

## Ontología Unificada para un Sistema Holónico de Manufactura

I. Simón-Marmolejo<sup>a,d,\*</sup>, Omar López-Ortega<sup>b</sup>, L. E. Ramos-Velasco<sup>c</sup>, M. Ortiz-Domínguez<sup>d</sup>

<sup>a</sup>Centro de Tecnología Avanzada, CIATEQ, Av. Diesel Nacional # 1 Ciudad Industrial Sahagún, CP. 43990, Cd. Sahagún, Hidalgo, México.

<sup>b</sup>AACyE, Universidad Autónoma del Estado de Hidalgo, Carretera Pachuca-Tulancingo, Km 4.5, CP. 42186, Mineral de la Reforma, Hidalgo, México.

<sup>c</sup>PEIA, Universidad Politécnica Metropolitana de Hidalgo, Boulevard Acceso a Tolcayuca 1009, Ex Hacienda San Javier, CP. 43860, Hidalgo, México.

<sup>d</sup>ESSAH, Universidad Autónoma del Estado de Hidalgo, Carretera Cd. Sahagún-Otumba s/n. Zona Industrial, CP. 43990, Tepeapulco, Hidalgo, México.

---

### Resumen

Los sistemas holónicos de manufactura son integrados por holones capaces de comportarse de una manera autónoma, cooperativa, auto-organizada y reconfigurable para adoptar estructuras distintas en condiciones de operación normales y de emergencia. Dichos holones cuentan con: (1) una representación del conocimiento, (2) una unidad de control distribuido y descentralizado, y (3) un módulo de coordinación. El objeto de interés de la presente investigación es la concepción de una ontología unificada en el dominio de manufactura, que garantice los requisitos en el formalismo del modelo de conocimiento de un sistema holónico. A diferencia de los modelos ontológicos encontrados en la literatura, el esquema de representación del conocimiento propuesto integra roles y comportamientos, mismos que son validados mediante un caso de estudio de una celda de manufactura de un laboratorio universitario. Los resultados muestran que al hacer uso de un vocabulario común, es posible representar coherentemente el conocimiento para que toda clase de holones en una holarquía puedan intercambiar, compartir y recuperar información.

#### Palabras Clave:

Sistemas y células de fabricación flexible e inteligente, técnicas de control inteligente, agentes, arquitecturas de control distribuido y descentralizado, sistemas de bases de datos.

### Unified Ontology for a Holonic Manufacturing System

#### Abstract

Holonic manufacturing systems are formed by holons that are capable of behaving in an autonomous, cooperative, self-organized and reconfigurable way to adopt different structures under normal and emergency operating conditions. These holons possess: (1) a representation of the world in which they live, (2) a distributed and decentralized control unit, and (3) a coordination module. The object of interest of the present research is the conception of a unified ontology in manufacturing domain, that guarantees the requirements in the formalism of the knowledge model of a holonic system. Unlike the ontological models found in the literature, the proposed knowledge representation scheme integrates roles and behaviors, which are validated through a case study of a manufacturing cell from a university laboratory. The results show that by using a common vocabulary, it is possible to represent knowledge coherently so that all kinds of holons in a holarchy can exchange, share and retrieve information.

#### Keywords:

Flexible and intelligent manufacturing systems and cells, smart control techniques, agents, distributed and decentralized control architectures, database systems.

---

\*Autor para correspondencia: isaiasm@uaeh.edu.mx

## 1. Introducción

Los *Sistemas Holónicos de Manufactura* (Holonics Manufacturing Systems, HMS), que integra toda la gama de actividades de fabricación, desde la recepción de pedidos hasta el diseño, producción y comercialización, son creados y modificados dinámicamente con el fin de adoptar estructuras distintas en condiciones de operación normales y de emergencia (Botti and Giret, 2008). Este tipo de sistemas son integrados por holones capaces de comportarse de una manera autónoma, cooperativa, auto-organizada y reconfigurable (Christensen, 1994). Los holones dotados de las habilidades antes mencionadas, necesitan estar informados del conocimiento cambiante del ambiente que los rodea y precisan de un mecanismo de comunicación para cooperar (Araújo et al., 2015).

Crear un ambiente de comunicación implica que los holones sean capaces de conocer y manejar descripciones del estado del mundo externo, las cuales consisten en una representación dinámica del entorno, y de otros holones con quienes comparte dicho entorno. El modelo correspondiente del *estado del mundo* y el *modelo social* serán considerados como modelos estructurales, que eventualmente pueden ser modelos por tareas o metas globales por alcanzar, e información sobre el grado del logro (Shen et al., 2001).

Al mismo tiempo en los HMS, es importante garantizar la compatibilidad entre los holones (por ejemplo, en problemas relacionados con interfaces y protocolos) para verificar que la semántica contenida es preservada durante el intercambio de mensajes entre los mismos. Así, la interoperabilidad en los HMS incrementa la necesidad de compartir ontologías de manufactura apropiadas (Borgo and Leitão, 2007), que sean acordadas previamente y que se expresen en un lenguaje basado en la lógica, que en consecuencia, permitan hacer distinciones detalladas, precisas, coherentes, sólidas y significativas entre las descripciones de la representación del conocimiento (Luther, 2008).

Las ontologías son necesarias para definir los conceptos esenciales y el vocabulario básico asociado con un dominio particular (Shen et al., 2001). De esta forma, una ontología ofrece la semántica y representación de datos que deriva de información sobre el comportamiento de cada holón para la formación de holarquías, negociación y colaboración (Koppensteiner et al., 2015). El medio semántico es un contexto donde todos los conceptos relevantes son importantes para la organización colaborativa. De esta manera, las ontologías proporcionan un mapeo explícito entre conceptos (Uschold and Gruninger, 1996), haciendo posible que la representación formal del conocimiento facilite su potencial para ser compartido e intercambiado (Coorea da Silva et al., 2002).

La principal contribución de este artículo es la especificación, conceptualización y formalización de una ontología unificada de dominio manufactura, para garantizar los requisitos en el formalismos del modelo de conocimiento y para verificar que la semántica contenida es preservada durante el intercambio de mensajes entre holones. Como segundo propósito, la ontología propuesta es implementada y estudiada haciendo uso de un lenguaje de programación formal en un caso de estudio particular. El objetivo del experimento es garantizar un lenguaje de contenido universal y de fácil acceso para intercambiar, compartir

y recuperar información entre holones capaces de actuar dentro de múltiples holarquías.

Las siguientes secciones del artículo son organizadas como sigue: La Sección 2 presenta el estado del arte de arquitecturas y ontologías de enfoque holónico. En la Sección 3 se da una propuesta de ontología unificada. En la Sección 4 se describe una celda de manufactura como un caso de estudio particular y se ilustran los resultados del uso de la ontología entre holones mediante protocolos de comunicación. La Sección 5 proporciona un análisis de resultados del trabajo realizado, y finalmente, la Sección 6 da las conclusiones y presenta investigaciones futuras.

## 2. Estado del Arte

Los principios (Holón: entidad constitutiva de ser tanto totalidad y parte al mismo tiempo) que dan sustento a los sistemas holónicos se propusieron por primera vez por Koestler (1990). Sin embargo, hoy en día el concepto está siendo empleado en el área de manufactura inteligente, donde se define cada unidad de fabricación (productos, órdenes, recursos, etc.) con diferentes holones que pueden utilizar distintos tipos de conocimiento (Botti and Giret, 2008).

### 2.1. Antecedentes en Arquitecturas de Enfoque Holónico

Una arquitectura holónica es vista como una técnica para diseñar, proyectar y construir HMS que buscan garantizar sistemas de control inteligentes con niveles de rendimiento mínimo en el caso de circunstancias imprevistas (Blanc et al., 2008), (Leitão and Restivo, 2008), (Ounnar and Pujó, 2012), (Tarun et al., 2013). Ejemplos de éstas, que integran y representan una amplia gama de entidades de un sistema de manufactura, el control empleado y la tecnología usada son: Product Resource Order Staff Architecture, "PROSA" (Brussel et al., 1998); Holonic Architecture for Reconfigurable Manufacturing System, "RMS-HA" (Wang et al., 2005); ADActive holonic CONTROL architecture for distributed manufacturing systems, "ADACOR" (Leitão and Restivo, 2006); Manufacturing Execution Systems y conceptos holónicos, "MES-HMS", (Blanc et al., 2008); "Arquitectura de Referencia en Capas" (Bravo et al., 2011); "Enfoque Holónico", (Araújo et al., 2015); "ADACOR<sup>2</sup>" (Barbosa et al., 2015); Estructura de sistemas de múltiples agentes (Multi-Agent Systems, MAS) organizada en dos niveles y complementada con el concepto Radio Frequency Identification-Information Management System, "MAS-DUO & RFID-IMS" (de las Morenas et al., 2015); Holonic Hybrid Control Model, "H2CM" (Indriago et al., 2016); entre otras arquitecturas.

A excepción de ADACOR, las arquitecturas carecen de una ontología de dominio manufactura que integre y clasifique de manera formal y completa los conceptos de un sistema de producción en beneficio del sistema holónico. Por otro lado, se indican las características y capacidades de los holones, pero en la mayoría de los casos no se hace notar la base de conocimientos necesaria para dotar a los holones con las características que los definen. Respecto a la comunicación entre holones, en algunos casos (ADACOR, MES-HMS, Enfoque Holónico y ADACOR<sup>2</sup>) se indica que ésta se hace a través de una red Ethernet, utilizando protocolos y un lenguaje de comunicación

específico. Sin embargo, la mayoría de los autores sólo mencionan el tema sin hacer notar la importancia del mismo en los sistemas holónicos.

## 2.2. Análisis Comparativo de Ontologías para HMS

En esta sección se describen ontologías prominentes en dominio de manufactura. Para el presente estudio, fueron seleccionados 7 proyectos específicos de la literatura. El análisis incluye características generales como: su cobertura en el dominio manufactura, el formalismo en los conceptos usados, enfoques basados en datos lingüísticos y normados, los niveles y jerarquía extendida, si cuenta o no con una evaluación y los alcances en la experimentación o aplicaciones asociadas a HMS/MAS. Se hace hincapié en las características que describen el contenido de la ontología incluyendo: conceptos, organización taxonómica, la representación de las relaciones “parte y todo” respecto a los holones o agentes involucrados, atributos y predicados necesarios para establecer relaciones entre conceptos en los protocolos de comunicación.

