



UNIVERSIDAD NACIONAL DE COLOMBIA

Modelo de Interoperabilidad Middleware para la Integración de Agentes Móviles Inteligentes con Redes de Sensores Inalámbricos

Ing. Alberto Alejandro Piedrahita Ospina

Universidad Nacional de Colombia

Facultad de Minas

Escuela de Sistemas

Medellín, Colombia

2012

Modelo de Interoperabilidad Middleware para la Integración de Agentes Móviles Inteligentes con Redes de Sensores Inalámbricos

Ing. Alberto Alejandro Piedrahita Ospina

Tesis presentada como requisito parcial para optar al título de:
Magíster en Ingeniería – Ingeniería de Sistemas

Director:

PhD. Demetrio Arturo Ovalle Carranza

Codirector:

MSc. Alcides de Jesús Montoya Cañola

Línea de Investigación:

Inteligencia Artificial

Grupo de Investigación:

Grupo de Investigación y Desarrollo en Inteligencia Artificial, GIDIA

Universidad Nacional de Colombia
Facultad de Minas, Escuela de Sistemas
Medellín, Colombia

2012

Lo que sabemos es una gota de agua, lo que ignoramos es el océano.

Isaac Newton

Agradecimientos

El autor expresa sus agradecimientos a:

DIME (Dirección de Investigaciones de la Universidad Nacional de Colombia - Sede Medellín), quien financió el proyecto de investigación titulado: “Modelo de sistema híbrido inteligente para el monitoreo de variables físicas usando redes de sensores inalámbricos y sistemas multi-agente” con código: 20201007312, de la Universidad Nacional de Colombia – Sede Medellín.

Facultad de Minas por la beca de facultad con exención de los derechos de matrícula concedida para llevar a cabo los estudios de maestría.

Demetrio Arturo Ovalle Carranza PhD., profesor de la Universidad Nacional de Colombia, director de la tesis; por sus valiosas y oportunas orientaciones, lo mismo que su dedicación para culminar con éxito el presente trabajo.

Alcides Montoya Cañola MSc., profesor de la Universidad Nacional de Colombia, por sus valiosos aportes en la co-dirección de la tesis.

Diego Luís Aristizábal Ramírez MSc., profesor de la Universidad Nacional de Colombia, por sus sabios consejos como amigo, su rectitud y responsabilidad hacia el trabajo que han demarcado mi camino.

Jaime Alberto Guzmán Luna PhD., por su conocimiento y por tener el honor de ser jurado de la presente tesis.

A mis compañeros de maestría Diana Carolina Restrepo Patiño, Tatiana Muñoz Hernandez, Silvia Restrepo y David Saldaña Santacruz, por todos los momentos que vivenciamos juntos.

A mis distinguidos amigos (as) y profesores.

Un agradecimiento especial a mi familia:

A mi madre Matilde Elena Ospina Ortiz, por el apoyo, voluntad y persistencia que presentó durante toda la vida para formarme como persona.

A mi hermana María Alejandra Piedrahita Ospina, por su amistad constante y por estar presente en los momentos más importantes de mi vida.

Finalmente un agradecimiento afectuoso a mi pareja Marylone Montoya Torres quien me ha brindado su amor y apoyo incondicional en todo momento.

Resumen

Las Redes de Sensores Inalámbricos (WSN, del inglés Wireless Sensor Network) son un conjunto de dispositivos esparcidos en un entorno determinado con el fin de percibir el entorno a su alrededor y obtener conocimiento de este. Este nuevo conocimiento del entorno es útil a nuevos sistemas de cómputo que necesitan observar el entorno de sus usuarios para ofrecer nuevas funcionalidades y satisfacer las nuevas demandas de la sociedad. Sin embargo, existe el problema de la heterogeneidad de plataformas y dispositivos que hacen sumamente engorrosa la integración de las WSN con sistemas cómputo habituales. Debido a esto en esta tesis de maestría se presenta un trabajo que resuelve esta dificultad a través de una capa intermedia "Middleware" que integra distintas tecnologías de WSN en una Red de Sensores Inalámbricos Integrada (IWSN, del inglés Integrated Wireless Sensor Network). La IWSN proporciona un conjunto de interfaces de programación (API) para el desarrollo de aplicaciones con capacidades de percibir su entorno.

