

Aplicaciones Industriales de los Sistemas Multiagente

Vicente J. Botti Navarro
Adriana Giret Boggino
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera s/n
46022. Valencia
{vbotti, agiret}@dsic.upv.es

1. Introducción

La tecnología de agentes/sistemas multiagente (SMA) está realizando importantes aportaciones en la resolución de problemas en diversos dominios (comercio electrónico, subastas electrónicas, medicina, bolsa, etc.), donde aproximaciones tradicionales no proporcionan soluciones suficientemente satisfactorias. En concreto, la industria de fabricación es uno de los dominios donde la tecnología de agentes/SMA proporciona una forma natural para resolver problemas que son inherentemente distribuidos.

Este capítulo tiene como objetivo realizar un estudio del estado del arte de la aplicación de la tecnología de agentes en el área de las empresas de fabricación. No es objetivo de este capítulo proporcionar un estudio exhaustivo de la teoría y tecnología de agentes/SMA. Estos estudios pueden ser encontrados en otros capítulos del libro o en otras publicaciones como (Wooldridge and Jennings 1995; Wooldridge 2002). Tampoco es objetivo de este capítulo proporcionar un estudio de temas relacionados con las industrias de fabricación como pueden ser: los entornos cooperativos para diseño de ingeniería (en Shen and Barthès 1996b, se puede consultar una detallada revisión); los sistemas de ingeniería simultáneos (en Molina et al 1995, se puede encontrar una revisión); o robótica multiagente (Jenkin et al 1996, proporciona una interesante taxonomía). Por ello nos centraremos en analizar los requerimientos de una empresa de fabricación actual (apartado 2), requerimientos que justifican la necesidad de aplicar nuevos paradigmas capaces de satisfacerlos. En el apartado 3 se presentan los dos paradigmas más prometedores para satisfacer estos requerimientos, los sistemas multiagente y los sistemas holónicos. Dado que los sistemas multiagente son tratados en otros capítulos, en este apartado nos centraremos en aquellos aspectos más relacionados con los sistemas de fabricación, mientras que los sistemas holónicos serán tratados con mayor detalle. En el apartado 4 se realiza un análisis comparativo de los sistemas multiagente y los sistemas holónicos, que nos lleva a concluir que ambos paradigmas tienen muchas características en común y sería posible fusionarlos en un único paradigma. Para finalizar, se proporciona una detallada relación de aplicaciones y proyectos de investigación de aplicaciones de los sistemas multiagente a los sistemas de fabricación, relación que va documentada con abundante bibliografía, lo que permite al lector poder profundizar en cualquiera de los temas de interés, dado que la limitación de este capítulo imposibilita una descripción más amplia.

2. La Empresa de Fabricación de Nueva Generación

La globalización, que afecta a todo el planeta, no iba a ser una excepción en las empresas de fabricación. La competitividad global y la rapidez con que cambian los requerimientos de los clientes están forzando importantes cambios en los estilos de

producción y configuración de las organizaciones de producción. Cada vez más, la planificación de producción, el scheduling y los mecanismos de control secuenciales y centralizados tradicionales no son suficientemente flexibles para responder a estos nuevos estilos de producción que cambian continuamente y a las variaciones que se producen continuamente en los requerimientos del producto. Las aproximaciones tradicionales no permiten una fácil extensión y reconfiguración de los sistemas de fabricación. La rigidez de la clásica organización jerárquica centralizada, puede también dar lugar a la paralización de la actividad de una fábrica por un fallo en un único punto de la cadena de producción. Por ello, las industrias de fabricación necesitan adoptar arquitecturas abiertas que integren sus actividades con suministradores y clientes en amplias redes de cadena de suministros. Para poder competir de forma efectiva en los mercados actuales, los fabricantes requieren interactuar con sus clientes, proveedores y servicios de forma rápida y económica. Aproximaciones clásicas como la ingeniería del conocimiento, con grandes bases de conocimiento, se convierten en inapropiadas debido a la naturaleza altamente distribuida de estos sistemas. La tecnología de agentes se presenta como una forma natural de superar tales problemas y diseñar e implementar entornos de fabricación inteligente distribuidos (Jennings et al 1995; Jennings and Wooldridge 1998). En la literatura podemos encontrar diversos trabajos que aplican la tecnología de agentes en la integración de empresas de fabricación, en la gestión de la cadena de suministro, en planificación de la fabricación, en scheduling y control, en manipulación de materiales, y sistemas de fabricación holónicos. En este capítulo, se proporciona una breve revisión de algunos de los proyectos relacionados con esta área, y se comentan algunos resultados más significativos en el desarrollo de sistemas de fabricación basados en agente.

2.1 Requerimientos de Empresas de Fabricación de Nueva Generación

Como se ha comentado en el apartado anterior, la globalización ha forzado a las empresas de fabricación a cambiar sus estrategias de producción para así poder adaptarse a la competitividad global, a las nuevas necesidades de innovación e introducción del producto, y responder rápidamente a los cambios del mercado. La nueva generación de sistemas de fabricación deberá tener como objetivo la optimización de factores como tiempo versus costes o calidad. Tales sistemas de fabricación deberán satisfacer los requerimientos fundamentales [<http://imsg.enme.ucalgary.ca/publication/abm.htm>]:

Integración de empresas

Con objeto de soportar la competitividad global y responder a los rápidos cambios del mercado, una empresa de fabricación tiene que estar integrada con sus sistemas de gestión (compras, pedidos, diseño, producción, planificación, control, transporte, recursos, recursos humanos, materiales, calidad, etc.) que son, en general, entornos software y hardware heterogéneos. Tal integración puede realizarse por medio de sistemas de planificación táctica, que están basados en sistemas distribuidos basados en el conocimiento para relacionar gestión de demanda directamente con recursos y planificación de capacidad.

Cooperación

Las empresas de fabricación requieren una plena cooperación con sus suministradores y clientes para así optimizar el suministro de materiales, fabricación de accesorios, comercialización del producto final, etc. Tal cooperación deberá realizarse de forma eficiente y deberá responder rápidamente a cualquier posible cambio.

En un sistema cooperativo, las cadenas dinámicas de eventos están embebidas en procesos de información concurrentes. Los requerimientos impuestos por los pedidos de clientes, decisiones de dirección, y etapas de diseño están integrados con las tareas de planificación de la producción y la asignación de recursos en una estructura compleja que incorpora decisiones de alto nivel en las actividades de planificación. Este es esencialmente un entorno de información-procesamiento concurrente. La cooperación es un requerimiento imperativo en cualquier modelo funcional de sistemas de fabricación avanzados.

Integración de persona con software y hardware

Personas y computadores deben ser integrados para trabajar colectivamente en diversas etapas de desarrollo del producto, con acceso al conocimiento y la información requerida. Para soportar estas necesidades y aumentar las capacidades de decisión del sistema hay que integrar fuentes heterogéneas de información. Se requieren entornos de comunicación bi-direccional que permitan una comunicación veloz y efectiva entre humanos y computadores, lo que facilitará su cooperación.

Agilidad

La globalización económica y las expectativas de expansión de mercados están transformando rápidamente los entornos de fabricación. Hay que realizar considerables esfuerzos en la reducción de ciclos del producto, para permitir responder con mayor rapidez a los deseos de los clientes. En esta nueva escala de transformación económica, las corporaciones están reorientando progresivamente sus estrategias a expandir su 'share' de mercado y a integrar la fabricación 'ágil' en sus capacidades de producción. La fabricación 'ágil' es la habilidad de adaptarse, en un entorno de fabricación, a continuos e inesperados cambios y por ello es un componente clave en las estrategias de fabricación para la competitividad global. Para conseguir agilidad, las capacidades de fabricación deben establecer asociaciones convenientes con socios heterogéneos. Idealmente, los socios son contratados 'on the fly' sólo por el tiempo requerido para realizar las tareas específicas. Este tipo de interacción puede también ser utilizada para planificar estrategias de 'a largo término'. La agilidad proporcionará mayor flexibilidad a la organización de fabricación sin incurrir en grandes o diversas inversiones industriales.

Escalabilidad

La escalabilidad es una importante propiedad para los sistemas de fabricación avanzados. Escalabilidad significa que en la organización pueden ser incorporados más recursos cuando son necesarios. Esta propiedad debería satisfacerse en cualquier nodo de trabajo del sistema y a cualquier nivel dentro de los nodos. La expansión de recursos debería ser posible sin romper enlaces organizacionales previamente establecidos.

Para identificar e incorporar nuevos componentes, se requieren registros de conocimiento organizacional. Cuando nuevos componentes físicos llegan al sistema, son creadas entidades representativas para actuar como imagen de los componentes a través de sus ciclos de vida. La habilidad de añadir nuevos componentes incrementalmente permite al sistema responder flexiblemente a una amplia variedad de solicitudes. Por ejemplo, el sistema puede añadir dinámicamente más inteligencia o incrementar su capacidad de fabricación para abastecer a un mercado que se expande rápidamente o reducir su capacidad para ajustar 'downwards' durante periodos de baja demanda. Cuando se eliminan del sistema componentes físicos, para mantenimiento o

cualesquiera otras razones, también son eliminados del registro del sistema. Mecanismos robustos de registro son necesarios para proporcionar la integración de nuevos componentes o la eliminación de los existentes.

Reconfiguración dinámica

Tanto las personas como las entidades artificiales que pertenecen a un sistema de fabricación necesitan estar alerta a los cambios del entorno. Cada etapa de la planificación de la fabricación se ve afectada por variaciones dinámicas que provienen tanto de fuentes internas como externas. En sistemas de fabricación convencionales, las entradas de los clientes activan una secuencia de eventos, empezando con operaciones de planificación. En este nivel, los pedidos son procesados de acuerdo a estados preestablecidos (los cuales incluyen la especificación del diseño del producto, gestión de materiales, planificación y disponibilidad de la capacidad de fabricación, y la preparación de los costes de producción). El proceso de planificación también activa requerimientos para la subcontratación de servicios externos.

Convencionalmente, el proceso de planificación avanza progresivamente a través de una serie de evaluaciones secuenciales que se corresponden con las condiciones del sistema en un futuro próximo. Por ello, posteriores variaciones del estado del entorno pueden invalidar estos planes. Este tipo de sistema secuencial resulta caro, ya que hay una tendencia a reutilizar recursos de cómputo en evaluaciones redundantes y repetitivas.

Eventualmente, hay una transición desde el proceso de planificación a una segunda área principal de control de fabricación, actividad en la que se asignan tiempos de ejecución a planes de fabricación. En este estado son introducidas muchas variaciones, estas afectan a la estabilidad del sistema y a su habilidad para ejecutar los planes de acuerdo a lo planificado. Esta compleja etapa restringe la habilidad del sistema de reconfigurarse para hacer frente a cambios dinámicos e inesperados.

La extensibilidad es posible en tales sistemas convencionales de fabricación, pero requiere una importante reconfiguración del sistema. Las interconexiones estrechamente acopladas entre los componentes existentes en el sistema hacen muy tedioso el ajustar cada módulo de procesamiento a nuevos componentes disponibles. Similarmente, la eliminación de componentes implica un considerable reajuste del sistema.

Capitalización y distribución del conocimiento

La adquisición y distribución eficiente del conocimiento de cada aspecto de la organización (finanzas, marketing, diseño y fabricación) junto con su uso efectivo, producirá notables avances en la investigación de mercados, desarrollo de productos y procesos, planificación y scheduling de la producción, y finalmente en respuesta a clientes.

Los problemas más obvios en procesamiento de la información, observados en sistemas convencionales de fabricación, son la gestión de información y control de producción muy centralizados, consecuencia de la necesidad de mantener una vista general del sistema con el fin de minimizar costes (y en consecuencia tener más ventajas en el mercado). Las bases de datos centralizadas son usadas comúnmente para acumular información del sistema que permita establecer planes de producción y prever futuros requerimientos. Potentes computadores centralizados procesan grandes cantidades de datos para crear planes de producción y schedules. Las transacciones entre diversos recursos son forzadas a realizarse a través de la unidad de control centralizado. Todas las actividades en sistemas de fabricación convencionales están limitadas por la

corrección y estabilidad de los componentes de procesamiento centralizados, produciendo una infraestructura frágil.

Aunque en el pasado los sistemas de procesamiento de la información secuenciales y centralizados han minimizado costes de hardware y software, su estructura central no es adecuada dada la inherente naturaleza distribuida del flujo de información concurrente en fabricación ágil.

La distribución del conocimiento de producción dará más valor a la modularidad del sistema y facilitará tanto la integración como la reconfiguración. El aumento de modularidad reduce la complejidad de organización del conocimiento manteniendo el conocimiento localmente (divide y vencerás). La información local puede ser procesada concurrentemente, evitando así las limitaciones del procesamiento secuencial de la información.

Ingeniería Concurrente

Asegurar la viabilidad de la fabricación del producto constituye el primer paso de la implementación de la ingeniería concurrente. Especificaciones geométricas y funcionales, disponibilidad de materias primas, y la capacidad y disponibilidad de recursos de fábrica tienen una gran influencia en la viabilidad de la fabricación. Un diseño puede ser fabricado bajo una combinación de requerimientos de producto y recursos de planta, pero no bajo otros. La selección y disponibilidad de material de stock a partir del cual el producto será fabricado influye en el número de etapas intermedias requeridas, y en consecuencia en el coste de producción. La capacidad y disponibilidad de recursos de planta influye en el plan de proceso a utilizar, y de nuevo en los costes de producción. Por ello, todos estos aspectos tienen que ser considerados simultáneamente en una ingeniería concurrente efectiva.

Entornos heterogéneos

Los sistemas de fabricación necesitan adaptarse a software y hardware heterogéneo tanto en sus entornos de fabricación como de información.

Interoperabilidad

Los entornos de información heterogéneos pueden utilizar diferentes lenguajes, representar datos con diferentes lenguajes de representación y modelos, y ejecutarse en diferentes plataformas. Los subsistemas y componentes de tales entornos heterogéneos deberán ejecutarse de forma eficiente.

Tolerancia a fallos

El sistema deberá ser tolerante a fallos tanto a nivel de sistema como de subsistema, así como detectar y recuperarse de fallos del sistema en cualquier nivel y minimizar sus impactos sobre el entorno de trabajo.

3. Sistemas Multiagente y Sistemas Holónicos. Aproximación para modelar sistemas de fabricación

Las Técnicas de Inteligencia Artificial han sido utilizadas en la Fabricación Inteligente desde hace más de 20 años. Sin embargo, los desarrollos recientes en el área de sistemas multiagentes han proporcionado nuevos e interesantes resultados de especial interés en esta área. En los últimos 10 años, se ha estado aplicando tecnología de agentes a la

integración de empresas de fabricación y administración de cadenas de suministro, planificación de producción, asignación de recursos y control, manipulación de materiales, y desarrollo de nuevos tipos de sistemas de producción tales como sistemas holónicos de producción.

Un sistema multiagente de control de fabricación usualmente requiere del uso de técnicas especiales de razonamiento y coordinación. Dependiendo de los objetivos de la producción y los tipos de procesos de producción, pueden ser necesarias diferentes arquitecturas de control y estrategias a fin de controlar de manera óptima el proceso de fabricación. Sin embargo, a pesar de las necesidades especiales de una aplicación de fabricación particular, cualquier sistema de control industrial debe satisfacer unos requisitos generales.

3.1 Requisitos Funcionales

Los sistemas de control de fabricación son sistemas grandes, complejos que son diseñados para desempeñar una tarea claramente definida en un ambiente estandarizado y bien estructurado. Aunque los procesos de producción experimentan varios cambios y perturbaciones, el grado de incertidumbre e impredecibilidad no es comparable a aquellos sistemas del espacio, del tráfico, o aplicaciones de servicio. Como consecuencia, las aplicaciones de fabricación requieren de menos deliberación mental y social que las aplicaciones multiagente típicas. Esto es particularmente verdadero para las categorías mentales tales como deseos, intenciones, o intenciones conjuntas, incluyendo su razonamiento asociado. Su uso es posible, pero no apropiado para la mayoría de las tareas de control de producción.

Más aún, todos los agentes de fabricación cooperan a fin de lograr los objetivos globales de fabricación. Con respecto a estos objetivos, un agente nunca rechaza de manera deliberada la cooperación con otro agente. Sólo rechaza su ejecución, cuando las acciones solicitadas son imposibles o fuertemente desventajosas para el proceso de producción. En este sentido, los agentes de producción son semi-autónomos. Esto nos lleva al primer requisito:

Requisito I: *Los sistemas de control de fabricación requieren agentes semi-autónomos. Los agentes deben razonar sobre el comportamiento del sistema de fabricación, pero no sobre sus propias actitudes mentales o aquellas de otras unidades de control.*

Este requisito no niega la creación de categorías mentales dentro de un agente, como las intenciones. Únicamente establece que el agente no tiene que razonar acerca de ellos.

El segundo requisito trata con el tipo de comportamiento que la unidad de control debe exhibir. Las unidades de control de producción están continuamente enfrentadas a alto grado de eventos repetidos que son conocidos, pero impredecibles. Este flujo de eventos debe ser manejado de manera efectiva y dentro de limitaciones temporales. El manejo de los eventos puede consecuentemente ser fijado de ante mano con la ayuda de rutinas, mientras que sólo el inicio y la ejecución de las rutinas deben ser realizadas de manera 'on-line'. El conjunto de eventos y su patrón de ocurrencia cambia con el tiempo pero sólo lentamente. Estos cambios a largo plazo son causados principalmente por cambios tecnológicos de productos y sistemas de producción.

Requisito II: *Las unidades de control de fabricación principalmente requieren de un comportamiento basado en rutinas que es al mismo tiempo efectivo y oportuno (timely). Este comportamiento puede ser tanto configurable o auto-adaptativo.*

3.2 Requisitos de Ingeniería del Software

Aparte de los requisitos funcionales, cualquier sistema de control que se supone será utilizado en un ambiente de fabricación productivo debe satisfacer estándares industriales generales. Estos estándares especifican, entre otras cosas, requisitos para fiabilidad, tolerancia a fallos, diagnóstico, y mantenibilidad. En particular, los sistemas de control deben lograr ciertos grados de fiabilidad que garantizan una operación continua. Esto es igualmente verdadero para el software de control. La confiabilidad del producto, en cambio, se logra únicamente si el proceso de desarrollo del software se realiza siguiendo una metodología de ingeniería, en lugar de desarrollarlo de manera ad hoc.

Los requisitos son los siguientes:

Requisito III: *Los métodos de programación deben proveer encapsulación de datos y procesos.*

Requisito IV: *Los programas de control deben tener una semántica clara. Adicionalmente, el comportamiento de un agente debería ser completamente especificado por su programa de control.*

Requisito V: *Un método o metodología de programación debería conducir directamente de una tarea de control a un programa de agente.*

3.3 Sistemas Multiagente

Un Sistema Multiagente (SMA) se compone de dos o más agentes que se comunican. Un agente es un sistema computacional autónomo y flexible, que es capaz de actuar en un entorno (Wooldridge and Jennings 1995). Flexible significa, que el agente es:

- ✓ Reactivo, reacciona al entorno en el cual se encuentra.
- ✓ Pro-activo, es capaz de cumplir su propia agenda (planes u objetivos).
- ✓ Social, es capaz de comunicarse con otros agentes a través de algún lenguaje.

Algunas propiedades que son atribuidas usualmente a los agentes en mayor o menor grado para resolver problemas particulares son (Nwana 1996, Franklin and Graesser 1996):

- ✓ Autonomía: los agentes pueden operar sin la directa intervención de humanos u otros agentes.
- ✓ Habilidad Social: los agentes son capaces de interactuar con otros agentes (humanos o no) a través de un lenguaje de comunicación de agentes.

- ✓ Racionalidad: un agente puede razonar acerca de datos percibidos a fin de calcular una solución óptima.
- ✓ Reactividad: los agentes son capaces de percibir estímulos del entorno y estos estímulos guían las acciones del agente en su entorno.
- ✓ Pro-actividad: los agentes no son sólo entidades que reaccionan a estímulos, sino también tienen un carácter emprendedor y pueden actuar guiados por sus propios objetivos.
- ✓ Adaptabilidad: esta característica está relacionada con el aprendizaje que un agente puede lograr y con su capacidad para cambiar su propio comportamiento basado en este aprendizaje.
- ✓ Movilidad: es la capacidad de un agente para moverse a través de una red.
- ✓ Veracidad: un agente no puede comunicar información falsa de manera deliberada.
- ✓ Benevolencia: un agente está dispuesto a ayudar a otros agentes si esto no está en contra de sus propios objetivos.

El estudio de Sistemas Multiagentes se inició hace cerca de 20 años, en el ámbito de la Inteligencia Artificial Distribuida (Distributed Artificial Intelligence - DAI). La DAI es un sub-campo de investigación de la Inteligencia Artificial (AI). La DAI estudia el comportamiento inteligente de grupo que se deriva a partir de la cooperación de entidades llamadas agentes. Estudia cómo un grupo de módulos cooperan para dividir y compartir el conocimiento del problema y cómo se desarrolla la solución. La DAI se centra en el comportamiento global, con un comportamiento prefijado de los agentes. Estudia las técnicas y el conocimiento necesarios para la coordinación y distribución del conocimiento y las acciones en un entorno multiagente.

El SMA estudia la coordinación del comportamiento inteligente entre un grupo de agentes inteligentes autónomos (posiblemente pre-existentes). Se centra en el comportamiento individual a partir del cual se deriva el comportamiento del sistema. Hoy en día los SMA forman un área de investigación muy activa y se los está empezando a utilizar en aplicaciones comerciales e industriales. Los SMA se centran en el comportamiento social de entidades inteligentes y se ocupan principalmente de estudiar modelos de comportamiento, estrategias de cooperación y coordinación, optimización del desempeño de tareas, aprendizaje a partir de experiencias propias, formación de coaliciones, etc.

