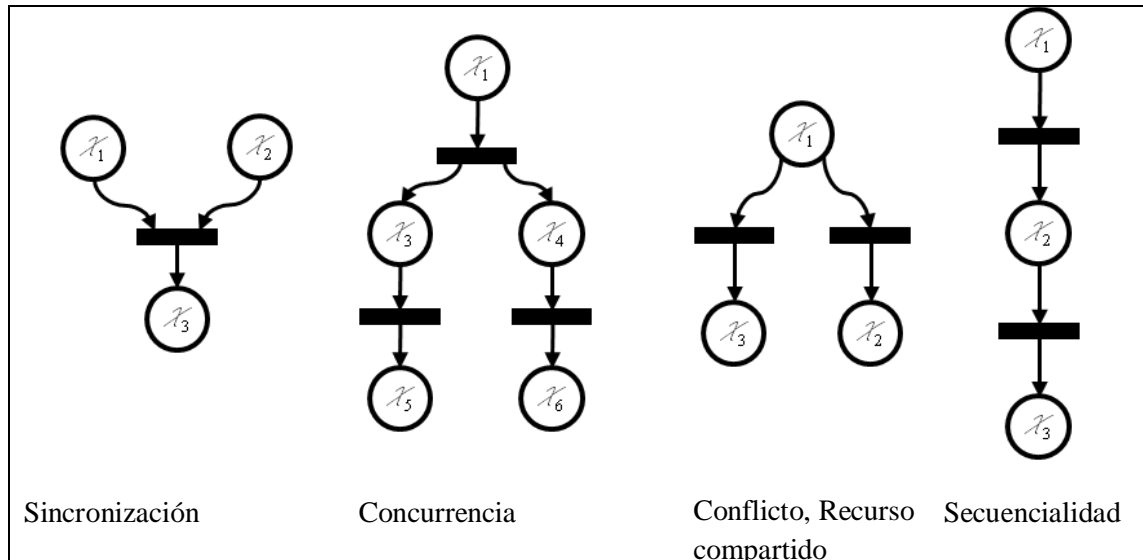


Figura 79. Capacidad de representación de las PN

- Permiten modularidad y reusabilidad. Esto permite una representación compacta debido a la representación de estados de forma distribuida, ventaja en comparación a una representación secuencial.
- La representación gráfica facilita la documentación y control del sistema.

Para modelar los sistemas a eventos discretos las PN se basa en el concepto de que las relaciones entre los componentes de un sistema pueden ser representadas por una red que permite el análisis de manera formal, obtener información del comportamiento dinámico del sistema modelado, capturar las relaciones de precedencia e interacción de eventos concurrentes y asíncronos. Son modelos lógicos con un fundamento matemático que permite el análisis cuantitativo y cualitativo del sistema (Vásquez, 2007). En esta línea, la robustez matemática del método de modelamiento brinda técnicas formales para analizar las propiedades y garantizar el desempeño del sistema. Típicamente se consideran las propiedades: vivacidad, seguridad y reversibilidad.

Para modelar la ejecución de una secuencia de eventos sobre el sistema modelado, es necesario asociar a las transiciones de una PN una etiqueta que relacione un operador a una transición. Además, las PN utilizadas para este trabajo tienen la capacidad de manejar variables con valores enteros, caracteres y booleanos en los estados o lugares definidos en el modelo. Bajo esta consideración los arcos deben tener la capacidad de reconocer inscripciones con expresiones que contengan constantes, variables y funciones ($f()$). Conjuntamente se considera la restricción de la dinámica del modelo mediante la asignación a una transición de una función de guarda.

En consecuencia, para representar estos conceptos de control y flujo de información, se realiza una extensión de las PN, definida como Redes de Petri Etiquetadas de Alto Nivel (High-Level labeled Petri Net-(HLLPN)).

Formalmente una HLLPN es una 9-tupla $N=\langle P,T,F,M_i,l,\mathcal{G},C,G,E\rangle$, donde P y T son un conjunto finito de lugares y transiciones respectivamente, $A\subseteq(P\times T)\cup(T\times P)$ es un conjunto finito de arcos representando el flujo de relaciones (flow relation), l es una etiqueta que asigna a cada transición un evento $l:T\rightarrow 2^\Sigma\cup\{\varepsilon\}$ donde $\Sigma=\Sigma_{nc}\cup\Sigma_c$ y ε es un evento silencioso (Zapata, 2008), \mathcal{G} es un conjunto finito no vacío de conjuntos numéricos-NC (R, Z, R^+ , B), C es una función que mapea a cada lugar P un conjunto numérico, tal que $C:P\rightarrow\mathcal{G}$. G relaciona una función de guarda que mapea a cada transición un predicado que representa un valor booleano de la forma: $\forall t\in T: [NC(G(t))=B \wedge NC(Var(G(t)))\subseteq\mathcal{G}]$, E es una función que mapea a cada arco una expresión de arco del tipo $C((p(a)))$ tal que: $\forall a\in A: [NC(E(a))=C(p)\wedge NC(Var(E(a)))\subseteq\mathcal{G}]$, M_i es la función de inicialización, que es definida desde P mediante una expresión cerrada tal que: $\forall p\in P: [NC(I(p))=C(p)]$, (Jensen,1997).