Palabras claves

Sistemas distribuidos, teleinformática, Redes de sensores

Abstract

Wireless Sensor Networks (WSN) are a set of devices spread over a given environment in order to perceive the environment around them and gain knowledge. This new knowledge of the environment is useful to new computing systems that need to monitor the environment for its users and to offer new features. However, there is the problem of heterogeneity of platforms and devices that make it extremely cumbersome the integration of WSN with standard computing systems. The propose in this work is to resolve this

difficulty through an intermediate layer "middleware" that integrates different technologies of WSN in an Integrated Wireless Sensor Network (IWSN). The IWSN provide a set of Application Programming Interfaces (APIs) in order to develop applications with capacities to perceive their environment.

Keywords

Distributed Systems, Tele-informatics, Sensor Networks.

Contenido

Agradecimientos	VII
Resumen	9
Palabras claves	9
Abstract	9
Keywords	10
Contenido	11
Lista de Figuras	15
Lista de Tablas	16
Glosario	17
1. INTRODUCCIÓN, OBJETIVOS Y METODOLOGÍA	20
1.1. Motivación	20
1.2. Aportes.....	22
1.3. Definición del problema de investigación	23
1.4. Hipótesis de investigación	23
1.5. Preguntas de Investigación.....	23
1.6. Objetivos	24
1.6.1. Objetivo general	24
1.6.2. Objetivos específicos.....	24
1.7. Alcance.....	24

1.8.	Metodología de trabajo	25
1.9.	Organización del documento	27
1.10.	Difusión de resultados.....	27
1.10.1.	Publicación en revistas indexadas y capítulos de libro.....	27
1.10.2.	Publicaciones en eventos internacionales	27
1.10.3.	Publicaciones en eventos nacionales	28
1.10.4.	Artículos en revisión	28
2.	AGENTES MÓVILES INTELIGENTES Y WSN.....	29
2.1.	Inteligencia Artificial Distribuida y Sistemas Multi-Agente	29
2.1.1.	Agentes de Software	29
2.1.2.	Agentes Móviles	31
2.1.3.	Modelo de razonamiento BDI (Belief, Desires and Intentions)	32
2.1.4.	Sistemas Multi-Agente.....	34
2.1.5.	Estándar para Tecnologías Basadas en Agentes	35
2.1.6.	Ontologías y SMA.....	36
2.2.	Redes de Sensores Inalámbricos ó WSN.....	37
2.2.1.	Factores que influyen en una WSN	39
2.2.2.	Programación de una WSN	40
2.3.	Estrategias para la Integración de Agentes Móviles y WSN: Estado del arte	41
2.3.1.	Agilla Framework.....	42
2.3.2.	Agent Factory Framework.....	42
2.3.3.	Agent Factory Micro Edition (AFME).....	43
2.3.4.	ActorNet Platform	43
2.3.5.	SoSAA framework	44

2.3.6.	Famiware.....	44
2.4.	Limitaciones encontradas en los trabajos relacionados	44
2.5.	Middleware	45
3.	MODELO PROPUESTO.....	46
3.1.	Consideraciones del modelo.....	46
3.1.1.	Heterogeneidad de dispositivos	46
3.1.2.	Interoperabilidad.....	47
3.1.3.	Escalabilidad	47
3.1.4.	Modularidad de programación	48
3.1.5.	Transparencia.....	48
3.1.6.	Formalización del entorno.....	48
3.2.	Modelo Planteado.....	49
3.2.1.	Arquitectura en Capas	49
3.2.2.	Modularidad del Modelo Propuesto	52
3.3.	Representación del entorno.....	57
4.	IMPLEMENTACIÓN DEL PROTOTIPO Y RESULTADOS.....	59
4.1.	Caso de Estudio	59
4.2.	Herramientas utilizadas	60
4.3.	Sistema Multi-Agente.....	60
4.4.	Escenario de Validación	63
4.5.	Resultados	65
4.6.	Comparación con trabajos relacionados	66
5.	CONCLUSIONES Y TRABAJO FUTURO.....	69
	REFERENCIAS BIBLIOGRÁFICAS.....	71