En DAI el problema a ser resuelto se formula de manera centralizada, y luego se distribuye a nodos de computación locales. En SMA, en cambio, el problema se origina y se resuelve en el nodo local y la solución global resultante es emergente.

En resumen, los SMAs son una tecnología software general motivada en cuestiones fundamentales de investigación acerca de autonomía, cooperación, formación de grupo, etc. Se ocupa de responder preguntas tales como: "¿Qué puede ser hecho?", y "¿Cómo puede ser realizado?", y se aplica a una gran variedad de dominios: comercio

electrónico, control inteligente de producción, robótica, recuperación de la información, etc.

3.4. Fabricación basada en Agentes

La fabricación basada en Agentes (Agent Manufacturing - AM) no es un campo de investigación con una visión única. Es mas bien una colección de trabajos que propone mejorar el control existente en la infraestructura de las industrias con la ayuda de técnicas orientadas a agentes. Muchos trabajos, por ejemplo, se centran sobre el control de un sistema de producción flexible, como fue desarrollado dentro de la iniciativa CIM (Computer Integrated Manufacturing).

La primera aplicación de técnicas de modelado y coordinación orientadas a agentes para el control en la fabricación fue el prototipo de sistema de control de fábrica YAMS (Parunak et al 1985). En YAMS, la empresa de producción es modelada como una jerarquía de unidades de producción, llamadas estaciones de trabajo o celdas de trabajo. Una planta de montaje, por ejemplo, consiste de un bloque, una bomba de aceite, una cabeza, y un sistema de ensamblado flexible (FMS). La jerarquía, sin embargo, registra sólo composición y no control. La distribución de tareas hacia abajo en la jerarquía se hace por negociación.

Algunos trabajos en AM se han concentrado en un aspecto del control de producción. Bussmann (Bussmann 1996), por ejemplo, presentó un algoritmo de coordinación para las tareas de asignación de recursos y transporte. Como en YAMS, las tareas se anuncian a las unidades de transporte. A diferencia de YAMS, estas unidades realizan sólo un análisis local de la tarea y devuelven el resultado del análisis a un coordinador. El coordinador luego sintetiza los resultados en una asignación de recursos global. Hahndel et al (Hahndel et al, 1994), han desarrollado un enfoque completamente descentralizado para planificación de tareas de producción que consisten de varios pasos. Un agente de máquina responsable por la ejecución de un paso de producción de una tarea específica determina mediante negociación otro agente máquina que es capaz de ejecutar el siguiente paso. Este agente luego toma la responsabilidad de la tarea y continúa el proceso de planificación.

Muchos de los trabajos en AM se han centrado en desarrollar soluciones novedosas para la parte de procesamiento de la información de una infraestructura existente, como sistemas de producción flexibles. Sin embargo, el mayor potencial para conceptos como SMA no se encuentra solo dentro del procesamiento de la información, sino en un enfoque completamente nuevo para la producción en general, tal como se lleva a cabo en los Sistemas Holónicos de Fabricación. Por una parte, los sistemas multiagentes como un campo activo de investigación han producido un gran número de enfoques para el modelado del comportamiento individual y social. Esta tecnología puede ser utilizada por HM u otro paradigma específico de la fabricación.

3.5. Holones

El concepto holónico fue desarrollado por el filósofo Arthur Koestler en 1967 (Koestler 1967) para explicar la evolución de los sistemas biológicos y sociales. Por un lado, estos sistemas desarrollan, durante la evolución, formas intermedias estables que son auto confiables. Mientras que por otra parte, es difícil, en un sistema viviente, distinguir

entre el todo y la parte. Esta observación llevó a Koestler a proponer la palabra "holon", combinación de la palabra griega 'holos' que significa todo y el sufijo griego 'on' que significa partícula o parte como en protón o neutrón. Koestler observó que, en los organismos vivos y en las organizaciones sociales, no existen entidades enteramente auto-suficientes y aisladas. Toda entidad de una organización, tal como una célula simple en un animal o una familia en una sociedad, se compone de unidades más básicas (plasma y núcleo, padres e hijos), mientras que al mismo tiempo forman parte de una unidad de organización mayor (un tejido muscular o una comunidad). La potencia de las organizaciones holónicas, o holarquías, es que permiten la construcción de sistemas muy complejos que son, no obstante, eficientes en el uso de recursos, altamente resistentes ante disturbios (tanto internos como externos), y adaptables a cambios en su entorno. Dentro de una holarquía, los holones pueden crear y modificar dinámicamente jerarquías. Más aun, los holones pueden participar en múltiples holarquías al mismo tiempo. Las holarquías son recursivas en el sentido que un holón puede ser una holarquía que actúa como una unidad autónoma y cooperativa en la primera holarquía.

La estabilidad de los holones y de las holarquías se debe a que los holones son unidades auto-confiabiles, las cuales tienen un grado de independencia y pueden manejar circunstancias y problemas sobre su nivel particular de existencia, sin tener que solicitar asistencia a niveles superiores. Los holones pueden a su vez recibir instrucciones de y, en cierto grado, ser controlados por holones de niveles superiores. La característica de auto-confiabilidad asegura que los holones son estables, y capaces de sobrevivir ante disturbios. La subordinación a holones de niveles superiores asegura la operación efectiva del todo mayor.

3.6. Sistemas Holónicos de Fabricación (Holonc Manufacturing Systems - HMS)

Los Sistemas Holónicos de Fabricación (Holonc Manufacturing Systems - HMS) fueron propuestos, a principios de 1990 como un nuevo paradigma para la fabricación y ha recibido desde entonces gran atención tanto de la investigación académica como de la industria, por el consorcio HMS. Un Holón es definido por el consorcio HMS como 'un bloque de construcción autónomo y cooperativo de un sistema de fabricación para transformar, transportar, almacenar y/o validar información y objetos físicos (Van Leeuwen and Norrie 1997). Otro concepto importante es el de 'Holarquía', que es definido como 'un sistema de holones que pueden cooperar para conseguir un objetivo'. Un Sistema de Fabricación Holónico (HMS) es 'una holarquía que integra el rango completo de actividades de fabricación desde reserva de pedidos hasta diseño, producción y marketing para obtener una empresa de fabricación ágil'. Por ello, un HMS es un sistema de fabricación donde elementos clave, tales como materiales en crudo, máquinas, productos, piezas, AGVs, etc. tienen propiedades de autonomía y cooperación (Christensen 1994; Deen 1994). En un HMS cada actividad de un holón es determinada por medio de la cooperación con otros holones, en oposición a ser determinada por mecanismos centralizados. En este tipo de sistemas, los agentes inteligentes llamados 'holones' tienen tanto una parte física como una parte software. Un holón puede ser parte de otro holón.

Con los mismos conceptos y arquitectura de sistema similar, los miembros del consorcio HMS han desarrollado sus propios testbeds para sus entornos hardware y

software existentes. Muchos de los resultados obtenidos sólo han sido transmitidos internamente en el consorcio. Sin embargo algunos resultados han sido publicados: manipulación de materiales (Christensen et al 1994), Scheduling y planificación de fabricación (Hasegawa et al 1994; Biswas et al 1995; Sousa and Ramos 1997), y control inteligente de fabricación (Brennan et al 1997; Wang et al 1998).

La aplicación de los conceptos holónicos a la fabricación estuvo motivada inicialmente por la incapacidad de los sistemas existentes de fabricación (i) para tratar con la evolución de los productos dentro de los proceso de producción existentes y (ii) para mantener un desempeño satisfactorio fuera de las condiciones normales de operación. Con respecto a la tecnología CIM (Computer Integrated Manufacturing), por ejemplo, se ha señalado que los sistemas de fabricación basados en los conceptos CIM son inflexibles, frágiles, y difíciles de mantener:

- ✓ Los sistemas CIM tienen una jerarquía de control fija que no soporta cambios.
- ✓ La reconfiguración y extensión de sistemas existentes es difícil.
- ✓ La eficiencia de producción no se mantiene fuera de las condiciones normales de producción.
- ✓ Los datos para diagnosis correspondientes a las maquinarias son difíciles de acceder.
- ✓ El control es completamente automatizado y la intervención humana se excluye.

Estas deficiencias están generalmente producidas por un control centralizado y jerárquico que crea una comunicación jerárquica rígida y un flujo de comandos autoritario top-down. HM trata de sobreponer estas deficiencias con la ayuda de autonomía, cooperación, y auto-organización.

Grupos de expertos en fabricación, científicos, e ingenieros de las naciones industriales líderes, trabajaron desde 1992 hasta 1994 para construir y verificar un ámbito de trabajo de colaboración internacional en los Sistemas Inteligentes de Fabricación (Intelligent Manufacturing Systems - IMS). Equipos experimentados de Australia, Canadá, Europa, Japón y Estados Unidos formaron parte de un estudio de viabilidad de dos años que empezó en Febrero de 1992. Este estudio de viabilidad probó que este tipo de colaboración internacional podría lograr resultados significativos en un tiempo relativamente corto.

El HMS es uno de los 6 casos de estudio dentro del programa IMS creado en 1992. El caso de estudio HMS se inició como un estudio de factibilidad de un año y consistió de 5 benchmarks que fueron utilizados para lograr un mejor entendimiento de los requisitos de los sistemas de producción del siglo 21 y desarrollar un enfoque para construir sistemas de producción futuros basados en estos requisitos. El éxito de los benchmarks llevó a la formación del HMS como proyecto internacional del IMS en 1994. El consorcio HMS consta de 31 socios de todas las regiones dentro del programa IMS, de los cuales 15 provienen de la industria, 10 de universidades y 6 de institutos de investigación.

El objetivo del proyecto HMS es desarrollar aspectos genéricos de la tecnología holónica y demostrar su adecuación y beneficios con la ayuda de varias aplicaciones prototipo para la industria. La tarea del consorcio HMS es traducir los conceptos desarrollados por Koestler para organizaciones sociales y organismos vivos en un conjunto de conceptos apropiados para las industrias de producción.

Los holones en un Sistema Holónico de Producción asisten al operador a controlar el sistema: los holones seleccionan de manera autónoma los parámetros apropiados, encuentran sus propias estrategias y construyen su propia estructura.

Las principales características de un holón, definidas por el HMS (HMS 1994), son:

- ✓ Autonomía: Cada holón debe ser capaz de crear, controlar y monitorizar la ejecución de sus propios planes y/o estrategias, y tomar las acciones correctivas apropiadas en contra de su propio mal funcionamiento.
- ✓ Cooperación: Los holones deben ser capaces de negociar y ejecutar planes mutuamente aceptados y adoptar acciones conjuntas en contra de mal funcionamiento (Christensen 1994; Deen 1994).
- ✓ Auto-Organización: El sistema debe ser capaz de acomodarse ante la incorporación de nuevos holones, la eliminación de holones existentes, o la modificación de las capacidades funcionales de los holones existentes, con una mínima intervención humana, donde los holones o sus funciones pueden ser proveídos por una variedad de fuentes diversas.

Visión del HMS

Un HMS está formado por unidades autónomas y auto confiables, llamadas holones. Cualquier unidad, por ejemplo una máquina, una cinta transportadora, una pieza de trabajo, o una orden de trabajo puede ser un holón siempre que la unidad sea capaz de crear y controlar la ejecución de sus propios planes y/o estrategias. Un holón contiene siempre una parte de procesamiento de información y opcionalmente una parte de procesamiento físico. Un scheduler es un ejemplo de un holón sin parte de procesamiento físico.

Los holones cooperan con otros holones durante el proceso de producción para lograr los objetivos de producción. La cooperación, en forma de coordinación y negociación, se desarrolla donde y cuando sea necesario, usualmente entre flujos de material y de información. El proceso de cooperación, en contraste con CIM, envuelve además a humanos. Ellos son vistos como recursos ordinarios que muestran comportamiento holónico autónomo y cooperativo. Sin embargo, la integración de humanos requiere de una interfaz hombre-máquina como un holón artificial. Por esta razón, Christensen (Christensen 1994) provee tres tipos principales de interfaz con un holón artificial: una interfaz de procesamiento físico, una interfaz inter-holón, y una interfaz humana.

Una arquitectura general de un holón se muestra en la figura 1 (arqui_general). La capa de procesamiento físico es el hardware que realiza la operación de fabricación, como por ejemplo ensamblado. Este es controlado por la capa de control físico. La toma de decisiones representa el kernel del holón y provee dos interfaces: la primera para la interacción con otros holones, y la segunda para la interacción con humanos.

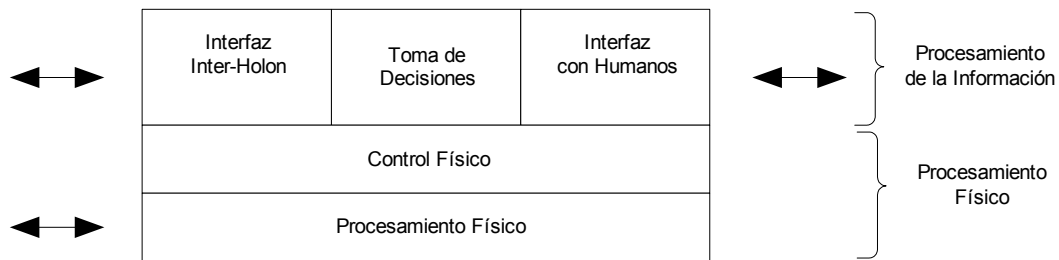


Figura 1. Arquitectura General.

Escenario Ideal en un HMS

Para ilustrar los conceptos de HMS, vamos a presentar una visión ideal de cómo un HMS debería operar. Al principio, el HMS consiste sólo de un conjunto de holones de recursos sin organización que forman un holón de fabricación. Cuando llega una orden de trabajo, el holón de fabricación crea un holón de orden de trabajo que inicia una negociación con holones de recursos para proporcionar ciertas operaciones de fabricación. Durante el proceso de negociación, el holón de orden de trabajo demanda propiedades específicas de la operación, tal como alta calidad o alto rendimiento, mientras que los holones de recurso tratan de lograr su máxima utilización. Al final de la negociación, los holones de recursos forman la línea de producción acordada y el holón de orden de trabajo inicia la creación de holones piezas de trabajo.

Los holones piezas de trabajo ingresan a la holarquía de fabricación (por ejemplo desde el stock) e inmediatamente negocian por recursos con el objetivo de ser procesados. Cada holón pieza de trabajo hace esto de manera individual y se centra en la siguiente operación. Una vez que las operaciones en un recurso hayan sido realizadas, la pieza de trabajo reinicia la negociación con las restantes operaciones. La organización completa de la holarquía de recursos asegura que la carga de piezas de trabajo está eficientemente distribuida sobre los recursos disponibles a fin de lograr el objetivo global de la holarquía.

En caso de disturbios, el holón de recurso afectado se excluye por cuenta propia de la holarquía y se dirige a una cabina de reparación. Los holones de recursos restantes se reorganizan a si mismos a fin de resistir la pérdida de capacidad. Desde el punto de vista del holón pieza de trabajo, el proceso continúa de manera normal, sólo que con menores holones de recursos con los que negociar. Luego de la reparación, el holón de recurso intenta unirse nuevamente a la holarquía de recursos.

Al final del proceso de la orden de trabajo, el holón de orden de trabajo se elimina y la holarquía de recursos se disuelve en holones de recursos que luego tratan de participar en nuevas holarquías de orden de trabajo.

3.7 Estado del Arte del HMS

De un estudio de la literatura especializada se han podido identificar cuatro grandes grupos que se dedican al desarrollo e investigación de los Sistemas Holónicos. A continuación se presenta un resumen de sus proyectos. En este resumen se podrá notar

que todos los trabajos se basan en arquitecturas de Holones que emplean Agentes como componentes principales.

PROSA (Product-Resource-Order-Staff Architecture):

PROSA (Brusel et al 1998), es la arquitectura holónica de referencia para los sistemas de fabricación de la PMA/K.U.Leuven (Universidad Católica de Leuven). Consiste principalmente en una arquitectura inter-holónica, que identifica los tipos de holones, sus responsabilidades, y la estructura en la cual interactúan. La arquitectura básica consiste de tres tipos de holones básicos: (i) orden de trabajo, (ii) producto, y (iii) recursos. 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 fabricación: (i) logística interna, (ii) plan de fabricación, y (iii) manejo de recursos. Para asistir, con conocimiento experto, a los holones básicos se pueden agregar holones ‘staff’. La estructura del sistema completo de fabricación es una holarquía dual descompuesta en una sub-holarquía de asignación de recursos (órdenes de trabajo, recursos y staff) y una sub-holarquía de control de proceso (producto y la parte de control de proceso de los recursos). Una de las técnicas de coordinación y de control de la holarquía se implementa utilizando conceptos inspirados en el comportamiento social de insectos, más específicamente hormigas. Además de esta técnica existen trabajos del grupo en los cuales la cooperación se implementa utilizando conceptos relacionados con los trabajos de Deen y Fletcher, que se presenta en la siguiente sección. Se distinguen tres categorías diferentes de dominios de cooperación: uno para conocimiento de producción, uno para procesamiento, y uno para ejecución de proceso. El protocolo de comunicación utilizado es el Contrat Net. Cuando una orden de trabajo ingresa al sistema, el holón de orden de trabajo se convierte en el administrador de la tarea, la divide en sub-tareas, anuncia cada sub-tarea, luego selecciona las ofertas de fabricación más apropiadas y asigna la tarea. El administrador de la tarea controla el progreso del contrato, y en caso de que el contratado falle en la culminación de la tarea, es libre de re-asignar la tarea.

Arquitectura basada en Agentes y bloques funcionales

El grupo de la Universidad de Keele, liderado por Deen y Fletcher, ha desarrollado una arquitectura para sistemas holónicos basada en agentes y bloques funcionales (Fletcher et al 2000).

Un holón de fabricación por lo general se compone de conocimiento y componentes software, además de un componente (opcional) hardware. Funcionalmente, un holón puede ser considerado como una composición de un sistema de control inteligente (cabeza) y un sistema de procesamiento (base).

Los elementos del sistema de control inteligente (ICS) se muestran en la figura 2. El PMC (control proceso/máquina) se encarga de ejecutar los planes de control para los procesos que están siendo controlados. El PMI (interfaz proceso/máquina) provee la interfaz lógica y física para el sistema de procesamiento a través de una red de comunicación. El HI (interfaz humana) provee la interfaz para humanos, tales como operadores, supervisores, personal de mantenimiento, etc. El IHI (interfaz inter holónica) se encarga de la comunicación inter-holónica. El sistema de procesamiento consiste de todos los componentes de procesamiento necesarios para realizar las actividades de producción. De esta manera el ICS permite a los holones ofrecer las habilidades de fabricación, como subsistemas autónomos en coordinación con el

ambiente y los demás holones. El sistema de procesamiento es responsable de las funcionalidades de fabricación de acuerdo a las reglas y estrategias operativas impuestas por el ICS.

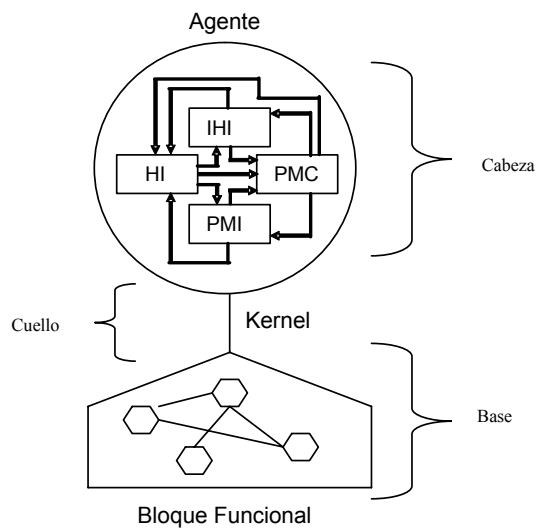


Figura 2. Estructura Interna de un holón.

En esta arquitectura integrada de agentes y bloques funcionales (IEC 61499)¹, los agentes son utilizados para manejar las estrategias de planificación de alto nivel (ICS-Cabeza), mientras que los bloques funcionales manejan el control de bajo nivel proceso/máquina en tiempo real (Base). Un kernel holónico se ejecuta por encima de la arquitectura de los bloques funcionales para proveer la interfaz necesaria entre los agentes y el control IEC 61499.

Las holarquías y la interacción entre los holones basados en agentes se organizan a través de una estructura llamada dominio de cooperación como se muestra en la figura 3. Un dominio de cooperación es un espacio lógico a través del cual: (i) los holones se comunican y operan, y (ii) se provee un contexto donde los holones pueden localizar, contactar e interactuar entre ellos. Un dominio de cooperación no puede existir por su propia cuenta debe poseer por lo menos un miembro holón, todos los dominios deben ser creados dinámicamente por la operación de los componentes funcionales de los holones. Un sistema holónico debe poseer por lo menos un dominio de cooperación. Un holón puede ser miembro simultáneamente de uno o más dominios de cooperación. Cada dominio está dirigido por un coordinador que es la interfaz del dominio con el exterior (otros dominios). Un holón puede unirse a un dominio de cooperación, consultar los atributos del mismo, intercambiar información entre los holones, y abandonar el dominio cuando haya finalizado su tarea. Para asignar actividades a los holones dentro de un dominio de cooperación se ejecuta un protocolo de cooperación.