La red $R(N, M_i)$ es una red marcada con marcaje inicial M_i . La notación $M_i[\sigma]M$ se utiliza para expresar que el disparo de σ en el marcaje M_i lleva a M (Zapata, 2008), donde σ representa una secuencia de disparos $\sigma = t_1 t_2 t_3 \dots t_n$

Los tipos de dato asignados a las marcas pueden ser inspeccionados mediante las expresiones de arco de llegada a las transiciones, lo que permite activarlas no sólo en función del número de marcas en los nodos lugar de entrada a las transiciones sino en función del valor del tipo de dato de las marcas disponibles en dichos nodos lugar. Al mismo tiempo estas expresiones modelan también los efectos de salida de la transición modificando los valores de los tipos de dato de las marcas de los nodos lugar de salida.

Las PN Etiquetadas de Alto Nivel permitan especificar el flujo de información que suele reflejarse mediante datos asignados a entidades cuyos valores cambian en función de los eventos que aparecen. Estas entidades en las PN Etiquetadas de Alto Nivel se representan como marcas de distintos tipos de dato en los correspondientes nodos lugar de la red. Con estos atributos, las PN Etiquetadas de Alto Nivel permiten construir modelos más compactos y paramétricos, lo que facilita considerablemente su mantenimiento e implementación. Estos modelos requerirían de estructuras con un número elevado de componentes si fueran desarrollados con el formalismo de las PN (Guasch et al., 2002).

Concretamente, al formalismo que represente el control de los DES se le exige del cumplimiento de los siguientes requerimientos:

➤ Entidades

Las entidades son objetos dinámicos en la simulación, que son creados y se mueven por el sistema, cambiando el valor de sus atributos, afectados por otras entidades y por el estado del sistema. Concretamente, las entidades comprenden los objetos, las personas, los conceptos, entre

otros, cuya existencia es reconocida por algún sistema de ontología, que define las relaciones de un conjunto de entidades dentro un dominio definido como sistema de estudio (Honderich, 2005).

➤ **Atributos**

Los atributos son características o propiedades de las entidades, permiten individualizar cada instanciación de una determinada clase de entidad sin más que asignar valores a sus atributos. Por ejemplo, algunos atributos que podrían definirse para el tipo de entidad “cliente” son: la prioridad con que debe ser atendido o determinados datos personales, como son el nombre y los apellidos, la edad, la nacionalidad, etc. En general, el valor de los atributos diferirá de un cliente a otro y es lo que permite diferenciarlos.

➤ **Variables**

Las variables representan características del sistema que son independientes de los tipos de entidades o del número de realizaciones existentes en determinado instante. Por tanto, las variables no están asociadas a entidades en concreto, sino que pertenecen al conjunto del sistema.

➤ **Recursos**

Los recursos pueden ser el personal (“operario”), las máquinas (por ejemplo, si las entidades son piezas que deben ser procesadas), el espacio (por ejemplo, en un almacén), etc. Una entidad captura un recurso cuando éste está disponible, a fin de obtener un servicio de él, y lo libera una vez ha terminado.

El recurso puede ser individual o estar compuesto por un grupo de elementos individuales, cada uno de los cuales se llama una unidad del recurso.

➤ **Actividades**

Las actividades son las tareas o acciones que tienen lugar en el sistema. Toda actividad está siempre delimitada por dos eventos, el de comienzo y el de fin de la actividad, por tanto tiene una duración temporal y, normalmente, precisa del uso de recursos.

Ejemplos de actividades son: la reparación de una máquina, el procesado de una pieza, el transporte de un cliente, etc.

➤ **Flujo de Información**

La necesidad de representación de flujos de conocimiento se deriva de la realidad, las especificaciones de las tareas de control (como criterios, restricciones, diferentes circunstancias concernientes con las influencias del entorno, entre otras) estas se expresan como reglas, estilo sistemas expertos. Entonces, un dominio orientado a una base de conocimiento (KB) expresando el conocimiento sobre las especificaciones de las tareas de control es requerido sea definido.

1.3.1 Propiedades de las Redes de Petri

La fortaleza del modelado de las PN radica en sus propiedades, que se dividen en dos grandes áreas, las dependientes del marcado inicial llamadas propiedades *dinámicas* o *del comportamiento* y las propiedades independientes del marcado, llamadas *estructurales* o *estáticas* (Soto, 2008).

Propiedades de Comportamiento

Las principales propiedades de comportamiento son la alcanzabilidad, la acotabilidad, la vivacidad, la reversibilidad, la cobertura y la persistencia.