Lista de Figuras

Figura 2.1. Ilustración de un agente en su entorno, tomada de (Wooldridge, 2009).....	30
Figura 2.2. Un agente en su entorno. Tomada de (Wooldridge, 1999).....	32
Figura 2.3. Representación gráfica de una WSN	38
Figura 3.1. Arquitectura multi-capas para el modelo middleware propuesto	51
Figura 3.2. Diagrama de Módulos: Capa Física	53
Figura 3.3. Interfaz gráfica del Administrador de Dispositivos	55
Figura 3.4. Servicios ofrecidos por el módulo DS.....	56
Figura 3.5. Ontología genérica del entorno	58
Figura 4.1. Entorno para el caso de estudio: Restaurante	60
Figura 4.2. Diagrama de Secuencia: Negociación entre agentes.....	63
Figura 4.3. Escenario utilizado en las pruebas realizadas.....	64
Figura 4.4. Resultado de negociación entre agentes: intersección de regiones	66

Lista de Tablas

Tabla 1-1 Metodología de trabajo	25
Tabla 4-1. Preferencias de los usuarios	64
Tabla 4-2. Tabla de comparación entre estrategias de integración de agentes con WSN	68

Glosario

AD: Administrador de dispositivos, en el modelo propuesto es un modulo quien tiene como tarea integrar, conocer y gestionar toda la infraestructura de dispositivos WSN existente.

ACL: acrónimo inglés para Agent Communication Languages, son los lenguajes a través de los cuales los agentes pueden establecer actos de comunicación y entender dicha comunicación.

Agente de Software: Programa auto contenido capaz de controlar su proceso de toma de decisiones y de actuar, basado en la percepción de su ambiente, con el fin de lograr de uno o varios objetivos. Poseen adicionalmente características de autonomía, deliberación y proactividad.

API: acrónimo inglés de *Application Programming Interface*, es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por los programadores en el desarrollo de una aplicación.

CD: Comunicador de dispositivos, en el modelo propuesto es un modulo quien se encarga de realizar todas las comunicaciones hacia los dispositivos de una tecnología en particular.

BDI: Es un modelo de razonamiento lógico que establece que cada agente posee Creencias (Beliefs), Deseos (Desires) e Intenciones (Intentions), y conforme a dicho conocimiento toma sus decisiones.

Bundle: es un módulo o componente de software que envuelve ciertas funcionalidades de manera oculta, conformando sistemas altamente desacoplados.

DS: En el modelo propuesto es el módulo encargado de ofrecer las funcionalidades de la WSN a través de interfaces y posteriormente servicios.

FIPA: acrónimo inglés para The Foundation for Intelligent Physical Agents, es un comité de la IEEE para la promoción de tecnologías basadas en agentes y la interoperabilidad de sus estándares con otras tecnologías.

GUI: acrónimo inglés para Graphics User Interface, es la interfaz gráfica través de la cual un usuario puede interactuar con un sistema informático.

IA: Abreviatura de Inteligencia Artificial, es la rama de las ciencias computacionales que se enfoca en el diseño de sistemas computarizados inteligentes que tengan capacidades humanas tales como: comprensión del lenguaje, aprendizaje, razonamiento, resolución de problemas, entre otros.

IAD: es la rama de la inteligencia artificial que busca resolver problemas complejos, que se caracterizan por tener una distribución funcional intrínseca y requieren de una solución distribuida.

Inteligencia Ambiental: es un área en desarrollo que busca sistemas sensibles al contexto para conseguir espacios y entornos inteligentes que de manera amigable y en lo posible transparente, interactúen con los usuarios.