¹ IEC, es la Comisión Internacional Electrotécnica, encargada de los estándares internacionales para todos los campos de la electro-tecnología. El IEC 61499 corresponde al estándar para los procesos industriales distribuidos. Básicamente consiste en una colección de controladores de dispositivos interconectados y en comunicación por medio de una red que puede estar organizada de manera jerárquica. Se basa en bloques funcionales, los cuales proveen soluciones software a problemas pequeños, por ejemplo el control de una válvula, o el control de una unidad mayor dentro de una fábrica, tal como toda una línea de producción.

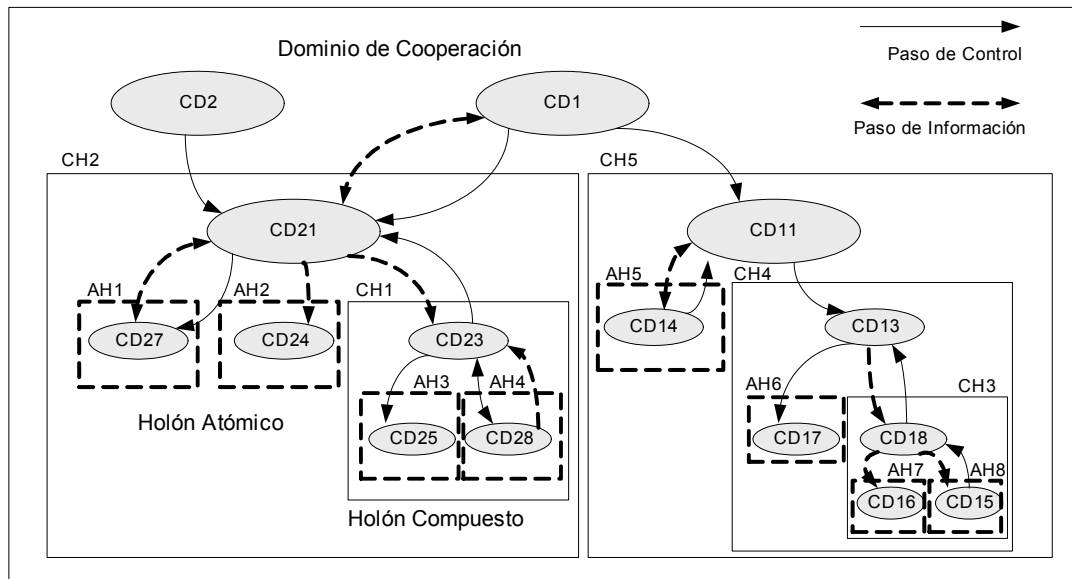


Figura 3. Dominio de cooperación

Deen y Fletcher proponen además un modelo computacional para redistribuir tareas (reorganización de la holarquía) basados en conceptos de mantenimiento del equilibrio de la temperatura. Durante el procesamiento de una tarea retrasada o con problemas, los holones pueden sufrir sobre carga, esto se interpreta en el modelo de temperatura como el calentamiento del holón. De esta manera cuando un holón determina que su temperatura sobrepasa un límite pre-establecido, éste informa a los demás holones de la situación. Si existe un holón frío que pueda realizar la tarea anunciada, entonces se inicia una negociación con el holón caliente para transferir la tarea. De esta manera cuando el sistema entero tiende a buscar un equilibrio de temperatura, se logra auto-organización de las holarquías que forman el sistema.

MetaMorph

El grupo del Departamento de Mecánica e Ingeniería Industrial de la Universidad de Calgary se encuentra desarrollando proyectos relacionados con modelos para control inteligente en sistemas de fabricación. Algunos de estos proyectos son: MetaMorph I, ABCDE (Agent Based Concurrent Design Environment), DIDE (Distributed Design Environment) y FBIICDE (Feature-Based Integrated and Intelligent Concurrent Design System). MetaMorph II (Maturana and Norrie 1997), es la arquitectura holónica basada en agentes en la que actualmente se encuentran trabajando.

La característica principal de los sistemas basados en MetaMorph es su forma y estructura cambiantes, ya que se adaptan dinámicamente a tareas emergentes y ambientes cambiantes. La arquitectura de MetaMorph utiliza el concepto de dominio de cooperación de Deen y Fletcher pero los llaman clusters virtuales dinámicos. A diferencia de PROSA en MetaMorph 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 es dual, 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 de recurso son utilizados para representar dispositivos y operaciones de fabricación.

La coordinación y la auto-organización se implementan mediante los clusters virtuales dinámicos. Enfoque similar al de los trabajos de Deen y Fletcher. Utilizando el mecanismo de los clusters virtuales dinámicos, los holones pueden participar dinámicamente en diferentes clusters (holarquías) y cooperar a través de dominios de cooperación. Los holones mediadores cumplen la misma función que los holones coordinadores de Deen y Fletcher, se utilizan como administradores de los clusters para coordinar la interacción entre holones. El cluster existe hasta que la tarea de cooperación desaparezca cuando la tarea haya sido completada. El ciclo de proceso para un cluster virtual se puede definir de la manera siguiente:

1. El holón primario recauda algunos o todos sus contratos en una nueva tarea. Luego de re-planificación y análisis, los requerimientos de cooperación se listan como tareas de cooperación.
2. Un holón mediador es creado para encontrar holones sub-contratados potenciales y para enviar a estos una invitación de puja.
3. Los holones invitados, deciden de manera autónoma cómo participar y envían ofertas para todas las tareas en las que están interesados.
4. Se recogen y evalúan todas las ofertas tanto por el holón primario o por el mediador. Luego de determinar la asignación óptima de la tarea, se establecen contratos directamente entre el holón primario y el sub-contratado. Luego el holón primario y todos los sub-contratados forman un cluster virtual.
5. A medida que la tarea sea resuelta, el cluster asociado, el mediador y los respectivos enlaces desaparecen.

Cualquier holón en la sociedad holónica puede pertenecer a más de un cluster, tal como se puede ver en la figura 4.

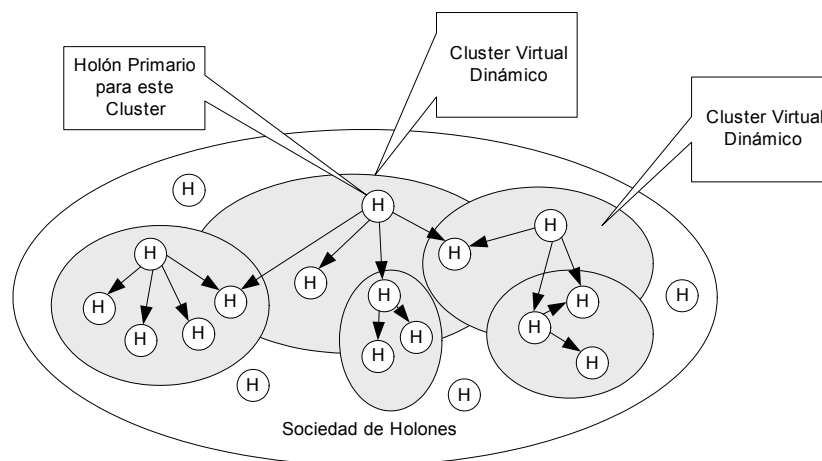


Figura 4. Sociedad de holones

Arquitecturas basadas en INTERRAP

El grupo del Centro Germano de Investigación para la Inteligencia Artificial (DFKI) y el grupo del Departamento de Ciencias de la Computación y Control Automático de la Universidad de Sibiu en Rumania, han desarrollado, respectivamente, dos arquitecturas muy similares para la implementación de los sistemas holónicos.

El grupo del DFKI (Fischer 1998) se ha basado completamente en la arquitectura de tres niveles concurrentes, INTERRAP de Müller (Müller 1996), ver figura 5. En ella la composición y configuración de las estructuras holónicas de la holarquía se realiza dentro del nivel de Planificación Cooperativa (CPL) que provee las funcionalidades para la comunicación, negociación y administración de las estructuras holónicas. Este grupo ha utilizado esta arquitectura en los siguientes dominios de aplicación: Suply Webs, HMS, Logística en Empresas Virtuales y Fuentes de Información basados en Agentes.

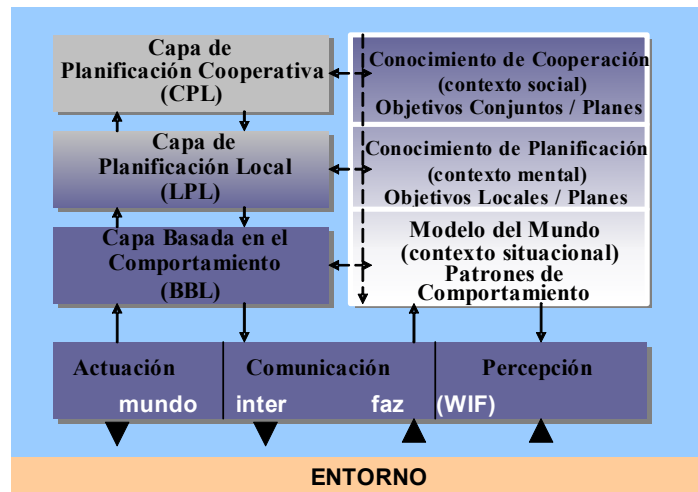


Figura 5. Interrap

Por su parte el grupo Rumano ha modificado ligeramente esta arquitectura, extendiendo el nivel de CPL en tres sub-niveles (i) integración (para cooperación vertical), (ii) cooperación (para integración horizontal); y (iii) monitoreo (para modificación de holarquías). La arquitectura utilizada por este grupo (Barbat and Zampfirescu 1999) ha sido, además, inspirada en los tipos de holones de PROSA. El área de aplicación en cambio no se limita únicamente a la producción, sino que además lo han aplicado al Robotic Soccer y a Agentes para el manejo de conocimiento en webs. Por cada holón que tiene una tarea asignada existe una Unidad de Ejecución del Plan (PnEU), además siempre existe exactamente un PnEU ocioso con un plan vacío que coordina la formación de un nuevo holón. Cuando una orden de trabajo ingresa al sistema se debe asignar la misma a un holón existente o a uno nuevo. El protocolo utilizado para la asignación de tareas es un Contrat Net extendido.

4. Sistemas Multiagente vs Sistemas Holónicos

Tanto Sistemas Multiagente como Sistemas Holónicos son paradigmas adecuados para modelar los Sistemas de Fabricación actuales, tanto uno como otro han sido foco de atención tanto de la universidad como de la industria. En la actualidad, existe un mal entendido acerca de la relación que existe entre estos dos enfoques. Algunos investigadores, tal como Bussmann (Bussmann 1998), Brennan y Norrie (Brennan and Norrie 2001), han intentado aclarar las diferencias filosóficas fundamentales entre los agentes y los holones. No obstante, una comparación exhaustiva de todas las propiedades de ambos paradigmas aun no ha sido analizada.

En este apartado, se realiza un estudio comparativo entre los sistemas holónicos (SH) y los SMA. Siguiendo la tendencia generalizada en el área de los sistemas de producción inteligente, indicaremos la motivación de ambos enfoques; luego identificaremos aquellas propiedades que serán contrastadas para hacer un análisis de cada uno, y; finalmente presentaremos una tabla con un resumen de la comparación. El objetivo principal de esta sección es averiguar si los holones y los agentes son diferentes o no, o si uno es un caso especial del otro.

Como ha sido señalado por Bussmann (Bussmann 1998), Brennan y Norrie (Brennan and Norrie 2001) entre otros, ambos enfoques difieren principalmente en la motivación. La investigación del HS está motivada en las tareas flexibles de fabricación. Por consiguiente, está orientada hacia los estándares de comunicación de bajo nivel y el comportamiento de bajo nivel. Por otra parte, la investigación en el área de los SMA está motivada en la programación de sistemas inteligentes distribuidos. Se centra en el comportamiento social de entidades inteligentes y se ocupa principalmente de la investigación de modelos de comportamiento, estrategias de cooperación y coordinación, optimización del desempeño de tareas, aprendizaje a partir de las propias experiencias, creación de coaliciones, etc. En resumen, a diferencia de SMA, que es un enfoque software amplio que puede ser utilizado además para el control inteligente distribuido, HS es, por definición, un enfoque específico de fabricación para sistemas de control distribuidos.

Como ha sido señalado en capítulos anteriores, los agentes son: autónomos, sociales, reactivos, pro-activos, racionales, móviles, pueden aprender, etc. (Wooldridge and Jennings 1995). Los holones, como paradigma, tienen las siguientes características básicas: autonomía, cooperación y reorganización. Además de estas características, que las podemos llamar "propiedades de comportamiento", los holones tienen "características estructurales". Una de ellas es la "recursividad", la cual permite a los holones estar formados internamente por entidades auto-similares (holones), que a su vez pueden estar formadas por holones, y así sucesivamente (hasta que se llega a un nivel atómico en el cual nuevas sub-divisiones son imposibles o de poca utilidad para el dominio de aplicación). Los holones pueden estar compuestos por holones los cuales 1) pueden ser holarquías, 2) pueden participar en varias holarquías simultáneamente, y 3) pueden llegar, salir, o cambiar - es decir, holones que forman holarquías dinámicas. Otra propiedad estructural importante, como ha sido definida por el consorcio HMS (HMS 1994), es que los holones usualmente poseen una parte de procesamiento de información con una parte opcional de procesamiento físico.

En las siguientes secciones haremos un análisis y comparación de cada propiedad de forma separada.

4.1 Autonomía

El comportamiento de las entidades autónomas puede estar basado tanto en sus experiencias propias así como en el conocimiento interno utilizado en la representación de las entidades del entorno particular en el cual actúa. Los agentes han sido utilizados con éxito en dominios en los cuales el grado de incertidumbre e impredecibilidad requiere unidades de procesamiento capaces de acción autónoma, sin la directa intervención de humanos u otros agentes.

Los sistemas de control de la fabricación son sistemas grandes y complejos que están diseñados para llevar a cabo una tarea claramente definida. Sin embargo, en un sistema de fabricación, raramente las cosas ocurren como se esperaban. Se puede necesitar que el sistema realice tareas adicionales que no fueron anticipadas y algunas veces se puede permitir que el sistema omita ciertas tareas. Los recursos disponibles para desempeñar tareas pueden no estar disponibles, y recursos adicionales pueden ser introducidos. El tiempo de inicio y el tiempo de procesamiento de una tarea también son objetos de variaciones. Una tarea puede tomar más o menos tiempo que el anticipado, y las tareas pueden llegar temprano o tarde en la cadena de producción. Estas son algunas de las razones por las que los holones son, por definición, entidades autónomas las cuales deben ser capaces de crear, controlar y monitorizar la ejecución de sus propios planes y/o estrategias, y de tomar acciones correctivas en contra de su propio malfuncionamiento (HMS 1994). En este sentido, podemos decir que los agentes y los holones comparten esta característica.

4.2 Reactividad

Un agente responde a eventos que ocurren en su entorno, estos eventos afectan tanto los objetivos del agente o los supuestos que soportan los procedimientos que el agente está ejecutando para lograr sus objetivos. Así, los efectos de los estímulos del entorno pueden ser cambios en los objetivos o supuestos del agente, o en las acciones del agente. De la misma manera, los holones necesitan reaccionar a cambios en sus entornos. Tales cambios afectan sus objetivos internos o pueden prevenir la ejecución de tareas planeadas actuales o futuras.

Supongamos un sistema de fabricación simplificado (ver el área gris en la sección de procesamiento de producción en la figura 6). Existen 3 tipos de holones: (i) holones orden de trabajo que representan las tareas en el sistema de fabricación; (ii) holones recurso que son recursos de fabricación del sistema, y; (iii) holones producto, los productos en si. Sea HO1 el holón orden de trabajo 1 que está procesando una tarea para producir HP1 (holón producto 1). El proceso de fabricación se lleva a cabo en etapas de procesamiento asignados a holones de recurso HR1, HR2 y HR3, tal que HR1 es la primera etapa de procesamiento, HR2 es la segunda y HR3 la tercera. Por alguna razón, HR1 se toma más tiempo de procesamiento que el tiempo inicial estimado. HR2 y HR3 tienen que darse cuenta de esta situación que previene la ejecución de sus planes actuales y reaccionar de alguna manera, por ejemplo, buscando otros holones orden de trabajo para aprovechar su potencial, o detenerse hasta que HR1 finalice su procesamiento (obviamente, la segunda alternativa es menos productiva para el rendimiento de todo el sistema, pero también es una acción reactiva). A pesar de que este ejemplo es muy simplificado y reducido, en él se refleja la propiedad reactiva de los holones.

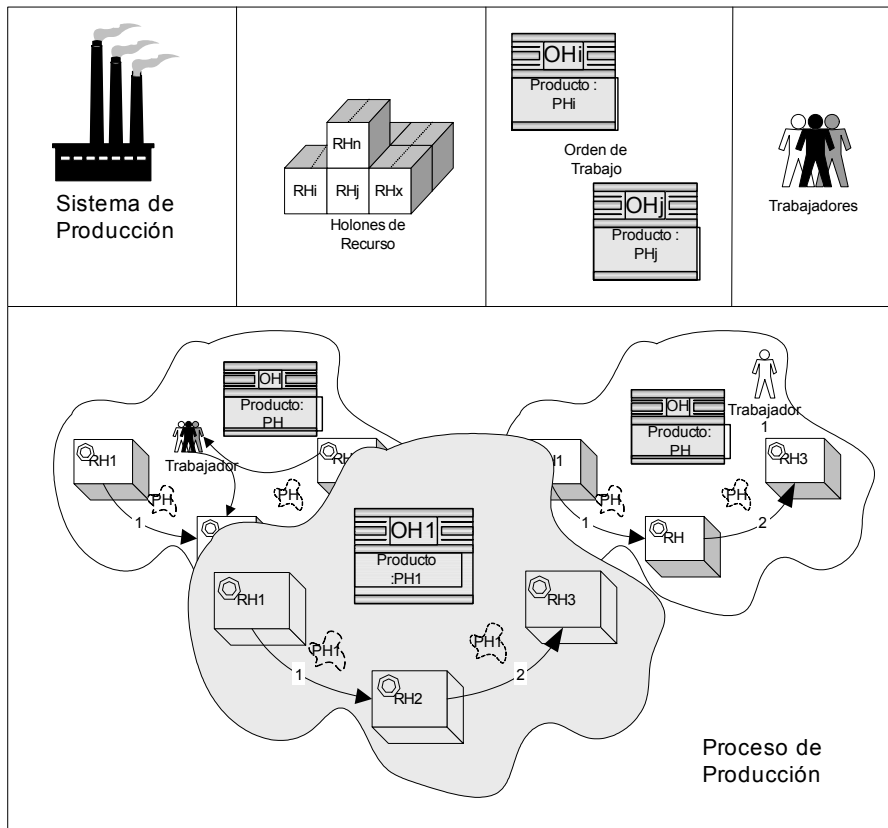


Figura 6. Ejemplo

Existen algunos problemas asociados con holones que están a su vez constituidos por holones. Uno de estos problemas es: ¿Quién realmente actúa y/o percibe en una holarquía? Un enfoque es tener por cada holarquía una cabeza que sea responsable de la interacción con el exterior (interacción a su nivel). Para la interacción interna, permitir a un holón interactuar con los demás holones de su holarquía. Este enfoque ha sido adoptado, entre otros por (Maturana and Norrie 1997; Fletcher et al 2000; Brussel et al 1998).

Un segundo enfoque es permitir a todo holón de una holarquía interactuar con cualquier holón, sin importar si los otros holones son de la misma holarquía o de otras holarquías. No existe una interfaz externa única, cada holón de la holarquía es una interfaz. El comportamiento de la holarquía emergerá como la composición del comportamiento de sus holones constituyentes. A pesar de que esta es la manera en que los sistemas biológicos están organizados, este es un enfoque muy complicado y difícil de implementar. Surgen algunas cuestiones asociadas con este enfoque: ¿Cuáles son los límites reales de una holarquía/holón?, ¿Cuál es la identidad de una holarquía/holón?, etc.

4.3 Pro-actividad

Los agentes no actúan simplemente en respuesta a su entorno, son capaces además de exhibir comportamiento dirigido por objetivos tomando la iniciativa (Wooldridge and Jennings 1995). No es difícil construir un sistema dirigido por objetivos, como lo han señalado Wooldridge y Jennings. Cuando uno escribe un procedimiento tal, uno lo

describe en términos de sus pre y post-condiciones. Los efectos del procedimiento son sus objetivos. Si las pre-condiciones se satisfacen cuando se invoca al procedimiento, luego se espera que el procedimiento se ejecute correctamente: el objetivo será alcanzado. Pero sistemas de este tipo asumen que el entorno no cambia mientras se ejecuta el procedimiento. De igual manera, se asume que el objetivo, es decir la razón para ejecutar el procedimiento, permanece válido al menos hasta que el procedimiento haya terminado.

Como lo hemos señalado en capítulos anteriores los dominios de aplicación de los agentes y de los holones cambian constantemente, impidiendo que éstos puedan ejecutar un procedimiento a ciegas sin preocuparse si aún son válidos los supuestos que soportan la ejecución del procedimiento. Esto implica que los agentes y holones necesitan un balance entre el comportamiento dirigido por objetivos (pro-activo) y el comportamiento reactivo. Los agentes y holones intentarán alcanzar sus objetivos de manera sistemática, pero además serán capaces de reaccionar a tiempo ante nuevas situaciones para que la reacción sea de alguna utilidad. Supongamos en el ejemplo anterior, que HR1, HR2 y HR3 tienen como objetivo: "maximizar su utilización". Antes que HR1 empieza a ralentizar su procesamiento, HR2 y HR3 tienen huecos de tiempo asignados a HP1, y huecos de tiempo asignados a otros productos para lograr una mayor utilización de su tiempo y potencia de procesamiento, es decir para lograr su objetivo. Cuando HR2 y HR3 descubren que HR1 se está tomando más tiempo del esperado, HR2 y HR3 deben re-asignar el hueco de tiempo asignado a HP1 así como los huecos de tiempo y potencia computacional asignados a los demás productos encontrados en su intento por alcanzar su objetivo. Este ejemplo simple refleja la propiedad pro-activa de los holones.