- **Marcado Alcanzable:** Un marcado M_n se dice alcanzable desde un marcado si existe una secuencia de disparos que transforma M_i en M_n .
- **Alcanzabilidad:** El problema de la alcanzabilidad para las redes de Petri será el problema de encontrar si M pertenece a $R(M_i)$ en una red dada $R(N, M_i)$.
- **Red de Petri Acotada:** Una red de Petri $R(N, M_i)$ se dice k -acotada o acotada si el número de tokens en cada lugar no es superior a un número finito k para cualquier marcado alcanzable desde M_i , es decir, $M(p) \leq k$ para todo lugar p y todo marcado M que pertenece a $R(M_i)$.
- **Red de Petri Segura:** Una red de Petri $R(N, M_i)$ se dice segura si está 1-acotada.
- **Interbloqueo:** Se dice que en una red de Petri ocurre un interbloqueo cuando se alcanza un marcado desde el que no se puede disparar ninguna transición.
- **Red de Petri Viva:** Una red de Petri $R(N, M_i)$ se dice que está viva (o equivalentemente se dice que es un marcado vivo para N) si, sea cual sea el marcado que se alcance desde M_i , existe una secuencia disparable que permite disparar cualquier transición de la red. La vivacidad garantiza, por tanto, la ausencia de interbloqueos. Esto implica que cualquier transición es eventualmente disparable en alguna secuencia de disparo.

Propiedades Estructurales

Las propiedades estructurales son aquellas que dependen de la estructura topológica de las redes de Petri. Son independientes del marcado inicial en el sentido de que dichas propiedades se cumplen para cualquier marcado inicial. Son propiedades estructurales la acotabilidad estructural, la vivacidad estructural, la controlabilidad, la conservatividad, la repetitividad y la consistencia (Soto, 2008).

Red de Petri Acotada Estructuralmente: Una red de Petri está acotada estructuralmente si está acotada para cualquier marcado inicial finito.

Lugar No Acotado Estructuralmente: Un lugar p en una red de Petri se dice no acotado estructuralmente si existe un marcado M y una secuencia de disparo σ desde M tal que p no esté acotado.

Red de Petri Estructuralmente Viva: Una red de Petri está estructuralmente viva si existe algún marcado inicial para el que está viva.

Red de Petri Controlable: Una red de Petri se dice completamente controlable si cualquier marcado es alcanzable desde cualquier otro marcado.

1.3.2 Métodos de análisis de propiedades

A partir de una Red de Petri $R(N, M_i)$, se puede obtener tantos nuevos marcados como transiciones habilitadas disparadas. Este proceso resulta en un árbol de marcados infinito para una PN no acotada. Para redes acotadas, el árbol de cobertura es llamado árbol de alcanzabilidad.

Formalmente un árbol de alcanzabilidad es una enumeración del conjunto de marcajes alcanzables con un marcaje M_i para una red de Petri $R(N, M_i)$ (Qinghua, 2008).

El conjunto de marcajes alcanzables desde el marcaje inicial M_i con diferentes secuencias de disparo de transiciones se representa como un grafo dirigido, donde los nodos corresponden a los marcajes generados y los arcos corresponden a las transiciones disparadas entre marcajes de la red. Un árbol de alcanzabilidad completo representa todas las posibles trayectorias en el espacio de estados discretos que describe el comportamiento de una red Petri. A través del árbol de alcanzabilidad es posible realizar la búsqueda de una ruta óptima entre dos marcajes en el grafo de estados resultante a partir de la evolución de una red de Petri. Esta capacidad es útil para programación de actividades en un sistema a eventos discretos, dado que es posible a analizar todas las posibles secuencias de disparo, que en el grafo de estado corresponden a programas factibles como solución al problema de asignación de actividades en un DES.

Cada nodo en el árbol de alcanzabilidad que representa un estado de la red de Petri, está definido en su interior por un número de nodo y en la parte inferior se referencia la cantidad de nodos predecesores y la cantidad de nodos sucesores, tal como se ilustra en la Figura 80.

