

IPCASCI (Intelligent business Processes Composition based on mAs, Semantic and Cloud Integration)

Arquitectura Multiagente para la composición ágil de Procesos de
Negocio

Índice de Contenidos

<u>1</u>	<u>INTRODUCCIÓN</u>	<u>4</u>
1.1	HIPÓTESIS DE PARTIDA Y OBJETIVOS	5
1.2	ESTRUCTURA DE LA MEMORIA	6
<u>2</u>	<u>COMPUTACIÓN EN LA NUBE</u>	<u>9</u>
2.1	INTRODUCCIÓN	9
2.2	CONCLUSION	10
<u>3</u>	<u>MAS Y ONTOLOGÍAS</u>	<u>13</u>
3.1	INTRODUCCIÓN	13
3.2	ONTOLOGÍAS	16
3.3	CONCLUSIÓN	17
<u>4</u>	<u>PROCESOS DE NEGOCIO</u>	<u>19</u>
4.1	INTRODUCCION	19
4.2	SERVICIOS WEB	19
4.3	SERVICIOS WEB SEMÁNTICOS	20
4.4	BPM Y BPEL	21
4.5	CONCLUSIONES	23
<u>5</u>	<u>IPCASCI</u>	<u>25</u>
5.1	INTRODUCCIÓN	25
5.2	CONCLUSIONES	28
<u>6</u>	<u>RESULTADOS Y CONCLUSIONES</u>	<u>31</u>
6.1	CASO DE ESTUDIO	31
6.2	EVALUACIÓN DEL CASO DE ESTUDIO	32
6.3	ANÁLISIS DEL CASO DE ESTUDIO	33
6.4	CONCLUSIONES GENERALES	33

7 REFERENCIAS **36**

7.1 REFERENCIAS WEB **51**

1 INTRODUCCIÓN

La industria del desarrollo del software está en continua evolución, buscando metodologías, lenguajes y herramientas cada vez más potentes, eficientes y seguras. En este proceso es imprescindible crear modelos, arquitecturas y tecnologías ágiles, que permitan la creación de nuevas herramientas de la forma más sencilla y económica posible. La reutilización de componentes es sin duda uno de los actores que de forma más clara pueden contribuir en este desarrollo, facilitando mecanismos eficientes para crear software de calidad (Schmid, 2011a) (Poulin 2006) (Poulin, 1997) (Lemley and O'Brien, 1997) (Shang *et al.*, 2012) (Rehesaar, 2011). La reutilización aumenta la fiabilidad del software (ya que se usa software que ha sido probado), la productividad en el desarrollo e implica una clara reducción de costes (Schmid, 2011b) (Xu *et al.*, 2011) (Rehesaar, 2011) (García *et al.*, 2007). Con el actual aumento del volumen y complejidad de los productos software, la reutilización es un campo que se tiene en alta consideración, constandingo incluso como etapa fundamental en modelos de diseño y calidad como CMMI (*Capability Maturity Model Integration*) (Osiecki *et al.*, 2011).

En este contexto la reutilización de servicios Web aparecen como una alternativa interesante a la reutilización clásica de código. Fundamentalmente, un servicio Web es un componente software que representa un servicio desplegado en la plataforma Web, y que soporta interacciones automáticas entre máquinas dentro de la red (Walsh, 2002). En este marco, aparece SOA (Service-oriented Architecture) como una nueva arquitectura para el desarrollo software convencional (Erl, 2009) (Ralyté, *et al.*, 2011) (Alizadeh, *et al.*, 2012). SOA presenta un nuevo método para crear aplicaciones distribuidas donde los servicios básicos pueden ser publicados, descubiertos y enlazados, para construir servicios más complejos. Las aplicaciones interactúan con los servicios a través de puntos en interfaces y no a nivel de implementación (Papazougou, 2008).

El desarrollo del software puede analizarse desde distintas perspectivas y ofrecernos gran variedad de alternativas a nivel metodológico, funcional, instrumental, etc. Uno de los paradigmas que han revolucionado esta industria en los últimos años es el de la computación en nube o "*cloud computing*" (Antonopoulos *et al.*, 2012) (Stage and Setzer, 2009) (Duy *et al.*, 2011) (Cloud Computing, 2011) (EuroCloud *et al.*, 2011). La nueva representa un nuevo concepto a nivel de distribución de servicios e información, que ofrece numerosas posibilidades a la hora de escalar soluciones, facilitar la utilización de terminales relativamente simples y económicos, interesantes modelos de pago por uso, accesibilidad multiplataforma y multidispositivo, etc. Sin embargo el desarrollo de este modelo tiene ciertas barreras relacionada con la complejidad de mantenimiento y de desarrollo de aplicaciones (Jaeger *et al.*, 2009) (Dölitzscher *et al.*, 2010). El número de ingenieros con experiencia en este campo es relativamente bajo y los tiempos de desarrollo son considerablemente altos para los costes que este sector es capaz de asumir hoy en día (Massonet *et al.*, 2011) (Schaaf *et al.*, 2010) (Sulistio *et al.*, 2009).

Desde este punto de vista, en este proyecto de investigación, se propone una metodología que facilita la construcción de procesos de negocio para entornos cloud de forma ágil y eficiente a partir de componentes ya desarrollados. Se trata de presentar una propuesta que facilite la creación de estos procesos en forma de servicios web, a partir de otros servicios ya funcionales de forma semi-automática en el marco de un proceso guiado por programadores relativamente inexperto. El proceso está guiado por una organización virtual gestionada por una arquitectura multi-agente (Rodríguez *et al.*, 2011) (Dignum *et al.*, 2005) (Escriva, *et al.*, 2006) que implemente el comportamiento inteligente necesario para la gestión del proceso utilizando una ontología (Maedche and Staab, 2001) (Chandrasekaran, 1999) (Noy and McGuinness, 2001) (Guarino, 1998) (López, 1999). El objetivo de este trabajo de investigación es presentar un modelo que permita la construcción de un proceso de negocio a partir de las especificaciones en formato texto (con ciertas restricciones) del proceso que estamos interesados en crear. El sistema multiagente basado en organizaciones virtuales facilitará la composición de los procesos utilizando el estándar BPEL (*Business Process Execution Language*). Este estándar permite a su vez componer servicios web de forma sencilla con la ventaja de tener una proyección directa desde diagramas BPMN (*Business Process Management Notation*) (Pedrinaci *et al.*, 2008)

1.1 hipótesis de partida y objetivos

Los servicios web son un estándar en la industria de la informática en la actualidad y forman parte de muchas de las aplicaciones que utilizamos todos nosotros hoy en día (Erl, 2009). También es un estándar en el desarrollo de aplicaciones para entornos cloud o en nube (Massonet *et al.*, 2011) (Schaaf *et al.*, 2010) (Sulistio *et al.*, 2009). Los servicios web son también elementos esenciales en el desarrollo de procesos de negocio. Estos servicios facilitan la construcción de software relativamente rápido y de forma eficiente. Estos dos aspectos pueden mejorarse definiendo modelos adecuados de reutilización. Con este trabajo se pretende definir un modelo que satisfaga los requisitos de creación de nuevos servicios a partir de la composición de servicios ya probados de forma eficiente y utilizando herramientas que la inteligencia artificial pone a nuestra disposición. Teniendo esto en cuenta, la hipótesis de este trabajo de investigación puede definirse como:

“Una arquitectura multiagente basada en organizaciones virtuales es una herramienta adecuada para facilitar la construcción en entornos cloud de procesos de negocio a partir de otros ya existentes, con la ayuda de un modelo ontológico y las herramientas que proporciona el estándar Business Process Execution”

En el contexto de esta propuesta, para generar un proceso de negocio nuevo a partir de los servicios de los que se dispone se parte de la especificación de los requisitos que debe cumplir el proceso. Esta especificación está formada por una descripción semi-libre de los requisitos que describen el servicio. La organización virtual, basada en un sistema multi-agente gestiona las tareas que requieren de un comportamiento inteligente. Este sistema analizará la entrada (descripción textual de la propuesta) para descomponerla en funcionalidad computable para su posterior tratamiento. Además, el sistema multi-agente

sirve como soporte para el descubrimiento y agrupación de los servicios web presentes en la definición del nuevo proceso de negocio. Los servicios web o procesos de negocio almacenados que podrán ser reutilizados se habrán concebido desde el punto de vista de las arquitecturas SOA, y tendrán asociado un componente ontológico que permitirá que el sistema multi-agente basado en organizaciones virtuales sea capaz de identificarlos para su reutilización. El modelo propuesto implica que una vez se hayan identificado los servicios que compondrán el proceso de negocio solución se lleve a cabo su composición usando el estándar BPEL. Este estándar permite componer servicios web de forma sencilla con la ventaja de tener un mapeado directo desde diagramas BPMN. Los objetivos que se pretenden alcanzar en el marco de este trabajo de investigación son:

- Desarrollar un modelo que permita crear procesos de negocio en entornos cloud de manera eficiente, ágil y económicamente rentable, a la vez que se facilita la reutilización de componentes y el desarrollo de sistemas informáticos a medida por personal poco experimentado.
- El modelo debe ser capaz de analizar semánticamente la definición textual del proceso de negocio que se desea implementar. La definición textual será semilibre pudiéndose utilizar el vocabulario definido por la ontología establecida para cada caso. La definición propuesta será analizada antes de llevar a cabo la búsqueda de servicios web.
- Se definirá un sistema multiagente basado en organizaciones virtuales que se encargue de la búsqueda de servicios de manera que descomponga la descripción del proceso de negocio en módulos, determine las conexiones entre dichos módulos y proceda a buscar servicios web que implementen la funcionalidad requerida. Para ello cada servicio web almacenado en el sistema, además de su especificación WSDL (*Web Service Description Language*), tendrá un contenido ontológico asociado, de forma que se conozca completamente la funcionalidad que ofrece.
- El modelo facilitará la creación de un diagrama BPD (*Business Process Diagram*) que se adapte a los requisitos del problema (Owen y Raj, 2003): A partir de la entrada ya analizada y los servicios encontrados, se automatizará el proceso de diseño de un diagrama BPD (perteneciente al estándar BPMN) de tal forma que su contenido pueda ser proyectado a un fichero BPEL. Cada actividad en el diagrama se corresponderá con un servicio Web encontrado. Este modelo podrá ser modificado o adaptado por el programador para que se ajuste a las necesidades del software requerido. A partir del diagrama BPD (que refleje las necesidades del proyecto), el modelo propuesto facilitará la creación de un archivo BPEL que realice un mapeado fiel y completo de dicho diagrama teniendo en cuenta los servicios concretos que implementan cada actividad.

1.2 Estructura de la memoria

Esta memoria está organizada en 6 capítulos de la forma que se indica a continuación:

En el capítulo introductorio se presenta la motivación con la que se lleva a cabo este trabajo y se definen la hipótesis de partida y los objetivos de la tesis. Se hace también una somera descripción de los elementos que componen el modelo propuesto.

En el segundo capítulo se hace una revisión del estado del arte sobre los sistemas de computación en la nube (*Cloud computing*). Se describen las partes y conceptos fundamentales de un sistema cloud, así como las ventajas que aporta a un entorno tecnológico global. También se ofrece una descripción de los principales proveedores cloud en la actualidad (*Windows Azure, Amazon y Google App Engine*) y un estudio de la tecnología en la que se basan los sistemas en la nube. Finalmente se analizan las ventajas de la reutilización en este tipo de entornos.

En el tercer capítulo se presentan los BPMs (*Business Process Management*) y todos los conceptos relacionados con los modelos, se revisan los servicios Web y su semántica. BPM está formado por un conjunto de sistemas software, herramientas y metodologías, enfocadas hacia como las empresas identifican, modelan, desarrollan, distribuyen y gestionan sus procesos de negocio. Dentro de BPM cabe destacar BPMN (*Business Process Management Notation*), que es un estándar gráfico que permite el modelado y representación de procesos de negocio. Finalmente, se presenta una descripción de BPEL (*Business Process Execution Language*) que es un estándar incluido en BPM y que permite la definición de servicios web a partir de la composición de servicios web existentes. Los servicios Web son una tecnología en alza que permite el despliegue de software vía Web ofreciendo servicios en línea. Se presenta los dos tipos más relevantes de servicios Web (SOAP y REST) y sus principales características. En el último apartado de este capítulo se presenta la arquitectura SOA, que es una arquitectura donde los servicios básicos pueden ser publicados, descubiertos y enlazados, para construir servicios más complejos. Finalmente se presenta el estado del arte de los servicios Web semánticos, centrándose en OWL-S, que es la propuesta más desarrollada hasta el momento.

En el cuarto capítulo se presenta el estado del arte sobre los sistemas multi-agente y las ontologías. La tecnología de agentes ha crecido ampliamente pasando de un estudio puramente académico, a la implementación con éxito en diversas áreas. Se explican los conceptos básicos de agentes, diferentes clasificaciones de agente, y algunas arquitecturas para la construcción de agentes. También se describen los sistemas multi-agente, que son sistemas en los que intervienen diferentes agentes para la consecución de un objetivo común. Se analizan las organizaciones virtuales de agentes y las ventajas que aportan. Finalmente se presenta un apartado sobre ontologías, en qué consisten, guías para su desarrollo, y algunos lenguajes para la creación de ontologías.

En el quinto capítulo se presenta el modelo propuesto IPCASCI (*Intelligent business Processes Composition based on mAs, Semantic and Cloud Integration*) que facilita la composición de procesos de negocio en entornos de computación en nube, de forma inteligente, con la ayuda de un sistema multiagente basado en organizaciones virtuales y con la definición de una ontología. Se presenta el modelo y analizan los procesos que incluye. El modelo se presenta como una alternativa a los modelos OWL-S en la definición de servicios Web semánticos, incluyendo los procesos de búsqueda, composición e invocación automáticos.

Por último, en el capítulo sexto se presentan los resultados, las conclusiones y las líneas futuras.

2 COMPUTACIÓN EN LA NUBE

2.1 Introducción

La computación en la nube (o en inglés *cloud computing*) facilita el desarrollo de sistemas informáticos distribuidos y la gestión de datos y recursos computacionales a través de una red escalable de nodos, centros de procesos de datos y servicios web (Massonet *et al.*, 2011) (Schaaf *et al.*, 2010) (Sulistio *et al.*, 2009). La definición de computación en la nube propuesta por el Instituto Nacional de Estándares y Tecnología (NIST) de los Estados Unidos especifica cinco características "esenciales" para la computación en la nube: autoservicio; accesibilidad desde el escritorio, portátil y teléfono móvil; recursos compartidos por varios usuarios y aplicaciones; recursos flexibles que se pueden redistribuir rápidamente según sea necesario; y servicios medidos. Estas características se combinan para hacer de la computación en la nube una especie de infraestructura o servicio público. La idea principal detrás de la computación en la nube no es nueva. John McCarthy en los años 60 ya previó que las instalaciones de computación se proporcionarían al público general como una utilidad (Padala *et al.*, 2009). La computación en la nube ha emergido como un paradigma para ofrecer soporte y entrega de servicios a través de Internet. En el aspecto tecnológico constituye una nueva manera de construir aplicaciones orientadas a Internet y enfocadas a un número masivo de usuarios. El enfoque tecnológico de *Cloud Computing* es el de construir servicios (ya sean aplicaciones, servicios de hosting o servicios de almacenamiento) que puedan ofrecerse a un gran número de usuarios empleando el mínimo volumen de recursos hardware; y que permitan incrementar la capacidad del servicio de manera sencilla al aumentar los recursos disponibles. Esta característica se denomina escalabilidad de un servicio (INSA, 2012).

Respecto al modelo de negocio, la computación en la nube introduce un cambio en la manera de explotar y comercializar los productos de la empresa (Chang *et al.*, 2010). El modelo de adquisición de bienes hardware o software se convierte en un modelo de suscripción (Nesse *et al.*, 2011) (Jaeger *et al.*, 2009), o lo que es lo mismo, un modelo de consumo de servicios. Esto quiere decir que en vez de adquirir los recursos computacionales pertinentes, se contratan a proveedores. Es una aproximación atractiva para las empresas ya que elimina los requisitos de planificación futura de recursos y permite empezar desde abajo e ir incrementando los recursos solamente cuando realmente se necesiten (Döhlitzscher *et al.*, 2010) (Zhang *et al.*, 2003). Algunas características que hacen de cloud computing una tecnología con potencial son: (i) Los recursos en un entorno cloud pueden asignarse y desasignarse en función de las necesidades en cada momento, (ii) Los recursos pueden compartirse y/o asignarse a cada cliente en función de las necesidades, (iii) Los servicios almacenados en la nube generalmente están basados en tecnología web, lo que los hace accesible a partir de una gran variedad de dispositivos con conexión a Internet, (iv) Al externalizar la infraestructura de servicios a la nube, el proveedor de servicios transfiere sus riesgos de negocio (como fallos del hardware) a los proveedores de la infraestructura, que habitualmente tienen más experiencia y están mejor equipados para manejar dichos riesgos.

La novedad de esta tecnología y las opciones que ofrece son inmensas y su desarrollo está siendo exponencial en los últimos años. Uno de los mayores problemas en su desarrollo está relacionado con la definición de los modelos de pago por uso y del de la construcción ágil y eficiente de aplicaciones sobre esta infraestructura (Cloud Computing, 2011) (EuroCloud *et al.*, 2011). Una opción a la hora de desarrollar software de forma eficiente en este contexto es la utilización de metodologías que faciliten el diseño y programación de procesos de negocio de forma guiada. Es esta metodología o modelo de gestión del trabajo facilita la reutilización de servicios webs/procesos de negocio de manera sencilla, se habrá superado uno de los mayores retos de la computación en nube. En este capítulo se presenta esta tecnología, sus conceptos básicos, los proveedores de servicios, sus herramientas, limitaciones, su taxonomía y su estructura a nivel de capas. Por último se analiza su potencial y se establecen los beneficios de construir procesos de negocio a través de la reutilización.

2.2 Conclusion

La computación en nube tiene gran potencial y es un área dentro del desarrollo del software en plena expansión. La reutilización no es una práctica habitual en este contexto por los motivos que se han mencionado en la sección anterior y que afectan a todos los tipos de reutilización. La definición de una metodología que haga eficiente el sistema de reutilización a la vez que mantenga altos estándares de calidad es uno de los desafíos de esta memoria y para ello es necesario analizar el grado de aplicabilidad de la propuesta y las perspectivas que esta metodología puede tener.