Los criterios para la selección de ontologías fueron: (1) obtener un conjunto de ontologías en dominio de manufactura con aplicaciones holónicas, (2) utilizar ontologías que son significativas en tamaño y relativamente bien desarrolladas, y (3) utilizar un conjunto bastante bien documentado de ontologías (por lo menos documentadas lo suficientemente como para poder integrarse a la mayoría de las variables que son analizadas, sin embargo, no todos los datos están disponibles para todos los proyectos en el estudio).

La arquitectura ontológica propuesta por Borgo and Leitão (2007), es presentada en el dominio de manufactura. Su cobertura hace referencia a los conceptos ADACOR acordes con la fundación ontológica DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) para la programación de la producción (especialmente organizados en la producción tipo job shop) y control del entorno. La propuesta reúne aspectos holónicos y de manufactura con bastante aserción: describe diagramas UML (Unified Modelling Language) de las clases propuestas, define los conceptos usados, y da ejemplos de predicados y atributos asociados. Sin embargo, a pesar de hacer una amplia investigación respecto a ontologías, sistemas inteligentes y HMS bajo la propuesta alineada ADACOR-DOLCE, el proyecto carece de cobertura y no es explícito.

Bajo otro enfoque en el trabajo desarrollado por Jules et al. (2015) se detalla una ontología concerniente a una red de manufactura en los términos de la arquitectura PROSA. El documento pone de relieve conocimientos base como conceptos y slots que constituyen a la propuesta, estos últimos asociados a un conjunto de reglas que permite la elección de recursos para la fabricación de un producto. Pese a dichas aportaciones, es evidente observar el uso de términos no estandarizados lo que hace ambigua la comprensión de conceptos.

La referencia Giraldo et al. (2013), hace una conjetura entre elementos de una ontología (conceptos, atributos y relaciones), agrupados en lo relativos a “Esquemas Preconceituales” (EP) y al tema HMS. En este caso, la ontología definida permite detectar conceptos holónicos en una organización e identifica los holones principales que están implícitos en los procesos de producción. Sin embargo, este trabajo se enfoca en listar y explicar

brevemente los elementos de la ontología propuesta, sin describir por completo una ontología de HMS, es decir no hace hincapié respecto a cómo estos conceptos se relacionan con los holones característicos.

Así mismo, también existen documentos donde se describe el proceso mediante el cual se creó una ontología de aplicación para HMS basados en la “Unidad de Producción” (UP) o HMS-UP, como es el caso del trabajo presentado por Checa and Rojas (2014). El proyecto puntualiza las fases necesarias para crear una ontología. Sin embargo, no se definen de manera clara y precisa cada uno de los términos, conceptos y relaciones necesarias para integrar un sistema de manufactura desde el enfoque holónico, no hay evidencia de tecnología computacional en el diseño de la misma y tampoco se muestran resultados o experimentos.

El análisis de las características generales y de las características que describen el contenido de la ontología de los artículos revisados y de algunos más como Uschold et al. (1998), Lemaignan et al. (2006) y Raileanu et al. (2014), se resume en la Tabla 1 clasificados según su contribución: abundante (●), mínima (○) o ninguna ( ), contrastando al final con la propuesta ontológica definida de forma resumida en las siguientes secciones de este documento.

Tabla 1: Evaluación de ontologías.

Contribuciones	Borgo and Leitão (2007)	Checa and Rojas (2014)	Giraldo et al. (2013)	Jules et al. (2015)	Lemaignan et al. (2006)	Raileanu et al. (2014)	Uschold et al. (1998)	Ontología Propuesta
Cobertura en dominio manufactura	●	○	○	○	●	○	●	●
Formalismos de conceptos	●	○	●	○	●	○	○	●
Datos lingüísticos y normados	●				●	●	○	●
Niveles y jerarquía extendida		○	●	○	●			●
Evaluación de ontología				●				○
Evidencia de aplicación en un HMS/MAS	●		○	●	●	●		○
Relaciones taxonómicas			○		●	○		●
Holones/Agentes	●	○	○	●	○	○		●
Atributos	○	○	○			●	○	●
Predicados	○	○	○	○			○	●

En conclusión, aunque los 7 artículos antes citados intentan integrar ontologías como base de información para los holones y en beneficio de los protocolos de comunicación entre los mismos, no existe una comprensión común del dominio manufactura, de modo que se aprecian confusiones conceptuales y terminológicas, por lo que hay necesidades en cuanto a la unificación de términos. Pocos autores hacen hincapié en las relaciones taxonómicas, es decir: cómo se estructura, complementa, relaciona y se define cada concepto dentro de una ontología. No es claro cómo se favorece la comunicación entre holones. No hay evidencia de cómo se beneficia la gestión de contenidos. Un lenguaje de contenido totalmente expresivo debe ser capaz de representar y distinguir entre los distintos tipos de elementos: predicados, términos, primitivas, conceptos, acciones de los holones, expresiones de consulta, variables, etc.

Las ontologías ocupan un lugar destacado dentro de los holones, sin embargo, lo antes mencionado requiere del dominio

global de los términos usados en sistemas de producción, proporcionando un análisis claro, sin ambigüedades y con un diseño simple e inequívoco que permita describir y representar una ontología unificada. Para hacer notorias las ventajas de la ontología propuesta y comparada en la Tabla 1, el siguiente apartado se refiere una ontología unificada como la principal aportación de la investigación, misma que integra roles y comportamientos de holones en un caso de estudio particular.

### 3. La Ontología Unificada Propuesta

La propuesta busca unificar el amplio conjunto de términos estandarizados en los sistemas de manufactura, para tal caso, se parte de una metodología específica hasta alcanzar la formalización deseada.

Para cumplir el objetivo planteado se analizan diversas metodologías: Methontology (Gómez-Pérez et al., 1996), Uschold and King's Ontology (Uschold et al., 1998), Ontology Development 101 (Noy and McGuinness, 2001), entre otras. Cada una de estas propuestas describen las fases del desarrollo de una ontología a diferentes niveles de conocimiento.

El presente proyecto, hace uso de una adaptación de la metodología "Methontology" dada por Corcho et al. (2003). La metodología se subdivide en tres áreas: 1) actividades de desarrollo (fases: especificación, conceptualización, formalización, implementación y mantenimiento) que implican diferentes niveles de conocimiento y que por el orden que siguen son denominadas "Ciclo de vida de la ontología"; 2) actividades de soporte (adquisición del conocimiento, integración, evaluación, documentación y gestión de configuración) tareas que se realizan durante todo el ciclo de vida de la ontología, y; 3) actividades de ejecución: necesarias en programación, control y aseguramiento de la calidad para la ontología, en el caso particular mediante un ambiente virtual.

Las actividades mostradas de manera general en el presente documento son dos de las cinco incluidas en las "actividades de desarrollo"; 1) formalización y 2) implementación. Para mayor información respecto a la metodología usada, consulte los trabajos de Corcho et al. (2003) y Fernández et al. (1997).

#### 3.1. Formalización de Conceptos

La formalización de una ontología es la fase donde se representa la conceptualización a través de un formato universal, comprensible por la comunidad en general. Una herramienta de modelado usual son los diagramas de clase UML, debido a su gran uso y la descripción de relaciones entre actores y entidades de manera gráfica. Los avances aquí mostrados hacen uso de clases "Agent Unified Modelling Language" (AUML), que son una extensión de los diagramas de clase UML.

Por otro lado, las ontologías integran conceptos a diferentes niveles de abstracción; a los primeros grupos de conceptos de primer orden se les agregan sub-classes (conceptos de segundo orden), seguido a esto axiomas y finalmente restricciones, las cuales brindan la posibilidad de inferir conocimiento, a esto se le conoce como conceptos de tercer orden.

En el documento presentado, se muestran los conceptos, taxonomías y relaciones de la ontología de orden superior, y se dan ejemplos de ontologías de segundo orden.

Así mismo, el formalismo de la estructura que integra esta clasificación se hace partiendo de ontologías existentes y está ligada a la propuesta "Extended Enterprise" establecida por López-Ortega and Ramírez-Hernández (2007), misma que fue desarrollada con el fin de integrar modelos de datos EXPRESS estandarizados bajo tres normas ISO; ISO 10303 (ISO, 1997), ISO 15531 (ISO, 1998) e ISO 13584 (ISO, 1999), conocidas como STEP, MANDATE y PLIB, respectivamente. Así los conceptos y taxonomías se describen en el siguiente apartado.

#### 3.2. Conceptos de Primer y Segundo Orden

En la ontología de primer orden es conceptualizado el conocimiento de una celda de manufactura identificando, agrupando y describiendo "conceptos" de un modo "taxonómico" y de manera independiente al contexto del dominio tratado (Temas evaluados en la Tabla 1). Esto es, los conceptos comunes a toda ontología de orden superior en el dominio Manufactura son: *Cliente*, *Producto*, *Orden*, *Operacion*, *Recurso*, *Holon*, *HolonAccion* y *Estado*, conceptos que servirán en lo subsiguiente para contener ontologías de orden inferior (véase Figura 1). Así, este vocabulario se irá integrando bajando de nivel en los diferentes grupos de conceptos.