4.4 Habilidad Social

Ya que los agentes no actúan usualmente aislados, sino en presencia de otros agentes o humanos, necesitan de habilidades sociales y comportamiento interactivo para comunicarse, cooperar, coordinarse y negociar con ellos. Una pregunta surge directamente: ¿Cómo los agentes son capaces de interactuar y especialmente de comunicarse con otros agentes? La respuesta es: Gracias a los lenguajes de comunicación de agentes (Agent Communication Language - ACL). Pero, ¿qué es un ACL? Un ACL, como ha sido propuesto por Austin (Austin 1962), "... es un medio a través del cual se comunican las actitudes con respecto al contenido del intercambio entre los agentes; sugiere cuándo el contenido de una comunicación es una aserción, una solicitud, una consulta, etc.". Estándares comunes para ACLs incluyen el Knowledge Query and Manipulation Language - KQML (Finin et al 1994) (Lenguaje de Manipulación y Consulta del Conocimiento) y una propuesta de FIPA (Foundation for Intelligent Physical Agents), basado en el lenguaje Arcol - ARtimis COmmunication Language (FIPA 1998). Estos desarrollos son en parte rivalizados por lenguajes de etiquetado de Internet altamente sofisticados tales como XML (W3C 2002), el cual también puede ser utilizado por agentes.

La habilidad social de un holón está incluida en su capacidad para cooperar (ver la siguiente sección), ya que los holones necesitan de un medio de comunicación con otros holones para ser capaces de cooperar. En el ejemplo anterior, los tres tipos de holones necesitan de habilidades sociales para intercambiar información acerca del proceso de fabricación, para permitir la ejecución de tareas. Por ejemplo, supongamos que HO1

necesita recursos para fabricar HP1, entonces HO1 interactúa con los holones de recursos HR1, HR2 y HR3 para obtener funcionalidades de procesamiento y propiedades específicas de la operación, tal como alta calidad o alto rendimiento. Por otra parte, HR1, HR2 y HR3 tratan de maximizar su utilización, mientras que HP1 al mismo tiempo se centra en la siguiente operación para lograr ser procesado por HR1, HR2 y HR3. Como se puede deducir a partir de este simple ejemplo, los holones necesitan interactuar entre sí, y al igual que los agentes, los holones necesitan de habilidad social. En los sistemas de fabricación, las personas y los computadores necesitan estar integrados, con acceso al conocimiento e información requeridos, para trabajar de manera colectiva a varios niveles del desarrollo del producto (HMS 1994). Estos requisitos llevaron a Christensen a proponer un bloque integrado de Interfaz Humana (ver figura 1) en su arquitectura general de holones. Cada holón debe ser capaz de cooperar con varios agentes especializados quienes proveen servicios de comunicación como un todo. Sin embargo, nada impide en la definición de agentes que estos tengan un bloque integrado de interfaz humana.

4.5 Cooperación

La cooperación es un medio de habilidad social. Las empresas de fabricación deben cooperar completamente con sus proveedores y clientes en suministros de material, fabricación de partes, comercialización del producto final, etc. Tal cooperación se debería realizar de forma tal que exista respuesta rápida de ambas partes. La cooperación es un requisito imperativo para cualquier modelo funcional para sistemas avanzados de fabricación (HMS 1994). Más aún, todas las unidades de fabricación cooperan para lograr los objetivos globales de fabricación. Con respecto a estos objetivos, un holón nunca rechaza de manera deliberada la cooperación con otro holón. Sólo cuando las acciones solicitadas son imposibles o fuertemente desventajosas para el proceso de fabricación, el holón rechaza sus ejecuciones (Bussmann 1998). La coordinación, negociación y otras técnicas de cooperación permiten a los holones interactuar de forma flexible con otros holones en una manera abstracta. Para la cooperación, existe un gran número de enfoques específicos en SMA, ver (O'Hare and Jennings 1996) y (Huhns and Singh 1998) para una apreciación global. Un ejemplo de escenario de cooperación ha sido presentado en la sección anterior.

4.6 Re-organización

Koestler define una holarquía como una jerarquía de holones la cual funciona (a) como un todo autónomo en supra-ordinación a sus partes, (b) como partes dependientes en sub-ordinación a controles de niveles mayores, (c) en coordinación con su entorno local. Esta definición se deriva en una jerarquía de holones la cual es una mezcla de organización jerárquica y horizontal. El HMS explora ampliamente principios holarquicos para crear colecciones integradas de varios niveles menores de holones. Como lo han señalado Brennan y Norrie (Brennan and Norrie 2001), la noción de holarquía puede ser implementada utilizando varias enfoques de arquitecturas SMA para federaciones tal como facilitadores, brokers o mediadores.

Los holones son capaces de actuar dentro de múltiples organizaciones (holarquías), las cuales son creadas y modificadas dinámicamente. Los entornos de fabricación reales son altamente dinámicos debido a situaciones diversas y cambiantes (HMS 1994): las tasas bancarias cambian por las noches, los materiales no llegan a hora, caída de

proveedores de energía, los facilitadores de producción fallan, los trabajadores están ausentes, nuevas órdenes de trabajo llegan y órdenes existentes se modifican o cancelan, etc. Tales cambios llevan a desviaciones del plan y asignación de recursos actuales. Es, por tanto, necesario para el sistema en ejecución adaptarse a tales entornos cambiantes. Este es el motivo por el cual las holarquías deben ser organizaciones abiertas, las cuales deben ser capaces de acomodar la incorporación de nuevos holones, la eliminación de holones existentes, la re-organización de la carga de trabajo, etc. Aparte de la cooperación, los holones requieren de técnicas para reorganizar el control de los procesos de producción. Estas técnicas monitorizan las acciones de otros componentes, es decir, acciones físicas y de comunicación, y analizan el proceso de control. De esta manera, los holones pueden identificar las posibilidades para mejorar e iniciar un proceso de negociación para la organización utilizando técnicas de cooperación estándar. Las implicaciones de la re-organización acordada se distribuyen a los componentes de los demás holones. En SMA, a diferencia de la cooperación, existe muy poco trabajo sobre re-organización.

4.7 Racionalidad

Un agente racional es uno que hace lo correcto, es decir una acción que causa que el agente sea el más exitoso (Russell and Norvig 1995). Las acciones que un agente ejecuta pueden ser entendidas como sus objetivos. Galliers propone, en (Galliers 1988), una definición de racionalidad: ...la suposición que un agente actuará para lograr sus objetivos y no actuará de tal manera que sus objetivos no puedan ser alcanzados - al menos en la medida que sus creencias se lo permitan. Un agente racional actúa de acuerdo a: la secuencia de percepciones, lo que el agente conoce acerca de su entorno, y las acciones que el agente puede realizar. Estas tres cosas determinarán el suceso del agente.

Aunque esta propiedad no aparece explícitamente en la definición de holones, puede ser derivada a partir de una definición más fuerte de autonomía: un sistema es autónomo en la medida que su comportamiento es determinado por sus propias experiencias (Russell and Norvig 1995). Y la suposición general que un holón siempre intenta obtener el mejor desempeño global del sistema.

4.8 Actitudes Mentales

En el área de DAI, ha sido propuesto un número de enfoques para especificar agentes racionales en término de actitudes mentales tales como conocimiento, creencias, deseos, objetivos, acuerdos, e intenciones. Sin embargo, no existe un consenso acerca de precisamente qué combinación de actitudes mentales es más adecuada para caracterizar a los agentes. No obstante, parece ser aceptado, por la mayoría, que las creencias deberían ser tomadas como una de las nociones básicas de la teoría de agentes (Wooldridge and Jennings 1995).

Como hemos señalado en secciones previas, a pesar de que los procesos de producción experimentan varios cambios y perturbaciones, el grado de incertidumbre e impredecibilidad no es comparable a aquella de otros dominios de aplicación de agentes. Como consecuencia, las aplicaciones de producción requieren menos deliberación mental y social que las aplicaciones típicas de sistemas multiagentes (Bussmann 1998). Las unidades de control de producción (holones) deben razonar

acerca del comportamiento del sistema de fabricación, pero no acerca de sus propias actitudes mentales o de la de otras unidades de control (holones).

4.9 Aprendizaje

Cuando se diseña un SMA, es generalmente imposible prever todas las situaciones potenciales que un agente puede encontrar y especificar de antemano el comportamiento de un agente de manera óptima. Para superar estos problemas de diseño, los agentes tienen que aprender a partir de y adaptarse a su entorno. Sen y Weiss presentaron, en (Sen and Weiss 1999), algunas clasificaciones estándar de Aprendizaje Máquina (Machine Learning - ML) para los distintos tipos de aprendizaje. Una de ellas es la siguiente:

De acuerdo con el método o estrategia de aprendizaje: aprendizaje por repetición, aprendizaje a partir de instrucciones o consejos, aprendizaje a partir de ejemplos y por práctica, aprendizaje por analogía, y aprendizaje por descubrimiento.

De acuerdo con la respuesta que está disponible para la entidad que aprende y que indica el nivel de desempeño alcanzado: aprendizaje supervisado, aprendizaje reforzado, y aprendizaje no supervisado.

Las unidades de control de fabricación (holones) deben ser capaces de adaptarse a cambios del entorno y manejar contextos emergentes. Como lo señalamos en la sección de re-organización, las holarquías pueden reorganizarse para tratar con circunstancias imprevistas. Esta capacidad puede ser mejorada con aprendizaje. Por ejemplo, (Shen et al 1998b), algunos objetos de aprendizaje son: combinaciones de recursos de fabricación para un área específica, comportamiento del sistema de fabricación, soporte en favor o en contra de una decisión, pre-condiciones y post-condiciones para acciones y tareas, tipos de conflictos, heurísticas para resolver conflictos y para negociar, etc.

4.10 Benevolencia

La propiedad de benevolencia es aquella por la cual el agente coopera con otros agentes cuando y donde sea posible. La benevolencia ciega no tiene lugar en el modelado de agentes autónomos para los cuales la cooperación ocurrirá sólo cuando sea considerada ventajosa en términos de motivaciones. El agente no puede gastar todo su tiempo en nuevas cooperaciones con otros agentes, sin tener en cuenta sus acuerdos actuales y motivaciones. Podemos decir que los holones son entidades benevolentes, porque cuando descubren un posible escenario de cooperación ellos cooperan.

4.11 Movilidad

Los agentes móviles extienden las capacidades de los sistemas distribuidos mediante la movilidad del código. Los agentes móviles son programas que pueden viajar a través de una red de computadoras y se pueden conectar a otros agentes y a plataformas de agentes para desarrollar sus tareas. El paradigma de agentes móviles ofrece un gran número de ventajas: La movilidad y autonomía hacen que las conexiones permanentes sean innecesarias. El uso de agentes es apropiado también en escenarios donde grandes volúmenes de datos deben ser transportados a través de una red mientras que el código de procesamiento es más bien pequeño. En tales casos, vale la pena considerar mover el

código hacia el dato. Las unidades de control de fabricación están dedicadas al control continuo de componentes de fabricación. La relación <controlador, unidad controlada> se asigna de antemano y es fija a lo largo de todo el proceso de fabricación. Más aún, toda la información de control necesaria se encuentra en la unidad controlada. Por tanto, los holones raramente necesitarán de movilidad para la ejecución de sus tareas (Bussmann 1998).

4.12 Recursión

La condición básica de los sistemas holónicos es que los holones son simultáneamente todo y parte (HMS 1994). Esto significa que los holones pueden contener niveles inferiores de holones, que pueden además estar contenidos en otros niveles superiores de holones, resultando en una arquitectura recursiva.

En la literatura especializada de sistemas multiagentes, no hemos encontrado referencias acerca de arquitecturas de agentes recursivos. No obstante, en la definición de agentes nada impide que los agentes tengan una estructura interna compuesta por entidades que a su vez sean agentes.

4.13 Procesamiento Físico y de la Información

En 1994 Christensen propuso una arquitectura general de holón (Christensen 1994) (Figura 1). Prácticamente todos los investigadores del SH adoptan esta arquitectura general. En esta arquitectura se puede observar una separación explícita entre la parte de procesamiento físico y la parte de procesamiento de la información. En el área SMA, no existe tal separación explícita, dado que un agente en su estructura interna posee componentes software que pueden realizar cualquier tipo de procesamiento. Bussmann, (Bussmann 1998), propone el uso de SMA como la tecnología que permite el procesamiento de la información en un SH. Empezando con la visión holónica, los SMAs pueden proveer las técnicas de razonamiento necesarias para desarrollar la arquitectura de procesamiento de la información de un holón y las técnicas de cooperación tal que los holones puedan interactuar con otros holones de la holarquía. Para la parte de procesamiento físico, Fletcher y Deen (Fletcher and Deen 2001) proponen bloques funcionales para manejar el control de tiempo real para la interacción de bajo nivel proceso/máquina.

En la siguiente tabla se presenta un resumen de la comparativa desarrollada en la presente sección.

Tabla 1: Holones Vs Agentes

Propiedad	Holón	Agente
Autonomía	Si	Si
Reactividad	Si	Si
Pro-actividad	Si	Si
Habilidad Social	Si. La Interfaz Humana es específica de cada holón.	Si. La Interfaz Humana se implementa generalmente por uno o varios agentes especializados.

Cooperación	Si. Los holones nunca rechazan de manera deliberada la cooperación con otro holón.	Si. El agente puede competir y cooperar.
Re-Organización	Si. Holarquías.	Si. Jerarquías, organización horizontal, heterarquías, etc. Las holarquías se pueden implementar utilizando varias enfoques para federaciones en SMA tales como facilitadores, o mediadores.
Racionalidad	Si	Si
Aprendizaje	Si	Si
Benevolencia	Si	Si
Movilidad	Los holones raramente necesitarán de movilidad para la ejecución de sus tareas.	Si
Recurción	Si	No existe ninguna arquitectura recursiva como tal, pero algunas técnicas son utilizadas para definir federaciones que simularan los diferentes niveles recursivos.
Procesamiento de la Información y Físico	Si. La separación es explícita, aunque la parte de Procesamiento Físico es opcional.	No existe una separación explícita.
Actitudes Mentales	Si. Los holones no necesitan razonar acerca de sus propias actitudes mentales o aquellas de otras unidades de control.	Si

5. Aplicaciones de los Sistemas Multiagente en Sistemas de Fabricación

En los últimos años, la tecnología de sistemas multiagente se ha estado aplicando en la integración de empresas de fabricación, en la gestión de la cadena de suministros, en el control de ejecución, planificación y scheduling de fabricación, en manipulado de materiales y gestión de inventario.

En (Shen and Barthès 1996b) podemos encontrar una revisión de los resultados clave relacionados con los sistemas cooperativos basados en agentes, tales como representación, gestión de ontologías, estructura de agente, arquitectura de sistema, comunicaciones, dinámicas de sistema, control de sistema global, resolución de conflictos, problemas de legado e interfaces externas. Muchos de estos resultados son también aplicables en sistemas de fabricación basados en agentes. En este apartado, se discuten los resultados clave especialmente relacionados con la fabricación basada en agentes [<http://img.enme.ucalgary.ca/publication/abm.htm>], como son la tecnología de

agentes para integración de empresas y gestión de la cadena de suministros, encapsulación, arquitectura de sistemas, reconfiguración de sistemas dinámicos, aprendizaje, diseño y viabilidad de fabricación, scheduling dinámico distribuido, integración de planificación y scheduling, scheduling y ejecución concurrente, arquitecturas de control de fábrica, herramientas y estándares para desarrollo de sistemas de fabricación basados en agente.

En los sistemas de fabricación distribuidos, los agentes pueden ser utilizados para [<http://imsg.enme.ucalgary.ca/publication/abm.htm>]:

- ✓ encapsular sistemas software existentes para así integrar las actividades de empresas de producción tales como diseño, planificación, asignación de recursos, simulación, ejecución, y distribución de productos, con aquellas de sus proveedores, clientes y socios dentro de un entorno abierto, inteligente y distribuido a través de una red (Fox et al 1993; Barbuceanu and Fox 1995^a; Peng et al 1998; Shen et al 1998b);
- ✓ representar recursos de producción tales como trabajadores, celdas, máquinas, herramientas, planes de trabajos, vehículos guiados, así como productos, partes, operaciones (Butler and Ohtsubo 1992; Parunak 1987; Shen et al 1998b) para facilitar la planificación de recursos de producción, asignación de recursos y control de ejecución;
- ✓ modelar servicios especiales en los sistemas de producción, tales como: Agente Servidor de Nombre en CIIMPLEX (Peng et al 1998) y Mediador de Empresa en MetaMorph (Maturana and Norrie 1996; Shen et al 1998a) para proveer registro y administración de servicios; agentes Facilitadores en PACT (Cutkosky et al 1993) y CIIMPLEX (Peng et al 1998) y agentes Mediadores en MetaMorph para facilitar la comunicación, cooperación y coordinación entre otros agentes; agentes de Base de Datos (Lin and Solberg 1992) y agentes de Información (Fox et al 1993) para proveer información de dirección;
- ✓ e incorporar un planificador y un scheduler dentro de la planificación de producción y sistema de asignación de recursos (Fox et al 1993).

En los sistemas Holónicos de Fabricación, los agentes se utilizan para modelar holones los cuales son entidades software y hardware (Deen 1994; Christensen 1994; Hasegawa et al 1994; Biswas et al 1999).

Recordemos que un sistema multiagente está constituido por dos o más agentes que se comunican, a diferencia de un sistema basado en agentes que es aquel en que se utiliza la abstracción de agente, pero no hay comunicación entre los agentes del sistema.

En [<http://imsg.enme.ucalgary.ca/publication/abm.htm>] se encuentra un detallado estudio del estado del arte de la aplicación de la tecnología de agentes/sistemas multiagente a los sistemas de fabricación. En los mismos se describen diversas aplicaciones y proyectos de investigación sobre la aplicación del paradigma de agente a los sistemas de fabricación. A continuación, se proporciona un resumen de estos proyectos y aplicaciones, así como una relación de bibliografía relacionada con el tema, organizados en función de sus aportaciones en diferentes áreas.

5.1. Ontología de conocimiento de fabricación

Todos los aspectos relevantes de un dominio de una organización pueden ser considerados en términos de conocimiento, esto es aplicable a la estructura y naturaleza de la propia organización, a los datos utilizados dentro de los diferentes componentes de la organización, al flujo de estos datos a través de la organización junto con los valores añadidos durante la ejecución de las tareas de dirección. Éste conocimiento será tanto factual, en forma de hechos sobre la organización y las tareas (generalmente datos que formarían parte de bases de datos del sistema de información), como normativo, en forma de reglas de conocimientos, que en función de los valores de dichos hechos determinarían funciones dentro de la organización, relacionadas directa o indirectamente con las tareas a realizar. Para representar este conocimiento se requieren técnicas de representación que nos permitan razonar de forma óptima y eficiente sobre las tareas de la organización, así como la representación de la información requerida en la ejecución de las tareas de línea (Davis and Oliff 1988).

Previamente, grandes o muy grandes bases de conocimiento han sido frecuentemente propuestas para aplicaciones de ingeniería que incluyen diseño, fabricación, operaciones, y mantenimiento porque estas actividades requieren cantidades extremadamente grandes y diversas clases de conocimiento (Forbus 1988). Pero según Tomiyama et al (1995), no solo la cantidad de conocimiento, sino también la calidad del conocimiento en términos de reusabilidad y compartibilidad es crucial, se persigue tanto la flexibilidad como la completitud del conocimiento. Una base de conocimiento puede realizar inferencias en circunstancias particulares pero no disponer de conocimiento suficiente para inferir soluciones en otras circunstancias. Por ello en los sistemas de fabricación actuales donde la robustez y la corrección son muy importantes, disponer de grandes o muy grandes bases de conocimiento no es suficiente.

El conocimiento en la fabricación moderna debe estar bien organizado y debe poder ser aplicado flexiblemente a diferentes clases de aplicaciones. En (Tomiyama et al 1995) se proponen tres diferentes tipos de arquitectura de compartición de conocimiento.

a) Bases de conocimiento independientes. En este caso, la ‘fortaleza’ del conocimiento es la suma de la de cada una de las bases de conocimiento independientes.

b) Bases de conocimiento integradas. Aquí, las bases de conocimiento pueden ser aplicadas a diversas situaciones, y la ‘fortaleza’ del conocimiento es casi la máxima. Sin embargo, esto requiere tener una plataforma con un lenguaje uniforme. El proyecto Cyc (Lenat and Guha 1989) es un ejemplo de esta aproximación.

c) Bases de conocimiento independientes pueden comunicarse y pueden formar una situación interoperable, aunque la ‘fortaleza’ del conocimiento puede ser más débil que en (b). La base de conocimiento completa es una federación o un conjunto de agentes inteligentes débilmente acoplados. Esta aproximación ha sido adoptada en proyectos como SHADE (McGuire et al 1993), PACT (Cutkosky et al 1993), DESIRE (Brazier and Treur 1996), and DIDE (Shen and Barthès 1997).

La segunda arquitectura (b), puede sufrir la carencia de una representación uniforme del conocimiento. A menos que sea diseñada con mucho cuidado, el lenguaje de la plataforma puede no cubrir todo. La tercera arquitectura (c) puede resolver este problema, si la arquitectura es suficientemente abstracta para incorporar los diferentes tipos de

ontologías que utilice cada agente. Sin embargo, no resuelve completamente, ya que la comunicación entre agentes requiere entender lo que los otros agentes dicen.