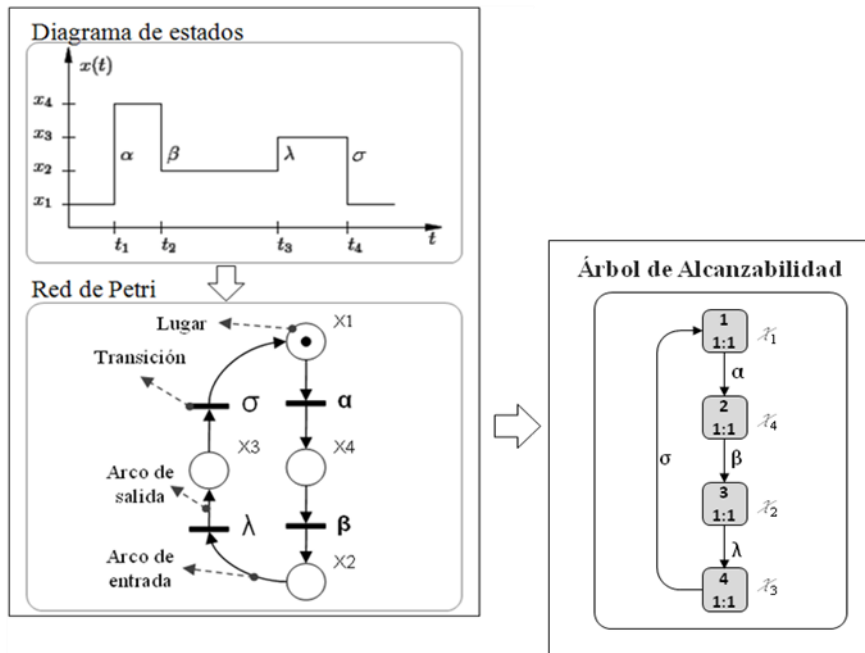


Figura 80. Árbol de alcanzabilidad

ANEXO 2

PROPIEDADES DINÁMICAS DE LAS REDES DE PETRI

Con el fin de evitar que en el proceso de elaboración de un sistema de control se pase a la fase de implementación con un modelo erróneo, se estudiarán algunas técnicas para llevar a cabo la verificación y validación funcional de los modelos hechos.

Para ello se introducirán las propiedades dinámicas definidas para las redes de Petri, las cuales están íntimamente ligadas a las propiedades de los grafos, estas propiedades son: Alcanzabilidad, Acotación, Reversibilidad, Vivacidad e Imparcialidad.

2.1 Definiciones Formales de Propiedades

En el presente trabajo los análisis serán hechos con base en simulaciones y Grafo de Alcanzabilidad, por lo tanto, las propiedades dinámicas serán dadas en términos de teoría de grafos simplificada y extendida al caso particular de las redes de Petri Binarias autónomas no jerárquicas, en el Capítulo 3 se extenderán las definiciones a las redes Jerárquicas.

2.1.1. Grafo de Alcanzabilidad

El Grafo de Alcanzabilidad de una PN [4] es un grafo dirigido: $\mathbf{OG} = (\mathbf{V}, \mathbf{A}, \mathbf{N})$ que satisface las siguientes condiciones:

- (i) \mathbf{V} es un conjunto de **Nodos** tal que: $\mathbf{V} = [\mathbf{M}_0]$
- (ii) \mathbf{A} es un conjunto de **Arcos** tales que: $\mathbf{V} \cap \mathbf{A} = \emptyset$
- (iii) \mathbf{N} es una **Función de Nodo** la cual está definida como:

$$\forall a \in \mathbf{A} : \mathbf{N}(a) \in [\mathbf{A} \rightarrow \mathbf{V} \times \mathbf{V}]$$
$$\mathbf{N}(a) = (\mathbf{M}_1, \mathbf{M}_2)$$

El Grafo de Alcanzabilidad tiene un nodo para cada marcaje alcanzable y un arco para cada paso que ocurre, el nodo fuente del arco es el marcaje inicial para el paso mientras que el nodo de destino es el marcaje final.

Proposición:

El Grafo de Alcanzabilidad satisface las siguientes propiedades:

(i) Cada secuencia de ocurrencia finita:

$$M_1 \xrightarrow{t_1} M_2 \xrightarrow{t_2} M_3 \cdots M_n \xrightarrow{t_n} M_{n+1}$$

Donde $M_i \in [M_0]$ y $t_i \in T$ para todo $i = 1, 2, \dots, n$

Con $[M_0]$ el conjunto de los marcajes alcanzables a partir de M_0 y T el conjunto de todas las transiciones.

Tiene asignado un camino dirigido entre cada par de nodos:

$$M_1(M_1, t_1, M_2)M_2(M_2, t_2, M_3)M_3 \cdots M_n(M_n, t_n, M_{n+1})M_{n+1}$$

Y cada camino dirigido finito tiene asignada una secuencia de ocurrencia.

2.1.2. Componentes Fuertemente Conectados

Un conjunto de nodos $V^* \subseteq V$ está fuertemente conectado [16] si y solo si:

(i) $\forall V_1, V_2 \in V^* \times V^* : DPF(V_1, V_2) \neq \emptyset$

DPF: es el conjunto de todos los caminos dirigidos finitos $DPF(V_1, V_2)$ que comienzan en V_1 y terminan en V_2 . En un Grafo de Alcanzabilidad, el **DPF** es un conjunto de arcos.

Un componente fuertemente conectado es el Subgrafo generado por un conjunto no vacío de nodos $V^* \subseteq V$, donde:

(ii) V^* está fuertemente conectado.

(iii) $\forall V' \subseteq V : (V' \text{ está fuertemente conectado} \wedge V^* \subseteq V') \Rightarrow V^* = V'$

El conjunto de todos los componentes fuertemente conectados (Strongly Connected Components) se denota por **SCC** por su siglas en inglés y se denomina Grafo de Componentes Fuertemente Conectados.