La cantidad de software potencialmente reutilizable depende del grado de funcionalidad común que haya entre los sistemas que lo comparten (Sametinger, 1997). Definiendo dominio como área de aplicación o campo de desarrollo de sistemas, se entiende que el grado de reutilización es elevado y eficiente entre sistemas pertenecientes al mismo dominio. La reutilización de software entre sistemas de un mismo ámbito de aplicación recibe el nombre de reutilización vertical o dependiente del dominio. La reutilización horizontal, o de ámbito general, se establece entre sistemas que no pertenecen al mismo dominio, lo que motiva que ésta sea, en general, más compleja y, consecuentemente, el grado de reutilización mucho menor. En este contexto esta memoria se presenta en la reutilización vertical de procesos de negocio implementados en la forma de servicios web y ejecutados en un entorno de computación en nube. Tradicionalmente se han diferenciado dos metodologías para enfocar la reutilización de software; la primera se basa en la obtención de un nuevo sistema por composición de elementos ya existentes y la segunda en la generación del nuevo sistema utilizando como base una estructura o modelo. La reutilización por composición es bastante intuitiva, se trata de conjugar elementos o componentes para construir un sistema nuevo. Es en esta en la que se base este trabajo de investigación. Aunque conceptualmente es sencilla, requiere un soporte técnico complejo como métodos de clasificación y selección optimizados según los componentes, y soporte para su adaptación e integración en el nuevo sistema. Se propone para ello la utilización de una arquitectura multiagente basada en organizaciones virtuales. La reutilización por generación es conceptualmente más compleja, ya que no es posible definir los componentes como entes autocontenidos y

concretos. En este caso se reutilizan procesos de generación obtenidos como resultado de una *codificación de estructuras*. Se suelen distinguir tres subtipos de metodologías: generadores de aplicación, generadores basados en el lenguaje y sistemas transformacionales.

La generación de sistemas se ha orientado hacia la reutilización de elementos pertenecientes a las primeras fases del ciclo de vida del software como son diseños, arquitecturas o requisitos, lo que la hace más apetecible en cuanto a su posible rentabilidad pero, desde luego, mucho más difícil de aplicar. Sin embargo, la reutilización por composición fue la primera en utilizarse, centrándose en elementos pertenecientes a las últimas fases del proceso software, como es el caso del código. En el caso de aplicar reutilización por composición, es necesario concretar cómo se pretenden integrar los componentes en el nuevo sistema, para eso será preciso hablar de: visibilidad de componentes y modificación de componentes.

Se denominan *cajas negras* a aquellos componentes de los que sólo se conoce su interfaz y su funcionalidad; como no se tiene acceso a su interior es imposible realizar ningún tipo de modificación, así que en este caso sólo es posible hablar de reutilización íntegra. Aquellos componentes que permiten el acceso a su interior son llamados *cajas blancas*, en ellos es posible realizar modificaciones para adaptar su funcionalidad a la requerida. Un tipo intermedio son las *cajas transparentes* (Sametinger, 1997), este tipo de componentes permiten que se conozca su interior, como en las cajas blancas, pero su modificación no está permitida, como en el caso de las cajas negras. La reutilización de cajas negras, aunque mucho más difícil, presenta muchas más ventajas ya que se incrementa la calidad y la fiabilidad del software generado, los componentes de este tipo han sido verificados y suelen estar certificados (Dunn y Knight, 1993; Knight y Dunn, 1998). Sin embargo, cuando se reutilizan cajas blancas es muy probable que se realicen modificaciones que provoquen que haya que repetir las tareas de verificación, decrementándose su fiabilidad. La reutilización por generación puede ser vista como un tipo de reutilización de cajas negras, ya que, aunque no se dispone del componente en sí, se tiene un programa generador y éste es reutilizado como si fuera una caja negra. A lo largo de esta investigación se identificará el tipo más adecuado para alcanzar los objetivos propuestos.

Según la definición que aquí se ha adoptado para la reutilización de software, toda información necesaria para el diseño y desarrollo de un sistema se considerará potencialmente reutilizable. En el caso que nos ocupa además es muy importante disponer de una ontología clara y una semántica bien definida. Así se tendrán tanto código como documentos, diseños, especificaciones, textos, etc. Las diferencias entre la reutilización de unos u otros radica, principalmente, en su grado de abstracción y la granularidad del componente, ambas características son fuente de controversia a la hora de decidir qué reutilizar (Prieto, 1996). Es necesario adoptar una posición de compromiso entre los beneficios que aporta la reutilización de determinado elemento, directamente proporcionales al nivel de abstracción y a su tamaño, y la facilidad con que éste es reutilizado, inversamente proporcional a los mismos factores. En esta memoria se presenta una metodología para hacer la reutilización de cajas blancas eficiente y sobreponerlos a los motivos económicos, técnicos y de organización que en

términos generales hacen que no se lleve a cabo la reutilización de software. En el próximo capítulo se van a definir las organizaciones de agentes que se encargarán de gestionar este proceso y en el cuarto capítulo se definirán el campo de los procesos de negocio y elementos afines que se utilizan en el modelo propuesto.

3 MAS Y ONTOLOGÍAS

3.1 Introducción

El concepto de agente ya ha sido ampliamente estudiado en nuestros días. Sin embargo, existe un pequeño vacío en la consideración de agente como entidad que opera dentro de una sociedad. Y se trata de un tema especialmente importante para un agente, pues una de sus características básicas es que se encuentra situado en un entorno cooperando con otros agentes. El presente capítulo estudia el concepto de agencia desde el punto de vista de su utilización en entornos cooperativos. Comienza con una revisión de la de noción de agente inteligente. A continuación se expone una descripción de los sistemas multi-agente, junto con la definición de sociedad de agentes. El objeto es poner de manifiesto las tendencias actuales del desarrollo de SMA desde el punto de vista organizativo. La teoría de la agencia se estudiado desde campos como la psicología, informática, sociología, medicina, economía etc., siendo diferentes sus características y comportamientos en cada uno. Respecto a las Ciencias de la Computación, el término agente cada vez es más conocido y se emplea en ámbitos tan diversos como Internet, sistemas distribuidos, inteligencia artificial o la interacción persona-computador (Corchado, 2000). Tanto la diversidad como el potencial de esta tecnología es extraordinariamente amplia y aunque en este capítulo se presenta una descripción relativamente genérica de la teoría de la agencia y de los sistemas multiagentes en último lugar se presenta el concepto de organización, ya que para desarrollar el modelo propuesto se empleará una organización virtual de agentes.

Los agentes se pueden describir desde muchos puntos de vista. Esto se debe a que es un área de investigación que ha atraído a científicos procedentes de áreas muy dispares: psicología, sociología, ingeniería del software, inteligencia artificial, etc. y cada uno de los miembros de estas comunidades tiende a ver el problema desde su perspectiva (Forner, 1993). Por tanto, realizar una definición de agente o agencia es complicado debido a esta diversidad de opiniones que existe en la comunidad científica (Franklin and Graesser, 1996). Incluso dentro del ámbito de la informática, no existe una definición de agente bien determinada. El testigo cedido por los investigadores iniciales en el campo de agentes ha sido tomado por otros nuevos en ámbitos tan variados como: inteligencia artificial distribuida, robótica, vida artificial, computación distribuida de objetos, interacción persona/ordenador, interfaces inteligentes y adaptativas, búsqueda y filtrado de información, adquisición de conocimiento etc. Debido a la proliferación de los diferentes tipos de agentes, ha ocurrido una explosión en el uso del término sin un correspondiente consenso sobre su significado (Bradshaw, 1997). Dentro de las diferentes definiciones, una definición formal y bastante aceptada entre la comunidad científica es la proporcionada por Labidi y Lejouad en (Labidi and Lejouad, 1993):

“Un agente es una entidad física o abstracta que puede percibir su ambiente a través de sensores, es capaz de evaluar tales percepciones y tomar decisiones por medio de mecanismos de razonamiento sencillos o complejos, comunicarse con

otros agentes para obtener información y actuar sobre el medio en el que se desenvuelve a través de efectores. “

Otra forma de definir el concepto de agente bastante extendida se basa en la caracterización sobre una serie de atributos que el agente debe poseer. De esta forma, en (Wooldridge and Jennings, 1995) se definen 4 atributos básicos que caracterizan el concepto de agente:

- Autonomía.
- Habilidades Sociales.
- Reactividad.
- Pro-actividad.

El desarrollo de sistemas informáticos inteligentes está guiado por la creación de entidades con comportamiento similar al de los humanos (Russel and Norvig, 1995). Al igual que los humanos, los agentes deben tener habilidades sociales y ser capaces de realizar trabajos o resolver problemas de forma distribuida (D'Inverno and Luck, 2004). Un agente puede verse como la evolución del concepto de objeto que incorpora características propias del comportamiento humano. Dentro de estas características se puede destacar la inteligencia o la capacidad de aprendizaje. El desarrollo en la tecnología de procesamiento de la información y de la computación, han hecho posible la construcción y el uso de agentes artificiales (Burgin and Dodig, 2009). La aparición del concepto de agente ha permitido que diversas disciplinas hayan convergido en un espacio común. Los sistemas compuestos de múltiples agentes se desarrollaron inicialmente en el ámbito de la Inteligencia Artificial Distribuida (O'hare *et al.*, 1996). En su origen, la IAD plantea la resolución de problemas distribuidos, donde un problema particular puede resolverse por un cierto número de elementos que cooperan y comparten conocimiento sobre el problema y su solución. De esta manera, estos sistemas forman parte de una de las tres categorías básicas de la IAD, siendo las otras dos: la Solución Distribuida de Problemas y la Inteligencia Artificial paralela. Como resultado, los sistemas compuestos de múltiples agentes heredan muchas de las motivaciones, metas y beneficios potenciales de la IAD (Nwana, 1995). El objetivo consiste en construir sistemas compuestos por múltiples entidades capaces de resolver problemas, de forma que interactúen entre ellas para mejorar su rendimiento (Jennings, 1993). Estos sistemas se conocen como Sistemas Multi-Agente (MAS), y son adecuados para solucionar problemas para los que hay múltiples métodos de resolución y/o múltiples entidades capaces de trabajar conjuntamente para solucionarlos (Chu-Carroll *et al.*, 1995). Así, los MAS podrán ser utilizados para resolver problemas que son difíciles o imposibles de resolver para un agente individual.

Los agentes inteligentes son objeto de gran interés por parte de múltiples campos de la informática y de la Inteligencia Artificial (IA) como paradigma para el desarrollo de aplicaciones (Jennings and Wooldridge, 1998). Los retos a los que se enfrentan en nuestros días los desarrolladores de aplicaciones en los sistemas de información son cada vez más complejos. La globalización y los cambios en la tecnología han causado que el mercado actual se encuentre en un estado de fluctuación constante. Las empresas que no sean capaces de adaptarse suficientemente de prisa serán dejadas atrás. En respuesta, muchas empresas están construyendo sistemas basados en

agentes. Estos sistemas emplean agentes software que distribuyen funcionalidad a través de redes informáticas. Además, los agentes no solo se adaptan a su entorno sino que evolucionan mediante el aprendizaje del entorno y emplean una variedad de enfoques computacionales, desde sistemas expertos OO, redes neuronales, algoritmos genéticos y demás. Este enfoque prepara a las empresas para un mercado creciente y complejo permitiéndoles responder con rapidez a los cambios producidos (Odell, 2010). La aplicación de los sistemas multi-agente no se restringe solamente al ámbito empresarial y académico, siendo Internet una de las áreas más importantes de desarrollo. Los agentes de Internet surgen a partir del problema de la búsqueda y filtrado de información en la Web. Estos agentes resultan útiles debido a la gran cantidad de información que es posible encontrar sobre cualquier tema en este medio (Burghoff *et al.*, 1996). Como el tiempo es, cada vez más, un factor determinante en la vida de las personas, y la posibilidad de acceso a la información es cada vez más amplia, los usuarios de Internet deben desarrollar habilidades y buscar herramientas para acceder oportuna y acertadamente a los datos que requieren para su desempeño personal. Los agentes virtuales son una buena alternativa para ahorrar tiempo de búsqueda de información e incluso para el análisis y atención de peticiones (Gutiérrez, 2008). Algunas aplicaciones de los sistemas multi-agente son:

- Agentes de filtrado de información: Encuentran el contenido de particular interés para un usuario usando diferentes fuentes de información.
- Agentes de entrega offline: Filtran que entregan información personalizada sin requerir necesariamente la conexión on-line a Internet.
- Agentes de búsqueda: Emplean robots que se mueven en el hiperespacio de la Web en nombre de los usuarios para proporcionar un servicio.

La computación en nube es otro campo en el que se están utilizando los agentes y sistemas multi-agente. La computación en la nube puede ofrecer una infraestructura de computación potente, predecible y escalable, para la ejecución de sistemas multi-agente que implementan complejas aplicaciones basadas en agentes, cuando el modelado y simulación de sistemas complejos deba ser proporcionada. Por otra parte, los agentes software pueden ser usados como componentes básicos para implementar comportamiento inteligente en la computación en la nube, haciéndola más adaptativa, flexible y autónoma en el manejo de recursos, distribución de servicios y ejecución de aplicaciones a larga escala (Talia, 2012). Un área en el que los sistemas multi-agente se están utilizando con más éxito actualmente es el mundo de los videojuegos. A medida que los videojuegos serios han aumentado su popularidad, han requerido de un comportamiento más natural y elaborado por parte de los personajes no jugables. Mientras más elaboradas sean las interacciones entre los personajes en un juego, más difícil será el diseño de dichos personajes sin el uso de herramientas especializadas orientadas a la implementación de agentes inteligentes (Dignum, 2011). Uno de los primeros intentos para conectar los agentes a los videojuegos fue realizada con los *Gamebots* (Adobbati, 2001). Los *Gamebots* proporcionan una infraestructura que permite la conexión de cualquier plataforma de agentes al juego *Unreal Tournament*. Actualmente se utilizan los agentes para emular el comportamiento humano de forma que la experiencia de juego refleje un mayor realismo.

Otra área de aplicación de los sistemas multi-agente es la minería de datos. Hay dos enfoques: donde los agentes realizan el proceso de minería de datos, o la minería de datos se usa para mejorar la característica de inteligencia de los agentes (Moemeng *et al.*, 2011).

3.2 Ontologías

La comunicación lleva implícita una serie de problemas como la inconsistencia del lenguaje, diferentes contextos, ambigüedades etc. Se pretende establecer un conocimiento compartido que solucione los problemas de comunicación y que sirva de base para (Uschold and Gruninger, 1996):

- Comunicación interpersonal con diferentes contextos y puntos de vista
- Interoperabilidad entre sistemas.
- Reusabilidad del conocimiento
- Facilitación del proceso de especificación de requisitos.
- Permitir la comunicación y entendimiento entre agentes software (Gruninger *et al.*, 2007).

En este contexto, aparece el concepto de ontología, inmerso en múltiples disciplinas y con significados dispares. En filosofía, el término ontología se refiere al estudio de las cosas que existen (Chandrasekaran and Josephson, 1999). En el siglo XVII el término ontología se usa como sinónimo de metafísica, concretamente como la rama de la metafísica que trata con la naturaleza del ser (Swartout and Tate, 1999). En Inteligencia Artificial habitualmente se atribuye el concepto de ontología a la especificación de una conceptualización (incluye la definición de términos y las relaciones entre ellos), preferiblemente formalmente y computable (Hendler, 2001). Las ontologías representan la tecnología esencial que permite y facilita la interoperabilidad a nivel semántico, proporcionando una conceptualización formal de la información que puede ser compartida y reutilizada (d'Aquin and Noy, 2011). Proporcionan una forma de representar y compartir el conocimiento utilizando un vocabulario común. Además, permiten usar un formato de intercambio de conocimiento y una reutilización de dicho conocimiento (Cantera *et al.*, 2007). Típicamente, una ontología proporciona el vocabulario que describe un dominio de interés y la especificación del significado de los términos usados en el vocabulario (Euzenat and Shvaiko, 2007).

Debido a los beneficios que conlleva la compartición de conocimiento y su modelado, las ontologías se están utilizando con éxito en diferentes áreas. Entre ellas, una de las que más relevancia tiene es la Web semántica, es más, la relación entre la web semántica y las ontologías ha suscitado mucho interés en los últimos años. La presentación de ontologías bien diseñadas mejora el efecto de los servicios de la Web semántica (Oh and Yeom, 2012). Otra área en la que las ontologías se usan ampliamente es los sistemas multi-agente. Al describir un conjunto de conceptos y las relaciones entre ellos, las ontologías pueden ser usadas para construir la arquitectura jerárquica del conocimiento de negocio y la lógica de la regulación de las negociaciones y actividades entre agentes (Wang *et al.*, 2012). El uso de ontologías en sistemas multi-agente puede realizarse para la descripción de objetos y la

comprensión común de sus relaciones. Promueven la eficiencia en la comunicación y colaboración entre sistemas multi-agente (Wang and Liu, 2012).

3.3 Conclusión

El objetivo de este trabajo de investigación es diseñar una metodología que facilite la reutilización de procesos de negocio de forma semi-automática en un entorno de computación en nube. La automatización de este proceso requiere de la utilización de mecanismos eficientes de inteligencia y de un proceso de estandarización. La gestión inteligente se llevará a cabo con una arquitectura distribuida de organizaciones de agente como la presentada en este capítulo. La gestión de la composición de procesos de negocio y la reutilización en general es más eficiente siempre que se aplique a procesos similares y/o que tengan una semántica común. La utilización de una ontología común. El proceso propuesto está guiado por una organización virtual gestionada por una arquitectura multi-agente que implemente el comportamiento inteligente necesario para la gestión del proceso utilizando una ontología.

El objetivo de este trabajo de investigación es presentar un modelo que permita la construcción de un proceso de negocio a partir de las especificaciones en formato texto (con ciertas restricciones) del proceso que estamos interesados en crear. El sistema multiagente basado en organizaciones virtuales tendrá el suficiente nivel de conocimiento e inteligencia para establecer la composición de los procesos utilizando el estándar BPEL (*Business Process Execution Language*). Este estándar permite a su vez componer servicios web de forma sencilla con la ventaja de tener una proyección directa desde diagramas BPMN (*Business Process Management Notation*). A lo largo del próximo tema se revisará el estándar BPEL y todos los elementos relacionados con el mismo. Así como los servicios webs que se utilizarán para la composición de los procesos de negocio.

4 PROCESOS DE NEGOCIO

4.1 INTRODUCCION

El objetivo de este trabajo es desarrollar una metodología que facilite la construcción de procesos de negocio para entornos cloud de forma ágil y eficiente a partir de componentes ya desarrollados. Se trata de facilitar la construcción de nuevos procesos de negocio a partir de otros implementados en forma de servicios web. El proceso está guiado por una organización virtual gestionada por una arquitectura multi-agente que implemente el comportamiento inteligente necesario para la gestión del proceso utilizando una ontología. En el tema anterior se analizaron los sistemas multiagentes que se aplicarán en el modelo y se describió un modelo ontológico. El sistema multiagente basado en organizaciones virtuales facilitará la composición de los procesos utilizando el estándar BPEL, estándar que permita a su vez componer servicios web de forma sencilla con la ventaja de tener una proyección directa desde diagramas BPMN. En este tema se presentan los servicios web y las arquitecturas SEO, el concepto de gestión de proceso de negocio y finalmente se analizarán los servicios web semánticos.