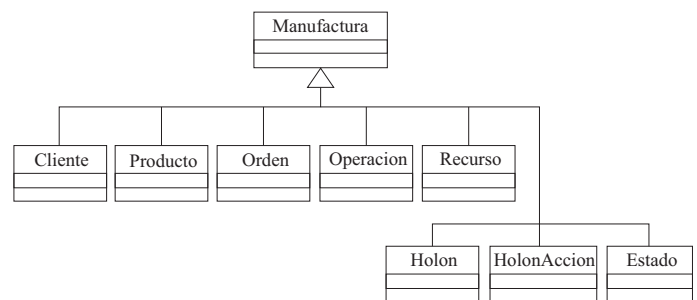
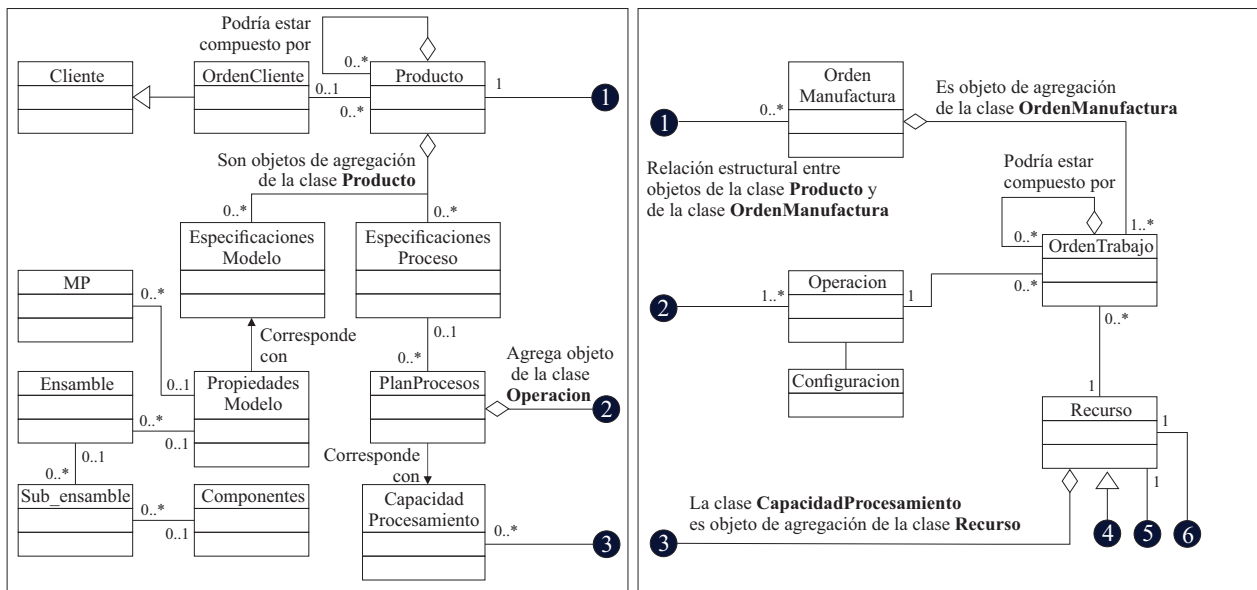


Figura 1: Conceptos. Orden superior.

##### a) Cliente Vs Producto

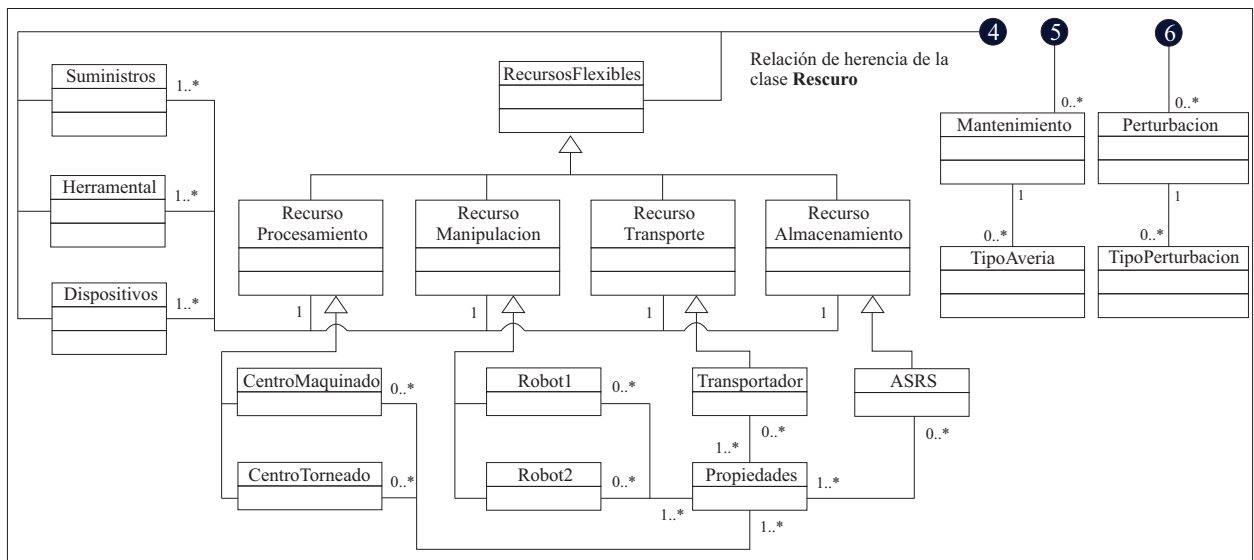
En un inicio (sección (a) de la Figura 2), es definida una clase "Cliente" agregándole alguno de los productos ofertados por el sistema "Producto", una cantidad de unidades requeridas (unidadesPorProducto), un valor de preferencia (Prioridad) y la fecha de solicitud (fechaSolicitud). Esta asociación de datos a una serie de productos es llamada orden del cliente (OrdenCliente) la cual incluye la referencia hacia un producto, su cantidad, prioridad y fecha de solicitud, todos ellos definidos como atributos de una orden del cliente.

Los productos son descritos por las especificaciones del modelo (EspecificacionesModelo) y por las especificaciones del proceso (EspecificacionesProceso). Las especificaciones del modelo contienen todas las propiedades de cada modelo específico (PropiedadesModelo), como; datos técnicos y características estructurales. Los datos técnicos incluyen la forma geométrica (Computer-Aided Design, CAD) e imágenes (Imagen) del producto, y las características estructurales de cada producto incluyen materia prima (MP) y sub-productos (Sub\_ensamble) con sus partes de ensamble (Componentes) que lo constituyen. Dichas especificaciones marcan la diferencia entre los distintos tipos de productos, por lo tanto, estas especificaciones incluidas en cada modelo delimitan los requerimientos



(a) Clases principales: Cliente, Producto

(b) Clases principales: Operacion, Orden



(c) Clase principal: Recurso

Figura 2: Ontología. Clases conceptuales de manufactura.

de calidad. Por otro lado, las especificaciones del proceso definen el proceso de manufactura o ensamble al cual deben ser sometidas las materias primas o los componentes de cada producto, a esto se le llama plan de procesos (PlanProcesos).

Ambas especificaciones antes descritas son necesarias para que un producto pueda ser fabricado (Manufacturado) y entregado como producto terminado (productoTerminado).

*b) Operacion Vs Orden*

Como se dijo, las especificaciones del proceso describen el plan de procesos (PlanProcesos), que es una lista de operaciones necesarias para producir un producto (por ejemplo: maquinados, ensambles o sub-ensambles, manipulación, transporte, mantenimiento o inspección) de entre una serie de tiempos de operación posibles.

Así bien, una operación (Operacion) es un trabajo que debe ser ejecutado en un orden específico con el fin de elaborar el

producto, y es caracterizada por un conjunto de información o atributos como: tiempo estimado de procesamiento (tiempoEstimadoProcesamiento), descripción (Descripcion), precedencia (Precedencia), requisitos técnicos (requisitosTecnicos), entre otros.

Un plan de producción se constituye por una o varias órdenes de clientes las cuales se convierten en órdenes de manufactura (OrdenManufactura). Una orden de manufactura es indexada a un objeto producto y comprende una lista de órdenes de trabajo. Una orden de trabajo (OrdenTrabajo) es la descripción de una operación (o trabajo) y por lo tanto es parte de un plan de proceso (obsérvese sección (b) de la Figura 2).

*c) Recurso*

La sección (c) de la Figura 2 muestra también cómo las órdenes de trabajo están destinadas a ser ejecutadas por recur-

son (Recurso) tales como fresadoras, tornos, robots, transportadores, entre otros. Cada recurso es una entidad que puede ejecutar una determinada gama de trabajos (CapacidadProcesamiento), siempre y cuando esté disponible (Disponibilidad), cuente con lo necesario (clases: Suministros, Herramental y Dispositivos) y no se exceda su capacidad (Capacidad). Los recursos que heredan estas características antes nombradas son denotados como recursos flexibles (RecursosFlexibles) de entre los cuales se clasifican como recursos de procesamiento (RecursoProcesamiento), recursos de manipulación (RecursoManipulacion), recursos de transporte (RecursoTransporte) y recursos de almacenamiento (RecursoAlmacenamiento). Cada uno de ellos es subclasificado según su tipo y modelo (por ejemplo en un: CentroMaquinado, CentroTorneado, Robot1, Robot2, Transportador, o un ASRS “Automatic Storage and Retrieval Systems”), y cuenta con una serie de propiedades (Propiedades) para dar servicio a las propias necesidades requisitadas por una orden de trabajo.

En la literatura revisada (Estado del Arte) con frecuencia son usados los términos perturbación y mantenimiento como factores afectivos a los recursos. Una emergencia normalmente ocurre de una perturbación (Perturbacion) inesperada activando un comportamiento adaptativo. Las perturbaciones pueden clasificarse como una clase (TipoPerturbacion) la cual tiene datos de la gama de perturbaciones que pudieran afectar al sistema, sus causas e impactos posibles. Por otro lado, el mantenimiento (Mantenimiento) es asociado a una lista de descomposturas posibles (TipoAveria). Según sea el factor que afecte a los recursos, se desencadena un mecanismo de adaptación que puede dar inicio a la auto-organización individual de los holones y que dependiendo de la gravedad del caso podría existir una reorganización de todos los recursos involucrados en el sistema (sección superior derecha de la Figura 2 (c)). Un buen ejemplo del tema perturbación y el uso de feromonas para incrementar los niveles de acción y propagar la necesidad de reorganización entre holones es dado por Barbosa et al. (2015), quien propone un sistema reconfigurable con capacidad evolutiva mediante un mecanismo de auto-organización.