JADE: Abreviatura de Java Agent Development Framework, es un entorno de desarrollo para construir aplicaciones basadas en agentes, soportado en las especificaciones de FIPA para sistemas multi-agente.

Middleware: Es básicamente una capa de programación intermedia que realiza el puente o la integración entre dos tecnologías con niveles de complejidad diferentes.

Ontología: es la descripción de los conceptos y relaciones entre ellos, que pueden formar parte del conocimiento de un agente o una sociedad de agentes.

OSGi: es un marco de trabajo modular que permite dividir sistemas complejos y monolíticos en sistemas flexibles y modulares para facilitar tanto su diseño e implementación como su posterior mantenimiento y actualización. Un sistema OSGi se encuentra dividido en n cantidad de bundles.

SMA: acrónimo para Sistema Multi-Agente, es una sociedad organizada compuesta por agentes que interactúan entre sí, con el fin de cooperar o competir con un fin determinado.

WSN: acrónimo inglés para Wireless Sensor Network, es un sistema distribuido compuesto por nodos desplegados en red, cada nodo es un dispositivo electrónico con capacidades de percepción y actuación.

WSNI: acrónimo inglés para Wireless Sensor Network Integrated, es una WSN propia del modelo propuesto conformada por distintas tecnologías de WSN que se integran entre si para componer una WSN más amplia y versátil.

1. INTRODUCCIÓN, OBJETIVOS Y METODOLOGÍA

En este capítulo de Tesis de Maestría, se pretende dar una perspectiva general a los contenidos de este trabajo de investigación. El propósito de esta tesis es proponer un modelo de interoperabilidad middleware para la integración de Sistemas Multi-Agente con Redes de Sensores Inalámbricos ó WSN (acrónimo inglés para Wireless Sensor Networks), con el fin de dotar agentes de software con capacidades para percibir el mundo, tomar decisiones y actuar sobre el entorno.

1.1. Motivación

La tendencia mundial actual en las áreas de Ciencias de la Computación, Ingeniería de Software, Interacción Hombre-Computador e Informática es desarrollar sistemas más amigables y conscientes de las necesidades de los usuarios (Sharp et al, 2007). Sin embargo, para lograr esto, es importante que los sistemas evolucionen a sistemas capaces de percibir y obtener información del entorno, para luego razonar y tomar decisiones, y posteriormente actuar sobre su entorno ejecutando acciones.

La Computación Ubicua (Ubiquitous Computing) (Weiser, 2002) es un área de la computación que promueve el desarrollo de una computación omnipresente, de modo que esté presente en la vida cotidiana de los usuarios. Actualmente, la computación ubicua se vale de dispositivos móviles para lograr su fin, tales como: Celulares, PDAs, Tablet, así como también la incorporación de tecnología en muchos de los artefactos y entornos de uso común, entre ellos: electrodomésticos, muebles, restaurantes, museos, transporte público, entre otros. Esto ha dado origen al concepto de Computación Invasiva (Pervasive Computing) (Estrin et al, 2002), el cual busca embeber tecnología en la mayor cantidad de objetos y espacios, logrando una sociedad más automatizada y comunicada, haciendo de esta forma más fácil las labores cotidianas de los usuarios.

La movilidad que se aprecia en la computación actual, conlleva a pensar que la ubicuidad ya es una realidad. Los bajos costos de los dispositivos móviles y chips electrónicos de radiofrecuencia, han facilitado una invasión de computación inalámbrica sin precedentes. Si bien la computación ubicua permite reconocer algunas bondades y un alto potencial para aplicaciones móviles, carece de un factor importante en la computación de próxima generación, la sensibilidad al contexto (traducción del inglés para context-aware), la cual

se refiere a sistemas computacionales que se adapten a las necesidades, gustos y preferencias de usuarios y adicionalmente a los cambios en el entorno. La inteligencia Ambiental o Aml (acrónimo inglés para Ambient Intelligence) (Preuveneers et al, 2004) enmarca sistemas sensibles al contexto para conseguir espacios y entornos inteligentes que de manera amigable y en lo posible transparente, interactúen con los usuarios.