4.2 SERVICIOS WEB

La web está pasando de ser una colección de páginas a una colección de servicios (Paolucci *et al.*, 2002). Durante muchos años, las empresas han interactuado usando aproximaciones ad hoc que aprovechaban la infraestructura básica de Internet. Sin embargo, desde hace unos años los servicios Web han aumentado en importancia, proporcionando un marco de trabajo sistemático y extensible para interacciones aplicación/aplicación. Este marco de trabajo está construido sobre los protocolos Web existentes y está basado en estándares XML. (Curbera *et al.*, 2002). Esta forma de trabajar ha cambiado la forma en que se conciben los sistemas software distribuidos (Newcomer, 2002).

Un servicio Web es un componente software que representa un servicio desplegado en la plataforma Web, y que soporta interacciones automáticas entre máquinas dentro de la red (Walsh, 2002). Además, tienen que estar descritos de forma que puedan ser descubiertos, asociados o compuestos (Le *et al.*, 2009). Para soportar el enfoque de los servicios Web, muchos lenguajes nuevos, la mayoría basados en XML, han sido diseñados como lenguajes de coordinación de negocio (WS-BPEL (OASIS, 2007)), lenguajes de descripción (WSDL (Curbera *et al.*, 2002)) y lenguajes de consultas (XPath (Clark and DeRose, 1999)) (Lapadula *et al.*, 2010). Cuando la mayoría de software y procesos estén soportados por los servicios Web, nuevos tipos de paradigmas de negocio, grupos de discusión, foros interactivos, y modelos de publicación, surgirán para aprovechar esta tecnología (Newcomer, 2002). Además, la composición de servicios Web ha emergido como una aproximación prometedora para la integración de aplicaciones de empresa dentro de límites organizacionales (Zeng *et al.*, 2003).

La computación orientada a servicios ha sido considerada como la nueva generación de computación distribuida, siendo ampliamente adoptada. SOA (*Service Oriented Architecture*) tiene como objetivo la puesta en práctica de la implementación de la computación orientada a servicios, usando los servicios Web como bloque principal de las aplicaciones (Besson *et al.*, 2011). Los tres elementos clave asociados a SOA son: WSDL (Web Service Description Language) que describe el servicio Web, SOAP (Simple Object Access Protocol) que es un protocolo de transporte para el intercambio de información, y UDDI (Universal Description, Discovery and Integration) que es un registro usado para almacenar el servicio, y su descubrimiento (Parimala and Saini, 2011).

Para servicios Web, SOAP ofrece la capacidad de comunicación básica, pero no nos dice qué mensajes se tienen que intercambiar para interactuar con un servicio con éxito. Esto lo completa WSDL, un formato XML desarrollado por IBM y Microsoft para describir los servicios Web como colecciones de puntos de comunicación que permiten intercambiar ciertos mensajes (Curbera *et al.*, 2002). WSDL describe una interfaz de un servicio Web. Consiste en mensajes que se intercambian entre el cliente y el servidor. Dichos mensajes están descritos de forma abstracta, y están ligados a un protocolo concreto de red y formato de mensaje. Las definiciones de servicios Web pueden ser mapeadas a cualquier lenguaje de implementación, plataforma, modelo de objetos, o sistema de mensajes (Tere and Jadhav, 2012). Los sistemas de descubrimiento de servicios Web han sido desarrollados para buscar un servicio Web adecuado a partir de un gran número de servicios Web publicitados (Le *et al.*, 2009). UDDI aparece como la forma de listar y publicar servicios Web aceptada.

Otra de las ventajas de SOA es su estrecha relación con los sistemas cloud. Las aplicaciones en sistemas de nube necesitan ser flexibles, y la adopción de SOA puede proporcionar a los desarrollos basados en Cloud Computing un diseño para el acceso a los servicios a través de un bajo acoplamiento y la habilidad de evolucionar fácilmente que de otro modo sería muy complejo (Arévalo, 2011). Además, se han propuesto arquitecturas cloud basadas en servicios web (SOCCA (*Service-Oriented Cloud Computing Architecture*)) de forma que los sistemas cloud puedan interactuar entre ellos (Wei-tek *et al.*, 2010).

4.3 SERVICIOS WEB SEMÁNTICOS

Un elemento clave para la realización de la Web Semántica es el desarrollo de un lenguaje para la codificación y descripción de dicho contenido Web. Tal lenguaje deberá tener una semántica bien definida, ser suficientemente expresivo para describir las complejas interrelaciones y restricciones entre objetos Web, y ser responsable de la manipulación automática y razonamiento con unos límites aceptables respecto a recursos y tiempo (McIlraith and Martin, 2003).

La tecnología de servicios Web constituye un marco de computación distribuida en la que se proporciona información y servicios bajo demanda, de una forma procesable por una máquina; incluyendo probablemente de un componente software para integrar los resultados proporcionados por diversos servicios (Sycara *et al.*, 2003).

Un servicio web semántico es un servicio web enriquecido con metainformación para facilitar su búsqueda y composición de forma automática. Las tecnologías empleadas en los servicios web semánticos emplean descripciones formales y el razonamiento automático para proporcionar u ofrecer las posibilidades descritas en el apartado anterior. Los servicios web semánticos suponen la integración de la web semántica y los servicios web. Un servicio web semántico extiende el concepto de servicio web, dotándolo de aspectos semánticos que permiten que puedan ser utilizados de forma autónoma por un sistema de información con acceso a la web (García, 2011).

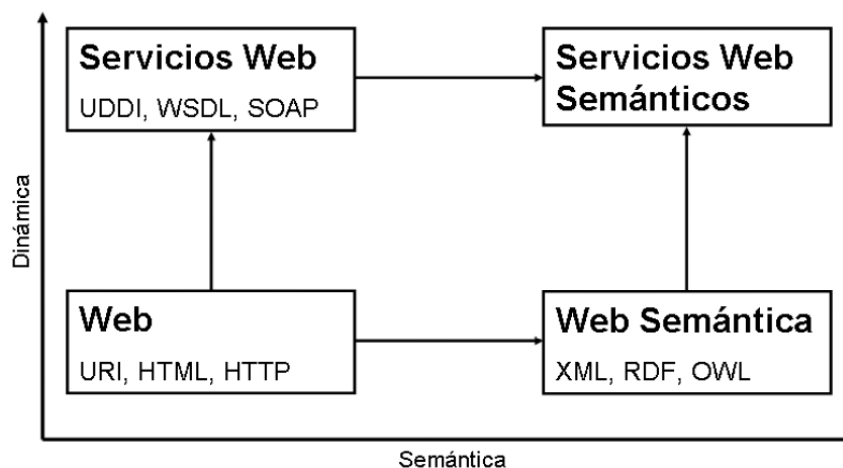


Figura 4.7 - Extensión semántica de los servicios Web

4.4BPM Y BPEL

Un proceso de negocio es una actividad del mundo real que consta de un conjunto de tareas lógicamente relacionadas, que cuando se realizan en la secuencia apropiada y siguiendo las reglas del negocio, producen una salida válida para la empresa (realizar un pago, realizar una extracción de efectivo de una cuenta bancaria, etc.) (Bazán, 2010). En este contexto, la flexibilidad de los sistemas de información se ha convertido en una preocupación importante para los analistas de empresa. De hecho, la constante evolución de los requisitos de una empresa necesita la implementación de un sistema de información flexible y adaptable, que pueda afrontar la modificación de los procesos de negocio (Radgui *et al.*, 2012).

Debido a la intensificación de la globalización, la administración efectiva de los procesos de negocio de una organización ha ido cobrando más importancia paulatinamente. Algunos factores que han provocado esta situación son (Simchi-Lecvi *et al.*, 2000):

- El aumento de la frecuencia de los bienes solicitados.
- La necesidad de que la transferencia de información sea rápida.
- Necesidad de tomar decisiones rápidamente.

- La necesidad de adaptarse al cambio en la demanda.
- Mayor competencia internacional.
- Demanda de ciclos temporales más cortos.

El desarrollo y análisis de procesos de negocio complejos requiere de métodos y herramientas avanzadas (Ligeza *et al.*, 2012). Como forma de gestionar los procesos de negocio surge BPM (Business Process Management) que son los métodos, técnicas y herramientas software, para diseñar, promulgar, controlar y analizar, los procesos operacionales relacionados con las personas, organizaciones, aplicaciones, documentos y otras fuentes de información (van de Aalst, 2003).

El interés de investigadores y profesionales ha crecido rápidamente sobre este tema. Una vasta variedad de paradigmas y metodologías procedentes de: teoría de control de organizaciones, ciencias de la computación, matemáticas, lingüística, semiótica, y filosofía, han sido adoptadas, haciendo de BPM un área multidisciplinar (Ko *et al.*, 2009). Sin embargo, este crecimiento sin precedentes se ha convertido en una espada de doble filo, derivando en que muchas de las tecnologías y terminologías BPM no están bien definidas, haciendo que muchos profesionales e investigadores que las usan no las terminan de entender (Havey, 2005) (Hill *et al.*, 2008). Todo ello ha ocasionado que los nuevos lenguajes y notaciones propuestos, habitualmente contengan características duplicadas para conceptos similares (Mendling y Neumann, 2005). Para modelar los procesos de negocio según BPM se suele utilizar el estándar gráfico BPMN (Business Process Management Notation) debido a su sencillez y potencia.

Un área intrínsecamente relacionado con BPM es la correspondiente a los servicios web y la arquitectura orientada a servicios. Los Servicios Web son una nueva generación de aplicaciones Web. Son componentes software autocontenidos, autodescriptivos y modulares, que pueden ser accedidos, localizados e invocados desde cualquier lugar en Internet. En este marco, aparece BPEL (*Business Process Execution Language*) como un estándar de facto para la ejecución de procesos de negocio que permite la composición e integración de diversos servicios web en un servicio por sí mismo (Viroli *et al.*, 2007). A medida que el uso de los servicios web aumenta, también lo hace la elección por parte de las empresas de "*Business Process Execution Language*" (BPEL), para el modelado de procesos de negocio dentro de la arquitectura de servicios web. Mediante el desarrollo de servicios web con BPEL, las empresas pueden implementar aspectos de la arquitectura orientada a servicios que previamente hubiesen sido difíciles de conseguir (Pasley, 2005).

Al comparar BPEL con estándares similares (XPDL y WSCI), se concluye que tiene una buena expresividad (Wohed, 2003) y actualmente es el único estándar que tiene motores de ejecución tales como Oracle BPEL Manager (2012), IBM BPEL4WS (2012) y ActiveBPEL (2012) (Morrison y Nugrahanto, 2007).

Como resultado, BPEL se ha implementado exitosamente en áreas muy dispares: software para el soporte de decisiones de diagnóstico (Morrison y Nugrahanto, 2007), modelado de aplicaciones clínicas (Morrison *et al.*, 2006), sistemas multiagente (Viroli *et al.*, 2007) etc. y su implementación en el ámbito empresarial sigue creciendo (2012).

4.5 CONCLUSIONES

Una buena gestión de los procesos de negocios es un aval para el buen desarrollo de una empresa. El análisis, basado en los objetivos, alcance y limitaciones, realizado a los estándares, metodologías y modelos para llevar a cabo el enfoque por procesos y la gestión de estos, demuestra que en muchos casos procesos de negocio diferentes incluyen elementos o componentes similares o exactamente igual. En especial si estos están creados a base de servicios web definidos en el marco de una ontología y con una semántica similar es relativamente sencillo identificar hasta que punto eso sucede sin un análisis excesivamente exhaustivo. El desarrollo y la integración de aplicaciones empresariales y de procesos de negocio requieren una metodología y modelo bien estructurada, para ser eficientes. En este capítulo se han mostrado los procesos de negocio como herramientas útiles, eficiente y metodológicamente probadas a la hora de implementar aplicaciones informáticas. Estos procesos pueden construirse con una tecnología tan sólida como la que aportan los servicios web. En este contexto parece necesaria la aplicación de una ontología y una semántica cuidada, para que el proceso de reutilización y composición de nuevos procesos se pueda realizar de forma ágil y eficiente. El siguiente capítulo se presenta un proceso disciplinado y sistemático para crear nuevos procesos de negocio de forma semiautomática, a partir de los ya existentes en un entorno de programación en nube.

5 IPCASCI

5.1 INTRODUCCIÓN

En este capítulo se presenta el modelo de construcción de procesos de negocio IPCASCI (*Intelligent business Processes Composition based on MAS, Semantic and Cloud Integration*). La industria del desarrollo de software requiere de la construcción rápida de nuevos productos que se adapten a las necesidades que surgen de un mercado en constante cambio. En este contexto, y como método para la reutilización de componentes software se presenta un nuevo modelo o metodología que facilita la reutilización de servicios Web en entornos cloud para componer procesos de negocio. Aquí se presenta una propuesta de arquitectura que, basándose en la tecnología de servicios web, permita: (i) Descubrir automáticamente servicios Web. (ii) Dotar de una descripción semántica a los servicios Web. (iii) Componer automáticamente servicios Web para generar nuevos servicios y (iv) Invocar automáticamente servicios Web.

Todo ello, se hace de forma que el proceso de automatización de la construcción de nuevos servicios, sea eficiente y tenga asociado un comportamiento “inteligente.” Con comportamiento “inteligente” nos referimos a que a partir de una entrada de requisitos realizada informalmente (textualmente), el sistema sea capaz de: (i) Analizar dicha entrada (ii) Encontrar los servicios que permitan cumplir los requisitos. (iii) Realizar una composición automática de los servicios Web y de los correspondientes procesos de negocio que definen. Se obtendrá como resultado nuevos servicios Web que implementan los requisitos aportados por el cliente, en un proceso automático de descubrimiento y composición. Existen diversas aproximaciones para la implementación de plataformas basadas en servicios Web semánticos. En (Su *et al.*, 2012) se presenta una solución basada en lógica difusa para el descubrimiento de servicios Web semánticos, en (Sycara *et al.*, 2003) se propone una solución basada en agentes y el lenguaje ontológico DAML-S, en (García *et al.*, 2012) se utiliza una propuesta basada en consultas SPARQL y el lenguaje ontológico OWL-S. En general, la mayor parte de las propuestas de arquitecturas de servicios Web semánticos se basan en el lenguaje OWL-S. La propuesta que se realiza en este documento difiere de las propuestas realizadas hasta el momento en que:

- Se diseña una plataforma global, embebida en un entorno cloud, cuya estructura está específicamente pensada para proporcionar una ejecución rápida y eficiente.
- La información semántica de los servicios Web es independiente de la construcción interna de las ontologías, lo que permitirá reutilizar cualquier ontología existente, independientemente del formato en el que estén construidas.
- Se cuenta con un sistema multi-agente basado en organizaciones virtuales que facilita el proceso de descubrimiento y el de análisis de los requisitos.
- Se construye el servicio Web solución a partir de una definición informal realizada por el cliente.

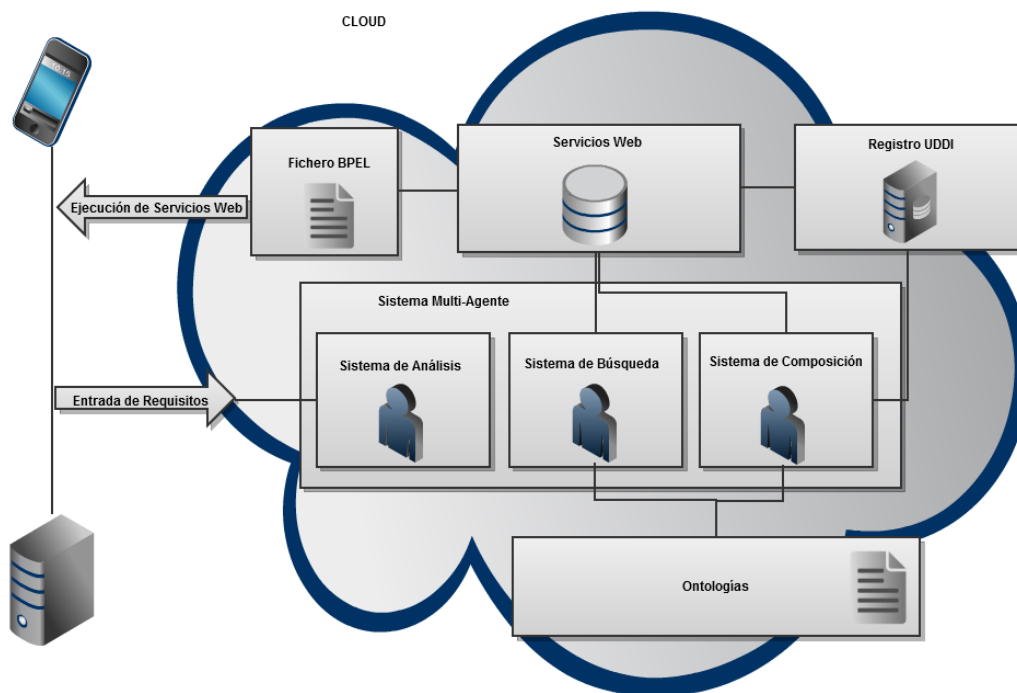


Figura 5.1 – Descripción de la plataforma

En la Figura 5.1 se muestra mediante un diagrama conceptual los diferentes elementos de los que se compone la plataforma:

- Sistema Cloud: Sistema que sigue el paradigma de computación en la nube en donde se incluyen los diferentes elementos de la plataforma proporcionando un entorno de ejecución y almacenamiento
- Servicios Web: Diferentes servicios Web registrados en la plataforma que se podrán utilizar para componer nuevos servicios Web.
- Registro UDDI: Sistema de registro en donde se encontrarán registrados los diferentes servicios Web de la plataforma.
- Sistema Multi-Agente basado en organizaciones virtuales: Aporta la funcionalidad que permitirá realizar el descubrimiento y composición de los servicios Web.
 - *Sistema de Análisis*: Analiza el contenido semántico introducido por el usuario y lo estructura de forma que sea analizable computacionalmente.
 - *Sistema de Búsqueda*: Se encarga de descubrir los servicios web que se ajustan a las restricciones tanto semánticas como de formato introducidas por el usuario.
 - *Sistema de Composición*: Una vez que se han determinado los servicios Web que cumplen los requisitos semánticos del usuario y sus relaciones, se pondrán de forma que obtenga un nuevo servicio Web.
- Ontologías: Diferentes ontologías que modelan el conocimiento semántico que podrá estar incluido en los servicios Web.

- Fichero BPEL: Composición de los servicios Web que cumplen los requisitos marcados por el usuario.