#### d) Holon

Después de la evaluación de los artículos mostrados en la Tabla 1 respecto al tema “Holones/Agentes”, se identificarán diferentes holones especificados estos mismos dentro de la clase *Holon* de la Figura 3. Esta última clase, contiene conceptos de segundo orden; cuatro holones básicos y seis complementarios. En un HMS, cada holón representa a una unidad lógica del sistema de manufactura de interés, clasificadas como dispositivos físicos y entidades no físicas (Unland, 2015). Los dispositivos físicos son representados por holones en todos los niveles de abstracción, por ejemplo, el holón recurso (HolonRecurso) puede simbolizar a recursos flexibles como: máquinas, robots, cintas transportadoras, un almacén, e incluso suministros, herramental o dispositivos, de manera semejante, un holón producto (HolonProducto) representa a los distintos productos, las materias primas, etc; Por otro lado, ejemplos de entidades no físicas representadas por holones son: holón orden (HolonOrden) que simboliza órdenes de clientes, órdenes de manufactura, órdenes de trabajo, etc. El holón configuración (HolonConfiguracion) figura planes de producción, programas globales de

producción, entre otros. Es así como, estos holones y algunos más como los holones complementarios de la Figura 3, representar a cada unidad lógica del sistema de manufactura.

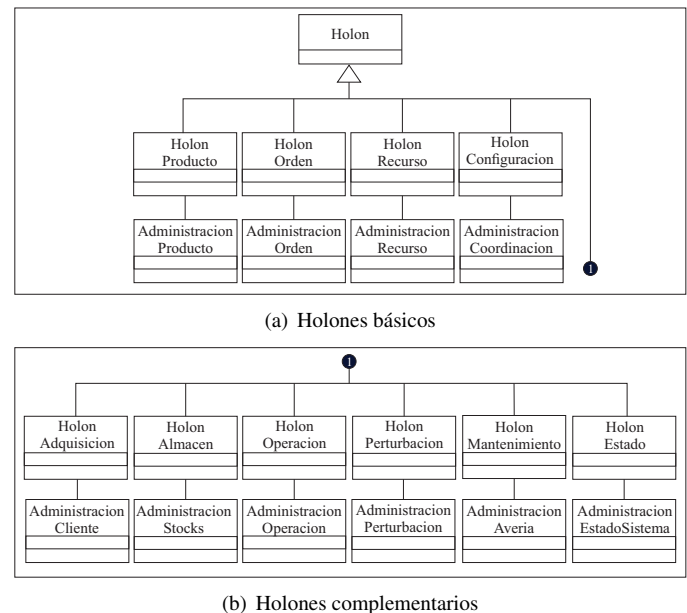


Figura 3: Holones. Tipología de holones básicos y complementarios.

La clase *HolonAccion* se menciona en el siguiente inciso.

#### e) HolonAccion

Las acciones del holón son conceptos especiales que indican acciones que pueden ser realizadas por este mismo. La Figura 4 ilustra acciones que ejercen los holones básicos sobre un HMS, y aunque éstas son de interés para un sistema de esta naturaleza, en el apartado no se hace hincapié de cómo operan los métodos de optimización, control, programación, configuración, supervisión, colaboración, negociación, acuerdos, auto-examinación, etc., con lo que la sección se limita a listar las acciones necesarias y que deben ser contempladas en la ontología.

##### HolonProducto

- Son parte de las órdenes del cliente (OrdenCliente).
- En coordinación con el holón almacén (HolonAlmacen), tiene conocimiento de las propiedades del modelo (PropiedadesModelo) como: materia prima (MP) o componentes (Componentes) necesarios para elaborar a cada producto solicitado.
- Hace diseño y (re)diseño modular del producto (EspecificacionesModelo).
- Verifica la calidad (EspecificacionesProceso).
- Intercambia información de la clase producto (Producto) y del plan de procesos (PlanProcesos) con el holón operación (HolonOperacion).
- Interactúa indirectamente con holones orden (HolonOrden) y configuración (HolonConfiguracion) durante la elaboración y (re)planeación de planes de procesos que incluyen la capacidad de procesamiento (CapacidadProcesamiento) de los holones recurso (HolonRecurso).
- Con apoyo del holón estado (HolonEstado), supervisa la tardanza, el progreso y el estado de un producto.

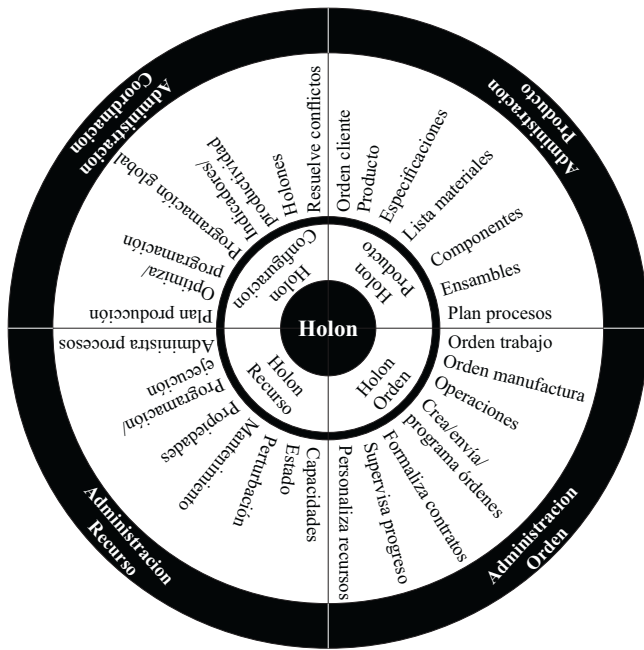


Figura 4: Holones básicos. Componentes y acciones.

**HolonOrden**

- En coordinación con el holón adquisición (HolonAdquisición), recibe órdenes del cliente (OrdenCliente) agrupadas como una orden de manufactura (OrdenManufactura).
- Administra la lista de productos y operaciones (Operación) necesarias para su elaboración.
- Crea y envía propuestas de órdenes de trabajo (OrdenTrabajo).
- Formaliza un contrato entre un holón recurso (HolonRecurso) y un holón producto (HolonProducto) mediante un acuerdo de la orden de trabajo.
- Programa, configura y monitorea acuerdos de trabajo con holones recurso.
- En colaboración con el holón estado (HolonEstado), supervisa la tardanza, el progreso y el estado de una orden de trabajo.
- Tiene la facultad de anular el acuerdo con un recurso.
- Evalúa al holón recurso por el cumplimiento correcto de una orden de trabajo.
- Penaliza al recurso por incumplimiento de contrato de orden de trabajo.

**HolonRecurso**

- Selecciona una propuesta de orden de trabajo la cual incluye una o más operaciones.
- Tiene conocimiento de las capacidades del recurso físico (RecursosFlexibles).
- Inicia y controla el procesamiento de una orden de trabajo.
- Junto con el holón mantenimiento (HolonMantenimiento) administra el plan y ejecución de mantenimiento (AdministraciónAvería).
- Tiene conocimiento de los tiempos de proceso, movimiento, transporte y almacén de un recurso físico para el cálculo de la fecha de entrega.

- Se auto-examina y examina a sub-recursos para percibir y reportar perturbaciones (TipoPerturbación) afectivas al recurso físico, en coordinación con el holón perturbación (HolonPerturbación).
- Con apoyo del holón estado (HolonEstado), supervisa la tardanza, el progreso y el estado de un recurso físico.

**HolonConfiguración**

- Propone al holón operación la ejecución de operaciones de manera optimizada e intercambian información relacionada con la asignación de recursos y del control de ejecución de dichas operaciones.
- Resuelve los conflictos entre los holones orden que requieren el mismo recurso (Recurso).
- Es responsable de la configuración del plan de proceso.
- Genera la programación global del sistema de manufactura.
- Evalúa el costo de (re)configuración de la holarquía.
- Ajusta la configuración (insuficiencia o falla de recursos).
- Es responsable de la negociación.
- Está a cargo de resolver los conflictos derivados de una perturbación reportada.
- Con ayuda del holón estado, supervisa la tardanza, el progreso y el estado del total de holones integrados al sistema.

f) *Estado*

Finalmente la clase *Estado* incluye información que describe el estatus de los distintos conceptos de primer orden y de orden inferior en un instante particular del tiempo (Figura 5).

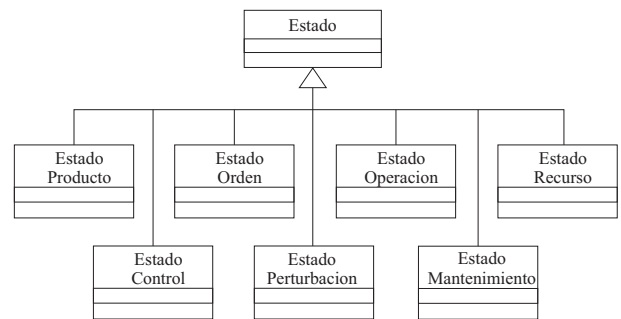


Figura 5: Estado. Monitoreo por instantes de tiempo.

3.3. *Atributos*

En los incisos antes descritos se han mencionado atributos asociados con clases características en un intento por introducir al lector en el nivel de detalle necesario en toda ontología (tema discutido en la Tabla 1). La Tabla 2, amplía el número de atributos asociados con algunas clases dadas en la Figura 2.

Es importante notar que los atributos alojan primitivas, variables o términos que formarán parte de la información compartida por los holones y que serán críticos en las decisiones que éstos tomen.

De esta manera, una clase *Cliente* admite datos del cliente que hace un pedido, el objeto *Producto* contiene datos de los diferentes modelos de productos. Una *Orden* posee datos de asociación entre los diferentes tipos de órdenes que capturan el aspecto logístico de la producción como: tiempos fijos, la cantidad y los recursos contratados. La clase *Operación* administra datos de los métodos y proceso de producción, además

de que captura el tipo de operación y la descripción técnica del funcionamiento. El concepto *Recurso* representa la capacidad y los registros históricos de funcionamiento, rendimiento, orden y producto. Estas últimas y algunas más pueden verse como ejemplos en la Tabla 2.

Tabla 2: Ejemplo de clases y atributos.