Explicación:

Un SCC es un subgrafo del Grafo de Alcanzabilidad, en el cual se especifican los grupos de nodos para los cuales es posible hallar un camino desde uno cualquiera de ellos hacia otro y viceversa.

2.1.3. Propiedades de Alcanzabilidad

Las siguientes propiedades prueban la Alcanzabilidad [4] y son válidas para todo $\mathbf{M}_1, \mathbf{M}_2 \in [M_0]$:

- (i) $V = [M_0]$
- (ii) $\mathbf{M}_2 \in [M_1] \Leftrightarrow \text{DPF}(\mathbf{M}_1, \mathbf{M}_2) \neq \phi$
- (iii) $\mathbf{M}_2 \in [M_1] \Leftrightarrow |\text{SCC}| = 1$

Explicación:

La propiedad (i) define si un nodo es o no alcanzable a partir de un marcaje inicial \mathbf{M}_0 , (ii) establece que un marcaje \mathbf{M}_2 es alcanzable a partir de un marcaje \mathbf{M}_1 sí y sólo si el Grafo de Alcanzabilidad tiene algún camino dirigido que permita ir de \mathbf{M}_1 a \mathbf{M}_2 , la propiedad (iii) precisa el caso en el cual el Grafo de Alcanzabilidad tiene un único componente fuertemente conectado, en esta situación, todos los marcajes del modelo son alcanzables y mas aún, todos son alcanzables desde cualquier otro marcaje.

2.1.4. Propiedades de Acotación

Sea $\mathbf{X} \subseteq \mathbf{T}$ una transición de una PN y $n \in \mathbb{N}$ un número entero positivo. Se tienen las siguientes reglas de prueba para la acotación:

- (i) n es una Cota Superior de marcas para \mathbf{X} sí y sólo si: $\forall \mathbf{M} \in [M_0] : |(M|X)| \leq n$
- (ii) n es una Cota Inferior de marcas para \mathbf{X} sí y sólo si: $\forall \mathbf{M} \in [M_0] : |(M|X)| \geq n$

En consecuencia una PN es acotada si el número de marcas en todos los lugares posee una cota superior.

De las definiciones (i) y (ii) se siguen las siguientes reglas de prueba, válidas para todo $\mathbf{X} \subseteq \mathbf{T}$ y todo $\mathbf{p} \in \mathbf{P}$:

(iii) Máxima Cota Superior(\mathbf{X}): $\text{MCS}(\mathbf{p}) = \max_{\mathbf{M} \in V} |M(\mathbf{p})|$

(iv) Mínima Cota Inferior(\mathbf{X}): $\text{MCI}(\mathbf{p}) = \min_{\mathbf{M} \in V} |M(\mathbf{p})|$

Explicación:

La definición anterior establece que una PN es acotada si el número de marcas en cada uno de los lugares es finito, esto es, la red tiene un número máximo de marcas $\mathbf{MCS(p)}$ para cada uno de los lugares, garantizándose así que el Grafo de Alcanzabilidad sea finito.

Si en una PN $\mathbf{MCS(p)} = \max_{M \in V} |M(p)| = 1$ la red se dice que es 1-acotada y por lo tanto es Segura [17].

Verificando que la red es acotada o segura, se puede garantizar que en el modelo no hay saturación en los lugares y por lo tanto los procesos se realizan de manera efectiva.

La red de Petri de la Figura 81 es acotada ya que el número máximo de marcas en cada uno de los lugares $\mathbf{MCS(p)} = 1$ con ello puede concluirse que la red es 1-acotada.

2.1.5. Propiedades de Reversibilidad

Sea $\mathbf{M} \in \mathbf{M}$ un marcaje y $\mathbf{X} \subseteq \mathbf{M}$ un conjunto de marcajes:

(i) \mathbf{M} es un *Marcaje Reversible* sí y sólo si: $\forall M' \in [M_0] : M \in [M']$

(ii) \mathbf{X} es un *Espacio Reversible* sí y sólo si: $\forall M' \in [M_0] : X \cap [M'] \neq \emptyset$

Explicación:

La propiedad (i) indica que si un marcaje \mathbf{M} es reversible, entonces siempre es posible ir del marcaje $\mathbf{M'}$ al marcaje \mathbf{M} , siendo $\mathbf{M'}$ un marcaje alcanzable desde el marcaje inicial $\mathbf{M_0}$. Por su parte (ii) indica que para que un conjunto \mathbf{X} sea un Espacio Reversible este debe tener por lo menos un elemento \mathbf{M} (pues de lo contrario será un conjunto vacío).

En la figura 3 se encuentra representado un sistema en el cual es posible siempre regresar a cualquier marcaje, todos los marcajes son alcanzables a partir del marcaje inicial $\mathbf{M_0}$ y este a su vez es siempre alcanzable desde cualquier otro, de esta manera no habrán bloqueos en el modelo, todos los marcajes son reversibles y la red es cíclica.