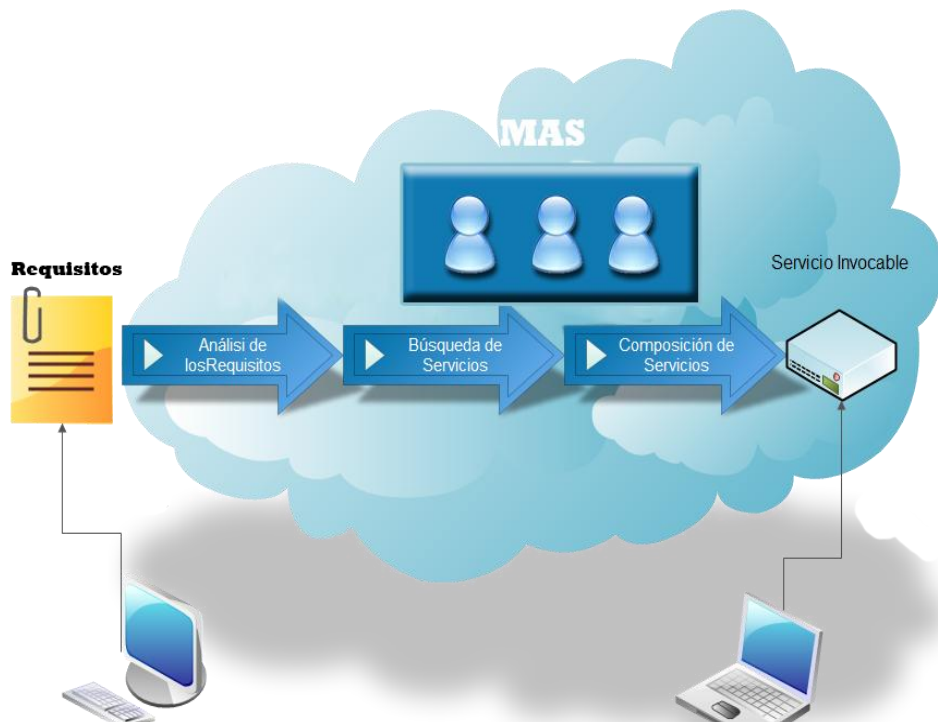


Figura 5.2 - Proceso de obtención del servicio solución

El proceso para generar un servicio *web* es el siguiente:

- El usuario introduce los requisitos del servicio que desea obtener mediante un sistema asistido (el sistema ofrecerá al usuario la posibilidad definir una serie de términos representativos de los requisitos del sistema – la lista de términos estará limitada por el repositorio de servicios *web* de la base de datos). El resultado será un conjunto de módulos relacionados, de tal forma que para cada módulo se tendrá la siguiente información:
 - Entradas del módulo.
 - Salida que produce.
 - Proceso realizado en el módulo (semántica).
 - Dominio del concepto semántica (*dominio ontológico*) que representa el módulo.
 - Precondición para su ejecución.
 - Relación con los módulos anteriores.
- Para cada módulo de la fase anterior, se realizará una búsqueda de los servicios Web que implementan los requisitos del módulo (la búsqueda será, además, por forma de invocación y contenido semántico).
- Se construirá un diagrama BPD reducido (no expresará todas las capacidades de modelado del estándar BPMN) en el que cada servicio Web se representa

mediante una actividad y se reflejan todas las interacciones producidas entre los diferentes servicios Web.

- Se presenta el diagrama BPD al usuario, puede darse el caso que para una actividad existan varios servicios que implementen su funcionalidad. En este caso, se seleccionará uno por defecto pero se le permitirá al usuario seleccionar el servicio que quiera dentro de los disponibles.
- A partir del diagrama BPD se realizará una composición BPEL de forma que el servicio solicitado por el usuario se pueda invocar directamente.

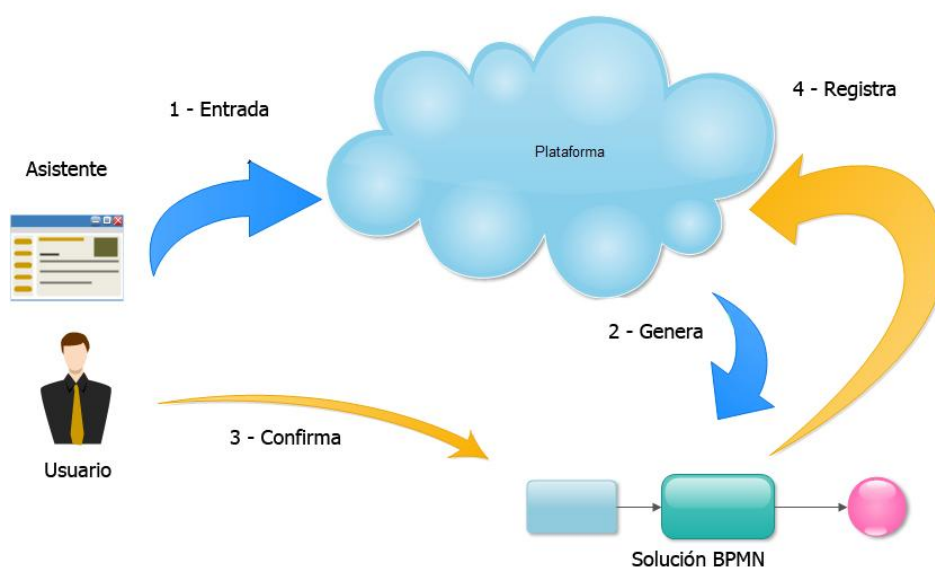


Figura 5.3- Proceso de obtención del servicio

5.2 CONCLUSIONES

En este capítulo, se ha presentado el proceso de construcción de una plataforma (IPCASCI) sobre un entorno *cloud*, la cual está orientada a crear nuevos servicios de negocios de forma semiautomática a partir de los servicios web existentes en dicha plataforma. En este sentido, la plataforma tiene como entrada un conjunto de especificaciones (que son convertidas a un formato XML) introducidas por el usuario, que son posteriormente convertidas, siguiendo el estándar BPMW, en un conjunto de módulos con información computable. Esta información es empleada para construir los agentes asociados a cada módulo, que se encargarán de representar la información introducida, formando un sistema multi-agente basado en organizaciones virtuales. Por tanto, los agentes se encargan de buscar los servicios web que responden a cada módulo y a través de un proceso de composición de servicios web, se crea un nuevo servicio de negocio como solución a las especificaciones introducidas por el usuario.

Como contribución principal de este modelo podemos destacar su implantación en un entorno *cloud*, trayendo consigo todas las ventajas y prestaciones que brinda este

paradigma (vea el Capítulo 2). Esto Permite a los usuarios, un mejor acceso y uso de servicios en la plataforma desde fuentes heterogéneas y sin costes para la infraestructura. Asimismo, la composición automática de servicios *web* para generar nuevos servicios (debido a la reutilización de componentes de software) y de acuerdo a las especificaciones del usuario, nos garantiza una autonomía en la plataforma, lo cual es deseable en cualquier sistema implantado por parte de una empresa. Consecuentemente, las dos contribuciones anteriores son la base de nuestra arquitectura, arrojando gran ventaja frente a otros enfoques.

Finalmente, a partir de la descripción exhaustiva de los componentes de la arquitectura propuesta, se han alcanzado los objetivos propuestos en el Capítulo 1 de esta memoria, por lo que en el capítulo siguiente, evaluaremos nuestra propuesta sobre un caso de estudio y analizaremos sus resultados, cerrando así, nuestra propuesta global del modelo IPCASCI.

6 RESULTADOS Y CONCLUSIONES

6.1 CASO DE ESTUDIO

En este apartado, vamos a presentar una implementación de nuestra propuesta (el modelo IPCASCI) sobre un caso de estudio práctico. El objetivo de este caso de estudio es materializar nuestra propuesta en un entorno real y de esta manera, desarrollar una herramienta que represente el caso de estudio. En este contexto, evaluaremos un conjunto de parámetros de la herramienta, tanto internos como externos, para alcanzar unos resultados que nos darán una visión general de la bondad de nuestra propuesta. De aquí que, podamos empezar por definir nuestro caso de estudio:

Se desea construir un servicio web para realizar reservas de libros en una biblioteca. El servicio web tendrá como entrada, el identificador del usuario que realiza la solicitud y del libro que solicita. El servicio web comprobará que el usuario está registrado en el sistema y que no le ha caducado la suscripción. Si todo es correcto, se procederá a comprobar la existencia de ejemplares disponibles del libro y que el usuario no ha sobrepasado su cuota de préstamos. Si todo es correcto, entonces el préstamo es registrado. En caso contrario, se enviará un correo electrónico al usuario indicando que no se ha podido realizar la solicitud y el motivo de la denegación.

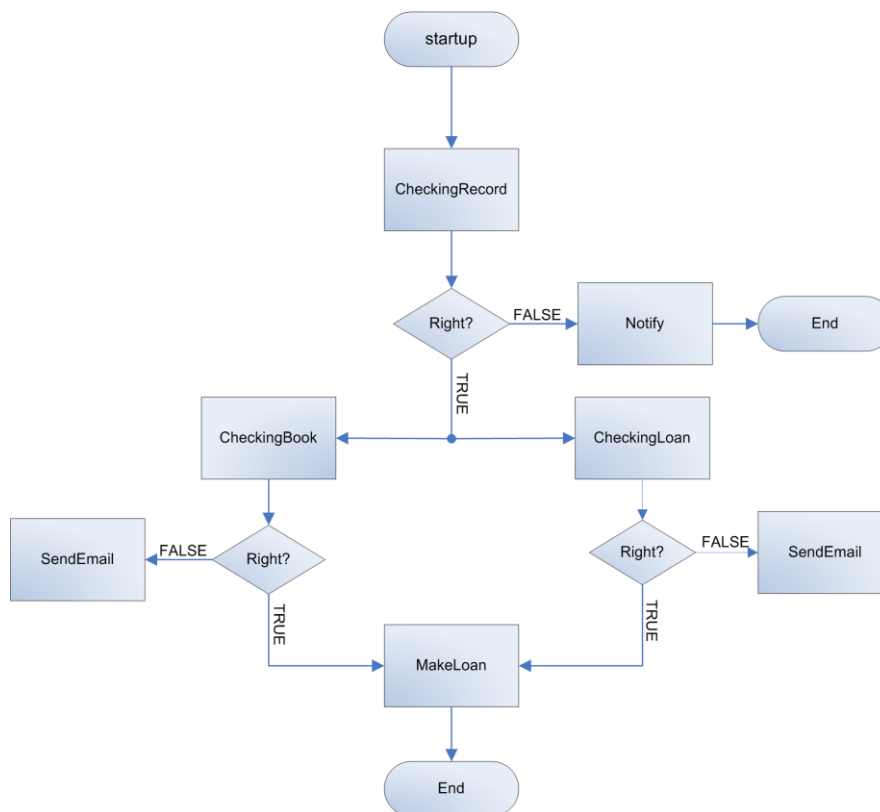


Figura 6.1: Diagrama de flujo de las especificaciones del caso de estudio que serán introducidas por el usuario.

Una representación del escenario propuesto en el caso de estudio, puede ser implementada mediante el diagrama de flujo en la Figura 6.1 (los componentes de este diagrama serán explicados más adelante). Nótese que este diagrama representa las especificaciones del usuario (a través de los módulos en forma de recuadros y sus relaciones, en forma de flechas) para construir el nuevo servicio *web*. De aquí que, en adelante, daremos las características de los subprocesos que intervienen en la herramienta para convertir las especificaciones del usuario en procesos y relaciones, que están dirigidas a crear el nuevo servicio *web* basándose en la composición de servicios existentes.

6.2 Evaluación del caso de estudio

En este apartado evaluaremos la herramienta *LibraryBookReserve* construida a partir del planteamiento de nuestro caso de estudio. La evaluación se llevará a cabo, siguiendo algunas de las pautas de la ingeniería del software, considerando cualidades internas, así como, externas. En este sentido, el logro de cualidades internas en un *software* permitirá una labor sencilla, ordenada y eficiente a los desarrolladores del sistema. Mientras que la cualidades externas evalúan los factores esperables y útiles (o aprovechables) por los usuarios. Partiendo entonces, de la clasificación anterior de las cualidades de un *software*, se tiene que la cualidad interna está asociada a la *prueba de caja blanca* (Pressman, 2005), que no es más que un examen de los detalles procedimentales del *software*. Así pues, se comprueban los caminos lógicos del *software*, suministrando casos de pruebas que ejecuten conjuntos específicos de condiciones o bucles. Además, se puede verificar el estado de la aplicación en varios puntos con el objetivo de determinar si el estado real coincide con el esperado. Por otra parte, la cualidad externa de un *software* está asociada a la *prueba de caja negra* (Pressman, 2005), que se refiere a las pruebas realizadas sobre la interfaz de la aplicación. O sea, los casos de prueba pretenden demostrar que las funciones de la aplicación son operativas y que se produce un resultado correcto, además de la integridad de la información externa. Una prueba de caja negra examina aspectos del modelo del sistema sin tener mucho en cuenta la estructura lógica interna del *software*.

En el caso particular de la evaluación de nuestra herramienta, utilizaremos la *prueba del camino básico* como prueba de caja blanca. Esta prueba se introduce para evaluar la lógica del diagrama de flujo seguido por la composición de servicios en la herramienta, por lo tanto, medirá la correctitud del nuevo servicio *web* creado de forma automática. Por otra parte, como prueba de caja negra, emplearemos un sistema de encuesta a usuarios que cubrirá atributos del software que están clasificados en las métricas de calidad ISO 9126 (Pressman, 2005). Ambos tipos de pruebas, se hicieron sobre los resultados obtenidos para 12 usuarios que crearon el servicio web (usando la herramienta) a partir de la composición de los servicios existentes, como explicado en la definición del caso de estudio. Cada uno de estos usuarios cumplimentaron también, la encuesta sobre la calidad de la herramienta.

6.3 Análisis del caso de estudio

En este capítulo hemos introducido un caso de estudio para mostrar una implementación de nuestra propuesta, *el modelo de construcción de procesos de negocios IPCASCI* (Capítulo 5). Como primer resultado a partir del caso de estudio, hemos construido una herramienta (*LibraryBookReserve*) que crea un servicio *web* (de forma semi-automático) para realizar reservas de libros en una biblioteca, para lo cual, nos basamos en la composición de los servicios existentes en nuestra arquitectura. Este resultado, la herramienta *LibraryBookReserve*, materializa también, nuestro modelo IPCASCI. Una vez construida nuestra herramienta, hemos introducido un proceso para su validación, evaluando tanto características internas (prueba de caja blanca) como externas (prueba de caja negra). Las pruebas de caja blanca, también conocida como *pruebas a pequeña escala*, se centran en la estructura de control de la aplicación, en nuestro caso, el diagrama de flujo a ejecutar por la herramienta en el proceso de la composición de servicios (diagrama BPMN). Las pruebas de caja negra, conocidas también como *prueba a gran escala*, amplían el enfoque y son diseñadas para validar los requisitos funcionales sin fijarse en el funcionamiento interno de una aplicación. En nuestro caso, el sistema de encuesta a 12 usuarios de la herramienta, para evaluar la calidad del producto final de nuestro modelo, *LibraryBookReserve*. Los resultados en este estudio, arrojarán que nuestra herramienta, en sentido general, se ajusta a los requerimientos del caso de estudio propuesto y que el servicio *web* construido a partir del diagrama BPMN (composición de servicios) se ajusta a las especificaciones introducidas por el usuario. Esto también demuestra la bondad así como la fiabilidad del diseño y de la estructura lógica llevada a cabo por las organizaciones virtuales multi-agente (introducidas en nuestra arquitectura), que están orientadas al descubrimiento/refinamiento de los servicios *web* que tendrán lugar en la composición de servicios. Además, como parte final, los resultados de la encuesta, también demostraron, que la herramienta cumple gran parte de los requisitos esperados en una aplicación orientada a servicios *web*.

6.4 Conclusiones generales

Tradicionalmente, el desarrollo de software se encuentra asociado a una serie de problemas, siendo los más destacados los retrasos en la fecha de entrega y el aumento descontrolado del coste del producto. Como solución a estos problemas, existen técnicas y herramientas pertenecientes al área de Ingeniería del Software, que permiten organizar, gestionar, planificar y asegurar la calidad, del proceso de desarrollo. Al usar estas técnicas, el riesgo de que el coste del proyecto se descontrole se reduce considerablemente. Sin embargo, llevar a cabo un modelo de proceso en el que se planifique el proyecto al completo (incluyendo análisis de costes, planificación temporal, ciclo de vida de desarrollo, y una documentación técnica extensa), es una tarea que lleva mucho tiempo y trabajo, por lo que es necesario valorar si su realización compensa su coste.

En contrapartida a un modelo de desarrollo muy controlado, planificado y documentado, existen metodologías que agilizan este proceso, haciendo que el ciclo de

desarrollo se reduzca en gran medida, pero sin contar con las ventajas que ofrece una documentación técnica completa.

La situación actual del mundo empresarial requiere que en muchas ocasiones sea necesario desarrollar software de forma rápida y a un coste reducido. Aún con metodologías ágiles, el coste del desarrollo de nuevos productos software sigue siendo alto para pequeñas y medianas empresas. Además, debido al rápido y constante cambio de la situación del mercado, se hace evidente la necesidad de nuevos productos software en un tiempo muy corto.

En este documento se presenta una propuesta de modelo que se adapta al entorno tecnológico del momento y que es accesible y útil para el mundo empresarial. Las principales características de esta propuesta son:

- Permite crear software a partir de componentes ya implementados.
- Es posible el acceso desde cualquier lugar y mediante cualquier plataforma.
- Uso rápido y sencillo.
- La creación del nuevo producto software se realiza de forma automática.
- Coste reducido y controlado.

Para cumplir estas características se ha optado por un modelo en la que los componentes software reutilizables son los servicios *web*. Además, el acceso y almacenamiento se realizará mediante un sistema en la nube, haciendo de esto una opción accesible para pequeñas empresas (ya que no tienen que adquirir grandes sistemas de almacenamiento), y un buen rendimiento.

Sin embargo, el principal atractivo de esta propuesta reside en la automatización de todo el proceso de construcción del software. El cliente solamente deberá introducir la especificación del software que desea construir (de forma sencilla y que requiera poca formación), siendo el modelo el encargado de construir el producto como un servicio web, a partir de los servicios que ya se encuentren almacenados en el sistema. Para llevar a cabo esta tarea, el sistema contará con un repositorio de servicios web, unas ontologías que permitan modelar el conocimiento sobre su funcionamiento, y un sistema multi-agente que, de forma inteligente, sea capaz de componer los servicios almacenados para satisfacer las necesidades del cliente.

A diferencia de las propuestas basadas en OWL-S, el modelo propuesto se basa en la asociación semántica de los servicios web con WSDL-S, y un sistema multi-agente encargado de realizarlas tareas de descubrimiento y composición.

Todo ello, desemboca en un sistema que se adapta a las necesidades actuales de desarrollo software, en las que es posible obtener nuevos productos en muy poco tiempo, a un coste reducido, y sin pasar por un proceso de desarrollo arduo y externo.

Resumiendo, se propone modelo sencillao e intuitivo, que permita componer servicios web a partir de otros servicios, usable por clientes sin conocimientos técnicos y de coste de uso relativamente bajo.