Cliente	+ idCliente: integer – nombreCliente: string – direccionCliente: string
OrdenCliente	+ idOrden: integer – fechaSolicitud: date – fechaEntrega: date – Prioridad: chart – listaProductosOrdenados: ArrayList<Productos> – unidadesPorProducto: Chart – EstadoOrden: string
Producto	+ idProducto: integer + nombreProducto: string – tipoProducto: string + costoProducto: numeric + cantidad: char + fechaVencimiento: date + EstadoProducto: string
SpecifModelo	+ idProducto: integer + idModel: integer + formaGeometrica: string + tipoMaterial: string + parametrosMfg: string + idPropModelo: integer
MP	+ idMp: integer – nombreMP: string + numeroCatalogo: integer – tipoMP: string – descMP: varchar + costoMP: numeric – EstadoMP: string
Operacion	+ idOperacion: integer + nombreOperacion: string + tiempoEstimadoProcesamiento : char + Precedencia: varchar + requerimientosTecnicos: varchar + operacionConcluida: Boolean + operacionEntregada: Boolean + operacionEnsamble: Boolean + EstadoOperacion: string
PlanProcesos	+ idEspecificacionesProceso: integer + idProducto: integer + listaOperaciones: ArrayList<Operacions> + secuenciaOperaciones: ArrayList<Secuenciado>
OrdenManufactura	+ idOrdenManufactura: integer + listaOrdenes: ArrayList<Ordenes> + numeroOrden: integer + listaProductosOrdenados: ArrayList<Productos> + numeroProductos: integer + fechaEntrega: data
OrdenTrabajo	+ idOrder: integer + idOrdenManufactura: integer + EstadoOrdenTrabajo: integer + idOperacion: integer + idEnsamble: integer + idSubEnsamble: integer + fechaEntrega: data + fechaProgramada: data
Recurso	+ idRecurso: integer + tipoRecurso: chart + EstadoRecurso: string

### 3.4. Predicados

Los predicados son expresiones sobre el estado de mundo y establecen relaciones entre conceptos, por lo que son utilizados eficazmente como el contenido de un mensaje en una interacción. La interacción entre las entidades distribuidas, es decir, holones, consiguen el control de manufactura holónica (Borgo and Leitão, 2007). Algunos ejemplos son:

- Producto( $a, b$ ): el producto  $a$  es parte de la orden del cliente  $b$ .
- MP( $a, b$ ): el producto  $a$  requiere de la materia prima  $b$ .
- Componente( $a, b$ ):  $a$  producto es un componente del ensamble  $b$ .
- Ensamble( $a, b$ ):  $a$  ensamble es una operación necesaria en el modelo  $b$ .
- EspecificacionesProceso( $a, b$ ): la especificación  $a$  corresponde con el producto  $b$ .
- AsignacionPlanProcesos( $a, b$ ): la producción de  $a$  requiere del plan de procesos  $b$ .
- PlanProcesosOperacion( $a, b$ ): el plan de procesos  $a$  incluye la operación  $b$ .
- PoseeRequisito( $a, b$ ):  $a$  operación requiere de la especificación del modelo  $b$ .
- EjecucionOperacion( $a, b$ ): la orden de trabajo  $a$  incluye la operación  $b$ .
- DisponibilidadRecurso( $a, b, t$ ): el recursos  $a$  está disponible en el momento  $t$  para la operación  $b$ .
- OrdenTrabajo( $a, b, t, s$ ): la operación  $a$  se asigna al recurso  $b$  en el tiempo  $t$ , a un costo  $s$ .

El siguiente apartado muestra de manera ejemplar el uso de la ontología en un caso de estudio particular.

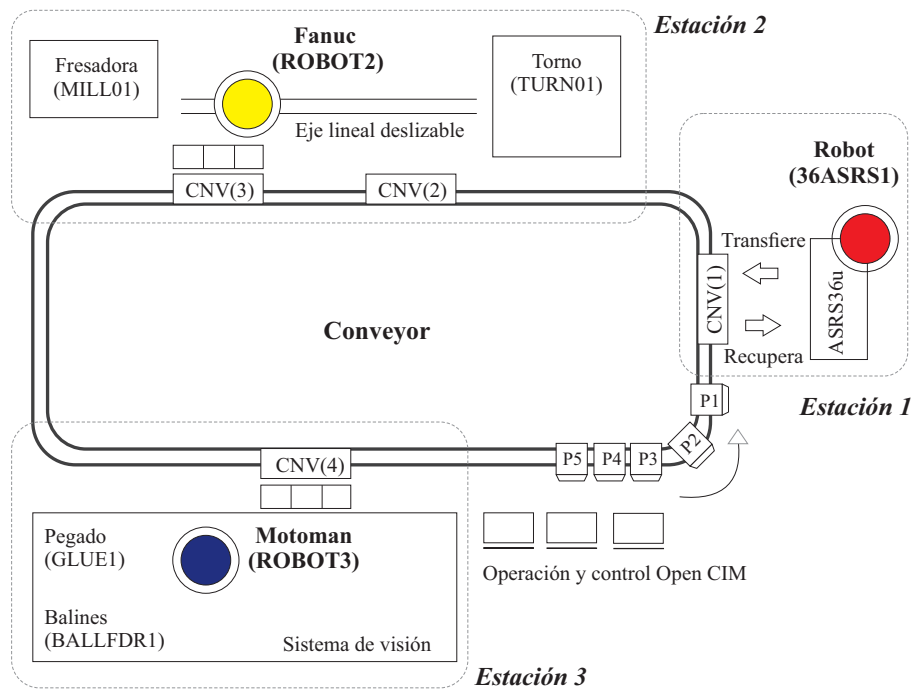
## 4. Caso de Estudio

La Celda de Manufactura Flexibles (CMF) de la Universidad Autónoma del Estado de Hidalgo (UAEH) que servirá como caso de estudio para la presente investigación, está equipada con tres estaciones de trabajo e integrada por una banda transportadora, y un software de operación y control de nombre Open CIM (véase la Figura 6).

Los recursos disponibles en la celda de manufactura se agrupan como sigue:

- Almacenamiento. El almacén automático (ASRS36u) de la estación 1 está integrado por un robot cartesiano (36ASRS1) dedicado a transferir y recuperar partes entre las celdas del almacén y los pallets detenidos en la estación de trabajo “CNV(1)” de la banda transportadora.
- Transporte. Se hace uso de un Conveyer de lazo cerrado integrado por cinco pallets (P1, P2, P3, P4, P5) para el transporte de materia prima o componentes entre recursos.
- Manipulación. Los robots “ROBOT2” y “ROBOT3” están encargados de la manipulación y ensamble en las estaciones 2 y 3, respectivamente.
- Procesamiento. Un centro de torneado (TURN01) y un centro de fresado (MILL01) manufacturan los productos en la estación 2.

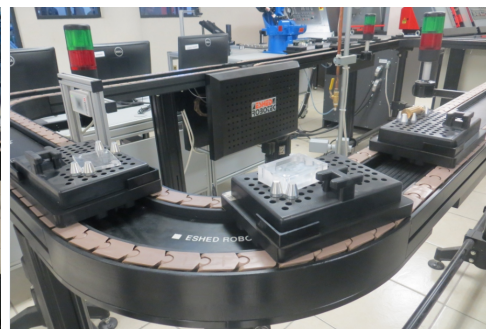




(a) Plano esquemático



(b) Imagen del robot Fanuc



(c) Imágenes del Conveyor

Figura 6: Celda de manufactura de la UAEH.

La orden de un cliente puede constituirse hasta por tres tipos de productos distintos: MILL\_TEST, TURN\_TEST y GAME-PROD. Los dos primeros productos requieren de materia prima, mientras el último de ellos necesita de un sub-ensamble por lo que hace necesaria la demanda de componentes.

#### 4.1. Integración y Programación de Ontología

En este apartado, son puestos a prueba ejemplos simples de roles y comportamientos de holones característicos haciendo uso de los conceptos propios de la celda de manufactura de la UAEH, en una plataforma JADE (Java Agent DEvelopment Framework).

La propuesta de la ontología unificada se integra mediante la herramienta PROTÉGÉ (formato, OWLSimpleJADEAbstractOntology.owl) bajo los términos recomendados por la Ontology Web Language (OWL) que utiliza clases para describir conceptos y predicados, y fija estos como parte de la aplicación ontológica. Para ello, se hace uso de un modelo interno JADE definido en la referencia Lemaignan et al. (2006), e ilustrado en la Figura 7.

Una ventaja importante al hacer uso de PROTÉGÉ es su

capacidad para reescribir archivos creados en PROTÉGÉ mediante código abierto como archivos Java haciendo uso de la herramienta Ontology Bean Generator.

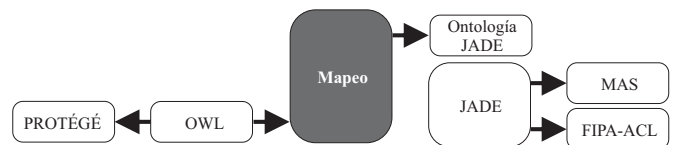


Figura 7: Mapeo entre OWL y JADE para el modelo ontológico propuesto (Lemaignan et al., 2006).

De esta manera, se genera una ontología con una implementación subyacente inmediata para asegurar las operaciones de lenguajes JADE entre los distintos holones (codificación, comprobación y decodificación de los mensajes), que es compatible con Java base para la plataforma de múltiples agentes JADE.

De aquí, la ontología *Manufactura* mostrada en la Figura 8, incluye elementos como: clases, atributos, instancias, propiedades o métodos como los agregados a la Figura 4 y restricciones.

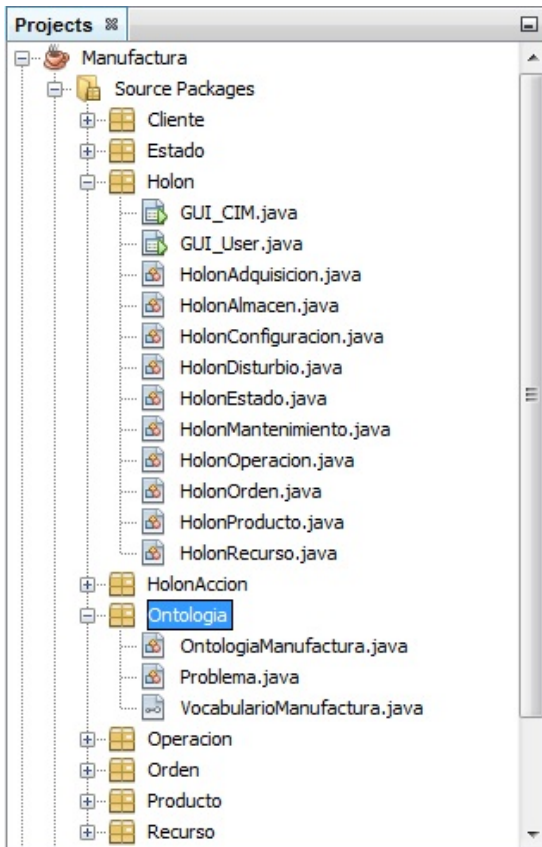


Figura 8: JADE. Programación de ontología.