Los mismos requerimientos de comunicación justifican el modelo de representación utilizado en (McGuire et al 1993). Este modelo es (Knowledge Interchange Format) (Genesereth and Fikes 1992), es una versión del cálculo de predicados de primer orden con extensiones que aumentan su expresividad. Las especificaciones de KIF definen la sintaxis y la semántica, la ontología define el vocabulario específico del problema. Los agentes intercambian sentencias en KIF utilizando el vocabulario compartido. Para poder compartir y reutilizar conocimiento se requiere definir un vocabulario común en el que representar dicho conocimiento. Una especificación de un vocabulario de representación para un dominio compartido de definiciones de discurso de clases, relaciones funciones y otros objetos es llamado ontología (Gruber 1993). La necesidad de ontologías compartidas es consecuencia directa de la naturaleza multidisciplinar de la ingeniería. Existen diferentes vistas de un diseño (función, ejecución, fabricación), cada una con un lenguaje diferente. Sin embargo, las diversas perspectivas se solapan normalmente, necesitando compartir información si el diseño debe realizarse concurrente y cooperativamente. Se requiere un vocabulario comprensible común para la información a compartir. En (Gruber 1993) se puede encontrar más información sobre ontologías y en (Fox et al 1996) se propone un aplicación de una ontología para modelar empresas.

En aplicaciones de diseño, es necesario representar conocimiento a diferentes niveles: conocimiento del dominio asociado con el vocabulario particular utilizado en el dominio de diseño, pero también conocimiento de ingeniería general (por ejemplo asociado con el uso de unidades de ingeniería). La representación particular de la correspondiente ontología requiere modelos, mecanismo, lenguajes y herramientas especiales.

En sistemas distribuidos, debe ser posible intercambiar conocimiento entre agentes, incluso si los agentes trabajan en especialidades diferentes. Diversos formalismos para intercambiar conocimiento han sido estudiados en diversos proyectos, y existen diversas propuestas tales como el lenguaje de representación del conocimiento KIF (Genesereth and Fikes 1992), sin embargo, en muchos de los sistemas existentes, el intercambio es realizado entre diseñadores humanos mediante e-mail.

Para que dos o más agentes puedan comunicarse es necesario que se entiendan tanto en el aspecto sintáctico como semántico de los posibles mensajes que se intercambien. A la hora de enviar un mensaje por parte de un agente se podría establecer un símil con el proceso de comunicación humana. Por una parte estaría el mensaje en sí, lo que se transmite, por otra parte estaría el sentido que se le quiere dar por parte del emisor a ese mensaje y, finalmente, la acción que realiza el receptor cuando percibe el mensaje. Este modelo de comunicación conocido como Speech Act Theory (Searle 1969) permite tener en cuenta estos tres aspectos y es empleado en el área de los sistemas multiagente como mecanismo base de comunicación.

De esta forma, con el uso de la Speech Act Theory se consigue indicar el sentido de un mensaje mediante el concepto de performativa, permitiendo restringir la semántica del acto comunicativo. Existen diferentes tipos de performativas con las que un agente

indica el sentido de su mensaje, como puede ser un requerimiento, una declaración, una pregunta, etc.

Los lenguajes de comunicación de alto nivel entre agentes incorporan esta teoría. Dichos lenguajes están constituidos fundamentalmente por un conjunto de performativas que expresan los distintos actos comunicativos que puede realizar un agente. Parece evidente pensar en la necesidad de una estandarización de la sintaxis y la semántica del lenguaje de comunicación empleado entre distintos agentes. Las propuestas más importantes en este sentido son KQML (Knowledge Query and Manipulation) (Finin et al 1994) y FIPA-ACL (Foundation for Intelligent Physical Agents - Agent Communication Language) (FIPA 2000a).

Estos lenguajes nos permiten construir mensajes y aportar un sentido al acto comunicativo entre un receptor y un emisor. Pero esto por sí sólo no es suficiente. Además, es necesario expresar el contenido de dicho mensaje en términos que entiendan los dos agentes. Para ello es necesario un lenguaje de contenido que permita expresar descripciones del dominio de forma entendible por ambos agentes. En esta línea, han aparecido lenguajes estandarizados de contenido de tipo lógico como KIF(Knowledge Interchange Format) (Genesereth and Fikes 1992) o FIPA-SL (FIPA Semantic Language) (FIPA 2000a). Por otra parte, también es necesario, que los agentes que traten de comunicarse empleen un vocabulario común y que los términos empleados en el contenido del mensaje signifiquen lo mismo tanto para uno como para el otro. Este vocabulario común viene a expresar la necesidad de que los agentes que se intenten comunicar empleen una misma ontología en dicho proceso. Una ontología puede verse como una especificación de conceptos y sus relaciones de un dominio concreto. La definición de una ontología como forma de representar los conceptos de interés de un determinado dominio, permite el entendimiento entre distintos agentes.

Protocolos de interacción

Anteriormente se ha comentado como los agentes pueden enviarse mensajes simples por medio de un lenguaje de comunicación estándar y del empleo de un lenguaje de contenido y una ontología común. Pero más complejo resulta el hecho de establecer conversaciones donde diferentes mensajes simples son intercambiados entre distintos agentes siguiendo un orden determinado.

Una conversación entre agentes es modelada mediante el concepto de protocolo de interacción, el cual establece las posibles secuencias de mensajes entre un emisor y un receptor en una conversación determinada. De esta forma, tanto uno como otro puede seguir el curso de una conversación analizando la secuencia esperada de mensajes.

En el caso más simple, un posible protocolo de petición de información se iniciaría con un mensaje de petición de datos por parte de un emisor, el receptor una vez recibida y analizada la petición podría contestarle bien que no ha entendido nada, bien que no quiere atenderle, o bien, le devolverá la respuesta a su petición. Cualquier otro tipo de mensaje sería considerado como sin sentido en esta conversación. Los protocolos de interacción pueden aumentar notablemente su complejidad permitiendo modelar conversaciones donde se establecen complicadas negociaciones entre agentes. Ejemplos de esta clase de interacciones serían establecimientos de contratos o determinados tipos de subastas.

La organización FIPA (<http://www.fipa.org>) ha desarrollado un conjunto de protocolos de interacción \cite{FIPAIP} que permiten representar un elevado conjunto de posibles situaciones de coordinación. Además dichos protocolos pueden ser modificados, refinados o empleados como componentes de protocolos más complejos.

5.2 Aproximaciones basadas en agente para modelar empresas de fabricación

Los requerimientos de un sistema de fabricación actual descritos en el apartado 2 requieren destrezas de fabricación descentralizada cuyos diseño, implementación, reconfiguración y viabilidad de fabricación permite la integración de los estados de producción en una red colaborativa dinámica. Tales destrezas pueden ser conseguidas por medio de aproximaciones basadas en agentes (Wooldridge and Jennings 1995) utilizando tecnología de compartición de conocimiento (Patil et al 1992).

A continuación se describen algunos proyectos de este dominio [<http://imsg.enme.ucalgary.ca/publication/abm.htm>].

El proyecto SHADE (McGuire et al 1993) estaba fundamentalmente interesado en aspectos de conpartición de información de ingeniería concurrente. Más que intentar modelar el proceso se diseño proporciona una infraestructura completa para prever colaboración por medio de máquina, basada en el conocimiento entre herramientas de ingeniería dispares. SHADE es distinta a otras aproximaciones en su énfasis en una aproximación distribuida al conocimiento de ingeniería más que en un modelo o conocimiento centralizado. Es decir, SHADE no sólo evita el requerimiento de conocimiento físicamente centralizado, sino que el vocabulario de modelado esta distribuido también, focalizando la representación del conocimiento en las necesidades de compartición de conocimiento.

PACT (Cutkosky et al 1993) fue un hito de demostración tanto de los esfuerzos en la investigación colaborativa como en la tecnología basada en agentes. La interacción de agentes se basa en conceptos y terminología compartida para comunicar conocimiento a través de disciplinas, una interlingua para transferir conocimiento entre agentes, y una comunicación y lenguaje de control que permite a los agentes solicitar información y servicios. Esta tecnología permite a los agentes que trabajan en diferentes aspectos de un diseño interactuar a nivel de conocimiento, compartir e intercambiar información sobre el diseño independientemente del formato en que la información está internamente codificada.

En SHARE (Toye et al 1993) se trataba sobre el desarrollo de entornos abiertos, heterogéneos, basados en redes para ingeniería concurrente. Utiliza un amplio rango de tecnologías de intercambio de información para ayudar en la colaboración de ingenieros y diseñadores en dominios mecánicos.

Recientemente PACT ha sido reemplazado por PACE (Palo Alto Collaborative Environment) [<http://cdr.stanford.edu/PACE/>] y SHARE por DSC (Design Space Colonization) [<http://cdr.stanford.edu/DSC/>].

FIRST-LINK (Park et al 1994) es un sistema de agentes semiautónomos de ayuda a especialistas en un aspecto del problema de diseño. NEXT-LINK (Petrie et al 1994) es una continuación del proyecto FIRST-LINK para comprobar la coordinación de agentes. Process-Link (Goldmann 1996), continuación de NEXT-LINK, proporciona

integración, coordinación y gestión de proyecto de servicios y herramientas CAD distribuidas que interactúan en un gran proyecto.

MADEFAST (Cutkosky et al 1996) es un proyecto financiado por DARPA DSO de demostración de tecnología desarrollado en el marco del ARPA MADE (Manufacturing Automation and Design Engineering). MADE es un programa DARPA DSO de larga duración para el desarrollo de herramientas y tecnologías que proporcionen soporte cognitivo al diseñador y permitir un orden de magnitud creciente en las alternativas exploradas en la mitad de tiempo que se tarda en explorar una única alternativa en la actualidad.

SiFA (Brown et al 1995), desarrollado en el Worcester Polytechnic, se centra en resultados de patrones de interacción, comunicación, y resolución de conflictos. DIDE propone la utilización de agents cognitivos autónomos para el desarrollo de entornos de diseño inteligentes (Shen and Barthès 1997). En AARIA (Parunak et al 1997a), los medios de fabricación (personal, máquinas, ..) son encapsulados como agentes autónomos. Cada agente interopera con otros agentes dentro y fuera de su fabrica. AARIA utiliza una mezcla de técnicas de scheduling heurísticas: scheduling forward/backward, scheduling de simulación, y scheduling inteligente. El scheduling es realizado por trabajo, por recurso, y por operación, tomándose decisiones de scheduling con objeto de minimizar costes en tiempos y cantidades de producción.

Saad et al (1995) proponen una aproximación de Reserva de la Producción utilizando un mecanismo de licitación basado en el protocolo de ContractNet para generar el plan y schedule de producción. Maturana et al (1996) describen una aproximación planificación-y-scheduling integrada que combina subtareas un clustering virtual de agentes con un protocolo de ContractNet modificado. RAPPID (Responsible Agents for Product-Process Integrated Design) (Parunak et al 1997b) del Industrial Technology Institute tiene como propósito desarrollar herramientas y método software basadas en agentes para utilizar dinámicas de mercado entre miembros de un equipo de diseño distribuido para coordinar el diseño basado en conjunto de un producto fabricado discreto. AIMS (Park et al 1993) fue desarrollado para proporcionar a los US una base industrial integrada capaz de responder rápidamente, con soluciones altamente customized, para requerimientos de cliente de cualquier magnitud, volviendo a colocar a los US como líder mundial en fabricación. ADDYMS (Architecture for Distributed Dynamic Manufacturing Scheduling) propuesta por (Butler and Ohtsubo 1992) es una arquitectura para scheduling dinámico en un entorno de fabricación. (Pan and Tenenbaum 1991) propusieron una arquitectura de agente software inteligente para integrar personas y sistemas informáticos en grandes empresas de fabricación geográficamente dispersas. Esta arquitectura esta basada en la visión de un gran número (10.000) asistentes informáticos, conocidos como Agentes Inteligentes. Los humanos que participan son encapsulados como un Asistente personal, un tipo especial de Agente Inteligente. (Roboam y Fox 1992) prpusieron Enterprise Management Network (EMN) para soportar la integración de actividades de la empresa de fabricación a través del ciclo de vida de producción.

El proyecto MetaMorph (Shen et al 1998a) utiliza una arquitectura mediador-central basada en agente híbrida para integrar socios, suministradores y clientes dinámicamente con la empresa principal a traves de sus respectivos mediadores dentro de una red de cadena de suministro vía Internet y Intranets. Los agentes pueden ser utilizados para

representar recursos (máquinas, herramientas, etc.) y piezas de fabricación, para encapsular los sistemas software existentes, para actuar como coordinadores de sistema/subsistema (mediadores), ejecutar una o más funciones de la cadena de suministro.

5.3. Integración de empresas y gestión de la cadena de suministro

Integración de Empresa significa que cada unidad de la organización tendrá acceso a la información relevante para su tarea y conocerá como sus acciones influirán en otras partes de la organización permitiéndole por ello elegir alternativas que optimicen los objetivos de la organización. La cadena de suministro de una empresa de fabricación es una red de suministradores, factorías, warehouses, centros de distribución y detallistas a través de la cual los materiales son adquiridos, transformados y entregados a los clientes (Fox et al 1993). Mejorar la gestión de la cadena de suministro es una estrategia clave para incrementar la posición competitiva y beneficios de la empresa. Por ello, las empresas tienden a adoptar arquitecturas abiertas para integrar sus actividades con las de sus suministradores, clientes y socios dentro de amplias redes de cadenas de suministro. La tecnología basada en agentes proporciona una forma natural de diseñar e implementar tales entornos. La tabla 2, recoge una relación de diversos proyectos en esta área.

Con el fin de adecuarse a la competitividad global y al rápido grado de reacción del mercado, una empresa de fabricación tiene que estar integrada con sus sistemas de gestión relacionados (facturación, pedidos, diseño, producción, planificación y scheduling, control, transporte, recursos, personal, materiales, calidad, etc.), sus socios, suministradores y clientes vía redes de comunicaciones (redes locales Internet o Intranet), que son, en general, entornos software y hardware heterogéneos.

La cadena de suministro de una empresa de fabricación es una red ww de suministradores, fabricas, warehouses, centros de distribución y minoristas a través de la cual materiales en crudos son adquiridos, transformados y entregados a los clientes (Fox et al 1993). Esta red integra también, en general, entornos heterogéneos. Las aproximaciones basadas en agente proporcionan una forma natural de diseñar e implementar integración de empresas y gestión de la cadena de suministros dentro de tales entornos. Puede que (Fox et al 1993) fuesen los primeros en proponer la organización de la cadena de suministro como una red de agentes inteligentes cooperantes. Una propuesta similar fue hecha por (Swaminathan et al 1996) utilizando una arquitectura multiagente para modelar la dinámica de la cadena de suministro. En ISCM (Fox et al 1993), cada agente ejecuta una o más funciones de la cadena de suministro y coordina sus acciones con otros agentes. En la biblioteca de cadena de suministro propuesta por (Swaminathan et al 1996), se distinguen dos categorías de elementos: elementos estructurales y elementos de control. Los elementos estructurales incluyendo elementos de producción (detallistas, centros de distribución, plantas, suministradores) y elementos de transporte son modelados como agentes. Los elementos de control (inventario, demanda, suministro, flujo y controles de información) son utilizados para ayudar en la coordinación del flujo de productos de una forma eficiente mediante la utilización de mensajes.

MetaMorph II (Shen et al 1998a) utiliza una arquitectura híbrida mediador-central basada en agentes para integrar socios, suministradores y clientes dinámicamente con la

empresa principal por medio de sus respectivos medidores dentro de una red de cadena de suministro vía Internet e intranets. En MetaMorph II, los agentes pueden ser utilizados para representar recursos de fabricación (máquinas, herramientas, etc.) y piezas, para encapsular los sistemas software existentes, para funcionar como coordinadores (mediadores) de sistemas/subsistemas, y para realizar una o más funciones de la cadena de suministro. Algunos investigadores también han propuesto aplicar la tecnología de agentes móviles a la integración de empresa y gestión de la cadena de suministro (Brugali et al 1998; Papaioannou and Edwards 1998; Yan et al 1998).

Los resultados de la investigación han demostrado que las aproximaciones basadas en agentes proporcionan las siguientes ventajas en la integración de empresa y gestión de la cadena de suministro:

- ✓ Incrementar el grado de reacción de la empresa a los requerimientos de mercado,
- ✓ Involucrar a los clientes en la optimización total de la cadena de suministro,
- ✓ Realizar la optimización de la cadena de suministro por medio de la asignación efectiva de recursos,
- ✓ Conseguir la optimización dinámica de gestión de materiales e inventario,
- ✓ Incrementar la efectividad del intercambio y realimentación de la información.

Sin embargo, los problemas de seguridad resultado de una arquitectura abierta de sistemas basados en agentes, en concreto cuando se utiliza Internet y tecnología de agentes móviles, son evidentes, aunque estos problemas no son específicos de los sistemas multiagente y puede que sean mitigados con futuras investigaciones.

Tabla 2. Proyectos que utilizan la tecnología de agents en integración de empresa y gestión de la cadena de suministro.

Proyecto	Grupo de Investigación	Dominio	Principales Características
ABMA	Architecture Tech. Co (Budenske et al 1998)	Integración de empresa	Arquitectura Middleware
ADE	Gensym Co (Mehra & Nissen 1998)	Gestión Cadena	Utiliza delegación basada en el manejo de eventos similar a JavaBeans
AIMS	Lockheed Martin (Park et al 1993)	Fabricación Agíl	Utiliza internet
ATP	NIST (Nist 1998)	Fabricación Agíl	Arquitectura Plug-and-play
CIIMPLEX	UMBC (Peng et al 1998)	Integración de empresa	Agentes de Servicio (Servidor de nombres, agente facilitador, agente gateway)
CLAIM	RMIT (Malkoun & Kendall 1997)	Integración de empresa	Método para integración de empresa utilizando

			agentes.
IA framework	EIT (Pan & Tenenbaum 1991)	Integración de empresa	Gran cantidad de asistentes informáticos (Agentes Inteligentes)
IAO	U. of Calgary (Kwok and Norrie 1994)	Fabricación Inteligente	Un sistema de objetos basado en reglas para el desarrollo de software de fabricación inteligente.
iDCSS	CSS, MIT (Klein 1995)	Ingeniería concurrente	Un modelo integrado que combina tecnologías de coordinación existentes.
ISCM	U. of Toronto (Fox et al 1993)	Gestión cadena suministro	Shell de construcción de agentes (ABS); Lenguaje de coordinación (COOL); agentes funcionales.
KRAFT	Consortio KRAFT (Gray et al 1998)	Transformación y reutilización del conocimiento	Mediadores como brokers de conocimiento.
Madefast	Stanford (Cutkosky et al 1996)	Ingeniería colaborativa	Utiliza internet
MADEsmart	Boeing (Jha et al 1998)	Ingeniería colaborativa	Agentes Wrapper para encapsulación.
MetaMorph I	U of Calgary (Maturana & Norrie 1996)	Fabricación Inteligente	Arquitectura mediador-central, clustering dinámico y clonación, aprendizaje
MetaMorph II	U of Calgary (Shen et al 1998a)	Fabricación Inteligente, Gestión cadena suministro.	Arquitectura híbrida, medidores como coordinadores de subsistemas e interfaces con el sistema principal.
	Politecnico di Torino (Brugali et al 1998)	Gestión cadena suministro.	Utiliza agentes móviles en un proceso industrial.
	Fleury et al 1996	Optimización del sistema de fabricación	Multiagente, simulated

			annealing, y simulación
	Loughborough (Papaioannou & Edwards1998)	Empresa virtual	Utiliza agents móviles
	Sandia Lab (Pancerella et al 1995)	Fabricación ágil	Agente definido como componente software encapsulado autónomo.
	CMU (Swaminathan et al 1996)	Gestión cadena suministro	Biblioteca de cadena de suministro con elementos estructurales (agentes) y elementos de control para la coordinación.
	ETH Zentrum (Wunderli et al 1996)	Sistemas CIM	Agentes de bases de datos para sistemas CIM
	Leipzig (Yan et al 1998)	Gestión de proyecto	Utiliza agents móviles.

5.4 Control, Scheduling y Planificación de la fabricación

Planificar es el proceso de seleccionar y secuenciar actividades cuya ejecución consigue uno o más objetivos y satisfacen un conjunto de restricciones del dominio. Scheduling es el proceso de seleccionar entre planes alternativos y asignar recursos y tiempos al conjunto de actividades del plan. Estas asignaciones deben satisfacer un conjunto de reglas o restricciones que reflejan las relaciones entre actividades y limitaciones de disponibilidad de un conjunto de recursos compartidos. Las asignaciones también afectan a la optimalidad de un schedule con respecto a criterios tales como costes, retrasos, etc. En resumen, scheduling es un proceso de optimización donde recursos limitados son asignados en el tiempo tanto entre actividades paralelas como secuenciales (Zweben and Fox 1994).

El scheduling de fabricación es un problema difícil, particularmente cuando se da en un entorno abierto y dinámico. En un sistema de fabricación, es extraño que el hacer algo produzca el efecto esperado. El conjunto de cosas a hacer es generalmente dinámico. Se puede requerir al sistema que realice tareas adicionales que no habían sido previstas, y algunas veces está permitido omitir ciertas tareas. Los recursos disponibles para ejecutar las tareas están sujetos a cambios. Ciertos recursos pueden convertirse en no disponibles, y pueden introducirse recursos adicionales. El tiempo de comienzo y el tiempo de procesamiento de una tarea están sujetos también a variaciones. Una tarea puede consumir más o menos tiempo del previsto, y las tareas pueden activarse más pronto o más tarde. Debido a sus aspectos altamente combinatorios, su naturaleza dinámica y su interés práctico para los sistemas de fabricación, el problema de scheduling ha sido ampliamente estudiado en la literatura mediante diversas técnicas:

heurísticas, técnicas de propagación de restricciones, formalismo de problema de satisfacción de restricciones, simulated annealing, búsqueda Taboo, algoritmos genéticos, redes neuronales, etc. Recientemente, también la tecnología de agentes ha sido utilizada en la resolución de este problema.