Alguna de las características más relevantes de esta propuesta son:

- Pago por uso: Debido a que el modelo se integra en un entorno Cloud, los clientes pagarán solamente por el uso que realicen de ella, abaratando los costes de su uso y haciéndola más accesible a pequeñas y medianas empresas.
- Escalabilidad respecto al tamaño: Al contarse con un Cloud, el sistema puede crecer respecto a la cantidad de servicios reutilizables almacenados sin que ello acarree un coste importante en componentes hardware. Además, se contará con un sistema de almacenamiento amplio, relevando de la responsabilidad a las empresas de adquirir grandes sistemas de almacenamiento.
- Escalabilidad respecto a las áreas de aplicación: Debido a la estructuración de las ontologías y el sistema multi-agente, se pueden añadir nuevas áreas de aplicación sin que ello conlleve mucho trabajo.
- Escalabilidad respecto a la funcionalidad: El sistema multi-agente se diseñará de forma que se pueda ampliar la funcionalidad proporcionada. Además, los servicios web mediante los cuales se accede al modelo, son extensibles.
- Ubicuidad: Se podrá acceder a al sistema desde cualquier lugar ya que la ejecución se realiza vía servicios web.
- Ejecución multiplataforma: El uso del sistema y su acceso se realiza mediante una máquina BPEL, que es un estándar aceptado por la mayoría de los principales fabricantes.
- Orientación a negocio: La ejecución se realiza mediante una máquina BPEL lo que permite que se adapte al proceso de negocio (BPM) específico de cada empresa.
- Funcionalidad reutilizable: Al usar servicios web ya implementados, y reutilizarlos para la composición de unos nuevos, se disminuye el tiempo necesario para obtener un producto que satisfaga las necesidades del cliente.
- Control de la calidad: Los servicios proporcionados se ajustan al proceso de negocio específico de la empresa dentro de un marco BPM. Esto, a su vez, permite que dicho proceso se englobe en estándares de calidad.
- Eficiencia energética: El consumo energético se ve reducido al estar el sistema empotrado en un sistema cloud, lo que además de abaratar costes, es una solución ecológicamente mejor que mantener servidores locales.
- Facilidad de uso: La plataforma que implemente el modelo estará diseñada para que su uso sea sencillo y de fácil aprendizaje. De esta forma, se podrán obtener resultados con muy poco esfuerzo. El manejo de la plataforma se realizará con interfaces gráficas intuitivas.
- Usable por usuarios no técnicos. La introducción de las especificaciones que debe cumplir la solución se realizará mediante un asistente que guiará al usuario en todo el proceso. Además, la modificación de la solución propuesta se realizará mediante un diagrama BPD (Business Process Diagram) perteneciente al estándar BPMN (Business Process Management Notation), que es relativamente sencillo y no requiere de un usuario con conocimientos informáticos avanzados. Todo ello

desemboca en que prácticamente cualquier persona pueda utilizar la plataforma con tan solo un pequeño periodo de formación.

7 REFERENCIAS

- Akkiraju R., Farrell J., Miller J., Nagarajan M., Schmidt M., Sheth A., Verma K. Web Service Semantics - WSDL-S (November 2005)
- Arkin A., Askary S., Bloch B., Curbera F., Goland Y., Kartha N., LiuK., Thatte S., Yendluri P. and Yiu A. Web Services Business Process Execution Language Version 2.0. OASIS (December, 2004).
- Bajo J., Zato C., de la Prieta F., de Luis A. y Tapia D. (2010): Cloud Computing in Bioinformatics. DCAI 2010: 147-155.
- Blanco S. (2004) Anotaciones semánticas en WebQuest. Tesis Doctoral, Universidad de Valladolid.
- Cao B., Li B., Xia Q. (2009). A Service-Oriented Qos-Assured and Multi-Agent Cloud Computing Architecture.
- DAML-S. DAML-S 0.7 Draft Release.(Accedido en 2012) <http://www.daml.org/services/daml-s/0.7/>
- García JA. IRIS & ME OWL-S Sistema de Recomendación en Dispositivo Móvil con OWL-S. Universidad de Salamanca, (2011)
- García JM., Ruiz D. and Ruiz-Cortés(2012). Improving semantic web services discovery using SPARQL-based repository filtering. Web Semantics: Science, Services and Agents on the World Wide Web Volume 17, December 2012, Pages 12–24
- GuideToBPEL. The BPEL Language. Obtenido de <http://www.radikalfx.com/bpel/language.html>. (Accedido en 2012).
- Huhn M., Kowalczyk, Ryszard, Maamar, Zakaria, Unland, Rainer. IOS Press; 2012. Special issue: development of service-based and agent-based computing systems.
- IBM (2008). Accedido en 2012. RESTful Web services: The basics. <http://www.ibm.com/developerworks/webservices/library/ws-restful/>.
- IBM. (2001) Understanding WSDL in a UDDI registry, Part 1. <http://www.ibm.com/developerworks/webservices/library/ws-wsdl/>
- Jan R. & Jan M. (2006) On the Translation between BPMN and BPEL: Conceptual Mismatch between Process Modeling Languages. In Latour, Thibaud & Petit, Michael (Eds.) The 18th International Conference on Advanced Information Systems Engineering. Proceedings of Workshops and Doctoral Consortium, Namur University Press, pp. 521-532.
- McCabe T. (1976) A Software Complexity Measure. IEEE Trans. Software Engineering, (SE-2): 308–320.
- Miller J., Verma K., Rajasekaran P, Sheth A., Aggarwal R., Sivashanmugam K. WSDL-S: Adding Semantics to WSDL - White Paper. (2004)
- Olson, Mike; Ogbuji, Uche (July 3, 2002). "The Python Web services developer: Messaging technologies compared". IBM developerWorks. Retrieved 2011-02-01.
- Ouvans C., Dumas M., Hofstede A., van der Aalst W. From BPMN Process Models to BPEL Web Services, (2006)
- Ouyang, C., van der Aalst W. Marlon D., & Hofstede A. (2006) Translating BPMN to BPEL.
- Pinzon C., Bajo J.,De Paz JF. And Corchado JM.: S-MAS: An adaptive hierarchical distributed multi-agent architecture for blocking malicious SOAP messages within Web Services environments. Expert Syst. Appl. 38(5): 5486-5499 (2011)
- Pressman R. (2005). Software Engineering: A Practitioner's Approach. Seventh Edition, Ph.D., McGraw-Hill Companies.
- Rodríguez S., Tapia D., Sanz E., Zato C., de la Prieta F. and Gil O. (2010). Cloud Computing Integrated into Service-Oriented Multi-Agent Architecture.
- Srinivasan N., Paolucci M. and Katia Sycara. Adding OWL-S to UDDI, implementation and throughput (2004)

- Su Z., Chen H., Zhu L., and Zeng Y. (2012) Framework of Semantic Web Service Discovery Based on Fuzzy Logic and Multi-phase Matching. *Journal of Information & Computational Science* 9: 1 (2012) 203–214
- Sycara K., Paolucci M., Ankolekar A. and Srinivasan N. (2003) Automated discovery, interaction and composition of Semantic Web services.
- Sycara K., Paolucci M., Ankolekar A. and Srinivasan N. Automated discovery, interaction and composition of Semantic Web services. (2003)
- Tapia D., Rodríguez S., Bajo J. and Corchado JM.: FUSION@, A SOA-Based Multi-agent Architecture. *DCAI 2008*: 99-107
- Tekli, J.M. Damiani, E. ; Chbeir, R. ; Gianini, G. SOAP Processing Performance and Enhancement Services Computing, *IEEE Transactions on* Volume: 5, Issue: 3 Page(s): 387 – 403 (2012)
- W3C OWL-S (Accedido en 2012) <http://www.w3.org/Submission/OWL-S/#5>.
- Wei R., Qiao L., Yang Z. A message interaction security mechanism based on SOA . *Systems and Informatics (ICSAI), 2012 International Conference on* Page(s): 1503 – 1506 (2012)
- White S. Using BPMN to Model a BPEL Process. (2005).
- Abdelkader A. Bakhta N. and Abdelkader O.M. Multi-Agents Model for Web-based Collaborative Decision Support Systems.
- Abu- Rahmeh O., Johnson P., and A. Taleb-Bendiab, (2008). A Dynamic Biased Random Sampling Scheme for Scalable and Reliable Grid Networks, *INFOCOMP - Journal of Computer Science*, ISSN 1807-4545, 2008, VOL.7, N.4, December, pp. 01-10.
- Adobbati, R., Marshall, A. N., Scholer, A., Tejada, S., Kaminka, G. A., Schaffer, S., and Sollitto, C. (2001) Gamebots: A 3D virtual world test-bed for multi-agent research. In *Proceedings of the second international workshop on Infrastructure for agents, MAS, and Scalable MAS*, Montreal, Canada.
- Agrawal R. et al. (2008). "The Claremont report on database research". *SIGMOD Record (ACM)* 37 (3): 9–19. doi:10.1145/1462571.1462573. ISSN 0163-5808.
- Akkiraju R., Farrel J., Miller J., Nagarajan M., Schmidt M., Sheth A. and Verma K. "Web Service Semantics - WSDL-S," A joint UGA-IBM Technical Note, version 1.0, April 18, 2005. <http://lsdis.cs.uga.edu/projects/METEOR-S/WSDL-S>
- Akkiraju R., Goodwin R., Doshi P., y Roeder S. (2003) A Method for Semantically Enhancing the Service Discovery Capabilities of UDDI.
- AlBreshne A., Fuhrer P. and Pasquier J. (September, 2009). *Web Services Technologies: State of the Art Definitions, Standards, Case Study*.
- Ali R., Bryl V., Cabri G., Cossentino M., Dalpiaz F., Giorgini P., Molesini A., Omicini R., Puviani M. and Seidita V. (2008) MENSA Project - Methodologies for the Engineering of complex Software systms: Agent-based approach, Tech. Rep. 1.2, UniTn.
- Alizadeh, K., Seyyedi, M., Mohsenzadeh, M., 2012. A Service Identification Method Based on Enterprise Ontology In Service Oriented Architecture. *International Journal of Information Processing and Management (IJIPM)* Volume3, Number2. April 2012.
- Andrews, T., Curbera, F., Dholakia, H., Golland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D. and Thatte, S. (2003). *Business Process Execution Language for Web Services, Version 1.1. Specification*, BEA Systems, IBM Corp., Microsoft Corp., SAP AG, Siebel Systems, New York, NY.
- Anton A. (1996) *Goal-Based Requirements Analysis*.
- Antonopoulos N., Anjum A. and Gillam L. (2012) Intelligent techniques and architectures for autonomic clouds: introduction to the itaac special issue *Journal of Cloud Computing: Advances, Systems and Applications* 2012, 1:18 (23 August 2012)
- Arévalo J.M. (2010) *Cloud Computing: fundamentos, diseño y arquitectura aplicados a un caso de estudio*.
- Argente E., Giret A., Valero S., Julian V. and Botti V. (2004) Survey of MAS Methods and Platforms focusing on organizational concepts. In: Vitria, J, Radeva, p and Aguilo, I (ed) *Recent Advances in Artificial Intelligence Research and Development, Frontiers in Artificial Intelligence and Applications*: 309–316
- Argente E., Julian V. and Botti V. (2006) Multi-agent system development based on organizations. *Electronic Notes in Theoretical Computer Science*, 150:55–71.
- Arkin, A., Askary, S., Bloch, B., et al. (eds.): *Web Services Business Process Execution Language Version 2.0. Committee Draft*, September 2005. Disponible en:

<http://www.oasis-open.org/committees/download.php/14616/wsbpel-specification-draft.htm>

- Armbrust M et al. (2010). "A View of Cloud Computing". *Communications of the ACM*, 53 (4): 50-58
- Arsanjani, A. (2004). Service-oriented modeling and architecture. How to identify, specify, and realize services for your SOA. IBM.
- Artikis A., Kaponis D. and Pitt J. (2009) Multi- Agent Systems: Semantics and Dynamics of Organisational Models, chapter in *Dynamic Specifications of Norm-Governed Systems*. IGI Globa, 2009.
- AWS (December, 2010) Overview of Amazon Web Services.
- B. Chandrasekaran and John R. Josephson. (1999). Ohio State University .V. Richard Benjamins, University of Amsterdam . What Are Ontologies, and Why Do We Need Them?
- Babak B., Jaferian P., and Reza M. (2006). Comparison of Global Computing with Grid Computing.
- Bajo J., Julian V., Corchado J.M., Carrascosa C., de Paz, Y. Botti V. and de Paz J. F. (2008) An Execution Time Planner for the ARTIS Agent Architecture. *Engineering Applications of Artificial Intelligence* Volume: 21 Issue: 5 Pages: 769-784 Published: AUG 2008
- Basili, V. R., Briand, L. C. y Melo, W. L. (1996). Assessing the Impact of Reuse on Quality and Productivity in Object-Oriented Systems. *Communication of ACM*.
- Baumer G., Breugst M., Choy S. and Magedanz T. (2000) Grasshopper: A Universal Agent Platform based on OMG MASIF and FIPA standars, in: *Agents Technology in Europe*.
- Bazán P. (2010). BPEL: una propuesta para el uso de Web Services. Universidad de la Plata.
- Ben van Eyle. (2001). Web Services – A Business Perspective on Platform Choice. www.theserverside.com.
- Berners-Lee, T., Hendler, J., Lassila, O. (2002) The Semantic Web. *Scientific American*.
- Besson F., Leal P., Kon F., Goldman A., and Milojicic D. (2011) Supporting. Test-Driven Development of Web Service Choreographies. *ACM*.
- Bizagi. Process Modeler BPMN Business Process Modeling Notation.
- Boella G., Hulstijn J., and Van Der Torre L. (2005) Virtual organizations as normative multiagent systems," in *HICSS IEEE Computer Society*.
- Boissier O. and Gateau B. (2007) Normative multi-agent organizations: Modeling, support and control. In *Normative Multiagent Systems*,
- Booth D., Champion M, Ferris C., McCabe F., Newcomer E. and OrchardD., Web Services Architecture, [Http://www.w3.org/TR/2003/WD-ws-arch-20030514/](http://www.w3.org/TR/2003/WD-ws-arch-20030514/), (14 May 2003). W3C Working Draft.
- Booth, D., Haas, H., McCabe, F., et al. (eds.): Web services architecture. W3C Working Group Note, 11 February 2004. Available at: <http://www.w3.org/TR/ws-arch/>
- Borgo, S., Gangemi, A., Guarino, N., Masolo, C., and Oltramari, A. (2002). *Ontology Roadmap*.
- Borst, P. (1997) Construction of Engineering Ontologies for Knowledge Sharing and Reuse. Ph.D. Dissertation. Tweente University.
- Bou E., López-Sánchez M. and Rodríguez-Aguilar J. A. (2007b) Towards self-configuration in autonomic electronic institutions. To appear In *Coordination, Organization, Institutions and Norms in agent systems*, Lecture Notes in Computer Science.Springer Verlag, 2007.
- Bou E., López-Sánchez M., and Rodríguez-Aguilar J. A. (2007a) Adaptation of autonomic electronic institutions through norms and institutional agents. To appear In *Engineering Societies in an Agents World*, Lecture Notes in Computer Science.Springer Verlag, 2007.
- Bou E., López-Sánchez M., Rodríguez-Aguilar J. A. and Sichman J. S. (2009) Adapting Autonomic Electronic Institutions in Heterogeneous Agent Societies. In *Organised*
- Bradshaw J.(1997) An introduction to software agents.
- Bratman M.E., Israel D. and Pollack M.E. (1988). Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4. pages 349- 355.
- Brescian P., Perini A., Giorgini P., Giunchiglia F. and Mylopoulos J. (2004) Tropos: An

Agent-Oriented Software Development Methodology, Autonomous Agents and Multi-Agent Systems 8 (3): 203-236.
doi:<http://dx.doi.org/10.1023/B:AGNT.0000018806.20944.ef>.

- Brogi A, Corfini S, Popescu R. (2003) Composition-oriented service discovery. Proceedings of 5th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)
- Brooks, R. (1986) A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation 2(1).
- Burghoff U., Pareschi R., Karch H., Noehmeier M. y Schlichter J. (April, 1996). 'Constraint based Information Gathering for a Network Publication system', PAAM 96, First International Conference and Exhibition on the Practical Application of Internet Agents and Multi-Agent Technology, London 22-24.
- Burgin M. and Dodig G. (2009). A Systematic Approach to Artificial Agents.
- Burt R. (1987) Social contagion and innovation: Cohesion versus structural equivalence. American J. of Sociology, 92:1287-1335, 1987.
- Busi N., Ciancarini P., Gorrieri R. and Zavattaro G. (2001): Coordinations Models: A Guide Tour. Coordination of Internet Agents: Models, Technologies, and Applications (Omicini, Zambonelli, Klush y Tolksdorf, editors). Springer-Verlag, 6-24.
- Busi, N.; Ciancarini, P.; Gorrieri, R.; y Zavattaro, G. Coordinations Models (2001). A Guide Tour. Coordination of Internet Agents: Models, Technologies, and Applications (Omicini, Zambonelli, Klush y Tolksdorf, editors). Springer-Verlag, 6-24.
- Buyya R., Yeo C., Venugopala S., Broberg J., and Brandic I. (December, 2010). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems.
- Cammarata S., McArthur D., and Steeb R., (1988). Strategies of Cooperation in Distributed Problem Solving, en Readings in Distributed Artificial Intelligence, Ed. Alan H. Bond and Les Gasser, Morgan Kaufmann.
- Canera J.M. , Hierro J.J and Romo P.A. (2007).La Web Semántica, la siguiente generación de Webs
- Carrascosa C., Bajo J., Julián V., Corchado J.M. and Botti V. (2008) Hybrid multi-agent architecture as a real-time problem-solving model. Expert Systems with Applications, vol. 34 (1) pp. 2-17. Pergamon-Elsevier Science LTD. doi:10.1016/j.eswa.2006.08.031, 2008
- Carrascosa C., Giret A., Julian V., Rebollo M., Argente E. and Botti V. (2009) Service Oriented MAS: An open architecture (Short Paper), Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009), Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10-15, 2009, Budapest, Hungary: 1291-1292
- Carrascosa C., Rebollo M., Soler J., Julian V. and Botti V. (2003) SIMBA Architecture for Social Real-Time Domains EUMAS 2003: The First European Workshop on Multi-Agent Systems pp. 0-0. (2003)
- Castelfranchi C., Miceli M. and Cesta A. (1992). Dependence relations among autonomous agent. Decentralized Artificial Intelligence, eds. E. Werner y Y. Demazeau 215.231. Amsterdam, Elsevier.
- Castelfranchi, C. (1995). Guarantees for autonomy in cognitive agent architecture. In: M Wooldridge and NR Jennings eds. Intelligent Agents: Theories Architectures, and Languages (LNAI Volume 890) pp 56-70 Springer-Verlag,
- Chandrasekaran B., Josephson J. R. and Benjamins R. (1999) What are Ontologies, and Why do we need them? in IEEE INTELLIGENT SYSTEMS 1999
- Chang F., Dean J. et al (2006) Bigtable: a distributed storage system for structured data. In: Proc of OSDI
- Chang V, Bacigalupo D, Wills G, De Roure D (2010) A Categorisation of Cloud Computing Business Models. 10th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing: 509-512.
- Chapell D. (2009). INTRODUCING WINDOWS AZURE. Microsoft Corporation.
- Chu-Carroll C. and Carberry S. (1995). 'Conflict Detection and Resolution in Collaborative Planning' INTELLIGENT AGENTS II: Agent Theories, Architectures and Languages, pp 111-127, Wooldridge, Michael J, Mueller, P and Tambe, Milind, (Ed.), Springer Verlag, Berlin.
- Clark J, DeRose S (1999) XML path language (XPath) Version 1.0. Tech rep, W3C.