Una ontología en JADE es una instancia de la clase `jade.content.onto.Ontology` a la que se han agregado esquemas que definen los tipos de conceptos, acción del holón y predicados relevantes para el dominio manufactura. Estos esquemas son instancias de las clases `ConceptSchema`, `AgentActionSchema` y `PredicateSchema` incluidas en el paquete `jade.content.schema`. Cada una de estas clases tiene métodos con los cuales es posible declarar principalmente nombres y slots definiendo la estructura de cada tipo de concepto, acción de los holones y predicados que son pertinentes para el dominio manufactura.

En este sentido, en el paquete `Ontologia` incluido en la Figura 8 se define la clase principal `OntologiaManufactura` la cual es una instancia de `jade.content.onto.Ontology` que a su vez agrega los tipos de conceptos y acción de los holones, e implementa la clase `VocabularioManufactura` misma que incluye un vocabulario con todos los conceptos que serán compartidos (slots y atributos de las diferentes clases existentes en la ontología, clasificados estos últimos como vocabulario básico y vocabulario para la ontología).

#### 4.2. Protocolo de Comunicación

JADE implementa especificaciones FIPA por lo que los holones hacen uso del Lenguaje de Comunicación entre Agentes (Agent Communication Language, ACL) definido por la FIPA (FIPA-ACL), acompañado de una selección de lenguajes para expresar contenido o Lenguaje Semántico (Semantic Language FIPA, FIPA-SL), de una ontología que indica el vocabulario de los símbolos utilizados en el contenido, y de un conjunto de protocolos de interacción que van desde el intercambio de un

sólo mensaje, hasta transacciones complejas (Caire and Cabanillas, 2010).

La FIPA-ACL define comunicación en términos de una función o acción llamada Acto Comunicativo (Communicative Acts, CA), realizado por el acto de comunicar. Por otro lado, una expresión de contenido FIPA-SL indica la sintaxis utilizada para expresar el contenido de un `ACLMessage` (clase o formato particular de los mensajes JADE). Tanto el emisor como el receptor deben ser capaces de codificar y analizar expresiones que cumplan con esta sintaxis para que la comunicación sea efectiva (Bellifemine et al., 2004).

En un principio el sistema de manufactura holónico se compone de un conjunto de holones, no organizados esperando a ser solicitados. Al ejecutar la aplicación es creado automáticamente un holón de nombre "HolonAdquisicion", responsable de solicitar mediante una interfaz gráfica "GUI\_User" al administrador de la celda de manufactura las características que definen a un nuevo cliente en el sistema (nombre, teléfono y dirección). Véase área superior de la Figura 9.

La información correspondiente al cliente es almacenada como un conjunto de parámetros en la clase "Cliente" con las siguientes variables: `private String idCliente`, `private String nombreCliente`, `private float telefonoCliente=0` y `private String direccionCliente`.

Posterior a ello, cuando el administrador solicita una o más órdenes nuevas "Create a new order" en la interfaz gráfica, es creado en automático un holón de nombre "HolonOrden" por cada orden con productos distintos solicitados, quienes interactúan tanto con la interfaz "GUI.User" como con el primer holón definido. Es entonces cuando se inicia con el protocolo de comunicación entre holones (Figura 10).

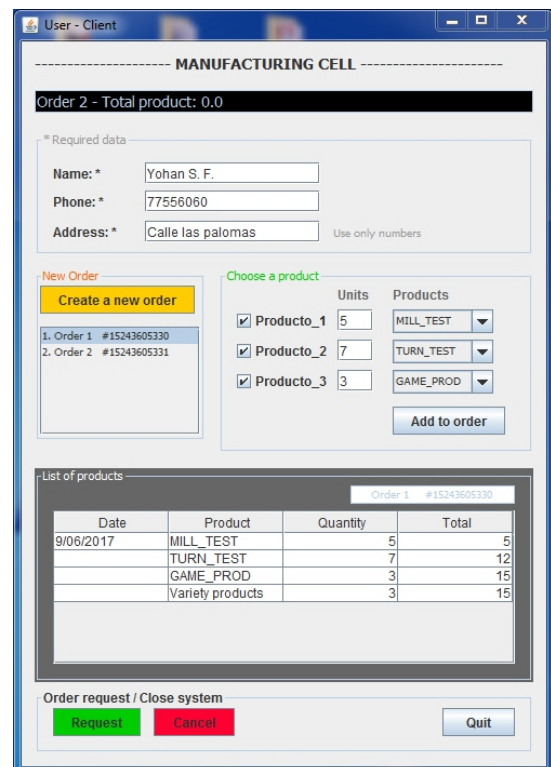


Figura 9: GUI.User. Definición de nuevos cliente y órdenes.

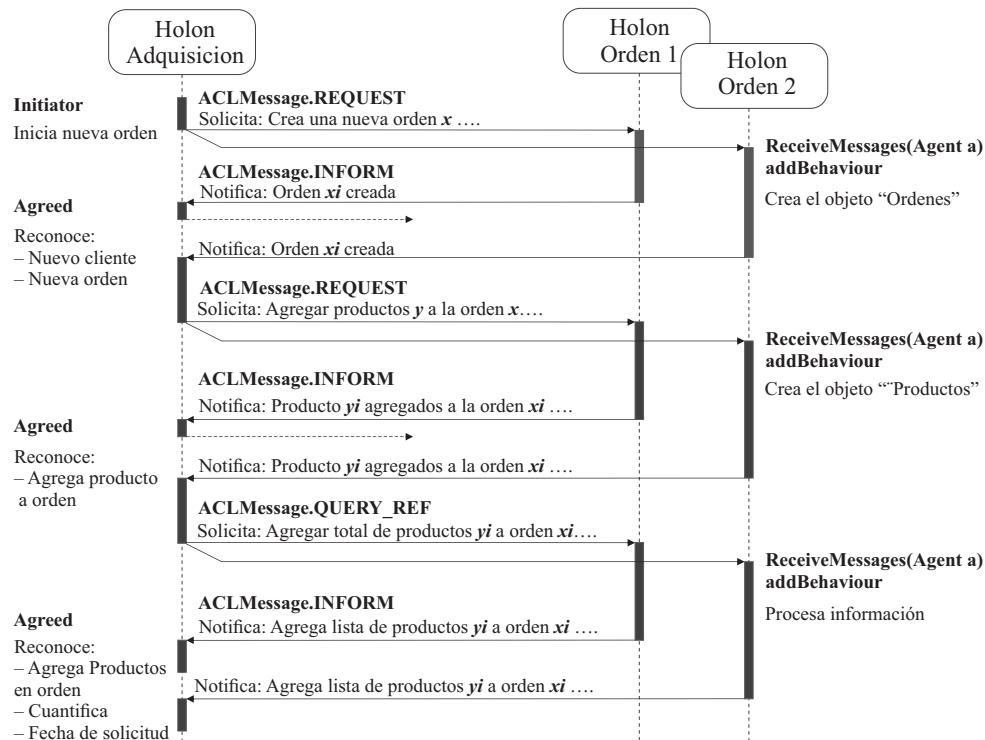


Figura 10: Protocolo de comunicación. HolonAdquisicion, HolonOrden 1 y HolonOrden 2.

Obsérvese en la Figura 10 que después de ser solicitada una nueva orden por el administrador del sistema, el HolonAdquisicion agrega el objeto *creaSolicitud* donde es instanciada la clase "AdministracionCliente" como una acción que incluye los atributos del cliente (*idCliente*, *nombreCliente*, etc.), mismo que es enviado mediante un mensaje tipo `ACLMessage.REQUEST` al HolonOrden. Con dicha información, el receptor actúa creando una nueva orden *x* del cliente asignándole un identificador numérico tanto al cliente como a la orden (*idCliente*, *idOrden*), notificando con un `ACLMessage.INFORM`.

Una vez creada la orden, el HolonAdquisicion agrega al objeto *solicitarProducto* donde es instanciada una nueva clase de nombre "AdministracionOrden". Posterior a ello, el remitente (HolonAdquisicion) envía el nuevo objeto al receptor (HolonOrden) solicitando realice la acción de agregar un producto *y* elegido por el cliente a la nueva orden *x* mediante un acto de petición (`ACLMessage.REQUEST`). Así, el receptor quien después de generar un producto (*generaProducto*) realiza otro acto comunicativo (`ACLMessage.INFORM`) pero esta vez regresando la clase *OrdenCliente* que agrega tanto las propiedades del cliente, la orden y un primer producto solicitado.

Finalmente, un último mensaje (`ACLMessage.QUERY_REF`) por parte del remitente es enviado al HolonOrden solicitando agregar a la orden una lista de todos los productos *yi* requeridos por el cliente (*listaProductosOrdenados*: `ArrayList<Productos>`), sus cantidades (*unidadesPorProducto*), la fecha de solicitud (*fechaSolicitud*), etc., respondiendo este último holón con un mensaje de tipo `ACLMessage.INFORM`.

El CA, que incluye un "performative", indica lo que el remitente desea lograr enviando el mensaje. Por ejemplo, si el performative es `REQUEST`, el remitente quiere que el receptor realice una acción, si es `INFORM`, el remitente quiere que el

receptor tenga conocimiento de un hecho, o bien, si fuese `PROPOSE` o `CFP` (Call for Proposals), el remitente requerirá entrar en una negociación.

De esta manera, es posible que los holones compartan objetos compuestos en JADE, sacando provecho de las capacidades para compartir e intercambiar conceptos de la ontología a través del paso asíncrono de mensajes.

#### 4.3. Envío de mensajes

El código mostrado en la Figura 11 muestra el envío de mensaje hacia un holón.