El control de fabricación está relacionado con estrategias y algoritmos para operar una planta de fabricación, teniendo en cuenta tanto los estados pasados como presentes de la planta de fabricación, y también la demanda del mercado. El problema de control de fabricación puede ser considerado a dos niveles: nivel alto y nivel bajo. A nivel bajo, los recursos de fabricación individuales son controlados para producir unidades de proceso esperadas por las funciones del control de nivel alto. El control de fabricación de nivel alto concierne a los recursos de fabricación disponibles para fabricar las cantidades deseadas de tipos de productos. En un sistema de fabricación basado en agentes, la tecnología de agentes es aplicada normalmente al control de fabricación de nivel alto, pero también es aplicada en el nivel bajo (Brennan et al 1997; Wang et al 1998).

Puede que Shaw fuese el primero en proponer la utilización de agentes en scheduling de fabricación y control de fábrica. Propuso que una celda de fabricación podría subcontratar trabajo con otras celdas mediante mecanismos de licitación (Shaw and Whinston 1983; Shaw 1988). YAMS (Yet Another Manufacturing System) (Parunak 1987) fue otro pionero de los sistemas de fabricación basados en agentes, donde cada fábrica y componente de la fábrica es representado como un agente. Cada agente tiene una colección de planes, que representan sus capacidades. El ContractNet es utilizado para la negociación entre agentes.

Durante la revisión de la literatura de investigación, hemos encontrado 30 proyectos que utilizan la tecnología de agentes en planificación de fabricación, scheduling y control de ejecución. La tabla 3 presenta un resumen de estos proyectos.

Scheduling Dinámico Distribuido.

En correspondencia con las dos distintas aproximaciones para encapsulación de agentes del apartado 6.2, se puede distinguir entre dos tipos de sistemas de scheduling dinámico distribuido:

1) Aquellos donde el 'scheduling' es un proceso de búsqueda incremental que puede implicar backtracking. Agentes, responsables de planificar pedidos, realizan búsquedas incrementales locales para sus pedidos y pueden considerar recursos múltiples. El 'schedule' global es obtenido uniendo los 'schedule's locales. Esto es muy parecido al 'scheduling' centralizado.

Sistemas en los que un agente representa un único recurso (una celda de trabajo, una máquina, una herramienta, un trabajador, etc.) y es responsable de planificar este recurso. Este agente puede negociar con otros agentes para realizar el 'scheduling' global.

Ejemplos de este primer tipo de sistemas de scheduling pueden ser encontrados en (Sadeh and Fox 1989; Sycara et al 1991; Burke and Prosser 1991; Fordyce and Sullivan 1994; Murthy et al 1997; McEleney et al 1998).

2) Para la segunda aproximación, el mecanismo de 'scheduling' es realizado generalmente por medio de la negociación entre agentes. Los protocolos de negociación incluyen mecanismo de evaluación (Fordyce and Sullivan 1994), Redes de Contrato de Smith (Smith 1980) o sus versiones modificadas, tales como el Protocolo de Red de Contrato Extendido (Extended Contract Net Protocol –ECNP-) propuesto por (Fischer et al 1995), la Red de Contrato

Dirigida por Mercado de (Baker 1991), la B-Contract-Net de (Scalabrin 1996), y el Leveled Commitment Contracting Protocol de (Sandholm and Lesser 1996). Otros ejemplos pueden ser encontrados en (Duffie and Piper 1986; Parunak 1987; Ow and Smith 1988; Shaw 1988; Butler and Ohtsubo 1992; Lin and Solberg 1992; Saad et al 1995; Shen and Norrie 1998; Ouelhadj et al 1998). El mecanismo de licitación puede ser orientado a la pieza (Duffie and Piper 1986; Ow and Smith 1988; Lin and Solberg 1992), orientado al recurso (Butler and Ohtsubo 1992; Baker 1991; Shen and Norrie 1998), o bi-direccional (Saad et al 1995).

Utilizamos aquí el término ‘scheduling dinámico’ para indicar que un sistema de scheduling de fabricación de tiempo real puede actualizar su schedule para adaptarse a situaciones cambiantes tales como la inserción de un nuevo pedido, fallos en las máquinas, retrasos en los trabajos, etc. ADDYMS desarrollo un mecanismo de scheduling dinámico para asignación local de recursos a nivel de celda de trabajo local (Butler and Ohtsubo 1992). En MetaMorph II, diversos mecanismos fueron desarrollados para scheduling dinámico y rescheduling combinando mecanismos de licitación basados en el protocolo de Contract Net con un mecanismo de mediación basado en la arquitectura Mediador (Shen and Norrie 1998). Sousa y Ramos (Sousa and Ramos 1997) han propuesto una arquitectura de scheduling dinámico utilizando holones.

Integración de Planificación y Scheduling

El concepto de planificación y scheduling de fabricación ha sido introducido anteriormente. Aproximaciones tradicionales a la planificación y el scheduling no consideran las restricciones de ambos dominios simultáneamente. A pesar de ser subóptimas estas aproximaciones han estado en boga debido a la no disponibilidad de una estructura unificada. Las aproximaciones basadas en agentes proporcionan una posible forma de integrar actividades de planificación y scheduling por medio de la coordinación a nivel de empresa entre sistemas de diseño de producto y el sistema de scheduling de recursos de la fábrica. MetaMorph I implementa este tipo de mecanismo por medio de coordinación a nivel de empresa entre Mediadores de Diseño y Mediadores de Recursos quienes en orden coordinan a los agentes de recurso a nivel de fábrica (Maturana et al 1996).

Los sistemas tradicionales alterna scheduling y ejecución. Por ejemplo, una firma desarrolla cada noche un schedule de sus operaciones de fabricación del siguiente día. El mundo real tiende a cambiar de forma que invalida los schedules anticipados. Los sistemas naturales no planifican anticipadamente, sino que ajustan sus operaciones en una escala de tiempo comparable a aquella en que su entorno cambia. MetaMorph II (Shen et al 1998a) propone utilizar Mediadores de Ejecución para coordinar la ejecución de máquinas, AGVs y trabajadores como sea necesario. Mediante tal aproximación, los recursos de fabricación (máquinas, interfaces para trabajadores, etc.) pueden ser conectados directamente al sistema de fabricación. Como el sistema de fabricación está en progreso, la información sobre la ejecución del proceso (progreso del schedule) es capturada por los Mediadores de Ejecución que ajustaran el schedule si es necesario.

Tabla 3. Proyectos que utilizan tecnología de agents en planificación de fabricación, scheduling y control de ejecución

Proyecto	Grupo de Investigación	Dominio	Principales Características
AARIA	ITI, U. of Cincinnati (Parunak et al 1998)	Scheduling y Control de Fabricación	Utiliza agentes autónomos para representar entidades físicas, procesos y operaciones.
ABACUS	UCB, UMIST (McEleney et al 1998)	Scheduling de Fabricación	Utiliza agentes funcionales; aproximación BDI para diseño de agentes.
ADDYMS	(Butler & Ohtsubo 1992)	Scheduling de Fabricación	Los agentes representan recursos físicos; scheduling de recursos local dinámico.
AMACOIA	U. of Neuchatel (Sprumont & Muller 1996)	Diseño de líneas flexibles de ensamblaje.	Utiliza simulated annealing para la búsqueda en el espacio del problema.
AMC	Sandia Lab (Goldsmith & Interrante 1998)	Scheduling de Fabricación	Utiliza agentes físicos: agentes máquina y agentes pieza.
ARMOSE	Odense U. (Overgaard et al 1994)	Robótica	Cada articulación del robot es modelada como un agente.
CAMPS	Miyashita 1998	Scheduling y planificación de Fabricación.	Metodología basada en reparaciones junto con mecanismos basados en restricciones.
CORTES	CMU (Sadeh & Fox 1989, Sycara et al 1991)	Scheduling de Fabricación	Técnicas micro-opportunistas para resolución de problemas de scheduling.
DAS	U. of Strathclyde (Burke & Prosser 1991)	Scheduling de Fabricación	Arquitectura jerárquica con agentes que representan recursos, grupos de recursos, y un proceso de

			scheduling completo.
I-Control	U of Calgary (Brennan et al 1997, Wang et al 1998)	Control de sistemas de fabricación.	Jerarquía de control dinámico parcial (PDCH); utiliza agentes para modelar bloques funcionales (IEC 1997).
IFCF	Purdue (Lin and Solberg 1992)	Control y Scheduling de Fabricación	Agentes recurso representan recursos físico; modelo de control tipo mercado.
LMS	(Fordyce & Sullivan 1994)	Scheduling de Fabricación	Utiliza agentes funcionales; protocolo de votación para la comunicación.
MAPP	U. of Minnesota (Hayes 1998)	Planificación de procesos	Utiliza arquitectura de blackboard.
MASCADA	Daimler-Benz AG, KULeuven (Bruckner et al 1998)	Control y Scheduling de Fabricación	Comportamiento emergente en control de fabricación; manejo proactivo de perturbaciones; agentes activables en caliente.
MASCOT	ITI (Parunak 1993)	Control y Scheduling de Fabricación	Una ontología compartida y un conjunto base de módulos realísticos.
Reagere	Penn State U.(Berry & Kumura 1998)	Control y Scheduling de Fabricación	Basado en una arquitectura de blackboard.
Sensible Agents	U of Texas at Austin (Barber et al 1998)	Scheduling de Fabricación	Implementado como objetos CORBA que se comunican a través del entorno de objetos ILU.
SFA	NCMS (Parunak 1996)	Control y Scheduling de Fabricación	Aplicaciones de fabricación reales.
YAMS	ITI (Parunak 1987)	Control y Scheduling de Fabricación	Una de las primeras aplicaciones del dominio.
	U. of Cincinnati (Baker 1991)	Control y Scheduling de Fabricación	ContractNet dirigida al Mercado; scheduling forward y backward.

	(Choi and Park 1997)	Scheduling de planta de Shipbuilding Yard	Un método económico para desarrollar sistemas de agentes inteligentes.
	Wisconsin (Duffie & Piper 1986)	Control y Scheduling de Fabricación	Los agentes representan recursos físicos, piezas, y personas; scheduling orientado a las piezas.
	DFKI (Fischer 1994)	Control y planificación de Fabricación	Arquitectura jerárquica por niveles.
	Toshiba Hasegawa et al 1994)	Planificación y Scheduling de Fabricación	Utiliza una aproximación HMS.
	Sandia Lab(Interrante & Goldsmith 1998)	Control y Scheduling de Fabricación	Agentes Tipo-A, Tipo -B y Tipo -C.
	Vanderbilt (Saad et al 1995)	Control y Scheduling de Fabricación	Aproximación de reserve de la producción; negociación centrada en máquina y piezas.
	(Kouiss et al 1997)	Scheduling de fabricación.	Cada agente representa un centro de trabajo.
	CMU (Liu & Sycara 1994, 1995)	Satisfacción de restricciones Job Shop y Optimización	CP&CR (Partición de restricciones y reacción coordinada) para satisfacción de restricciones; Anchor & Ascend para optimización distribuida de restricciones.
	(Murthy et al 1997)	Scheduling de fabricación.	Scheduling de fabricación.
	U. of Toulouse 1 (Ouelhadj et al 1998)	Control y Scheduling de Fabricación	Los agentes representan recursos físicos.
	CRAN GGP Patriti et al 1997, Schaefer et al 1996)	Control y Scheduling de Fabricación	Arquitectura por niveles; utiliza diferentes mecanismos a niveles diferentes.
	ISEP/IPP (Sousa &	Scheduling de	Utiliza una

	Ramos 1997)	Fabricación	aproximación HMS; scheduling dinámico.
	HKUST (Tseng et al 199)	Control y Scheduling de Fabricación	Modelo tipo Mercado para fabricación con agentes que representan recursos.

5.5 Encapsulación de agentes.

Entre las diferentes aproximaciones utilizadas para la encapsulación de agentes en sistemas de fabricación basados en agentes, dos aproximaciones son inconfundibles: la aproximación de descomposición funcional y la aproximación de descomposición física.

En la aproximación de descomposición funcional, los agentes son utilizados para encapsular módulos asignados a funciones tales como adquisición de pedidos, planificación, scheduling, manipulación de material, gestión de transporte, y distribución de producto. No hay relaciones explícitas entre agentes y entidades físicas. Ejemplos de este tipo de aproximaciones son ISCM (Fox et al 1993), CIIMPLEX (Peng et al 1998), ABACUS (McEleney et al 1998), y LMS (Fordyce and Sullivan 1994).

En la aproximación de descomposición física, los agentes son utilizados para representar entidades del mundo físico, tales como trabajadores, máquinas, herramientas, instalaciones, productos, piezas, características, operaciones, etc. Hay una relación explícita entre un agente y una entidad física. Ejemplos de esta aproximación pueden ser encontrados en MetaMorph I & II (Maturana and Norrie 1996; Shen et al 1998a), ADDYMS (Butler and Ohtsubo 1992), AIMS (Park et al 1993), AARIA (Parunak et al 1998), YAMS (Parunak 1987).

La aproximación funcional tiende a compartir diversas variables de estado entre diferentes funciones. Distintos agentes deben compartir algunas variables de estado, esto puede producir problemas de consistencia e interacciones no previstas. La aproximación física define naturalmente distintos conjuntos de variables que pueden ser manejados eficientemente por agentes individuales con interacciones limitadas.

Sin embargo, la aproximación funcional es muy útil en la integración de sistemas ya existentes (herramientas CAD, sistemas MRP, etc.) y en la resolución de problemas de legado. Incluso, en un sistema de fabricación que utilice la aproximación física, la aproximación funcional puede resultar útil. Estos agentes que encapsulan algunas funciones especiales pueden ser utilizados para proporcionar tales servicios en el nivel de sistema, como pueden ser facilitadores en PACT (Cutkosky et al 1993), agentes broker en CIIMPLEX y AIMS, mediadores en MetaMorph I & II, y monitores del sistema en DIDE (Shen and Barthès 1996a) y en IFCF (Lin and Solberg 1992).

5.6 Organización Multiagente – Arquitecturas de Sistema.

Las diferentes arquitecturas propuestas en la literatura para sistemas de fabricación basados en agentes pueden ser clasificadas en tres categorías: la aproximación Jerárquica, la aproximación de Federación y la aproximación de Agente Autónomo.

Una empresa de fabricación moderna característica consta de una cantidad, frecuentemente distribuidas físicamente, unidades semi-autónomas, cada una con un grado de control sobre los recursos locales o con diferentes requerimientos de información. Para tal situación real, diversas aplicaciones industriales reales basadas en agente aún utilizan la arquitectura jerárquica, aunque esto puede ser criticado por su apariencia centralizada. Ejemplos de esta aproximación pueden ser encontrados en HMS (Van Leeuwen and Norrie 1997; Bussmann 1998), ADDYMS (Butler and Ohtsubo 1992), DAS (Burke and Prosser 1991), y en la Estructura de Planificación y Control de Fischer (1994).

Tres aproximaciones han sido propuesta para las arquitecturas de federación: Facilitadores, Brokers y Mediadores. En la aproximación de Facilitador, algunos agentes relacionados son combinados dentro de un grupo. La comunicación entre agentes se realiza siempre por medio de una interfaz llamada facilitador. Cada facilitador es responsable de proporcionar una interfaz entre una colección local de agentes y agentes remotos, usualmente proporcionando dos servicios principales: (1) direccionando los mensajes de salida a los destinos apropiados; (2) traduciendo los mensajes de entrada para ser consumidos por sus agentes. CIIMPLEX (Peng et al 1998), PACT (Cutkosky et al 1993) y otros proyectos basados en SHADE (McGuire et al 1993; Park et al 1994; Petrie et al 1994) utilizan esta aproximación.

Los Brokers (o agentes broker) son similares a los facilitadores con algunas funciones adicionales como son la monitorización y la notificación. La diferencia funcional entre un facilitador y un broker es que el facilitador es responsable un grupo designado de agentes, mientras que cualquier agente puede contactar con cualquier broker del mismo sistema para encontrar agentes de servicio que realicen una tarea especial. Agentes broker pueden ser encontrados en AIMS (Park et al 1993) y CIIMPLEX (Peng et al 1998).

La aproximación Mediador es otro tipo de arquitectura de federación. Además de las funciones de un facilitador y un broker, un mediador asume el papel de coordinador del sistema proporcionando la cooperación entre agentes inteligentes y el aprendizaje a partir del comportamiento de los agentes. Una descripción completa del concepto de mediador puede ser encontrada en (Gaines et al 1995). Aplicaciones que utilizan mediadores en sistemas de fabricación inteligente pueden ser encontradas en (Gaines et al 1995; Maturana and Norrie 1996; Shen et al 1998a; Ouelhadj et al 1998).

Las arquitecturas multiagente de federación (la aproximación Facilitador, la aproximación Broker y la aproximación Mediador) pueden coordinar la actividad del sistema multiagente vía la ‘facilitación’ como medio de reducir sobrecargas, asegurar la estabilidad, y proporcionar escalabilidad. La aproximación federación promete ser una buena base sobre la que desarrollar arquitecturas de sistema multiagente escalable y abierta (Genesereth and Ketchpel 1994).

La aproximación de Agente Autónomo es diferente. Aunque se han propuesto diferentes definiciones de agentes autónomos, argumentamos que un agente autónomo debería tener, al menos, las siguientes características: (1) no es controlado ni gestionado por ningún otro agente software ni humano, (2) puede comunicarse/interactuar directamente con cualquier otro agente del sistema y también con otros sistemas externos, (3) tiene conocimiento sobre otros agentes y sus entornos, (4) tiene sus propios objetivos y un conjunto asociado de motivaciones. DIDE utiliza esta aproximación para desarrollar sistemas de diseño de ingeniería basados en agente (Shen and Barthès 1996a). AARIA también utiliza esta aproximación, pero con protocolos de negociación fijos (Parunak et al 1998).

Desde nuestra experiencia, la aproximación de agente autónomo es muy adecuada para desarrollar sistemas de diseño inteligentes distribuidos donde las herramientas de ingeniería existentes son encapsuladas como agentes y conectadas al sistema para proporcionar servicios especiales, y el sistema consta de un pequeño número de agentes. Este tipo de arquitectura es también muy útil para desarrollar sistemas de múltiples robots autónomos. En la arquitectura de mediador, se impone una jerarquía estática o dinámica para cada tarea específica, lo que proporciona simplicidad y manejabilidad computacional. Este tipo de arquitectura es completamente adecuada para desarrollar sistemas distribuidos de fabricación que son complejos, dinámicos, y constituidos por gran cantidad de agentes recurso. Una combinación de las aproximaciones anteriormente mencionadas como una aproximación híbrida fue propuesta en MetaMorph II (Shen et al 1998a) para desarrollar sistemas dinámicos de fabricación más flexibles, dinámicos y escalables.

5.7 Reconfiguración dinámica del sistema

Los entornos de fabricación del mundo real son altamente dinámicos debido a la diversidad de situaciones que cambian frecuentemente: cambio de intereses bancarios, cambio de situaciones políticas, retrasos en la recepción de materiales, fallos en las funcionalidades de producción, llegan nuevos pedidos y los pedidos existentes son cancelados o cambiados. Tales situaciones cambiantes producen desviaciones de los planes y schedules existentes. Es por ello necesario que la arquitectura del sistema reúna tales requerimientos y que el sistema en funcionamiento se adapte a tales entornos cambiantes.

En AIMS (Park et al 1993), utilizando Internet para conectar agentes, pueden añadirse o deshabilitarse servicios en cualquier instante, con un impacto incremental sobre la capacidad, simplemente informando a los directorios y brokers adecuados. MetaMorph I & II (Maturana and Norrie 1996; Shen et al 1998a) también proporcionan esta característica utilizando un mecanismo de registro tanto a nivel de sistema como a nivel de mediador. AARIA (Parunak et al 1998) proporciona tal característica por medio de su aproximación de Agente Autónomo, y ATP (NIST 1998) por medio de su arquitectura plug-and-play.

5.8 Aprendizaje en Sistemas de Fabricación Basados en Agentes

Para muchas tareas de aplicación es extremadamente difícil o incluso imposible determinar correctamente a priori el repertorio de comportamientos y actividades concretas de un sistema de fabricación basado en agentes, es decir, en tiempo de diseño

y antes de su uso. Esto requeriría, por ejemplo, que se conozca a priori que requerimientos del entorno emergerán en el futuro, que agentes estarán disponibles cuando haya una emergencia, y como tienen que interactuar los agentes disponibles en respuesta a estos requerimientos. Estos problemas fruto de la complejidad de los sistemas basados en agentes pueden ser evitados o al menos reducidos dotando a los agentes de la habilidad de aprender, es decir, con la habilidad de mejorar la funcionalidad futura del sistema total o de parte del sistema. Por esto, el aprendizaje es una de las técnicas clave en la implementación de sistemas de fabricación basados en agente. Una discusión más detallada sobre aprendizaje en sistemas de fabricación basados en agentes puede ser encontrada en (Shen et al 1998b).

5.9 Evaluación de diseño y manufacturabilidad

Especificaciones geométricas y funcionales, disponibilidad de materiales en crudo, y la funcionalidad y disponibilidad de recursos de planta tienen cada vez una mayor influencia en la viabilidad de fabricación. Un diseño puede ser fabricable bajo una combinación de requerimientos de producto y recurso de planta, pero no bajo otros. Cualquier sistema de fabricación integrado basado en agentes debería proporcionar evaluaciones dinámicas de manufacturabilidad durante el proceso de diseño de producto.