- Available at <http://www.w3.org/TR/xpath/>
- Cloud Computing. Retrieved at 03. June 2011 [[http:// fclose.com/ b/ cloud-computing/ article/ mrcc-a-distributed-c-compiler-system-on-mapreduce/](http://fclose.com/b/cloud-computing/article/mrcc-a-distributed-c-compiler-system-on-mapreduce/)]
 - Cohen P., Levesque H. (1990) Intention is Choice with Commitment. *Artificial Intelligence* 42(2-3), March 1990, pp. 213-261.
 - Cole A. (March, 2011). Load Balancing in the Cloud. *ITBusiness Edge*.
 - Conte R. and Paolucci M. (2001) Intelligent social learning. *Artificial Society and Social Simulation*, 4(1):1-23, 2001.
 - Corchado J.M., (2000). Inteligencia Artificial Distribuida. El concepto de agente.
 - Cossentino M. (2005) From requirements to code with the passi methodology, *Agent Oriented Methodologies IV*: 79-106.
 - Costello R. (2002). Building Web Services the REST Way
 - Criado N., Argente E., Julian V. and Botti V. (2009) Designing Virtual Organizations. 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS2009). Volume 55:440-449.
 - Crosby S. and Brown D. (2007) The Virtualization Reality. *Queue - Computer Architecture*, 4 (10).
 - Curbera F., Duftler M., Khalaf R., Nagy W., Mukhi N. and Weerawarana S. (March-April, 2002). Unraveling the Web Services Web : An Introduction to SOAP, WSDL, and UDDI. *IEEE Internet Computing*, Vol 6 Issue:2 pp 86-93.
 - d'Aquin M. and Noy N. (September, 2011). Where to publish and find ontologies? A survey of ontology libraries. *Web Semantics: Science, Services and Agents on the World Wide Web*.
 - D'Inverno M. and Luck M. (2004). *Understanding Agent Systems*. Springer Verlag. ISBN: 354040700.
 - Deloach S. (2009) *Multi-Agent Systems: Semantics and Dynamics of Organizational Models*, IGI Global, Ch. Organizational Model for Adaptive Complex Systems: 1-26.
 - Dignum F. (March 2011). Agents for games and simulation.
 - Dignum V. (2004) A model for organizational interaction: based on agents, founded in logic, PhD. Thesis, 2004.
 - Dignum V. and Dignum F. (2007) A logic for agent organization. In *Proc. FAMAS@Agents'007*.
 - Dignum V., Vazquez-Salceda J. and Dignum F. (2005) OMNI: Introducing Social Structure, Norms and Ontologies into Agent Organizations. *Proceeding of 3rd International Workshop on Programming Multi-Agent Systems*, Utrecht, The Netherlands 2005, pp. 181-198.
 - Dignum V., Vazquez-Salceda J. and Dignum F. (2005) Omni: Introducing social structure, norms and ontologies into agent organizations. *LNAI 3346*.
 - Dölitzscher F., Reich C. and Sulistio A. (2010) Designing Cloud Services Adhering to Government Privacy Laws. In: *Proceedings of 10th IEEE International Conference on Computer and Information Technology (CIT 2010)*. Furtwangen, Germany. pp 930-935
 - Doran J. E., Franklin N., Jennings N. R., and Norman T. (1997) On cooperation in multi-agent systems. *The Knowledge Engineering Review*, 12(3):309-314, 1997.
 - Drogul A. and Ferber J. (1992). Using Reactive Multi-Agent Systems in Simulation and Problem Solving. In *Distributed Artificial Intelligence: Theory and Praxis* (eds N.M. Avouris and L. Gasser).
 - Dunin-Keplicz B. and Verbrugge R (2003) Calibrating Collective Commitments. In: *Marik, V., Müller, J., Pechoucek, M. (Eds.): Multi- Agent Systems and Applications*, LNAI 2691, Springer, 2003, pp. 73 - 83.
 - Dunn, M. F. y Knight, J. C. (1993). Certification of Reusable Software Parts. Technical Report CS-93-41, University of Virginia.
 - Duong D. V. and Grefenstette J. (2005), "The emulation of social institutions as a method of coevolution" in *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, ACM Press, pp. 555-556.
 - Durfee, E. H. (1999). Distributed problem solving and planning. In G. Weiß, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*.
 - Duy T., Sato Y. and Inoguchi Y. (2011) A prediction-based green scheduler for datacenters in clouds. *IEICE Trans Inf Syst E94-D(9)*:1731-1741
 - Eisen M. (April, 2012). Introduction to Virtualization. The Long Island Chapter of the

- IEEE Circuits and Systems (CAS) Society
- Erl T. (2006) Service-Oriented Architecture, Concepts, Technology, and Design. s.l. : Prentice Hall Indiana. 0-13-185858-0.
 - Erl, T., 2009. SOA Design Patterns. The Prentice Hall Service-Oriented Computing Series
 - Escrivá M., Palanca J., Aranda G. and GarcíaFornes A. (2006) A Jabberbased multiagent system platform. In Proc. of AAMAS06, 12821284, 2006
 - Escrivá M., Palanca J., Aranda G., García A., Julian V. and Botti V. (2006) A Jabber-Based Multi-Agent System Platform. Proceeding of 5th International Joint Conference on Autonomous Agents and Multiagent Systems, Hakodate, Japan, May, 2006, pp. 1282-1284.
 - Esteva M. (2003) Electronic Institutions: from specification to development Ph. D. Thesis, Technical University of Catalonia, 2003.
 - Farquhar A. (1997). Ontolingua Tutorial. University of Stanford.
 - Fedak G., Germain C., N'eri V., and Cappello F (2001). XtremWeb: a generic global computing system. In Proceedings of the 1st IEEE International Symposium on Cluster Computing and the Grid, pages 582-587.
 - Ferber J., Gutknecht O. and Michel F. (2004) From Agents to Organizations: an Organizational View of Multi-Agent Systems, in: P. Giorgini, J. Muller, J. Odell (Eds.), Agent-Oriented Software Engineering VI, Vol. LNCS 2935 of Lecture Notes in Computer Science, Springer-Verlag: 214-230.
 - Ferber M., Rauber T., and Hunold S. (2010). BPEL Remote Objects: Integrating BPEL Processes into Object-Oriented Applications.
 - Ferber, J. (1994). Simulating with Reactive Agents, In Hillebrand, E. & Stender, J. (Eds.), Many Agent Simulation and Artificial Life, Amsterdam: IOS Press, 8-28
 - Fergusson, I. (1995). Integrated Control and Coordinated Behavior: A case for agent models. M. Wooldridge, N. Jennings (Eds.): Intelligent Agent: Theories, Architectures and Languages, LNAI 890, pp. 203 - 218, Springer.
 - Fernandez M., Gomez-Perez A. and Juristo N. (1997) "Methodology: From Ontological Art Towards Ontological Engineering" from AAAI Technical Report ss- 97-06
 - Fielding R. (2000) Architectural styles and the design of network-based software architectures. PhD Thesis, University of California, Irvine.
 - Finin, Labour and Mayfield. (1997). KQML as an agent communication language" in Bradshaw, J., Software agents", The MIT Press.
 - Fischer, M. (1994). A Survey of METATEM, the Language and its Applications. In: Gabbay, D., Ohlbach, H. (Eds.): Proceedings of the 1st. International Temporal Logic Conference, LNAI 827, Springer.
 - Foner L. N. (1993). What's an agent, anyway? A sociological case study. Agents Memo 93-01, Agents Group, MIT Media Lab.
 - Foster I, Yong Zhao, Raicu I and Lu S (2008) Cloud Computing and Grid Computing 360-Degree Compared. Grid Computing Environments Workshop GCE '08: 1-10.
 - Foster I. and Kesselman C. (1998). The Grid: blueprint for a new computing infrastructure. Morgan Kaufmann.
 - Foster I., Kesselman C., and Tuecke S. (2001). The anatomy of the grid: Enabling scalable virtual organizations, Int. J. High Perform. Comput. Appl, vol. 15, no.~3: 200-222
 - Fox M. and Gruninger M. (1998) Enterprise Modeling. AI Magazine, AAAI Press, Fall, 1998, pp. 109-121.
 - Franklin S. and Graesser A. (1996). Is it an Agent, or Just a Program? A Taxonomy for Autonomous Agent.
 - G. Dudek, M. Jenkin, R. Milius, and D. Wilkes. (1993). A taxonomy for swarm robots. In Proceedings of IEEE/RSJ Conference on Intelligent Robots and Systems.
 - Galvez Rojas S. Fundamentos de WS-BPEL. Curso Tecnologías Emergentes Multiplataformas. Universidad de Málaga - Departamento de Lenguajes y Ciencias de la Computación.
 - Garca-Magario I., Gómez-Sanz J. and Fuentes R. (2009) Ingenias development assisted with model transformation by-example: A practical case. In: 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009), pp. 40 - 49

- García F. (2007). "Sistema basado en tecnologías del conocimiento para entornos de servicios web semánticos". Universidad de Murcia.
- García I. (xxxx) Lógicas para la red: La Web Semántica. Trabajo de Lógicas para la Inteligencia Artificial. 5º Ingeniería Informática Superior. Universidad de Salamanca.
- García J.A. (2012). IRIS & ME OWL-S Sistema de Recomendación en Dispositivo Móvil con OWL-S .Trabajo Fin de Master. Universidad de Salamanca.
- García J.A. IRIS & ME OWL-S Sistema de Recomendación en Dispositivo Móvil con OWL-S .Análisis e implementación del protocolo semántico de comunicación OWL-S en dispositivos móviles para sistemas de recomendación(2012)
- Garcia V. C., Lucrédio D., Alvaro A., de Almeida E. S., de Mattos R. P., de Lemos S. R. (2007) Towards a maturity model for a reuse incremental adoption. In: The 1st Brazilian Symposium on Software Components, Architecture and Reuse, Campinas, São Paulo, Brazil, pp. 61-74 (2007)
- Gasser L. and Ishida, T. (1991) A dynamic organizational architecture for adaptive problem solving. In: Proc. of AAAI-91, 185-190, 1991
- Genesereth, M., Nilsson, N. (1987). Logical Foundations of Artificial Intelligence, Morgan Kaufman..
- Genesereth, MR and Ketchpel, SP. (1994). Software Agents. Communications of the ACM 37(7) 48-53.
- Giampapa J. A. and Sycara K. (2002) "Team-Oriented Agent Coordination in the RETSINA Multi-Agent System," Tech. Report CMU-RI-TR-02-34, Robotics Institute, Carnegie Mellon University, December 2002. Presented at AAMAS 2002 Workshop on Teamwork and Coalition Formation.
- Gil, AB. (2011). Recuperación inteligente de contenidos digitales educativos.Tesis Doctoral Universidad de Salamanca.
- Gil, O. (2009). "iSumillerService: Servicio Web Semántico de consulta de vinos españoles". Proyecto de Fin de Carrera INGENIERÍA INFORMÁTICA.
- Giner P., Torres V., Pelechano V. (2010). Bridging the Gap between BPMN and WS-BPEL. M2M Transformations in Practice project DESTINO TIN2004-03534.
- Giret A., Julian V., Rebollo M., Argente E., Carrascosa C. and Botti V. (2009) An Open Architecture for Service-Oriented Virtual Organizations. Seventh international Workshop on Programming Multi-Agent Systems.PROMAS 2009: 23- 33.
- Goldner S. and Papproth A. (2011). Extending the BPMN Syntax for Requirements Management Business Process Model and Notation Lecture Notes in Business Information Processing, , Volume 95, Part 2, 142-147, DOI: 10.1007/978-3-642-25160-3_13.
- Gomez-Perez A. and Rojas-Amaya D. (1999). Ontological reengineering for reuse. In Knowledge Acquisition, Modeling and Management: 11th European Workshop, EKAW '99, Dagstuhl Castle, Germany, May 1999. Proceedings, volume 1621 of Lecture Notes in Computer Science, page 139. Springer Berlin.
- Governor J., Nickull D., and Hinchcliffe B. (2009). Web 2.0 Architectures Specific Patterns of Web 2.0: Chapter 7.
- Greenwood,D. and Calisti, M. (2004). Engineering Web service - agent integration . Systems, Man and Cybernetics, 2004 IEEE International Conference on. Page(s): 1918 - 1925 vol.2
- Grosz B. y Sidner C. (1990). Plans for discourse. Intentions for Communication, pp 417-444, MIT Press.
- Grosz B., Kraus S. (1996) Collaborative Plans for Complex Group Actions. Artificial Intelligence 86, 1996, pp. 269-358.
- Gruber, T. R. (1995). Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human-Computer Studies, Special Issue on the Role o formal Ontology in the Information Technology, 43(5/6), 907-928.
- Gruninger M. Bodenreider O. Olken F. Obrst L. and Yim P. (2007). Ontology Summit 2007 - Ontology, taxonomy, folksonomy: Understanding the distinctions.
- Gruninger, M. and Fox, M.S. (1995). Methodology for the Design and Evaluation of Ontologies. In: Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI-95, Montreal.
- GTI-IA. (2009) An Abstract Architecture for Virtual Organizations: The THOMAS project. <http://www.fipa.org/docs/THOMASarchitecture.pdf>

- Guarino N. (1998), Formal Ontology in Information Systems. Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998. Amsterdam, IOS Press, pp. 3-15.
- Gutiérrez F. Comprenda las técnicas de Internet, Artículo 179 en PYME
- Gutknecht O. and Ferber J. (2000) MadKit: a generic multiagent platform. In Proceedings of the Fourth international Conference on Autonomous Agents (Barcelona, Spain, June 03 - 07, 2000). AGENTS '00. ACM, New York, NY, 78-79.
- H. J. Muller. (1996). Negotiation principles. In John Wiley & Sons, editor, Foundations of Distributed Artificial Intelligence, pages 211{230, Norway.
- H. Van Dyke and Parunak. (1996). Applications of distributed artificial intelligence in industry. In G. M. P. O'Hare and N. R. Jennings, editors, Foundations of Distributed AI. JohnWiley & Sons.
- Hashimi S. (2003). Service-Oriented Architecture Explained. Published on ONDotNet.com (<http://www.ondotnet.com/>)
- Hass H. and Brown A. (2004) Web Services Glossary W3C Working Group W3C
- Havey, M. (2005). Essential Business Process Modeling, 1st ed., O'Reilly Media, Sebastopol, CA.
- Hawley A. (2009). Virtual Infrastructure: What Is Required for the Cloud?.ORACLE VM.
- Hendler, J., Berners-Lee, T., and Miller, E. (2002). Integrating Applications on the Semantic Web. Journal of the Institute of Electrical Engineers of Japan, 122(10), 676-680.
- Hernandez L., Botti V., Garcia-Fornes A. (2006) A deliberative scheduling technique for a real-time agent architecture. Engineering Applications of Artificial Intelligence Volume: 19 Issue: 5 Pages: 521-534 Published: AUG 2006
- Hill J.B., Sinur, J., Flint D. and Melenovsky M.J. (2006). Gartner's position on business process management, Business Issues, Gartner, Stamford, CT.
- Hill, J.B., Pezzini, M. and Natis, Y.V. (2008). Findings: confusion remains regarding BPM terminologies, Vol. ID No. G00155817, Gartner Research, Stamford, CT.
- Hoare C. A. R. Communicating sequential processes. Communications of the ACM, 21:666-677. 1978.
- Horling B. and Lesser V. (2005). Using ODML to Model Multi- Agent Organizations. In Proc. of the IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology
- Howden N. (2001). JACK intelligent agents—summary of an agent infrastructure. In Proceedings of IEEE international conference on autonomous agents, Montreal.
- Hubner J. F., Sichman J. S. and Boissier, O. (2004) Using the Moise+ for a cooperative framework of mas reorganisation. In: LNAI Proc. of the 17th Brazilian Symposium on Artificial Intelligence (SBIA'04). Volume 3171, 506–515, Springer ,2004.
- Hubner J., Sichman J. and Boissier O. (2006) S-Moise+: A middleware for developing organised multi-agent systems. In: Proc. Int. Workshop on Programming in MAS, LNCS, vol 3913: 64–78
- Huhns M. and Stephens L. (1999) Multiagent Systems and Societies of Agents. In: Weiss, G. (Ed.), Multi-agent Systems: a Modern Approach to Distributed Artificial Intelligence, MIT Press, 1999.
- Huhns, M. and Larry M., Multiagent (1999). Systems and Societies of Agents, en Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, editado por Gerhard Weiss, The MIT Press, Cap 2, pp. 121-164.
- Huth A. And Cebula J. (2011). The Basics of Cloud Computing. US-CERT.
- IEEE1517-2009 D2. (2009) IEEE1517 Standard for Information Technology - Software Life Cycle Processes - Reuse Processes: 2009 D2. Software Engineering Standards Committee of the IEEE Computer Society, USA (2009)
- Iglesias Martínez I. A. (2010) Modelado automático del comportamiento de agentes inteligentes. Phd Tesis, . Departamento de Informática. Universidad Carlos III de Madrid. 2010.
- ISO (1991, 1994). Quality Management and Quality Assessment Standards, Part 3: Guidelines for the ISO 9001 to the Development, Supply and Maintenance of Software. ISO 9000-3, International Standards Organization.
- J. Euzenat and P. Shvaiko. (2007). Ontology matching. Springer.
- J. van Diggelen. (2007). Achieving Semantic Interoperability in Multi-agent Systems: A Dialogue-based Approach, PhD Thesis.
- Jacques Ferber. (1999). Multi-Agent Systems: An Introduction to Distributed Artificial