Primeramente se busca al holón que incluya ciertas características (`buscarServidor()`).

Seguido a esto, se crea el nuevo mensaje (`ACLMessage msg = new ACLMessage(performative)`), al cual se le incluye tanto el lenguaje (`msg.setLanguage(codec.getName())`) como la ontología usada (`msg.setOntology(ontology.getName())`). La clase `codec` para los lenguajes FIPA-SL<sub>n</sub>, implementa la interfaz `Codec` (`private Codec codec = new SLCodec()`) y permite la conversión de ida y vuelta entre cadenas y frames de acuerdo con la gramática SL, de esta forma, se especifica el lenguaje semántico o vocabulario de los símbolos usados con el cual los parámetros contenidos son expresados (`import jade.content.lang.sl.*;`).

La clase `ontology`, implementa la interfaz `Ontology` (`private Ontology ontology = OntologiaManufactura.getInstance()`), misma que define los métodos para convertir objetos de la clase ontológica de manufactura en/a partir de descriptores abstractos (`import jade.content.onto.*`). Más específicamente, la ontología valida la información a convertir desde un punto de vista semántico mientras que el `codec` realiza la traducción en cadenas (o secuencias de bytes) de acuerdo con las reglas sintácticas del lenguaje de contenido relacionado.

Finalmente, en la Figura 11 se define un comportamiento en espera de respuesta por parte del holón receptor (`addBehaviour(new EsperarServidorResponda(this))`).

```

void enviarMensaje(int performative, AgentAction action) {
// -----
if (servidor == null) buscarServidor();
if (servidor == null) {
    alertGui("¡No se puede localizar al Holon!, Operación
abortada");
    return;
}
ACLMessage msg = new ACLMessage(performative);
msg.setLanguage(codec.getName());
msg.setOntology(ontology.getName());
try {
    getContentManager().fillContent(msg, new
Action(servidor,
action));
    msg.addReceiver(servidor);
    send(msg);
    alertGui("Contactando servidor... Por favor espere!");
    addBehaviour(new EsperarServidorResponda(this));
}
catch (Exception ex) { ex.printStackTrace(); }
}

```

Figura 11: JADE. Método usado en el envío de mensajes.

Los términos individuales contenidos en un mensaje pueden referenciarse explícitamente a una ontología, con lo que, la ontología indica y da significado al vocabulario de los símbolos utilizados en el contenido del mensaje. Tanto el emisor como el receptor deben atribuir el mismo significado a estos símbolos para que la comunicación sea efectiva, con lo que, un holón puede iniciar la comunicación con cualquier otro holón en cualquier momento que desee y también puede ser el objeto de una comunicación entrante en cualquier momento (Bellifemine et al., 2004).

#### 4.4. Holarquías, Negociación y Colaboración

Atendiendo a las consideraciones antes mencionadas, la nueva orden del cliente creada ahora propiedad del holón orden (HolonOrden) es compartida tanto con un nuevo holón producto (HolonProducto) como con el holón configuración (HolonConfiguracion).

Hecho lo anterior, en las arquitecturas holónicas de la literatura revisada (Estado del Arte) el holón configuración en coordinación con los holones orden y producto podrían empezar a negociar con holones recurso la prestación de determinadas órdenes de trabajo. Durante el proceso de negociación, el holón orden exigirá propiedades específicas de la operación por realizar, tales como calidad o rendimiento, mientras que los holones recurso tratarán de maximizar su utilización. Al final de la negociación, los holones recurso se organizan para configurar y formar la línea de producción necesaria.

Seguido a esto, a los holones producto se les permite la entrada en la holarquía de manufactura, quienes buscan negociar de inmediato recursos físicos con el fin de conseguir ser procesados. Una vez que estas operaciones sean realizadas en un recurso físico, la pieza de trabajo reiniciará la negociación con las

(próximas) operaciones restantes. Del mismo modo que la estructura *ADACOR*<sup>2</sup> en estado estacionario, los holones podrían estar organizados en una estructura jerárquica, la actividad general del holón configuración juegan el papel de coordinación y optimizan la programación de sus subordinados organizados en grupos, su trabajo es mantener una negociación permanente entre holones orden y holones recurso asegurando que la carga de holones producto se distribuyan de manera eficiente en los recursos físicos disponibles para alcanzar los objetivos globales de esta holarquía. El sistema *ADACOR*<sup>2</sup> funciona conforme a esta configuración hasta que se detecta una perturbación. Si un holón recurso detecta la perturbación en un recurso físico, éste analiza primeramente el problema e intenta reorganizar al recurso físico a nivel local mediante la realización de un autodiagnóstico, si la recuperación del mismo no tiene éxito, su factor de autonomía se incrementa y se propaga la necesidad de reorganización a otros holones en el sistema, compartiendo la carga de trabajo como sucede en los sistemas biológicos cuando existe un daño. Más información respecto a cómo operan estos mecanismos, puede consultarse en: Brussel et al. (1998), Wang et al. (2005), Leitão and Restivo (2006), Barbosa et al. (2015), Araújo et al. (2015).

## 5. Análisis de Resultados

### 5.1. Conjetura Ontológica

El intercambio y reutilización del conocimiento requieren de un marco común para apoyar la interoperabilidad de las ontologías creadas de manera independiente. Los resultados de la investigación muestran que existe una gran diversidad en la manera en que se conciben las ontologías y en la forma en que representan al dominio de manufactura. Al identificar las similitudes y diferencias entre ontologías existentes, se vislumbra el rango de alternativas en la creación de un marco estándar para el diseño de ontologías disponibles para HMS.

### 5.2. Conjetura Modelo Experimental

El ejercicio descrito muestra que la comunicación es un componente clave en los holones, tanto con los usuarios del sistema, recursos del sistema, con ellos mismos y con cualquier otro hólón. En particular se aprecia cómo los holones interactúan unos con otros usando un lenguaje de comunicación específico y protocolos de interacción predefinidos.

Así, todas las conceptualizaciones (definiciones, categorizaciones, jerarquías, propiedades, herencia, etc.) de una ontología pueden ser procesables por el ordenador, puesto que la profundización semántica proporciona una descripción lógica y formal que puede ser interpretada tanto por las personas, como por las máquinas.

Cada uno de estos ejemplos de comportamientos obedece a las propias necesidades de los diferentes holones. Por consiguiente, esta clase de holones abstractos, que tienen ya implementadas las funcionalidades básicas de los agentes inteligentes (búsqueda de otros agentes, comunicación, gestión de comportamientos, gestión de la línea de espera de mensajes, etc.), constituyen un sistema de múltiples agentes que interactúan para resolver los problemas que están más allá de las capacidades individuales y el conocimiento individual de cada agente abstracto extendiéndose para crear las clases que representan cada

tipo holón de un sistema de manufactura (holones producto, orden, recurso, configuración, etc.), añadiendo en cada caso los comportamientos e interacciones de cada recurso flexible.

Con ello, en entornos de manufactura distribuida cada holón tiene conocimiento parcial y total del sistema (parte y todo), con lo que el control holónico sobre los sistemas de manufactura surge como un todo integrado de la interacción entre los holones distribuidos, donde cada uno de estos contribuye con su conocimiento local.

## 6. Conclusiones y Trabajo Futuro

Este artículo presentó una ontología unificada en dominio de manufactura en beneficio de los HMS. Las aportaciones al hacer uso de la ontología se pueden resumir como sigue:

- La ontología actúa con claridad como repositorio para la organización correcta del conocimiento en dominio manufactura y sirve de herramienta para la adquisición de información.
- Al hacer uso de un vocabulario común, es posible representar coherentemente el conocimiento para toda clase de holones.
- El formalismo que brinda la ontología propuesta, permite hacer uso de un lenguaje de comunicación específico, acompañado de una selección de lenguajes de contenido para el intercambio de conocimiento.
- La ontología brinda la posibilidad de crear conjuntos de protocolos de interacción claves que van desde el intercambio de un sólo mensaje hasta transacciones complejas, extendiendo el conocimiento a los diferentes holones.
- La ontología permite la recuperación y reutilización del conocimiento de manera específica y precisa.

Las expresiones formales en JADE tanto en tiempo de desarrollo como de ejecución muestran cómo en un inicio cada hólón ejemplificado posee conocimiento parcial del sistema o una lista de habilidades necesarias para cooperar con otros holones indirectamente (actuando sobre el medio ambiente) o de manera directamente (vía comunicación y negociación), para alcanzar sus objetivos o para obtener información adicional sobre el sistema, compartiendo conocimiento para transformar el conocimiento local en conocimiento global, siendo esta la característica principal de HMS.

Por otro lado, al hacer uso de la ontología propuesta, se pudo observar que es complejo el hecho de formalizar, implementar y experimentar con una ontología propia y aplicada a un sistema de inteligencia artificial distribuida, agentes inteligentes que son programados para que reflejen las capacidades de holones. Este hecho nos impulsa hacer un esfuerzo por crear y mantener un programa de mantenimiento que mejora constantemente la consistencia de la ontología en este dominio.

Finalmente, un problema que surge de manera natural es proponer una arquitectura de control inteligente haciendo uso del paradigma de manufactura holónico en las áreas de manufactura antes mencionadas, que permita vincular flexibilidad, modularidad y control descentralizado en un sistema de manufactura flexible.

## Agradecimientos

Isaías Simón Marmolejo agradece el apoyo recibido del Programa para el Desarrollo Profesional Docente (PRODEP), del Centro de Tecnología Avanzada (CIATEQ) y de la Universidad Autónoma del Estado de Hidalgo (UAEH).