2.1.6. Propiedades de Vivacidad

Sea $\mathbf{M} \in \mathbf{M}$ un marcaje y $\mathbf{X} \subseteq \mathbf{T}$ un conjunto de transiciones:

(i) \mathbf{M} es un marcaje *Muerto* sí y sólo si ningún elemento se habilita en dicho marcaje, es decir:

$$\forall x \in \mathbf{T} : \neg M[x]$$

(ii) \mathbf{X} es un elemento *Muerto* en \mathbf{M} sí y sólo si ningún elemento de \mathbf{X} se puede habilitar en \mathbf{M} , es decir:

$$\forall M' \in [M] \forall x \in X : \neg M'[x]$$

(iii) X es un elemento vivo sí y sólo si no hay marcajes alcanzables en los cuales X sea muerto, es decir:

$$\forall M' \in [M_0] \exists M'' [M'] \exists x \in X : M''[x]$$

Explicación:

Como consecuencia de la propiedad (i) se dice que un marcaje es muerto, si el nodo correspondiente en el Grafo de Alcanzabilidad es terminal, es decir, no tiene arcos de salida. Las propiedades (ii) y (iii) indican las condiciones en las cuales las transiciones de un modelo resultan ser muertas o vivas, respectivamente.

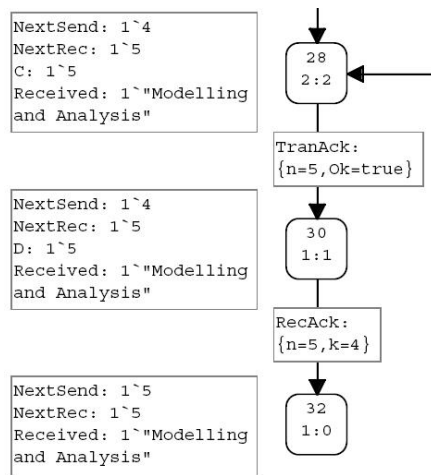


Figura 81. Fragmento de un Grafo de Alcanzabilidad

El subgrafo presentado en la Figura 81 es una porción de un Grafo de Alcanzabilidad. El nodo 32 es un estado de bloqueo ya que no posee arcos de salida, este ilustra claramente lo que es un nodo terminal.

2.1.7. Propiedades de Imparcialidad

Sea $X \subseteq T$ un conjunto de transiciones y σ una secuencia infinita de ocurrencias:

(i) X es *Imparcial* para σ sí y sólo si X tiene infinitamente muchas ocurrencias, es decir:

$$OCC_X(\sigma) = \infty$$

(ii) X es *Objetivo* para σ sí y sólo si un número infinito de habilitaciones implica un número infinito de ocurrencias, es decir:

$$\text{ENB}_X(\sigma) = \infty \Rightarrow \text{OCC}_X(\sigma) = \infty$$

(iii) X es *Justo* para σ sí y sólo si una habilitación de manera persistente genera al menos una ocurrencia, es decir:

$$\forall i \geq 1 : (\text{ENB}_{X,i}(\sigma) \neq 0 \Rightarrow \exists k \geq i : [\text{ENB}_{X,k}(\sigma) = 0 \vee \text{OCC}_{X,k}(\sigma) \neq 0])$$

Se utilizó la notación $\text{OCC}_X(\sigma)$ para denotar el número total de ocurrencias de los elementos de X en σ mientras que el símbolo $\text{ENB}_X(\sigma)$ es usado para indicar el número total de habilitaciones de los elementos de X, de igual manera, $\text{OCC}_{X,k}(\sigma)$ expresa el número de elementos de X que ocurren en el k-ésimo paso y $\text{ENB}_{X,i}(\sigma)$ expresa el número de elementos de X que se habilitan en el i-ésimo paso.

Las siguientes son consecuencias directas de la anterior definición:

- Como consecuencia de la propiedad (i), una transición es *Imparcial* para una secuencia σ de ocurrencia infinita si dicha transición ocurre infinitamente. Esto significa que la transición siempre se habilitará y ocurrirá cuando la red esté evolucionando, así, una transición imparcial sólo dejará de habilitarse y ocurrir cuando el sistema esté detenido.

Las transiciones **T1** y **T6** de la Figura 82 tienen la propiedad de que siempre que la marca presente en **P1** evolucione, estas se habilitarán y ocurrirán. Claramente las dos transiciones están obligadas a ocurrir siempre, en consecuencia, son *Imparciales*.

- De acuerdo con la propiedad (ii), una transición es *Objetiva* si cada vez que esta se habilita, ocurre. Lo anterior no implica que la transición siempre se habilite en una secuencia de ocurrencia determinada.

T3 y **T5** en la Figura 82 son mutuamente excluyentes, de manera que cuando ocurre **T3** no puede ocurrir **T5** y viceversa, sin embargo, cada vez que una de ellas se habilita, ocurre. **T3** y **T5** son entonces *Objetivas*.