- Intelligence. Addison-Wesley Longman Publishing Co., Boston, MA, USA.
- Jaeger P. T., Lin J., Grimes JM. and Simmons S. N. (2009) Where is cloud? *Geography, Economics, Env Jurisdiction Cloud Comput* 14(5)
 - Jennings N. (1993). *Commitments and Conventions: The Foundation of Coordination in Multi- Agent Systems*.
 - Jennings N., Wooldridge M. (1988). (eds.), *Agent Technology: Foundations, Applications and Markets*, Springer,
 - Jiao W.P. Mei H. (2004) Automated adaptations to dynamic software architectures by using autonomous agents. *Engineering Applications of Artificial Intelligence* Volume: 17 Issue: 7 Pages: 749-770 Published: OCT 2004
 - K. Gottschalk et al., "Web Services Architecture Overview: The Next Stage of Evolution for E-Business," <http://www-106.ibm.com/developerworks/library/w-ovr>.
 - Kaelbling, L. P. (1991). A situated automata approach to the design of embedded agents. *SIGART Bulletin*, 2(4):85-88
 - Karlsson, E. A. (1996). *Software Reuse: A Holistic Approach*. Wiley Series in Software based Systems. John Wiley and sons.
 - Keil F. C. (1989). *Concepts, Kinds, and Cognitive Development*. MIT Press.
 - Kemsley S. (2007). *Business Process Design*. TIBCO.
 - Kennedy S., Molloy O., Stewar R., Jacob P., Maleshkova M., and Doheny F. (2012). A Semantically Automated Protocol Adapter for Mapping SOAP Web Services to RESTful HTTP Format to Enable the Web Infrastructure, Enhance Web Service Interoperability and Ease Web Service Migration.
 - Khondoker M. R. and Mueller P. (2010) Comparing Ontology Development Tools Based on an Online Survey IN: *Proceedings of the World Congress on Engineering 2010 Vol I*
 - Kittock J. E. (1993) Emergent conventions and the structure of multi-agent systems. In *Lectures in Complex systems: the proceedings of the 1993 Complex systems summer school*, Santa Fe Institute Studies in the Sciences of Complexity Lecture Volume VI, Santa Fe Institute, pages 507–521. Addison-Wesley, 1993.
 - Klusch M. y Sycara K. (2001): Brokering and matchmaking for coordination of agent societies: a survey. En *Coordination of Internet Agents: Models, Technologies, and Applications* (Omicini et al.) Springer, 197-224
 - Knight, J. C. y Dunn, M. F. (1998). Software Quality through Domain-Driven Certification. *Annals of Software Engineering*, 5:293–315.
 - Ko R., Lee S. and Lee E. (2009) .Business process management (BPM) standards: a survey". *Emerald Business Process Management Journal* Vol. 15 No. 5,pp 744-791.
 - Kortz D. and Gray R. (1999). *Mobile Agents and the future of the Internet*.
 - Koskela M. and Haajanen, J. (2007). *Business process modeling and execution: tools and technologies report for the SOAMeS project*, VTT Research Notes No. 2407, VTT Technical Research Centre of Finland, Espoo.
 - Krueger, C. W. (1992). Software Reuse. *ACM Computing Surveys*, 24(2):131–183.
 - Labidi, S. and Lejouad, W. (1993). *Del'intelligence artificielle distribu'eeaux syst'emes multiagents*.
 - Lakkaraju K. and Gasser L. (2008) "Norm Emergence in Complex Ambiguous Situations." In *Proceedings of the AAI Workshop on Coordination, Organizations, Institutions and Norms AAI*, Chicago, July, 2008.
 - Lapadula A., Pugliese R. and Tiezzi F. (2010) A WSDL-based type system for asynchronous WS-BPEL Processes. *Form Methods Syst Des* 38: 119–157.
 - Lee C.H. and Hwang S. Y. (2009). *A Model for Web Services Data in Support of Web Service Composition and Optimization*. ISBN: 978-0-7695-3708-5.
 - Lee J.M. (1997) *Riemannian Manifolds. An introduction to Curvature*. Springer- Verlag, New York, Inc. 1997
 - Lemley M. and O'Brien D. (1997) Encouraging Software Reuse. *Stanford Law Review*, 49, 255.
 - Lésperance, Y., Levesque, H., Lin, F., Marcu, D., Reiter, R., and Scherl, R. (1996). *Foundations of a Logical Approach to Agent Programming*. In:Wooldridge, M., Müller, J., Tambe, M. (Eds.): *Intelligent Agents II*, LNAI 1037, Springer.
 - Li K., Verma K., Mulye R., Rabbani R., Miller J. and Sheth A. (2006) *Designing Semantic Web Processes: The Wsdl-S Approach*. *Semantic Web Services, Processes*

- and Applications Semantic Web and Beyond, Volume 3, Part II, pp 161-193.
- Ligeza A., Kluza K. and Potempa T. (2012). AI Approach to Formal Analysis of BPMN Models. Towards a Logical Model for BPMN Diagrams. Preprints of the Federated Conference on Computer Science and Information Systems pp. 959–962
 - López F. (1999) Overview of Methodologies for Building Ontologies In Proceedings of IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5) in Stockholm Sweden, August 2.
 - Lowe-Norris A.G. and Denn R. (2000). Windows 2000 Active Directory, O'Reilly & Associates, Sebastopol, CA.
 - Lu J., Chen G., and Yu X. (2011). Modelling, Analysis and Control of Multi-Agent Systems: A Brief Overview”.
 - Luck M. and McBurney P. (2008) Computing as interaction: Agent and agreement technologies, in: IEEE SMC Conference on Distributed Human-Machine Systems: 1–6.
 - Luo J., Montrose B., Kim A., Khashnobish A. and Kang M. Adding OWL-S Support to the Existing UDDI Infrastructure.(2006)
 - M. Wooldridge and N. R. Jennings. (1995). Intelligent Agents: Theory and Practice. In KNOWLEDGE ENGINEERING REVIEW 10(2).
 - Maamar, Z. , Mostefaoui, S.K. , and Yahyaoui, H. (2005). Toward an agent-based and context-oriented approach for Web services composition . Knowledge and Data Engineering, IEEE Transactions on. Vol 17, pp 686-697
 - Maedche A. and Staab S. (2001) Learning Ontologies for the Semantic Web in Semantic Web Workshop 2001 Hongkong, China
 - Maes, P. (1990).(Ed.): Designing Autonomous Agents. MIT Press.
 - Maes, P. (1991). The agent network architecture (ANA). SIGART Bulletin, 2(4):115-120.
 - Mahmoud Q. (2005) Service-Oriented Architecture (SOA) and Web Services: The Road to Enterprise Application Integration (EAI). ORACLE.
 - Malone, T. and Crowston, K. (March, 1994). The Interdisciplinary Study of Coordination. ACM Computing Surveys 26(1).
 - Mancarella S. (January 2011). Business Process Modelling Notation – A tutorial. OMG SOA Healthcare. SPARX Systems.
 - Martin D., Burstein M., McDermott D., McIlraith S., Paolucci M., Sycara K., McGuinness D. Sirin E. and Srinivasan N. Bringing Semantics to Web Services with OWL-S. (2006)
 - Mas, A. (2005). Agentes software y sistemas multiagente: conceptos, arquitecturas y aplicaciones. ISBN 84-205-4367-5 , pp. 29-64.
 - Massonet P., Naqvi S., Ponsard C., Latanicki J., Rochwerger B. and Villari M. (2011) A Monitoring and Audit Logging Architecture for Data Location Compliance in Federated Cloud Infrastructures. In Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW). 2011 IEEE International Symposium on. 1510-1517
 - Mataric M. J. (1997) “Learning Social Behavior” in Robotics and Autonomous Systems, vol. 20, pp. 191-204, 1997
 - Mathes M., Gartner J. , Dohmann, H. , and Freisleben B. SOAP4IPC: A Real-Time SOAP Engine for Industrial Automation . Parallel, Distributed and Network-based Processing, 2009 17th Euromicro International Conference on. Pp 220 - 226
 - McAllester D. and Rosenblitt D. (1991) Systematic nonlinear planning. In Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91), volume 2, pages 634–639, Anaheim, California, USA, 1991. AAA Press/MIT Press
 - McIlraith, S., Son, T.: Adapting golog for composition of semantic web services. In: Proceedings of the 8th International Conference on Principles of Knowledge Representation and Reasoning (KR), pp. 482–493. Toulouse, France, April 2002
 - McIlraith, S.A. and Martin, D.L. Bringing semantics to Web services Volume: 18 , Issue: 1 Page(s): 90 - 93 (Jan.-Feb. 2003)
 - Mell P. and Grance T. (2011). NIST Definition of Cloud Computing v15, csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc .
 - Mendling J. and Neumann G. (2005). “A comparison of XML interchange formats for business process modeling”, Workflow Handbook, Future Strategies, Lighthouse Point, FL.
 - Menzel C. (2003). Reference Ontologies —Application Ontologies: Either/Or or Both/And?. Texas A&M University .

- Milenkovic M., Robinson H., Knauerhase R., Barkai D., Garg S., Tewari V., Anderson T., and Bowman M. Toward Internet distributed computing. *IEEE Computer*, 36(5):38–46, 2003.
- Moemeng C., Wang C., and Longbing C. (2011). Obtaining an optimal MAS configuration for Agent-Enhanced Mining Using Constraint optimization.
- Molesini A., Omicini A., Denti E. and Ricci A. (2006) SODA. 2006. A roadmap to artefacts, *Engineering Societies in the Agents World VI LNAI 3963*: 49–62.
- Morrison I. and Nugrahanto S. (2007). *Decision Support With BPEL and Web Services*.
- Morrison I., Lewis B. and Nugrahanto S. (2006). *Modelling in Clinical Practice with Web Services and BPEL*. Chapter X.
- N. Noy and D. McGuinness. (2001). *Ontology development 101: A guide to creating your first ontology*. Technical Report SMI-2001-0880, Stanford Medical Informatics (SMI), Department of Medicine, Stanford University School of Medicine.
- Nakrani S., and Tovey C., On Honey Bees and Dynamic Server Allocation in Internet Hosting Centers. *Adaptive Behavior* 12, pp: 223-240 (2004).
- Nau, D., Au, T.C., Ilghami, O., et al.: SHOP2: An HTN planning system. *J. Artif. Intell. Res.* 20, 379–404 (2003)
- Nesse PJ, Undheim A, Solsvik FH, Dao M, Salant E, Lopez JM, Elicegui JM (2011) Exploiting cloud computing: A proposed methodology for generating new business. 15th International Conference on Intelligence in Next Generation Networks (ICIN): 241-246.
- Newcomer E. (2002). *Understanding Web Services, XML, WSDL, SOAP and UDDI*. Independent Technology Guides. David Chapell: Series Editor. ISBN: 0-201-75081-3
- Nitzsche J., Van Lessen T., Karastoyanova D., and Leymann F. (2007). *BPEL for Semantic Web Services (BPEL4SWS)*. Institute of Architecture of Application Systems, University of Stuttgart.
- Noy N. F. and McGuinness D. L. (2001) *Ontology Development 101: A Guide to Creating Your First Ontology at Stanford University*. SMI-2001-0880, March 2001.
- Nwana H. S. (1996). *Software Agents: An Overview*. Knowledge Engineering Review.
- Nwana H. S., Ndumu D. T., Lee L. C. and Collis J. C. (1999) ZEUS: A Toolkit and Approach for Building Distributed Multi-Agent Systems *Agents 1999*: 360-361
- O'hare G. M. and Jennings N. R. (1996). *Foundations of Distributed Artificial Intelligence*. Willey-Interscience.
- OASIS (April, 2007) *.Web Services Business Process Execution Language Version 2.0"OASIS Standard*.
- Odell J. (October 2010). *Agent Technology: An Overview*,.
- Omicini A., Ricci A. and Viroli M. (2004) Coordination artifacts: Environmentbase coordination for intelligent agents. In *Proceedings of 3rd international Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, 286—293, 2004.
- Osiecki L., Phillips M., and Scibilia J. (2011) "Understanding and Leveraging a Supplier's CMMI Efforts: A Guidebook for Acquirers (Revised for V1.3)," *Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical Report CMU/SEI-2011-TR-023*, 2011.
- Ossowski S. (2001): *Agent Coordination by Constraint Optimisation*. *Electronic Notes in Theoretical Computer Science* 48, Elsevier.
- Ossowski, S.: *Co-ordination in Artificial Agent Societies*, LNAI 1535, Springer, 1998.
- Ouyang C., Dumas M., ter Hofsetede A. and van der Aalst W. (2006) . From BPM Process Models to BPEL Web Services: The BPMN to BPEL Way .
- Owen M. and Raj J. (2003). *BPMN and Business Process Management Introduction to the New Business Process Modeling Standard*.
- OWL-S: Semantic Markup for Web Services. Disponible en: <http://www.w3.org/Submission/OWL-S/>. Accedido en (2012)
- Padala P, Hou K-Y et al (2009) Automated control of multiple virtualized resources. In: *Proc of EuroSys*.
- Panait L. and Luke S. (2005). *Cooperative Multi-Agent Learning: The State of Art*. George Mason University.
- Paolucci M., Kwamura T., Payne T., and Sycara K. (2002). *Importing the semantic Web in UDDI*. Revised Papers from the International Workshop on Web Services, E-

Business, and the Semantic Web . pp 225-263

- Papazoglou M. (2008) Web Services: Principles and Technology. s.l. : Prentic Hall.
- Parimala N. and Saini A. (2011). X-WSDL: An Extension to WSDL and its mapping to X-UDDI.
- Pasley J. (May/June 2005). How BPEL and SOA Are Changing Web Services Development Published by the IEEE Computer Society, IEEE Internet Computing. Pp 60- 67.
- Pattison H. E., Corkill D. D. and Lesser V. R. (1987) Distributed Artificial Intelligence, chapter Instantiating Descriptions of Organizational Structures, pages 59–96. Pitman Publishers, 1987.
- Pautasso C., Zimmerman O., and Leymann F. (2008). Restful web services vs. "big" web services: making the right architectural decision. Published in: WWW '08 Proceedings of the 17th international conference on World Wide Web. Pp 805-814.
- Pavon J. and Gomez-Sanz J. J. (2003) Agent oriented software engineering with INGENIAS. In Proceedings of CEECMAS, volume 2691, pages 394– 403, 2003.
- Pavon J., Gomez-Sanz J. and Fuentes R. (2005) The INGENIAS Methodology and Tools, Idea Group Publishing, article IX: 236–276.
- Pedrinaci C., Brelage C., van Lessen T., Domingue J.; Karastoyanova d. and Leymann F. (2008). Semantic business process management: scaling up the management of business processes. In: 2nd IEEE International Conference on Semantic Computing (ICSC) 2008, 4-7 Aug 2008, Santa Clara, CA, USA.
- Peiró, J. M. (1990): «Las Nuevas Tecnologías» en Organización. Nuevas perspectivas psicosociales, Barcelona, PPU S.A. Penberthy y Weld, 1992] Penberthy, J. S. and Weld, D., UCPOP: A Sound, Complete, Partial-Order Planner for ADL. Third International Conference on Knowledge
- Poulin J. S. (1997) Measuring software reuse - principles, practices, and economic models. Addison-Wesley-Longman 1997, isbn 978-0-201-63413-6, pp. I-XIX, 1-195
- Poulin J. S. (2006) PoulinThe Business Case for Software Reuse: Reuse Metrics, Economic Models, Organizational Issues, and Case Studies, Lecture Notes in Computer Science Volume 4039, 2006, p 439
- Prieto, R. (1993). Status Report: Software Reusability. IEEE Software, 10(3):61– 66.
- Prieto, R. (1996). Reuse as a New Paradigm for Software Development. En Sars- har, M., editor, Systematic Reuse: Issues in Initiating and Improving a Reuse Program, London. Springer Verlag.
- Radgui M., Saidi R. and Moulina S. (September 2012) . A Pattern for the Decomposition of Business Processes. Special Issue of International Journal of Computer Applications (0975 – 8887) on Software Engineering, Databases and Expert Systems – SEDEXS.
- Rahman S., Bhardwaj A., Pathak M., and Rathore S. (2012). INTRODUCTION OF KNOWLEDGE MANAGEMENT ARCHITECTURE USING MULTI AGENT.
- Ralyté, J., Mirbel, I., Deneckère, r., 2011. Engineering Methods in the Service-Oriented Context. 4th IFIP WG 8. 1 Working conference on method engineering. Paris, France.
- Randles M, Lamb D. and Taleb-Bendiab A. (2010). A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing. 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops
- Rao A. and Georgeff M. (June, 1995). Bdi agents: From theory to practice. In Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95), pages 312-319, Menlo Park, California. AAAI Press.
- Rao J. (2012). SemanticWeb Service Composition via Logic-based Program Synthesis. ISBN 82-471-6464-7
- Rao, A. and Georgeff, M. (1992). An Abstract Architecture for Rational Agent. In: C. Rich, W. Swartout, B. Nebel (Eds.): Proc. of KR'92, Morgan Kaufmann.
- Ravichandran T. (1999) Software reuseability as synchronous innovation: a test of four theoretical models. European Journal of Information Systems 8, 83–199
- Razavi R., Perrot J., & Guelfi N. (2005). Adaptive modeling: an approach and a method for implementing adaptive agents. Lecture Notes in Artificial Intelligence , 3446, 136-148.
- RedHat. (2007). Virtualization Guide. Red Hat Virtualization.