Los sistemas sensibles al contexto son la próxima generación de la computación, y proveerán a los usuarios un mayor número de servicios para satisfacer sus necesidades. No obstante, para lograr dichos sistemas es necesario diseñar un soporte computacional y de instrumentación que permita ver y percibir el mundo a través de perceptores, que pueden ser sensores ó cámaras, con los cuales se extraiga información y conocimiento del entorno.

En este aspecto hay una brecha grande en cuanto a la integración de tecnologías computacionales tradicionales y el soporte de instrumentación necesario. No se tienen modelos de integración óptimos, lo cual hace difícil la tarea del desarrollador a la hora de construir sistemas sensibles al contexto.

En el área de la electrónica, las comunicaciones y la computación, una tecnología relativamente reciente son las WSN (Mainwaring et al, 2002), las cuales son conjuntos de pequeños dispositivos electrónicos inalámbricos denominados nodos, los cuales cuentan con un importante componente en instrumentación electrónica para medir variables físicas a su alrededor. Las variables físicas más comunes que son monitoreadas mediante nodos de WSN son: temperatura, humedad, luminosidad, aceleración, magnetismo, entre otras.

Actualmente existe una gran diversidad de dispositivos destinados a conformar una WSN, existen desde plataformas comerciales con potentes capacidades, hasta diseños electrónicos libres con menores prestaciones. Los dispositivos varían en aspectos tales como procesamiento, memoria, almacenamiento, protocolo de comunicación, fuente de energía, sin embargo, la principal diferencia se encuentra en sus capacidades para interactuar con su entorno, ya que poseen diferentes sensores y actuadores. Toda esta diversidad hace de los sistemas de WSN ambientes sumamente heterogéneos haciendo muy difícil la tarea del desarrollador, quien tendrá que aprender y dominar distintos lenguajes de programación de bajo nivel para la programación de diferentes plataformas WSN.

En esta Tesis de Maestría se propone un modelo de interoperabilidad middleware para la integración de sistemas computacionales tradicionales y Redes de Sensores Inalámbricos, de modo que se selle la brecha que existe entre ambas tecnologías y se suministre al desarrollador de herramientas, con el fin de facilitar la fabricación de Aplicaciones Sensibles al Contexto. Adicionalmente, dicho modelo favorece la Integración de Agentes Móviles Inteligentes los cuales provienen del campo de la IAD (Inteligencia Artificial distribuida) con WSN.

1.2. Aportes

Los principales aportes de esta tesis de maestría son:

- Un modelo de interoperabilidad middleware con una arquitectura multicapas para la integración de Sistemas Multi-Agente con WSN. El modelo posee las siguientes características:
 - Conformidad a los estándares FIPA (Foundation for Intelligent Physical Agents) para Sistemas Multi-Agente (FIPA, 2011).
 - Interoperabilidad en la utilización de distintos dispositivos WSN, resolviendo así la heterogeneidad que existe en el mercado. La heterogeneidad de dispositivos se presenta tanto en recursos limitados de Procesamiento y Almacenamiento, así como también en características y capacidades, tales como la diversidad de sensores y actuadores.
 - Ontología del entorno a través de la cual se representa el conocimiento proveniente del entorno y se establecen los formalismos de comunicación para los agentes que compongan el modelo.
- Un prototipo en el área de la Inteligencia Ambiental, el cual a partir de las preferencias del usuario configurar el entorno para hacer los espacios más amigables. Así por ejemplo, si un usuario tiene como preferencia un cierto nivel de temperatura, el entorno será ajustado a dicho nivel de temperatura.

1.3. Definición del problema de investigación

El principal problema por el cual se emprende este trabajo de investigación es la falta de un soporte de interoperabilidad para la integración de Agentes Móviles y WSN, con el fin de que los agentes de software puedan interactuar en el entorno físico. Otra parte fundamental del problema consiste en la imposibilidad de los modelos actuales de proveer un soporte de inteligencia artificial claro y acorde a los estándares de agentes de software existentes, tales como FIPA (FIPA, 2011). Lo cual se vuelve necesario si lo que se pretende es que la WSN adquiera un grado de inteligencia tal que pueda resolver las situaciones que le presente el entorno, acoplarse a las características del contexto y adaptarse a las necesidades del usuario.