- Según la propiedad (iii), una transición es *Justa* si ante múltiples habilitaciones esta ocurre finitas veces, en consecuencia, no siempre que la transición se habilita, ocurre, pero debe ocurrir al menos una vez en la evolución de la red. Esta propiedad es característica de las transiciones de salida de los lugares donde hay conflicto.

En la PN de la figura 3, **T2** y **T4** se habilitan simultáneamente pero solo una de las dos ocurrirá, de esta manera, es posible por ejemplo que en 100 ciclos completos, **T4** ocurra 99 veces mientras que **T2** lo haga solo una vez. Lo mismo puede suceder en sentido inverso.

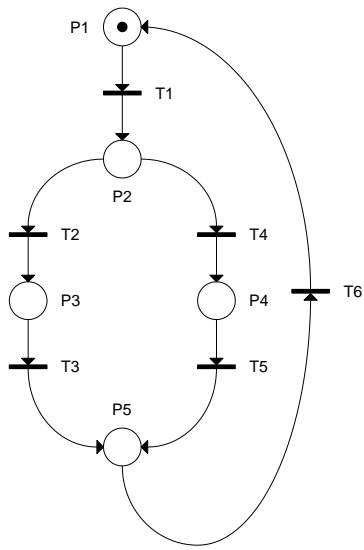


Figura 82. Red de Petri ilustrativa

ANEXO 3

REPORTES DE VALIDACIÓN Y VERIFICACIÓN DE MODELOS

3.1 Estructura del reporte

Estadísticas (Statistics): Muestra información general sobre el número de elementos del Grafo de Alcanzabilidad (Occ Graph) y el grafo de Componentes Fuertemente Conectados (Scc Graph).

Propiedades de Acotación (Boundedness Properties): Detalla la máxima y mínima distribución de marcas en cada uno de los lugares.

Integer Bounds: Indica el número máximo y mínimo de marcas que puede tener cada uno de los lugares de la red.

Multiset Bounds: Indica los valores que pueden tener las marcas en cada lugar (colores asignados a ellas). **Upper** es el máximo multiconjunto de marcas que pueden tener los lugares, mientras que **Lower** es el mínimo.

Propiedades de Reversibilidad (Home Properties): Especifican la capacidad que tiene el modelo para regresar a un estado determinado, siguiendo una secuencia de ocurrencia.

Marcaje Reversible (Home Marking): Muestra a cuáles estados es posible siempre regresar.

Propiedades de Vivacidad (Liveness Properties): Especifica el comportamiento dinámico de los marcajes y las transiciones del modelo.

Marcajes Muertos (Dead Markings): Indica qué marcajes no tienen transiciones habilitadas. En Design/CPN un Marcaje Muerto es, por sus características, un Marcaje Reversible.

Instancias de Transiciones Muertas (Dead Transitions Instances): Detalla cuáles son las transiciones del modelo, que nunca se habilitan para todo marcaje alcanzable. Si una transición se habilita por lo menos una vez para algún marcaje, ésta no es una transición muerta.

Instancias de Transiciones Vivas (Live Transitions Instances): Muestra las transiciones que siempre se pueden habilitar una vez más. Si una transición no es muerta pero se dispara finitas veces, se dirá que es una *transición parcialmente viva*. En este caso el reporte será:

Dead Transitions Instances: **None**

Live Transitions Instances: **None**

Propiedades de Imparcialidad (Fairness Properties): Brindan información acerca de la habilitación y ocurrencia de cada una de las transiciones de acuerdo a la semántica del modelo.

En una misma instancia, cada transición de manera inherente posee una de las características de Imparcialidad (Impartial), Objetividad (Fair) o Justo (Just), se explican en más detalle en el Holón Recurso

CPN Tools state space report for:
 D:\Universidad\Maestria\Ejecución de tesis\Escribir
 Tesis\Validación y Verificación\Holon_Recurso.cpn

Statistics

State Space

Nodes: 10
 Arcs: 23
 Secs: 0
 Status: Full

Scc Graph

Nodes: 1
 Arcs: 0
 Secs: 0

Boundedness Properties

Best Integer Bounds

	Upper	Lower
Holon_Recurso'Arranque 1	1	0
Holon_Recurso'Degradado 1	1	0
Holon_Recurso'Disponibilidad_degradada 1	1	0
Holon_Recurso'Disponible 1	1	0
Holon_Recurso'Falla 1	1	0
Holon_Recurso'Mtto_Programado 1	1	0
Holon_Recurso'Mtto_preventivo 1	1	0
Holon_Recurso'Normal 1	1	0
Holon_Recurso'Parada 1	1	0
Holon_Recurso'Reservado 1	1	0