- Rehesaar H. (2011) Capability Assessment for Introducing Component Reuse, Lecture Notes in Computer Science Volume 6727, 2011, pp 87-101
- Rehesaar H. (2011) Capability Assessment for Introducing Component Reuse, Top Productivity through Software Reuse, Lecture Notes in Computer Science Volume 6727, 2011, pp 87-101
- Rodríguez S. (2010). Modelo Adaptativo para organizaciones virtuales de agentes. Tesis Doctoral. Universidad de Salamanca.
- Rodríguez S., de Paz Y., Bajo J. and Corchado J. M. (2011) Social-based Planning Model for Multiagent Systems. [PDF] Expert Systems with Applications vol.: 38 num.: 38 pag.: 13005-13023
- Rodríguez S., Pérez-Lancho B., De Paz J.F., Bajo J. and Corchado J.M. (July, 2009). Ovamah: Multiagent-based Adaptive Virtual Organizations. 12th International Conference on Information Fusion, Seattle, Washington, USA.
- Romero, R. (2007). Tesis Doctoral "Especificación OWL de una ontología para teleeducación en la Web Semántica". Universidad Politécnica de Valencia.
- Rosenberg D. (November, 2011). Are databases in the cloud really all that different?, CNET.
- Rosenschein, J.; y Zlotkin, G. (1994) Rules of Encounter - Designing Conventions for Automated Negotiation among Computers. MIT Press.
- Rosu M.C. (July, 2007). A-SOAP: Adaptive SOAP Message Processing and Compression. Web Services, 2007. ICWS 2007. IEEE International Conference. Pp 200-207
- Ruckhaus E. (2006) RDF y RDF-Schema.
<http://www ldc.usb.ve/~ruckhaus/materias/ci7453/clase4.pdf>
- Rueda S., García J., and Simari G. R. (2002) Argumentbased negotiation among bdi agents. Journal of Computer Science and Technology, 2(7):1-8, October 2002.
- Russell S. and Norvig P. (1995). Artificial Intelligence: A Modern Approach. Englewood Cliffs, NJ: Prentice-Hall.
- Saffre F., R. Tateson, J. Halloy, M. Shackleton and J.L. Deneubourg.(march, 2008). Aggregation Dynamics in Overlay Networks and Their Implications for Self-Organized Distributed Applications. The Computer Journal.
- Sametinge, J. (1997). Software Engineering with Reusable Components. Springer Verlag.
- Sanderson D. (November, 2009). Programming Google App Engine. O'Reilly Google Press.
- Sansores C., Pavón J. (2005) Simulación social basada en agentes. Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial, ISSN 1137- 3601, Vol. 9, N.º. 25, 2005 , pags. 71-78
- Schaaf M., Koschel A., Grivas S .G. and Astrova I. (2010) An active DBMS style activity service for cloud environments. Cloud Computing 2010 : The First International Conference on Cloud Computing, GRIDs, and Virtualization in Cloud Computing 2010 in ComputationWorld 2010 IARIA. 80-85
- Schein E. H. (2004) Organizational culture and leadership. Jossey-Bass. 1985. Edición: 3 - 2004 - 437 páginas
- Schelling T. C. The Strategy of Conflict. Harvard University Press, Cambridge, MA.1960
- Schmid K. (2011a) 12th International Conference on Software Reuse, ICSR 2011, Pohang, South Korea June 13-17, 2011. Series: Lecture Notes in Computer Science, Vol. 6727
- Schmid K. (2011b) Top Productivity through Software Reuse 12th International Conference on Software Reuse, ICSR 2011, Pohang, South Korea, June 13-17, 2011. Proceedings
- Serrano J M., and Ossowski S. (2004) "RICA-J -- A Dialogue-Driven Software Framework for the Implementation of Multiagent Systems", JISBD Taller en Desarrollo de Sistemas Multiagente (DESMA-2004), {M}álaga, pp. 48-61, Nov, 2004.
- Shang R. D., Mohan K., Lang K. R., Vragov R. (2012) A market mechanism for software component reuse: opportunities and barriers ICEC, 2012
- Shapiro, R. (2002). A comparison of XPDL, BPML, and BPEL4WS, OASIS XML Cover Pages, Rotterdam.
- Sherif K. and Vinze A. S. (2003) Barriers to adoption of software reuse: a qualitative study. Information and Management 41(2), 159-175 (2003)

- Shoham Y. and Tennenholtz M. (1992) Emergent conventions in multi-agent systems. In Proceedings of Knowledge Representation and Reasoning, pages 225– 231, 1992.
- Shoham Y. and Tennenholtz M. (1997) On the emergence of social conventions: Modeling, analysis, and simulations. *Artificial Intelligence*, 94(1-2):139–166, 1997.
- Silberschatz, Galvin (1994). *Operating System concepts*, chapter 17 Distributed file systems. Addison-Wesley Publishing Company. ISBN 0-201-59292-4.
- Simchi-Levi D., Kaminsky P. and Simchi-Levi E. (2000). *Designing and Managing the Supply Chain: Concepts, Strategies, and Case Studies*, McGraw-Hill/Irwin, New York, NY.
- Singh A. and Malhotra M. (2012). Agent Based Framework for Scalability in Cloud Computing. *International Journal of Computer Science & Engineering Technology (IJCSET)*
- Sirin E., Parsia B., Cuenca B., Kalyanpur A. and Katz Y. (2003) “Pellet: A Practical OWL-DL Reasoner” at University of Maryland, MIND Lab.
- Sowa, J. F. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove, CA: Brooks Cole Publishing Co.
- Spencer H. (1896) *The Study of Sociology*. New York. D. Appleton and company. 1896
- Srinivasan N., Paolucci M., and Sycara K. (July, 2004). Adding OWL-S to UDDI, implementation and throughput. Proceedings of the first international workshop on Semantic Web Services and Web Process Composition – SWSWPC 2004.
- Stage A. and Setzer T. (2009) Network-aware migration control and scheduling of differentiated virtual machine workloads. In: Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing (CLOUD '09). IEEE Computer Society, Washington, DC, USA. pp 9-14
- Stone P. (1997). Reactive vs. Deliberative agents.
- Stone P. and Veloso M. (2000). Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383.
- Stroustrup, B. (1996). Language-technical Aspects of Reuse. En Sitaraman, M., editor, Fourth International Conference on Software Reuse, páginas 11–19, Orlando, Florida. IEEE Computer Society Press.
- Sulistio, A, Reich C. and Dölitzscher F. (2009) Cloud Infrastructure & Applications - CloudIA. In: Proceedings of the 1st International Conference on Cloud Computing (CloudCom'09), Beijing, China. pp pp 583-588
- Swartout B., Ramesh P., Knight K. and Russ T. (1997) Toward Distributed Use of Large Scale Ontologies. In: AAAI'97 Spring Symposium on Ontological Engineering. Stanford University, California.
http://ksi.cpsc.ucalgary.ca/KAW/KAW96/swartout/Banff_96_final_2.html
- Swartout, W., Tate, A. (1999). Ontologies. *IEEE Intelligent Systems*, 14(1), 18-19.
- Sycara K., Paolucci M., Ankolekar A. and Srinivasan N. (2003) Automated discovery, interaction and composition of Semantic Web services.
- Sycara K., Paolucci M., van Velsen and M. Giampapa J. (2003) The RETSINA MAS Infrastructure. *Journal of Autonomous Agents and Multi-Agent Systems* 7(1 & 2), Kluwer Academic Publishers, 2003.
- T. W. Malone. (1988) What is coordination theory? In National Science Foundation Coordination Theory Workshop. MIP, EE.UU.
- Talia D. (2012). *Cloud Computing and Software Agents: Towards Cloud Intelligent Services*
- Tambe M. (1997) Towards Flexible Teamwork. *Journal of Artificial Intelligence Research* 7, 1997, pp. 83 – 124.
- Tan M. (1993). Multiagents Reinforcement Learning: Independent vs. Cooperative Agents Multiagents Reinforcement Learning: Independent vs. Cooperative Agents,.
- Tao J., Kunze M., Castellanos A. C., Kramer, D. and Karl, W. (2008) Scientific Cloud Computing: Early Definition and Experience High Performance Computing and Communications, 2008. HPCC '08. 10th IEEE International Conference, pp. 825 – 830.
- Taylor F. W. (1991) *Principios y métodos de gestión científica*
- Tere G.M. and Jadhav B.T. (2012). Designing Application Framework using WSDL. International Conference & Workshop on Recent Trends in Technology, (TCET).
- Toma, I., Foxvog, D. (2006). Non-functional properties of web services WSMO working draft.

- Uschold M. and Gruninger M. (1996). *Ontologies: Principles, Methods and Applications*. Knowledge Engineering Review.
- van der Aalst W.M.P. de Beer, H.T. and van Dongen B.F. (2005). *Process Mining and Verification of Properties: An Approach Based on Temporal Logic*, Springer, New York, NY.
- van der Aalst W.M.P., Dumas M., ter Hofstede A.H.M., Russell, N., Verbeek, H.M.W. and Wohed, P. (2005). *Life after BPEL?*, Lecture Notes in Computer Science, Vol. 3670, pp. 35-50.
- van der Aalst, W.M.P. (2003). "Patterns and XPD: a critical evaluation of the XML process definition language, QUT Technical Report, FIT-TR-2003-06, Queensland University of Technology, Brisbane.
- van der Aalst, W.M.P. (2005). Don't go with the flow: web services composition standards exposed, *IEEE Intelligent Systems*, Vol. 18 No. 1, pp. 72-6.
- van Gorp P., Dijkman R. (September, 2012) A visual token-based formalization of BPMN 2.0 based on in-place transformations.
- van Heijst G., Schreibe A., and Wielinga B. J. (1997). Using Explicit Ontologies in KBS Development. *International Journal of Human and Computer Studies*, 46(2/3), 183-292.
- Vaquero L, Rodero-Merino L, Caceres J, Lindner M (2009). A break in the clouds: towards a cloud definition. *ACM SIGCOMM computer communications review*.
- Vidal J.M. (March, 2010). *Fundamentals of Multi-Agent Systems with Net Logo Examples*.
- Villatoro D. and Sabater-Mir J. (2008c) Towards the Group Formation through Social Norms. *Proceedings of the Sixth European Workshop on Multi-Agent Systems (EUMAS08)*
- Villatoro D. and Sabater-Mir J. (2008a) Categorizing Social Norms in a Simulated Resource Gathering Society. *Proceedings of the AAAI Workshop on Coordination, Organizations, Institutions and Norms (COIN @ AAAI08)*
- Villatoro D. and Sabater-Mir J. (2008b) Mechanisms for Social Norms Support in Virtual Societies. *Proceedings of the Fifth Conference of the European Social Simulation Association (ESSA08)*
- Viroli M., Denti E. and Ricci A. (January, 2007) Engineering a BPEL orchestration engine as a multi-agent system" *Science of Computer Programming* 66 226–245.
- Von Martial F. (1992): *Co-ordinating Plans of Autonomous Agents*. LNAI 610, Springer.
- W3C OWL-S (Accedido en 2012) <http://www.w3.org/Submission/OWL-S/#5>.
- Walsh A.E. (2002) "UDDI, SOAP, and WSDL: The Web Services Specification," 1st edition ed. 2002, Pearson Education.
- Walsh A.E. (2002) "UDDI, SOAP, and WSDL: The Web Services Specification," 1st edition ed. 2002, Pearson Education.
- Wang B. and Liu L. (2012). *Ontology-Based Multi-Agent Diagnostic System of Enterprise Management*.
- Wang L., and Lazewsky G. (2008). *Scientific Cloud Computing: Early Definition and Experience*. Service Oriented Cyberinfrastructure Lab, Rochester Institute of Technology.
- Wang, G., Wong, T.N., and Wang, X. (2012). An Ontology based Approach to Organize Multi-Agent Assisted Supply Chain Negotiations, *Computers & Industrial Engineering*, doi: <http://dx.doi.org/10.1016/j.cie.2012.06.018>.
- Weerawarana, S., Curbera, F., Leymann, F., Storey, T. and Ferguson. (2005) *D.:Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging and More*. Prentice-Hall, Upper Saddle River.
- Wegner P. (1997) Why Interaction is more Powerful yhan Computing. *Communications of the ACM* 5(40), 80-91.
- Wegner P. (1997). Why Interaction is more Powerful than Computing. *Communications of the ACM* 5(40), 80-91.
- Wei-TekTsai, Sun X., and Balasooriya, J. *Service-Oriented Cloud Computing Architecture* (April, 2010). *Information Technology: New Generations (ITNG)*, 2010 Seventh International Conference pp 684 - 689.

- Werner E. (1989). "Cooperating agents: A unified theory of communication and socialstructure". Distributed Artificial Volumen II, pp 3-36. Pitman Publishing: London andMorgan Kaufmann: San Mateo, CA.
- Weyns D. and Georgeff M. (2010). Self-Adaptation Using Multiagent Systems. Software, IEEE Volume: 27, Issue: 1.
- White S. (2005). Using BPMN to Model a BPEL Process. IBM Corp., United States.
- White S. (October, 2006). Introduction to BPMN. IBM Software Group.
- Wiederhold G. (1992) Mediators in the Architecture of Future Information Systems. IEEE Computer Magazine, March 1992.
- Wohed P., van der Aalst W. M. P. Dumas M. and ter Hofstede A. H. M. (October, 2003). Analysis of Web services composition languages: The case of BPEL4WS. In Proceedings of the 22nd International Conference on Conceptual Modeling (ER'2003), Chicago (pp. 200-215).
- Wooldridge M. (2002). An Introduction to Multiagent Systems. Chichester, England, John Wiley & Sons, ISBN 047149691X.
- Xu Y., Singh N. and Deshpande S. (2011) Reuse by Placement: A Paradigm for Cross-Domain Software Reuse with High Level of Granularity, Top Productivity through Software Reuse, Lecture Notes in Computer Science Volume 6727, 2011, pp 69-77
- Yukio S., Silva H.m and Barthès J.P. (April 2011) . Agent and multi-agent applications to support distributed communities of practice: a short review.
- Zambonelli F. and Parunak H. (2002) From design to intention: Signs of a revolution. In Proc. 1st Int. Joint Conference on AAMAS: 455- 456.
- Zambonelli F., Gleizes M.P., Mamei M. and Tolksdorf R. (2004) Spray computers: frontiers of self-organisation for pervasive computing. Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises - WETICE04 (págs. 397-402). IEEE Computer Society
- Zambonelli F., Jennings N. R., and Wooldridge M. (2003) Developing multiagent systems: The Gaia methodology. ACMTransactions on Software Engineering and Methodology, 12(3): 317-370, July 2003.
- Zeng L., Benatallah B., Dumas M., Kalagnanam J., and Sheng Q. (2003). Quality Driven Web services Composition.
- Zhang Q., Cheng L., and Boutaba R. (April, 2010). Cloud computing: state-of-the-art and research challenges. J Internet Serv Appl.
- Zsu and M.T. and Valduriez P. (1999). Principles of Distributed Database Systems, 2nd edition. Prentice-Hall,

7.1 REFERENCIAS WEB

- Active endpoints. (accedido en 2009) In Depth. <http://www.activevos.com/>
- ActiveBPEL (Accedido en 2012). <http://swik.net/ActiveBPEL>
- Amazon Elastic Computing Cloud, (Accedido 2012). aws.amazon.com/ec2
- Amazon Security. Amazon Web Services Security. (Accedido 2012). <http://aws.amazon.com/es/security/>
- Answers. (Accedido 2012). Cloud Computing. <http://www.answers.com/topic/cloud-computing>
- Apache Cassandra. (Accedido en 2012). <http://cassandra.apache.org/>
- Azure Platform. (2011) Microsoft Azure. Introducing Microsoft Azure Platform.
- Bluelock. (Accedido en 2012). <http://www.bluelock.com/>
- CAR (2010). Arquitectura Orientada a Servicios (SOA) Cómo reformular la Arquitectura Corporativa para alcanzar el alto rendimiento. Un estudio publicado por el Centro de Alto Rendimiento de Accenture (CAR)
- CherryPy. (Accedido 2012). <http://www.cherrypy.org/>
- Citrix. (Accedido en 2012). <http://www.citrix.es/lang/Es-es/home.asp>
- CouchDB.(Accedido en 2012). <http://couchdb.apache.org/>
- Django. (Accedido 2012). <http://django.es/>
- Dropbox. (Accedido 2012). <https://www.dropbox.com/>
- Elasticdrive Project . (Accedido 2012). <http://www.elasticdrive.com/>

- EuroCloud Deutschland_eco eV (2011) Eurocloud star audit saas certificate. <http://www.saas-audit.de>
- FIPA (2012) <http://fipa-os.sourceforge.net/>
- FIPA (2012a) <http://www.fipa.org>
- FlexiScale Cloud Comp and Hosting, (Accedido 2012). www.flexiscale.com
- GoGridCloud Hosting (Accedido 2012), Cloud Computing and Hybrid Infrastructure from GoGrid, <http://www.gogrid.com>
- Google App Engine, (Accedido 2012). URL <http://code.google.com/appengine>
- Google Developers. ¿Por qué App Engine?. (Accedido en 2012). <https://developers.google.com/appengine/whyappengine?hl=es>
- GramaticasFormales (Enero, 2011). Web Services. <http://gramaticasformales.wordpress.com/2011/01/24/web-services/>
- IBM BPEL4WS (Accedido en 2012). <http://www.ibm.com/developerworks/webservices/library/ws-bpelcol4/>
- IBM. (2009) WebSphere Business Integration Adapters. http://publib.boulder.ibm.com/infocenter/wbihelp/v6rxmx/index.jsp?topic=/com.ibm.wbia_adapters.doc/doc/webservices/webservices17.htm
- IBM. (January, 2008). How service-oriented architecture (SOA) impacts your IT infrastructure Satisfying the demands of dynamic business processes. IBM Global Technology Services
- IBM. IBM Blue Cloud project (Accedido 2012). <http://www-03.ibm.com/press/us/en/pressrelease/22613.wss/>
- INSA, NEBUSENS, DIVISAIT, JLLORENS and USAL (2012) Análisis del Estado del Arte y la Problemática.
- JADE (2012) <http://jade.tilab.com/>
- Joyent. (Accedido en 2012). <http://joyent.com/>
- Microsoft (2010). Service Oriented Architecture Infrastructure Business agility through service virtualization.
- Microsoft SQL Server (Accedido en 2012). <http://www.microsoft.com/es-es/sqlserver/default.aspx>
- MongoDB. (Accedido en 2012). <http://www.mongodb.org/>
- MySQL. (Accedido en 2012). <http://www.mysql.com/>
- Nimbus. Nimbus Project (Accedido 2012). <http://workspace.globus.org/clouds/nimbus.html/>
- North K. (Accedido en 2011) "SQL, NoSQL or SomeSQL?", Dr. Dobb's,
- OASIS (August, 2006). Reference Model for Service Oriented Architecture 1.0 Committee Specification.
- OASIS WSBPEL TC (2007) Web services business process execution language version 2.0. Tech rep, OASIS. Available at <http://docs.oasis-open.org/wsbpel/2.0/OS/>
- Oracle BPEL Manager (Accedido en 2012) <http://www.oracle.com/technetwork/middleware/bpel/overview/index.html>
- Oracle Database (Accedido en 2012). <http://www.oracle.com/es/products/database/index.html>
- OWL (2012) <http://www.w3.org/TR/owl-features/>
- Point, Tutorials. (2009) What are Web services. http://www.tutorialspoint.com/webservices/what_are_web_services.htm
- Pylons. (Accedido 2012). <http://www.pylonsproject.org/>
- Rackspace, Dedicated Server (Accedido 2012), Managed Hosting, Web Hosting by Rackspace Hosting, <http://www.rackspace.com>
- S3. Amazon Simple Storage Service (Accedido 2012).. <http://aws.amazon.com/s3/>
- Salesforce CRM (Accedido 2012), www.salesforce.com/platform
- SAP Business ByDesign (Accedido 2012), www.sap.com/sme/solutions/businessmanagement/businessbydesign/index.epx
- SearchCloudComputing (2012). <http://searchcloudcomputing.techtarget.com/photostory/2240149038/Top-10-cloud-providers-of-2012/1/Introduction#contentCompress>
- Terremark. (Accedido en 2012). <http://www.terremark.es/default.aspx>
- Thomas (2010) <http://thomas-tin.usal.es/>

- UDDI (2002). UDDI Technical Committee. Universal Description, Discovery and Integration (UDDI). <http://www.oasis-open.org/committees/uddi-spec/>
- VMWare (Accedido en 2012). <http://www.vmware.com/es/>
- W3C (May, 2000) Simple Object Access Protocol (SOAP) 1.1 W3C Note
- W3C. Christensen E., Curbera F., Meredith G. and Weerawarana S. (March, 2001). Web Services Description Language (WSDL) 1.1
- Web2Py (Accedido 2012). <http://www.web2py.es/welcome/default/index>
- WidowsServer. Microsoft Windows Server. (Accedido en 2012). <http://www.microsoft.com/es-es/server-cloud/ws2012/default.aspx>
- Windows Azure (Accedido 2012)., www.microsoft.com/azure
- WSDL (2009) <http://www.w3.org/TR/wsdl>