La arquitectura MetaMorph I proporciona un mecanismo para evaluaciones inmediatas de manufacturabilidad (Maturana and Norrie1996). Como una pieza de un producto es diseñada progresivamente por instanciaciones repetida de características, la manufacturabilidad es evaluada por agentes recurso para cada instanciación. Mediadores de Diseño y Mediadores de Recurso aseguran la coordinación entre piezas de diseño agentes recurso. Los subsistemas de diseño (o agentes de diseño) interactúan con agentes recurso vía Mediadores de Recurso para obtener evaluaciones de manufacturabilidad durante el proceso de diseño de producto. Este proceso no sólo asegura la manufacturabilidad de un producto, sino que da lugar a una identificación incremental de los planes generales de procesos.

5.10 Arquitecturas de Control de Fábrica

Las arquitecturas de control para sistemas de fabricación distribuidos de tiempo real necesitan comúnmente satisfacer requerimientos tales como autonomía, seguridad, tolerante a fallos, interoperabilidad, reconfigurabilidad, y otras funcionalidades de tiempo real. En contraste con los sistemas de fabricación tradicionales que utilizan arquitecturas centralizadas de control, los sistemas de fabricación distribuidos basados en agentes utilizan arquitecturas descentralizadas. Los controles descentralizados tienen tres arquitecturas características: jerárquicas, oligárquica y heterárquica (Brennan et al1997). Generalmente, la aproximación de sistemas de control de gran escala jerárquicos implica descomponer el sistema global en subsistemas más pequeños que tiene interacciones más débiles con cada uno de los otros y también un menor grado de autonomía cooperativa. En un sistema jerárquico, los subsistemas o elementos de un nivel más bajo reciben instrucciones de aquellos que están en el siguiente nivel más alto, lo que restringe su autonomía. La aproximación oligárquica proporciona caminos de comunicación menos rígidos y soporta un mayor grado de autonomía. La aproximación heterárquica soporta grados aún mayores de autonomía, ofrece expectativas de reducir la complejidad, reducir los costes de desarrollo de software, alta

modularidad, alta flexibilidad, y mejora la tolerancia a fallos (Duffie and Piper 1986), pero consume mucho más tiempo durante la toma de decisiones, especialmente cuando un gran número de agentes están conectados.

Duffie y Prabhu (Duffie and Prabhu 1994) propusieron una técnica de scheduling oportunística heterárquica para control de celda de trabajo de máquina. Baker (1991) desarrollo un Contract Net dirigido por el Mercado para sistemas de fabricación multiagente heterárquicos. Lin y Solberg (1992) propusieron una estructura genérica para controlar el flujo de trabajo en sistemas de fabricación. Basándose en un modelo de mercado y una combinación de mecanismo de objetivo y precios, la estructura permite objetivos de trabajo heterogéneos, admite prioridades de trabajo, reconoce recursos múltiples, y permite múltiples pasos de negociación entre piezas y recursos. Brennan et al (Brennan et al 1997) propusieron una Jerárquica de Control Dinámica Parcial combinando arquitecturas jerárquicas y heterárquicas. El uso de jerarquías dinámicas parciales favorece la reconfigurabilidad y puede proporcionar mejores prestaciones del sistema que cualesquiera aproximaciones de control único.

5.11 Herramientas y estándares

Muchos de los trabajos de investigación y proyectos revisados en este capítulo utilizan lenguajes de programación de tradicionales, como C++, Java, Lisp, SmallTalk, Prolog, para desarrollar sistemas de fabricación basadas en agentes. Para un uso más amplio de la tecnología de agentes en el desarrollo de sistemas de fabricación basados en agentes, son necesarias potentes herramientas de desarrollo de agentes. Recientemente, hay ya algunas comercialmente disponibles. ABS (Agent Building Shell) está desarrollada en in EIL de la Universidad de Toronto especialmente para el desarrollo de agentes de empresa cooperativos (Barbuceanu and Fox 1995a; 1995b; 1996). Está siendo utilizado para desarrollar aplicaciones multiagente en el área de integración de cadena de suministro de empresas de fabricación. VoyagerTM de ObjectSpace proporciona un Object Request Broker (ORB) basado en Java diseñado para agents móviles. ObjectSpace, junto con AMD (Advanced Micro Devices), gana el premio de I+D de 1997 NIST ATP (Advanced Technology Program) de \$4.9M para crear un Sistema de Fabricación Intensificado en Agentes VoyagerTM(ObjectSpace 1997). ZEUS es una herramienta desarrollada por British Telecom para la construcción de sistemas multiagente distribuidos (Nwana et al 1998). ADE de (Agent Development Environment) utiliza su entorno de desarrollo de software de fabricación inteligente G2 (Gensym 1997). ADE ha sido aplicado al desarrollo de sistemas basados en agents para la cadena de suministro (Mehra and Nissen 1998). También está siendo utilizado en el proyecto NCMS Shop Floor Agents (SFA) (NCMS 1998).

Otras herramientas generales de desarrollo de agentes son los Aglets SDK de IBM(IBM 1998), OdysseyTM de General Magic (General Magic 1997), ADT (Agent Development Toolkit)de SRI y OAA (Open Agent Architecture) (Martin et al 1998), JATLite de Stanford (Stanford 1997), Cybele de AARIA team (Baker et al 1997), y DESIRE (Brazier et al 1998).

El amplio uso de la tecnología de agentes en la industria depende de la disponibilidad de herramientas de desarrollo y plataformas que protejan a los desarrolladores de la necesidad de desarrollar funcionalidades básicas en cada sistema. Tales herramientas y plataformas, sucesivamente, presumen la existencia de estándares que reflejan los

acuerdos de desarrolladores sobre que funcionalidades básicas deberían reunir y cómo deberían ser presentadas. Algunos esfuerzos han sido dedicados en proporcionar estándares para sistemas basados en agentes, pero no se puede encontrar estándares aceptados para desarrollar sistemas de fabricación basados en agentes. KQML (Finin et al 1993) intenta ser un lenguaje de comunicación entre agents común, con KIF (Genesereth and Fikes 1992) como formato de contenido común. Algunos estándares tradicionales han sido también utilizados en desarrollos de sistemas basados en agentes, tales como, CORBA (Common Object Request Broker Architecture) (OMG 1998) para comunicación entre agentes, y STEP (Bjork and Wix 1991) para proporcionar semántica de mensajes en aplicaciones de fabricación. Recientemente, dos consorcios están focalizados en la formalización de estándares específicamente de soporte de agentes. La Foundation for Intelligent Physical Agents (FIPA), establecida in 1996 como un consorcio www, promueve el desarrollo de especificaciones de tecnologías de agentes genéricas que maximicen la inteoperatividad dentro y a través de aplicaciones basadas en agentes (FIPA 1998). El National Industrial Information Infrastructure Protocols (NIIP) es un consorcio de compañías de los Estados Unidos de America formado para desarrollar protocolos software de industria abiertos que hagan posible, para productores y suministradores, interoperar efectivamente como si fuesen parte de la misma empresa (NIIP 1994).

6. Referencias

- Austin, J.L. (1962). How to Do Things With Words. Oxford University Press. Oxford.
- Baker, A.D. (1991). Manufacturing Control with a Market-Driven Contract Net. PhD thesis, Rensselaer Polytechnic Institute, NY. (<http://www.ececs.uc.edu/~abaker/>)
- Baker, A.D., Parunak, H.V.D. and Erol, K. (1997). Manufacturing over the Internet and into Your Living Room: Perspectives from the AARIA Project. Working Paper, Department of Electrical & Computer Engineering and Computer Science, University of Cincinnati. (<http://www.ececs.uc.edu/~abaker/>)
- Balasubramanian, S., Maturana, F. and Norrie, D. (1996). Multi-agent planning and coordination for distributed concurrent engineering. International Journal of Cooperative Information Systems, 5(2-3): 153-179.
- Barbat, B. and Zampfirescu, C. B. (1999). A3ckm: Anthropocentric Agent Architectures for Complex Knowledge Management. Acta Universitatis Cibiniensis. XXXVIII, pp 23-28.
- Barber, S., White, E., Goel, A., Han, D., Kim, J., Li, H., Liu, T.H., Martin, C.E. and McKay, R. (1998). Sensible Agent Problem-Solving Simulation for Manufacturing Environments. In Proceedings of AI & Manufacturing Research Planning Workshop, Albuquerque, NM, The AAAI Press, pp. 1-9. (<http://www.lips.utexas.edu/>)
- Barbuceanu, M. and Fox, M. (1995a). The Architecture of an Agent Based Infrastructure for Agile Manufacturing. In Proceedings of IJCAI'95, Montreal, Canada. (<http://www.ie.utoronto.ca/EIL/ABS-page/ABSintro.html>)
- Barbuceanu, M. and Fox, M. (1995b). COOL: A Language for Describing Coordination in Multi Agent Systems. In Proceedings of ICMAS'95, San Francisco, CA, The AAAI press/The MIT Press, pp17-24. (<http://www.ie.utoronto.ca/EIL/ABS-page/ABS-intro.html>)

Barbuceanu, M. and Fox, M. (1996). Capturing and Modeling Coordination Knowledge for Multi-Agent Systems. *International Journal of Cooperative Information Systems*, 5(2-3). (<http://www.ie.utoronto.ca/EIL/ABS-page/ABS-intro.html>)

Barbuceanu, M. and Fox, M. (1997). Integrating Communicative Action, Conversations and Decision Theory to Coordinate Agents. In *Proceedings of Autonomous Agents'97*, Marina del Rey, CA. (<http://www.ie.utoronto.ca/EIL/ABS-page/ABS-intro.html>)

Berry, N.M. and Kumura S. (1998). Evaluating the Design and Development of Reagere. In *Working Notes of the Agent-Based Manufacturing Workshop*, Minneapolis, MN. (<http://www.ie.psu.edu/people/faculty/kumara.htm>)

Biswas, G., Bagchi, S. and Saad, A. (1995). Holonic Planning and Scheduling for Assembly Tasks. TR CIS-95-01, Center for Intelligent Systems, Vanderbilt University. (A complementary reference to Saad et al 1995.) (<http://shogun.vuse.vanderbilt.edu/CIS/IMS/index.html>)

Bjork, B. and Wix, J. (1991). An Introduction to STEP. Technical Report, VTT Technical Research Centre of Finland and Wix McLelland Ltd., England. (<http://www.nist.gov/sc4/www/stepdocs.htm>)

Brazier, F., Jonker, C., Treur, J. and Wijngaards, N. (1998). Compositional Design of Generic Design Agent. In *Proceedings of AI & Manufacturing Research Planning Workshop*, Albuquerque, NM, The AAAI Press, pp. 30-39. (<http://www.cs.vu.nl/vakgroepen/ai/projects/desire/>)

Brazier, F.M.T. and Treur J.J. (1996). Modelling Project Coordination in a Multi-Agent Framework. In *Proceedings of WET ICE'96*, Stanford, CA.

Brennan, R.W., Balasubramanian, S., and Norrie, D.H. (1997). Dynamic Control Architecture for Advanced Manufacturing Systems. In *Proceedings of International Conference on Intelligent Systems for Advanced Manufacturing*, Pittsburgh, PA. (<http://imsg.enme.ucalgary.ca/>)

Brennan, R. W. and Norrie, D. H. (2001). Agents, Holons and Functions Blocks: Distributed Intelligent Control in Manufacturing. *Journal of Applied Systems Studies Special Issue on Industrial Applications of Multi-Agent and Holonic Systems*. 2(1), pp 1-19.

Brown, D., Dunskus, B., Grecu, D. and Berker, I. (1995). SINE: Support for Single Function Agents. In *Proceedings of AIENG'95*, Applications of AI in Engineering, Udine, Italy.

Brückner, S, Wyns, J, Peeters, P, Kollingbaum, M. (1998). Designing Agents for the Manufacturing Process Control. In *Proceedings of Artificial Intelligence and Manufacturing Research Planning Workshop - State of the Art & State of the Practice*, AAAI Press, Albuquerque, New Mexico, pp. 4046. (<http://www.mech.kuleuven.ac.be/pma/project/mascada/welcome.html>)

Brugali, D., Menga, G. and Galarraga, S. (1998). Inter-Company Supply Chains Integration via Mobile Agents. In *Proceedings of PROLAMAT'98*, Trento, Italy. (<http://www.polito.it:80/~brugali/>)

Brussel, V., Wyns, J., Valckenaers, P., Bongaerts, L. and Peeters, P. (1998). Reference Architecture for Holonic Manufacturing Systems: PROSA. *Computers In Industry*, 37, pp 255-274.

- Budenske, J., Ahamad, A. and Chartier, E. (1998). Agent-Based Architecture for Exchanging Modeling Data between Applications. In Working Notes of the Agent-Based Manufacturing Workshop, Minneapolis, MN. (<http://www.atcorp.com/>)
- Burke, P. and Prosser, P. (1994). The Distributed Asynchronous Scheduler. Intelligent Scheduling, Zweben, M. and Fox, M.S., eds., Morgan Kaufman Publishers, San Francisco, CA, pp. 309-339. (<http://www.cs.strath.ac.uk/biography/pat/>)
- Bussmann S. (1998). An Agent-Oriented Architecture for Holonic Manufacturing Control. In Proceedings of First International Workshop on IMS, Lausanne, Switzerland, pp. 1-12. (<http://www.daimler-benz.com/>)
- Bussmann, S. (1996). A Multi-Agent Approach to Dynamic, Adaptive Scheduling of Material Flow. In J.W. Perram, J.-P. Müller (eds.): Distributed Software Agents and Applications (MAAMAW'94), LNAI 1069, Springer-Verlag, pp 191-205. 1996.
- Butler, J. and Ohtsubo, H. (1992). ADDYMS: Architecture for Distributed Dynamic Manufacturing Scheduling. Artificial Intelligence Applications in Manufacturing, Famili, A., Nau, D.S., and Kim, S.H., (eds.), The AAAI Press, pp. 199-214.
- Choi, H.S. and Park, K.H. (1997). Shop-floor scheduling at shipbuilding yards using the multiple intelligent agent system. Journal of Intelligent Manufacturing, 8(6), 505-515.
- Christensen, J.H. (1994). Holonic manufacturing systems: initial architecture and standards directions. In Proceedings of First European Conference on Holonic Manufacturing Systems, Hanover, Germany. (<http://www.automation.rockwell.com/>)
- Christensen, J.H., Struger, O.J., Norrie, D. and Schaeffer, C. (1994). Material Handling Requirements in Holonic Manufacturing Systems. In Proceedings of the 1994 International Material Handling Research Colloquium, Grand Rapids, MI, The Material Handling Industry of America, 22 pp. (<http://www.automation.rockwell.com/>)
- Cutkosky, M.R., Engelmores, R.S., Fikes, R.E., Genesereth, M.R., Gruber, T.R., Mark, W.S., Tenenbaum, J.M. and Weber, J.C. (1993). PACT: An Experiment in Integrating Concurrent Engineering Systems. IEEE Computer, 26(1), 28-37. (<http://cdr.stanford.edu/PACE/>)
- Cutkosky, M.R., Tenenbaum, J.M. and Glicksman J. (1996). Madefast: Collaborative Engineering over the Internet. Communication of the ACM, 39(9), 78-87. (<http://cdr.stanford.edu/html/MADEFAST/home.html>)
- Davis, J. and Oliff, M. (1988). Requirements for the Integration of Manufacturing Planning Islands Using Knowledge Based Technology. Expert Systems and Intelligent Manufacturing, Oliff, M. (editor), North-Holland, pp. 25-42.
- Decker, K. (1995). Environment centered analysis and design of coordination mechanisms. Ph.D. Thesis, Dept. of Computer Science, University of Massachusetts, Amherst.
- Decker, K., Sycara, K. and Williamson, M. (1997). Cloning for Intelligent Adaptive Information Agents. Multi-Agent Systems: Methodologies and Applications, Lecture Notes in Artificial Intelligence 1286, Zhang, C. and Lukose, D., (eds.), Springer, pp. 63-75.
- Deen, S.M. (1994). A cooperation framework for holonic interactions in manufacturing. In Proceedings of the Second International Working Conference on Cooperating Knowledge Based Systems (CKBS'94), DAKE Centre, Keele University. (<http://www.keele.ac.uk/depts/cs/Research/Dake/home.html>)

Duffie, N.A. and Piper, R.S. (1986). Non-Hierarchical Control of Manufacturing Systems. Journal of Manufacturing Systems, 5(2), 137-139. (http://www.engr.wisc.edu/me/faculty/duffie_neil.html)

Duffie, N.A. and Prabhu, V.V. (1994). Real-time distributed scheduling of heterarchical manufacturing systems. Journal of Manufacturing Systems, 13(2), 94-107. (http://www.engr.wisc.edu/me/faculty/duffie_neil.html)

Finin, T., Fritzson, R., McKay, D. and McEntire, R. (1993). KQML - A Language and Protocol for Knowledge and Information Exchange. Tech. Report, University of Maryland, Baltimore. (<http://www.cs.umbc.edu/kqml/>)

Finin, T., Fritzson, R., McKay, D. and McEntire, R. (1994). KQML as an Agent Communication Language. Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94). ACM Press. Gaithersburg, MD, USA. N. Adam and B. Bhargava and Y. Yesha, pp 456-463.

FIPA. (1998). Foundation for Intelligent Physical Agents. <http://drogo.cselst.stet.it/fipa/>

FIPA (2000a): FIPA Communicative Act Library Specification. <http://www.fipa.org/specs/fipa00037/>.

FIPA (2000b): FIPA SL Content Language Specification. <http://www.fipa.org/specs/fipa00008/>.

Fischer, K. (1994). The Design of an Intelligent Manufacturing System. In Proceedings of the 2nd International Working Conference on Cooperating Knowledge-based Systems, University of Keele, pp. 83-99. (<http://www.dfki.uni-sb.de/mas/>)

Fischer, K., Muller J.P, Pischel, M. and Schier, D. (1995). A Model for Cooperative Transportation Scheduling. In Proceedings of ICMAS'95, AAAI Press/MIT Press, San Francisco, CA, pp. 109-116. (<http://www.dfki.uni-sb.de/mas/>)

Fischer, K. (1998). An Agent-Based Approach to Holonic Manufacturing Systems. In L. M. Camarinha-Matos, H. Afsarmanesh and v. Marik (Eds.) Intelligent Systems for Manufacturing. Multi-Agent Systems and Virtual Organisations. Kluwer Academic Publishers, pp 3-12.

Fletcher, M., Garcia-Herreros, E., Chritensen, J.H., Deen, S.M. and Mittmann, R. (2000). An Open Architecture for Holonic Cooperation and Autonomy. Proceeding of HoloMAS'2000. IEEE Computer Society

Fletcher, M. and Deen, M. S. (2001). Fault-tolerant holonic manufacturing systems. Concurrency and Computation: Practice and Experience. 13(1), pp 43-70.

Fleury, G., Goujon, J-Y., Gourgand, M. and Lacomme, P. (1996). Multi-Agent Approach for Manufacturing Systems Optimization. In Proceedings of PAAM'96, London, pp. 225-244.

Forbus, K.D. (1988). Intelligent computer-aided engineering. AI magazine, 9(3):23-36.

Fordyce, K. and Sullivan, G.G. (1994). Logistics Management System (LMS): Integrating Decision Technologies for Dispatch Scheduling in Semiconductor Manufacturing. Intelligent Scheduling, Zweben, M. and Fox, M.S., eds., Morgan Kaufman Publishers, San Francisco, CA, pp. 473-516.

Fox, M.S., Barbuceanu, M., and Gruninger, M., (1996). An Organization Ontology for Enterprise Modelling: Preliminary Concepts for Linking Structure and Behaviour. *Computers in Industry*, 29:123-134.

Fox, M.S., Chionglo, J.F., and Barbuceanu, M., (1993), The Integrated Supply Chain Management System. Internal Report, Dept. of Industrial Engineering, Univ. of Toronto. (<http://www.ie.utoronto.ca/EIL/iscm-descr.html>)

Franklin, S. and Graesser, A. (1996). It is an agent, or just a Program?: A Taxonomy for Autonomous Agents. *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*. Springer-Verlag.

Gaines, B.R., Norrie, D.H., and Lapsley, A.Z. (1995). Mediator: an Intelligent Information System Supporting the Virtual Manufacturing Enterprise. In *Proceedings of 1995 IEEE International Conference on Systems, Man and Cybernetics*, New York, pp. 964-969. (<http://imsg.enme.ucalgary.ca/>)

Galliers, J. R. (1998). A Theoretical Framework for Computer Models of Cooperative Dialogue, Acknowledging Multi-Agent Conflict. Open University

General Magic (1997). *Odyssey Information*.
<http://www.genmagic.com/technology/odyssey.html>

Genesereth, M. and Ketchpel, S. (1994) Software agents. *Communications of the ACM*, 37(7), 48-53. (<http://logic.stanford.edu/people/genesereth/>)

Genesereth, M.R., Fikes, R.E. (1992): Knowledge Interchange Format, Version 3.0, Reference Manual, Computer Science Department, Stanford University, Technical Report Logic-92-1. (<http://www.cs.umbc.edu/agents/kse/kif/>)

Gensym. (1997). Agent Development Environment (ADE): Overview. Project Report, NCMS. (<http://www.gensym.com/>)

Goldmann, S. (1996). Procura: A Project Management Model of Concurrent Planning and Design. In *Proceedings of WET ICE'96*, Stanford, CA.