En esta tesis de maestría se pretende atacar el problema de la interoperabilidad para la integración de Agentes Móviles Inteligentes y las WSN.

1.4. Hipótesis de investigación

La hipótesis de investigación en esta tesis de maestría es el diseño y construcción de un modelo de interoperabilidad Middleware que permita la integración de ambas tecnologías: Sistemas Multi-Agente y WSN. El modelo resuelve en primer lugar la diversidad de las WSN, y en segundo lugar propone un mecanismo para integrar una programación de alto nivel orientada a agentes móviles inteligentes con las capacidades de interacción con el entorno de una WSN, para proveer una inteligencia del contexto a nuevas aplicaciones.

1.5. Preguntas de Investigación

Luego de definir la hipótesis a abordar, surgen las siguientes preguntas de investigación:

- ¿Cuál es la mejor forma de proveer inteligencia a una WSN utilizando agentes de software sin perder de vista su carácter distribuido?
- ¿Cómo lograr que un nodo de WSN pueda tomar decisiones y actuar sobre su entorno basándose en la información que percibe del entorno?
- ¿Cómo modelar agentes de software que se ajusten a las limitaciones de recursos de una WSN, sin perder su capacidad de deliberar?

- ¿Cómo integrar una programación de alto nivel propia de agentes de software al interior de los nodos de una WSN, los cuales poseen una programación de bajo nivel?
- ¿Cómo diseñar una WSN inteligente aplicable a un caso de estudio en el área de Inteligencia Ambiental?

1.6. Objetivos

A continuación se presentan los objetivos planteados en esta tesis de maestría:

1.6.1. Objetivo general

Desarrollar un modelo de interoperabilidad middleware que permita la integración de WSN con Sistemas Multi-Agente y construir un prototipo en el área de Inteligencia Ambiental que valide el modelo propuesto.

1.6.2. Objetivos específicos

A continuación se listan los objetivos específicos:

- Caracterizar los conceptos principales de Sistemas Multi-Agentes, Agentes Móviles y WSN.
- Caracterizar las estrategias de integración que han sido propuestas en la comunidad científica para incluir agentes de software en WSN.
- Proponer un modelo de interoperabilidad middleware para integrar agentes de software conformes a FIPA en el interior de un Nodo de WSN.
- Proponer un Sistema Multi-Agente conforme al estándar FIPA basado en el modelo propuesto.
- Desarrollar un prototipo que valide el modelo propuesto y realizar pruebas para un caso de estudio específico.

1.7. Alcance

En el trabajo presentado en esta tesis de maestría se diseñó un modelo para integrar el funcionamiento de una WSN con agentes de software conformes con el estándar FIPA, adicionalmente los agentes de software adquieren las capacidades que ofrecen las WSN

para percibir el entorno por medio de sensores, para posteriormente razonar y tomar decisiones con la información adquirida, y en caso de ser necesario actuar sobre éste con actuadores.

En esta tesis de maestría se diseñó y desarrolló un prototipo que valida el modelo middleware de integración propuesto. El sistema cumple con la nueva tendencia en el desarrollo de software de aplicaciones conscientes o sensibles al contexto. La aplicación de acuerdo a las preferencias del usuario consigue modificar el entorno en el que éste se encuentra.

En esta tesis no se considera trabajar sobre las capacidades de una WSN ni de sus dispositivos, como lo sería incrementar sus recursos (procesamiento, memoria, energía, entre otros.) ó aumentar sus capacidades de percepción y actuación desarrollando nuevos componentes en instrumentación, tales como sensores y actuadores.