Best Upper Multi-set Bounds

```
Holon_Recurso'Arranque 1
                      1`m
Holon_Recurso'Degradado 1
                      1`m
Holon_Recurso'Disponibilidad_deagradaada 1
                      1`m
Holon_Recurso'Disponible 1
                      1`m
Holon_Recurso'Falla 1
                      1`m
Holon_Recurso'Mtto_Programado 1
                      1`m
Holon_Recurso'Mtto_preventivo 1
                      1`m
Holon_Recurso'Normal 1
                      1`m
Holon_Recurso'Parada 1
                      1`m
Holon_Recurso'Reservado 1
                      1`m
```

Best Lower Multi-set Bounds

```
Holon_Recurso'Arranque 1
                      empty
Holon_Recurso'Degradado 1
                      empty
Holon_Recurso'Disponibilidad_deagradaada 1
                      empty
Holon_Recurso'Disponible 1
                      empty
Holon_Recurso'Falla 1
                      empty
Holon_Recurso'Mtto_Programado 1
                      empty
Holon_Recurso'Mtto_preventivo 1
                      empty
Holon_Recurso'Normal 1
                      empty
Holon_Recurso'Parada 1
                      empty
Holon_Recurso'Reservado 1
                      empty
```

Home Properties

Home Markings
All

Liveness Properties

Dead Markings
None

Dead Transition Instances
None

Live Transition Instances
All

Fairness Properties

Holon_Recurso'Fin_Mtto_Preventivo 1
Fair
Holon_Recurso'Iniciar 1
Just
Holon_Recurso'Iniciar_1 1
Just
Holon_Recurso'Iniciar_2 1
Just
Holon_Recurso'Mtto 1 Just
Holon_Recurso'Mtto_1 1 Just
Holon_Recurso'Mtto_2 1 Fair
Holon_Recurso'Mtto_Preventivo 1
Just
Holon_Recurso'Poner_Disponibilidad_Degradada 1
Just
Holon_Recurso'Poner_Disponible 1
Just
Holon_Recurso'Poner_Disponible_1 1
Just
Holon_Recurso'Poner_en_Degradado 1
Just
Holon_Recurso'Poner_en_Falla 1
Just
Holon_Recurso'Poner_en_Falla_1 1
Just
Holon_Recurso'Poner_en_Falla_2 1
Just


```

Holon_Recurso'Poner_en_Falla_3 1
      Just
Holon_Recurso'Poner_en_Falla_4 1
      Just
Holon_Recurso'Poner_en_Falla_5 1
      Just
Holon_Recurso'Reservar 1
      Just
Holon_Recurso'Secuencia_Arranque 1
      Just
Holon_Recurso'Secuencia_Parada 1
      Just
Holon_Recurso'Sin_Falla 1
      Fair
Holon_Recurso'sp_set 1 Just

```

3.2 Misión del recurso

CPN Tools state space report for:
D:\Universidad\Maestria\Ejecución de tesis\Escribir
Tesis\Validación y Verificación\Modulo_Mision.cpn
Report generated: Sat Aug 08 21:08:26 2009

Statistics

State Space

```

Nodes: 8
Arcs: 18
Secs: 0
Status: Full

```

Scc Graph

```

Nodes: 3
Arcs: 2
Secs: 0

```

Boundedness Properties

Best Integer Bounds

	Upper	Lower
Mision'Abortado 1	1	0

Mision'Avance_Normal	1	1	0
Mision'Desviada	1	1	0
Mision'Incumplimiento	1	1	0
Mision'Incumplimiento_autorizado	1	1	0
Mision'Riesgo	1	1	0
Mision'Sin_Compromiso	1	1	0
Mision'Terminada	1	1	0

Best Upper Multi-set Bounds

Mision'Abortado	1	1`m
Mision'Avance_Normal	1	1`m
Mision'Desviada	1	1`m
Mision'Incumplimiento	1	1`m
Mision'Incumplimiento_autorizado	1	1`m
Mision'Riesgo	1	1`m
Mision'Sin_Compromiso	1	1`m
Mision'Terminada	1	1`m

Best Lower Multi-set Bounds

Mision'Abortado	1	empty
Mision'Avance_Normal	1	empty
Mision'Desviada	1	empty
Mision'Incumplimiento	1	empty
Mision'Incumplimiento_autorizado	1	empty
Mision'Riesgo	1	empty
Mision'Sin_Compromiso	1	empty
Mision'Terminada	1	empty

Home Properties

Home Markings

[3]

Liveness Properties

Dead Markings
[3]
Dead Transition Instances
None
Live Transition Instances
None

Fairness Properties

Mision'Autorizacion 1 Just
Mision'Autorizacion_1 1
Just
Mision'preguntar 1 Just
Mision's0 1 Fair
Mision'sa 1 Just
Mision'sa_1 1 Just
Mision'sb 1 Just
Mision'sd 1 Just
Mision'sd_1 1 Just
Mision'sd_2 1 Just
Mision'sd_3 1 Just
Mision'sn 1 Just
Mision'sn_1 1 Just
Mision'sn_2 1 Fair
Mision'sn_3 1 Just
Mision'sp_1 1 Just
Mision'spd 1 Just
Mision'sr 1 Just