## Referencias

- Araújo, J. A., del Olmo-Martínez, R., Laviós, J. J., de Benito-Martín, J. J., 2015. Programación y control de sistemas de fabricación flexibles: un enfoque holónico. *RIAI - Revista iberoamericana de automática e informática industrial* 12 (1), 58–68.  
DOI:10.1016/j.riai.2014.11.005.
- Barbosa, J., Leitão, P., Adam, E., Trentesaux, D., 2015. Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution. *Computers in industry* 66, 99–111.  
URL: <http://dx.doi.org/10.1016/j.compind.2014.10.011>
- Bellifemine, F., Caire, G., Greenwood, D., 2004. Developing multi-agent systems with JADE. Wiley Series in Agent Technology, Series Editor: Wooldridge, M., Liverpool University, UK.
- Blanc, P., Demongodin, I., Castagna, P., 2008. A holonic approach for manufacturing execution system design: An industrial application. *Engineering applications of artificial intelligence* 21 (3), 315–330.  
DOI:10.1016/j.engappai.2008.01.007.
- Borgo, S., Leitão, P., 2007. Foundations for a core ontology of manufacturing. In: Sharma, R., Kishore, R., Ramesh, R. (Eds.), *Ontologies, A Handbook of principles, concepts and applications in information systems*. Springer Science+Business Media, NY, USA, Ch. 27, pp. 751–775.  
DOI:10.1007/978-0-387-37022-4\_27.
- Botti, V., Giret, A., 2008. ANEMONA, A multi-agent methodology for holonic manufacturing systems. Springer: Departamento de sistemas informáticos y computación (DSIC), Valencia, Spain.  
DOI:10.1007/978-1-84800-310-1.
- Bravo, C., Aguilar-Castro, J., Ríos, A., Aguilar-Martín, J., Rivas, F., 2011. Arquitectura basada en inteligencia artificial distribuida para la gerencia integrada de producción industrial. *RIAI - Revista iberoamericana de automática e informática industrial* 8 (4), 405–417.  
URL: <http://dx.doi.org/10.1016/j.riai.2011.09.013>
- Brussel, H. V., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P., 1998. Reference architecture for holonic manufacturing systems: PROSA. *Computers in industry* 37 (3), 255–274.  
DOI:10.1016/S0166-3615(98)00102-X.
- Caire, G., Cabanillas, D., April 2010. JADE TUTORIAL, application-defined content languages and ontologies. Support provided by JADE, Italia S.p.A., 9th Edition.  
URL: <http://jade.tilab.com/doc/tutorials/CLOntoSupport.pdf>
- Checa, D., Rojas, O., 2014. Ontología para los sistemas holónicos de manufactura basados en la unidad de producción. *Revista colombiana de tecnologías de avanzada* 1 (23), 134–141.  
URL: [http://www.academia.edu/8676921/ONTOLOGÍA\\_PARA\\_LOS\\_SISTEMAS\\_HOLÓNICOS\\_DE\\_MANUFACTURA\\_BASADOS\\_EN\\_LA\\_UNIDAD\\_DE\\_PRODUCCIÓN](http://www.academia.edu/8676921/ONTOLOGÍA_PARA_LOS_SISTEMAS_HOLÓNICOS_DE_MANUFACTURA_BASADOS_EN_LA_UNIDAD_DE_PRODUCCIÓN)
- Christensen, J. H., December 1994. Holonic manufacturing systems: Initial architecture and standards directions. In: *At first European Conference on Holonic Manufacturing Systems*. Hannover, Germany, pp. 1–20.  
URL: <http://ho1ob1oc.com/papers/hannover.pdf>
- Corcho, O., Fernández-López, M., Gómez-Pérez, A., 2003. Methodologies, tools and languages for building ontologies. Where is their meeting point? *Data & Knowledge Engineering* 46 (1), 41–64.  
URL: [http://dx.doi.org/10.1016/S0169-023X\(02\)00195-7](http://dx.doi.org/10.1016/S0169-023X(02)00195-7)
- Correa da Silva, F. S., Vasconcelos, W. W., Robertson, D. S., Brilhante, V., de Melo, A. C. V., M., F., Agustí, J., 2002. On the insufficiency of ontologies: problems in knowledge sharing and alternative solution. *Knowledge-Based Systems* 15 (3), 147–167.
- de las Morenas, J., García, A., Martínez, F., P., G., 2015. Implementación del control en planta de un centro de distribución automatizado mediante agentes físicos y RFID. *RIAI - Revista iberoamericana de automática e informática industrial* 12 (1), 25–35.  
URL: <http://dx.doi.org/10.1016/j.riai.2014.11.002>
- Fernández, M., Gómez-Pérez, A., Uristo, N., 1997. Methontology: From ontological art towards ontological engineering. *AAAI Technical Report*, 33–40.  
URL: <http://oa.upm.es/5484/>

- Giraldo, G., Arboleda, A., Zapata, G., 2013. Enfoque ontológico para detectar conceptos holónicos en las organizaciones. *Revista facultad de ingeniería Universidad de Antioquia* 69, 53–66.  
URL: <http://www.redalyc.org/articulo.oa?id=43029812004>
- Gómez-Pérez, A., Fernández, M., Vicente, A. d., August 1996. Towards a method to conceptualize domain ontologies. In: *At 12th European Conference on Artificial Intelligence*. Budapest, Rumanía, pp. 41–51.  
URL: <http://oa.upm.es/7228/>
- Indriago, C., Cardin, O., Rakoto, N., Castagna, P., Chacòn, E., 2016. H2CM: A holonic architecture for flexible hybrid control systems. *Computers in industry* 77, 15–28.  
DOI:10.1016/j.compind.2015.12.005.
- ISO, 1997. ISO 10303 - Industrial Automation Systems and Integration - Product data representation and exchange - Part 1: Overview and fundamental principles. ISO - International Standards Organization, Geneva, Switzerland.  
URL: <https://www.iso.org/standard/20579.html>
- ISO, 1998. ISO 15531 - Industrial Automation Systems and Integration - Industrial Manufacturing Management Data - Part 1: General overview. ISO - International Standards Organization, Geneva, Switzerland.  
URL: [http://hsevi.ir/RI\\_Standard/View/2880](http://hsevi.ir/RI_Standard/View/2880)
- ISO, 1999. ISO 13584 - Industrial Automation Systems and Integration - Parts Library - Part 1: Overview and Fundamental Principles. ISO - International Standards Organization, Geneva, Switzerland.  
URL: <https://www.iso.org/standard/25103.html>
- Jules, G., Saadat, M., Saeidlou, S., 2015. Holonic ontology and interaction protocol for manufacturing network organization. *IEEE Transactions on systems, man, and cybernetics: systems* 45 (5), 819–830.  
DOI:10.1109/TSMC.2014.2387099.
- Koestler, A., 1990. *The ghost in the machine*. Arkana.
- Koppensteiner, G., Grabler, R., Miller, D., Merdan, M., 2015. Virtual enterprises based on multiagent systems. In: *Leitão, P., Karnouskos, S. (Eds.), Industrial agents: Emerging application of software agents in industry*. Elsevier Inc., MA, USA, Ch. 7, pp. 121–136.  
URL: <http://dx.doi.org/10.1016/B978-0-12-800341-1.00007-3>
- Leitão, P., Restivo, F., 2006. ADACOR: a holonic architecture for agile and adaptive manufacturing control. *Computers in industry* 57 (2), 121–130.  
DOI:10.1016/j.compind.2005.05.005.
- Leitão, P., Restivo, F., 2008. Implementation of a holonic control system in a flexible manufacturing system. *IEEE systems, man, and cybernetics society* 38 (5), 699–709.  
DOI:10.1109/TSMCC.2008.923881.
- Lemaignan, S., Siadat, A., Dantan, J. Y., Semenenko, A., June 2006. MASON: A proposal for an ontology of manufacturing domain. In: *At IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications*. pp. 195–200.  
DOI:10.1109/DIS.2006.48.
- López-Ortega, O., Ramírez-Hernández, M., 2007. A formal framework to integrate express data models in an extended enterprise context. *Intelligent manufacturing* 18, 371–381.  
DOI:10.1007/s10845-007-0035-7.
- Luther, M., 2008. *Ontologies*. In: *Berndt, H. (Ed.), Towards 4G Technologies. Services with Initiative*. John Wiley & Sons, Ltd., Chippingham, England, Ch. 7, pp. 141–164.  
DOI:10.1002/9780470010334.
- Noy, N. F., McGuinness, D., March 2001. *Ontology development 101: A guide to creating your first ontology*. [online] Knowledge Systems Laboratory, Stanford University, Stanford, CA, EE. UU.  
URL: <http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html>
- Uunnar, F., Pujo, P., 2012. Pull control for job shop: holonic manufacturing system approach using multicriteria decision-making. *Intelligent manufacturing* 23 (1), 141–153.  
DOI:10.1007/s10845-009-0288-4.
- Raileanu, S., Borangiu, T., Radulescu, S., 2014. Towards an ontology for distributed manufacturing control. In: *Borangiu, T., Trentesaux, D., Thomas, A. (Eds.), Service Orientation in Holonic and Multi-Agent Manufacturing and Robotics*. Vol. 544 of *Studies in Computational Intelligence*. Springer International Publishing, Ch. Part II, pp. 97–109.  
DOI:10.1007/978-3-319-04735-5.7.
- Shen, W., Norrie, D. H., Barthès, A., 2001. *Multi-agent systems for concurrent intelligent design and manufacturing*. Taylor & Francis Group, NY, USA, Ch. 11, pp. 215–227.
- Tarun, K. J., Bipradas, B., Soumen, P., Bijan, S., Jyotirmoy, S., 2013. Dynamic schedule execution in an agent based holonic manufacturing system. *Manufacturing systems* 32 (4), 801–816.  
DOI:10.1016/j.jmsy.2013.07.004.
- Unland, R., 2015. *Industrial agents*. In: *Leitão, P., Karnouskos, S. (Eds.), Industrial agents: Emerging application of software agents in industry*. Elsevier Inc., Amsterdam, Netherlands, Ch. 2, pp. 23–44.  
URL: <http://dx.doi.org/10.1016/B978-0-12-800341-1.00002-4>
- Uschold, M., Gruninger, M., 1996. *Ontologies: principles, methods and applications*. *Knowledge Engineering Review* 11 (2), 93–136.  
DOI:10.1017/S0269888900007797.
- Uschold, M., King, M., Moralee, S., Zorgios, Y., 1998. *The enterprise ontology*. *The knowledge engineering review* 13 (1), 31–89.
- Wang, F., Xu, D., Tan, M., Wan, Z., 2005. A holonic architecture for reconfigurable manufacturing systems. *IEEE International conference on industrial technology*, 905–909  
DOI:10.1109/ICIT.2005.1600764.