Goldsmith, S.Y. and Interrante, L.D. (1998). An Autonomous Manufacturing Collective for Job Shop Scheduling. In *Proceedings of AI & Manufacturing Research Planning Workshop*, Albuquerque, NM, The AAAI Press, pp. 69-74. (<http://www.sandia.gov/>)

Gray, P.M.D., Embury, S.M., Hui, K. and Price, A. (1998). An Agent-Based System for Handling Distributed Design Constraints. In *Working Notes of the Agent-Based Manufacturing Workshop*, Minneapolis, MN (<http://www.csd.abdn.ac.uk/~pgray/>)
(<http://www.csd.abdn.ac.uk/~apreece/Research/KRAFT.html>)

Gruber, T. (1993). A Translation Approach to Portable Ontology Specification. *Knowledge Acquisition*, 5(2), pp. 199-220.

Hasegawa, T., Gou, L., Tamura, S., Luh, P.B., and Oblak, J.M. (1994). Holonic Planning and Scheduling Architecture for Manufacturing. In *Proceedings of the 2nd International Working Conference on Cooperating Knowledge-based Systems*, University of Keele. (<http://www.toshiba.co.jp/>)

Hayashi, H. (1993). The IMS International Collaborative Program. In *Proceedings of 24th ISIR*, Japan Industrial Robot Association. (<http://www.ims.org/>)

Hayes, C.C. (1998). MAPP: an Agent Organization for Process Planning. In Working Notes of the Agent-Based Manufacturing Workshop, Minneapolis, MN. (<http://www.me.umn.edu/>)

Huhns, M.N. and Singh, M.P.". (1998). Readings in Agents. Morgan Kaufman Publ

IBM. (1998). Aglets Software Development Kit. IBM, <http://www.trl.ibm.com/aglets/>

HMS. (1994). Press Release. HMS Requirements. HMS Server. <http://hms.ifw.uni-hannover.de/>.

IEC Technical Committee, (1997). Function Blocks for Industrial-Process Management and Control Systems, Part-1: Architecture. IEC-TC65/WG6 Committee Draft.

Interrante, L. and Glodsmith, S. (1998). Emergent Agent-Based Scheduling of Manufacturing Systems. In Working Notes of the Agent-Based Manufacturing Workshop, Minneapolis, MN. (<http://www.sandia.gov/>)

Jenkin, M.R.M., Milios, E. and Wilkes, D. (1996). A taxonomy for multi-agent robotics. Autonomous Robotics, 3(4), 375-397.

Jennings, N.R. and Wooldridge, M.J. (1998). Applications of Intelligent Agents. Agent Technology: Foundations, Applications, and Markets. Jennings, N.R. and Wooldridge, M.J (Eds.), Springer, pp. 3-28. (<http://www.elec.qmw.ac.uk/dai/>)

Jennings, N.R., Corera, J.M. and Laresgoiti, I. (1995). Developing Industrial Multi-Agent Systems. In Proceedings of ICMAS'95, San Francisco, The AAAI press/The MIT press, pp. 423-430. (<http://www.elec.qmw.ac.uk/dai/>)

Jha, K.N., Morris, A. Mytych, E. and Spering, J. (1998). MADEsmart: Agents for Design, Analysis, and Manufacturability. In Working Notes of the Agent-Based Manufacturing Workshop, Minneapolis, MN. (<http://www.boeing.com/>)

Koestler, A. (1967). The Ghost in the Machine. Arkana Books, London, UK.

Kouiss, K., Pierreval, H., and Mebarki, N. (1997). Using multi-agent architecture in FMS for dynamic scheduling. Journal of Intelligent Manufacturing, 8(1), 41-47.

Finin, T., Fritzson, R., McKay, D. and McEntire, R.. (1994). KQML as an Agent Communication Language. Proceedings of the 3rd International Conference on Information and Knowledge Management {CIKM}'94. ACM Press. Gaithersburg, Maryland, pp 456--463

Kwok, A.D. and Norrie, D.H. (1994). A Development System for Intelligent Agent Manufacturing Software. Integrated Manufacturing Systems, 5(4-5), 64-76. (<http://imsg.enme.ucalgary.ca/>)

Lenat, D.B. and Guha, R.V. (1989). Building Very Large Knowledge-Based Systems: Representation and Inference in the Cyc Project. Addison-Wesley, Reading, MA, USA.

Lin, G.Y.-J. and Solberg, J.J. (1992). Integrated Shop Floor Control Using Autonomous Agents. IIE Transactions: Design and Manufacturing, 24(3), 57-71. (<http://IE.www.ecn.purdue.edu/IE/>)

- Liu, J. and Sycara, K.P. (1994). Distributed problem solving through coordination in a society of agents. In Proceedings of the 13th International Workshop on DAI. (<http://www.cs.cmu.edu/afs/cs/user/katia/www/katia-home.html>)
- Liu, J. and Sycara, K.P. (1995). Exploiting Problem Structure for Distributed Constraint Optimization. In Proceedings of ICMAS95, San Francisco, CA, pp. 246-253. (<http://www.cs.cmu.edu/afs/cs/user/katia/www/katia-home.html>)
- M. Klein. iDCSS: Integrating Workflow, Conflict and Rationale-based Concurrent Engineering Coordination Technologies. Concurrent Engineering: Research and Application, 3(1), 21-27, 1995. (<http://ccs.mit.edu/>)
- Malkoun, M.T. and Kendall, E.A. (1997). CLAIMS: Cooperative Layered Agents for Integrating Manufacturing Systems. In Proceedings of PAAM'97, London. (<http://www.cse.rmit.edu.au/~rdsek/>)
- Martin, D.L., Cheyer, A.J. and Moran, D.B. (1998). Building Distributed Software Systems with the Open Agent Architecture. In Proceedings of PAAM'98, London. (<http://www.ai.sri.com/~oaa/>)
- Maturana, F. and Norrie, D. (1996). Multi-Agent Mediator Architecture for Distributed manufacturing. Journal of Intelligent Manufacturing, 7, 257-270. (<http://imsg.enme.ucalgary.ca/>)
- Maturana, F., Balasubramanian, S. and Norrie, D.H. (1996). A Multi-Agent Approach to Integrated Planning and Scheduling for Concurrent Engineering. In Proceedings of the International Conference on Concurrent Engineering: Research and Applications, Toronto, pp. 272-279. (<http://imsg.enme.ucalgary.ca/>)
- Maturana, F.P. and Norrie, D.H. (1997). Distributed decision-making using the contract net within a mediator architecture. Decision Support Systems 20, pp 53-64.
- McEleney, B., O'Hare, G.M.P. and Sampson, J. (1998). An Agent Based System for Reducing Changeover Delays in a Job-Shop Factory Environment. In Proc. of PAAM'98, London. (<http://www.co.umist.ac.uk/>)
- McGuire, J., Huokka, D., Weber, J., Tenenbaum, J., Gruber, T., and Olsen, G. (1993). SHADE: Technology for Knowledge-Based Collaborative Engineering. Journal of Concurrent Engineering: Applications and Research, 1(3).
- Mehra, A. and Nissen, M. (1998). Intelligent Supply Chain Agents using ADE. In Proceedings of AI & Manufacturing Research Planning Workshop, Albuquerque, NM, The AAAI Press, pp. 112-119. (<http://www.gensym.com/>)
- Miyashita, K. (1998). CAMPS: a constraint-based architecture for multi-agent planning and scheduling. Journal of Intelligent Manufacturing, 9(2), 147-154.
- Molina, A., Al-Ashaab, A.H., Ellis, T.I.A., Young, R.I.M., and Bell, R. (1995). A Review of Computer-Aided Simultaneous Engineering Systems. Research in Engineering Design, 7(1), 38-63.
- Müller, J. P. (1996). The design of intelligent agents: a layered approach. Volume 1177 of Lecture notes in artificial intelligence. Springer.

Murthy, S., Akkiraju, R., Rachlin, J. and Wu, F. (1997). Agent-Based Cooperative Scheduling. In Proceedings of AAAI Workshop on Constraints and Agents, AAAI Press, pp. 112-117.

NCMS. (1998). Shop Floor Agents. NCMS. (<http://www.ncms.org/3portfolio/1ProjectPortfolio/TechProjPort.htm>- Search by 'Agent')

NIIP. (1994). About the NIIP Consortium. <http://www.niip.org/about-NIIP-text.html>

NIST (1998) Advanced Technology Program. <http://www.atp.nist.gov/>

Nwana, H. S. (1996). Software Agents: An Overview. Intelligent Systems Research. AA&T, BT Laboratories.

Nwana, H.S., Ndumu, D.T. and Lee, L.C. (1998). ZEUS: An Advanced Tool-Kit for Engineering Distributed Multi-Agent Systems. In Proceedings of PAAM'98, London. (<http://www.labs.bt.com/projects/agents/>)

ObjectSpace. (1997). Agent-Enhanced Manufacturing System Initiative: Project Brief. ObjectSpace, <http://www.atp.nist.gov/www/comps/briefs/97050018.htm>.

O'Hare, G.P.M and Jennings, N.R. (1996). Foundation of Distributed Artificial Intelligence. John Wiley & Sons

OMG. (1998). OMG Formal Documentation. <http://www.omg.org/corba/>

Ouelhadj, D., Hanachi, C. and Bouzouia, B. (1998). Multi-Agent System for Dynamic Scheduling and Control in Manufacturing Cells. In Working Notes of the Agent-Based Manufacturing Workshop, Minneapolis, MN. and error handling.)

Overgaard, L., Petersen, H. G. and Perram, J. W. (1994). Motion Planning for an Articulated Robot: A Multi-Agent Approach. In Proceedings of Modelling Autonomous Agent in a Multi-Agent World, Odense University, pp. 171-182. (<http://www.imada.ou.dk/>) (<http://www.amrose.spo.dk/>)

Ow, P.S. and Smith, S.F. (1988). A Cooperative Scheduling System. In Proceedings of the 2nd International Conference on Expert Systems and the Leading Edge in Production Planning and Control, pp. 43-56.

Pan, J.Y.C. and Tenenbaum, M.J. (1991). An intelligent agent framework for enterprise integration. IEEE Transactions on Systems, Man, and Cybernetics, 21(6), 1391-1408.

Pancerella, C., Hazelton, A. and Frost, R. (1995). An autonomous agent for on-machine acceptance of machined components. In Proceedings of SPIE International Symposium on Intelligent Systems and Advanced Manufacturing, Philadelphia, PA.

Papaoannou, T. and Edwards, J. (1998). Mobile Agent technology Enabling the Virtual Enterprise: a Pattern for Database Query. In Working Notes of the Agent-Based Manufacturing Workshop, Minneapolis, MN. (<http://msiri.lboro.ac.uk/>)

Park, H., Cutkosky, M., Conru, A. and Lee, S.H. (1994). An Agent-Based Approach to Concurrent Cable Harness Design, AIEDAM, 8(1). (<http://cdr.stanford.edu/FirstLink/FirstLink.html>)

Park, H., Tenenbaum, J. and Dove, R. (1993). Agile Infrastructure for Manufacturing Systems (AIMS): A Vision for Transforming the US Manufacturing Base. Defense Manufacturing Conference.

Parunak, H.V.D., Baker, A.D. and Clark, S.J. (1997a). The AARIA Agent Architecture: An Example of Requirements-Driven Agent-Based System Design. In Proceedings of the First International Conference on Autonomous Agents, Marina del Rey, CA.

Parunak, H.V.D., Irish, B.W., Kindrick, J. and Lozo, P.W. (1985). Fractal Actors for Distributed Manufacturing Control. Proc. Of 2nd Conference on AI Applications (CAIA'85). pp 653-660.

Parunak, H.V.D., Ward, A., Fleischer, M. and Sauter, J. (1997b). A Marketplace of Design Agents for Distributed Concurrent Set-Based Design. In Proceedings of the Fourth ISPE International Conference on Concurrent Engineering: Research and Applications, Troy, Michigan.

Parunak, V.D. (1987). Manufacturing Experience with the Contract Net. Distributed Artificial Intelligence, Huhns, M.N. ed., Pitman, pp. 285-310. (<http://www.erim.org/~van/>)

Parunak, V.D. (1993). MASCOT: A virtual factory for research and development in manufacturing scheduling and control. Tech. Memo 93-02, Industrial Technology Institute. (<http://www.erim.org/~van/>)

Parunak, V.D. (1996). Workshop Report: Implementing Manufacturing Agents. Sponsored by the SFA project of NCMS in conjunction with PAAM'96. NCMS. (A report on a related workshop held at PAAM'96, with summary of a number of related papers presented at PAAM'96.) (<http://www.erim.org/~van/>)

Parunak, V.D., Baker, A. and Clark, S. (1998). The AARIA Agent Architecture: From Manufacturing Requirements to Agent-Based System Design. In Working Notes of the Agent-Based Manufacturing Workshop, Minneapolis, MN. (<http://www.aaria.uc.edu/>)

Patil, R., Fikes, R., Patel-Schneider, P., McKay, D., Finin, T., Gruber, T. and Neches, R. (1992). The DARPA Knowledge Sharing Effort: Progress Report. Rich, C., Nebel, B., and Swartout, W. (eds.), Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference, Cambridge, MA, Morgan Kaufmann.

Patriti V., Schaefer K., Ramos M., Charpentier P., Martin P. and Veron M. (1997). Multi-agent and manufacturing: A multilevel point of view. In Proceedings of CAPE'97, Detroit. (<http://www.cran.u-nancy.fr/>)

Peng, Y., Finin, T., Labrou, Y., Chu, B., Long, J., Tolone, W.J. and Boughannam, A. (1998) A Multi-Agent System for Enterprise Integration. In Proceedings of PAAM'98, London. (<http://www.cs.umbc.edu/lait/research/ciimplex/>)

Petrie, C., Cutkosky, M., Webster, T., Conru, A. and Park, H. (1994). Next-Link: An Experiment in Coordination of Distributed Agents. Position paper for the AID-94 Workshop on Conflict Resolution, Lausanne. (<http://cdr.stanford.edu/NextLink/NextLink.html>)

Roboam, M. and Fox, M.S. (1992). Enterprise Management Network Architecture. Artificial Intelligence Applications in Manufacturing, Famili, A., Nau, D.S. and Kim, S.H., (eds.), The AAAI Press, pp. 401-432.

Rothermel, K. and Popescu-Zeletin, R. (eds.) (1997). Mobile Agents, Lecture Notes in Computer Science 1219, Springer.

- Russell, S. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice Hall. Englewood Cliffs, New Jersey. ISBN 0131038052.
- Saad, A., Biswas, G., Kawamura, K., Johnson, M.E. and Salama, A. (1995). Evaluation of Contract Net-Based Heterarchical Scheduling for Flexible Manufacturing Systems. In *Proceedings of Intelligent Manufacturing Workshop at IJCAI'95*, Montreal, pp. 310-321. (<http://shogun.vuse.vanderbilt.edu/CIS/IMS/index.html>)
- Sadeh, N. and Fox, M.S. (1989). CORTES: An Exploration into Micro-Opportunistic Job-Shop Scheduling. In *Proceedings of Workshop on Manufacturing Production Scheduling at IJCAI-89*, Detroit. (<http://agile.cimds.ri.cmu.edu/icll/index.html>)
- Sandholm, T. and Lesser, V. (1996). Advantages of a Leveled Commitment Contracting Protocol. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, Portland, OR, pp. 126-133. (<http://dis.cs.umass.edu/>) (<http://siesta.cs.wustl.edu/~sandholm/>)
- Scalabrin, E. (1996). *Conception et Réalisation d'environnement de développement de systèmes d'agents cognitifs*. PhD Thesis, Université de Technologie de Compiègne, France. (<http://www.hds.utc.fr/~barthes/JPB-rech.html>)
- Schaefer, K., Patrity, V., Charpentier, P. Martin, P. and Spath, D. (1996). The Multi-Agent Approach in Scheduling and Control of Manufacturing Systems. In *Proceedings of PAAM'96*, London. (Initial proposal and research work related to that presented in (Patrity et al 1997).) (<http://www.cran.u-nancy.fr/>)
- Searle, J. (1969). *Speech Acts - An Essay in the Philosophy of Language*. Cambridge University Press.
- Sen, S. and Weiss, G. (1999). Learning in multi-agent systems. In G. Weiss, editor, *Multi-Agent Systems: A modern Approach to Distributed AI*. MIT Press. Cambridge, MA, pp 259-298.
- Shaw, M.J. (1988). Dynamic Scheduling in Cellular Manufacturing Systems: A Framework for Networked Decision Making. *Journal of Manufacturing Systems*, 7(2), 83-94.
- Shaw, M.J. and Whinston, A.B. (1983). *Distributed Planning in Cellular Flexible Manufacturing Systems*. Tech. Report, Management Information Research Center, Purdue University.
- Shen, W. and Barthès, J.P. (1995). DIDE: A Multi-Agent Environment for Engineering Design. In *Proceedings of the First International Conference on Multi-Agent Systems*, San Francisco, USA, The AAAI press/The MIT press, pp. 344-351.
- Shen, W. and Barthès, J.P. (1996a). An Experimental Multi-Agent Environment for Engineering Design. *International Journal of Cooperative Information Systems*, 5(2-3), 131-151. (<http://www.hds.utc.fr/~barthes/JPB-rech.html>)
- Shen, W. and Barthès, J.P. (1996b). Computer Supported Cooperative Environments for Engineering Design: A Survey. Tech. Report 96-122, CNRS UMR Heudiasyc, Université de Technologie de Compiègne, France. (<http://imsg.enme.ualgary.ca/CIG/papers/csceed.pdf>)
- Shen, W. and Norrie, D.H. (1998) An Agent-Based Approach for Dynamic Manufacturing Scheduling. In *Working Notes of the Agent-Based Manufacturing Workshop*, Minneapolis, MN. (<http://ksi.cpsc.ualgary.ca/KSI/KSI.html>) (<http://imsg.enme.ualgary.ca/>)

Shen, W., Maturana, F. and Norrie, D.H. (1998b). Learning in Agent-Based Manufacturing Systems. In Proceedings of AI & Manufacturing Research Planning Workshop, Albuquerque, NM, The AAAI Press, pp. 177-183. (<http://imsg.enme.ucalgary.ca/>)

Shen, W., Xue, D., and Norrie, D.H. (1998a). An Agent-Based Manufacturing Enterprise Infrastructure for Distributed Integrated Intelligent Manufacturing Systems. In Proceedings of PAAM'98, London, UK. (<http://imsg.enme.ucalgary.ca/>)

Smith, R.G. (1980). The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. IEEE Transactions on Computers, C-29(12), 1104-1113.

Sousa, P. and Ramos, C. (1997). A Dynamic Scheduling Holon for Manufacturing Orders. Journal of Intelligent manufacturing, 9(2), 107-112. (<http://www.ipp.pt/ipp/>)

Sprumont, F. and Muller, J.P. (1996). AMACOIA: A Multi-Agent System for Designing Flexible Assembly Lines. In Proceedings of PAAM'96, London, UK, pp. 573-585. (<http://iun.unine.ch/Research/IA/projects/amacoia/amacoia.html>)

Stanford. (1997). JATLite. CDR, Stanford University, <http://java.stanford.edu/>

Swaminathan, J.M., Smith, S.F. and Sadeh, N.M. (1996). A Multi-Agent Framework for Supply Chain Dynamics. In Proceedings of NSF Research Planning Workshop on AI & Manufacturing, Albuquerque, NM. (<http://www.cs.cmu.edu/afs/cs.cmu.edu/project/ozone/www/supply-chain/supply-chain.html>)

Sycara, K.P., Roth, S.F., Sadeh, N. and Fox, M.S. (1991). Resource Allocation in Distributed Factory Scheduling. Intelligent Scheduling, Zweben, M. and Fox, M.S., eds., Morgan Kaufman Publishers, San Francisco, CA, pp. 29-40 (<http://www.cs.cmu.edu/afs/cs/user/katia/www/katia-home.html>)

Tseng, M.M., Lei, M., and Su, C. (1997). A collaborative control system for mass customization manufacturing. Annals CIRP, 46(1), 373-376. (<http://iesu5.ust.hk/research/gmrg/member/tseng/>)

Van Leeuwen, E.H. and Norrie, D.H. (1997). Intelligent manufacturing: holons and holarchies. Manufacturing Engineer, 76(2), 86-88. (<http://imsg.enme.ucalgary.ca/>)

Wang L., Balasubramanian S. and Norrie D. (1998). Agent-based Intelligent Control System Design for Real-time Distributed Manufacturing Environments. In Working Notes of the Agent-Based Manufacturing Workshop, Minneapolis, MN. (<http://imsg.enme.ucalgary.ca/>)

Wooldridge, M. and Jennings, N. (1995). Intelligent Agents: Theory and Practice. Knowledge Engineering Review, 10(2), 115-152. (<http://www.elec.qmw.ac.uk/dai/>)

Wooldridge, M. (2002). An Introduction to MultiAgent Systems. John Wiley & Sons (Chichester, England). ISBN 0 47149691X.

Wunderli, M., Norrie, M.C., and Schaad, W. (1996). Multidatabase agents for CIM systems. International Journal Computer Integrated manufacturing, 9(4), 293-298.

W3C. (2002). XML. World-Wide Web Consortium XML. <http://www.w3.org/>. World-Wide Web Consortium XML. <http://www.w3.org/>.

Yan, Y., Kuphal, T. and Bode, J. (1998). Application of Multi-Agent Systems in Project Management. In Working Notes of the Agent-Based Manufacturing Workshop, Minneapolis, MN. (<http://www.uni-leipzig.de/wifa/>)

Zweben, M. and Fox, M.S. (1994). Intelligent Scheduling. Morgan Kaufman Publishers, San Francisco, CA.