En esta tesis de maestría no se busca desarrollar una nueva plataforma para la construcción y desarrollo de Sistemas Multi-Agente, con el fin diseñar nuevos agentes más óptimos para una WSN ó más enfocados al conocimiento del entorno, simplemente se pretende proponer una capa de interoperabilidad middleware entre Sistemas Multi-Agente y WSN.

1.8. Metodología de trabajo

Una vez se plantearon los objetivos que señalan el alcance de este trabajo de investigación, se presenta en la Tabla 1-1 la metodología descrita a través de 5 fases que permitieron el logro de los objetivos propuestos.

Tabla 1-1 Metodología de trabajo

FASE	OBJETIVOS	ACTIVIDADES
Fase 1: Conformación del marco teórico revisión del estado del arte.	<p>Caracterizar los conceptos principales de Sistemas Multi-Agentes, Agentes Móviles y WSN.</p> <p>Caracterizar las estrategias de integración que han sido propuestas en la comunidad</p>	<p>1.1 Revisión Bibliográfica sobre Sistemas Multi-Agente: características, arquitecturas, metodologías y plataformas.</p> <p>1.2 Revisión Bibliográfica sobre WSN.</p> <p>1.3 Revisión Bibliográfica de diferentes estrategias de integración entre agentes</p>

	científica para incluir agentes de software en WSN.	<p>móviles y WSN.</p> <p>1.4 Revisión Bibliográfica sobre el modelo de razonamiento de para agentes de software, conforme a FIPA.</p> <p>1.5 Revisión Bibliográfica sobre las distintas plataformas de WSN, encontrando aquellas que presenten mejores rendimientos.</p>
Fase 2: Diseño del modelo propuesto.	Proponer un modelo de interoperabilidad middleware para integrar agentes de software conformes a FIPA en el interior de un Nodo de WSN.	<p>2.1 Seleccionar una metodología para la construcción de SMA.</p> <p>2.2 Seleccionar una plataforma de WSN con recursos suficientes para introducir un agente de software dentro de ella.</p> <p>2.3 Definir un caso de estudio de manera que se apunte hacia éste en búsqueda de evaluar el rendimiento del modelo propuesto.</p>
Fase 3: Construcción del modelo propuesto y Diseño de un SMA para validar dicho modelo.	Proponer un Sistema Multi-Agente conforme al estándar FIPA basado en el modelo propuesto.	<p>3.1 Diseñar y modelar los agentes que serán implementados, para llevar a cabo las diferentes tareas en el sistema.</p> <p>3.2 Diseñar y modelar las relaciones que existen entre cada agente.</p> <p>3.3 Modelar la arquitectura Multi-Agente, así como la comunicación e interacción con su entorno.</p>
Fase 4: Implementación y validación del modelo propuesto.	Desarrollar un prototipo que valide el modelo propuesto y realizar pruebas para un caso de estudio específico.	<p>4.1 Implementar la metodología seleccionada para desarrollar el sistema Multi-Agente propuesto y proveer toda la documentación necesaria como un soporte del trabajo realizado.</p> <p>4.2 Implementar el Sistema.</p>

		4.3 Validar el sistema.
--	--	-------------------------

1.9. Organización del documento

A continuación se presenta la estructura del documento de tesis donde se presenta el desarrollo de tesis basado en el cumplimiento de los objetivos específicos. En el capítulo 1 se encuentra la introducción, los objetivos y la metodología utilizada en la construcción de este trabajo. Posteriormente el capítulo 2 presenta el marco teórico y estado del arte de la integración de Agentes Móviles Inteligentes y WSN. Por su parte en el capítulo 3 se plantea el modelo propuesto para la integración de Agentes Inteligentes con WSN. El capítulo 4 desarrolla un caso de estudio con el cual se valida el modelo propuesto y se analizan los resultados obtenidos. Por último el capítulo 5 presenta las conclusiones y el trabajo futuro de esta tesis de maestría.

1.10. Difusión de resultados

A continuación se presentan las publicaciones en revistas científicas, capítulos de libro y memorias en congresos nacionales e internacionales que se han realizado y han permitido la difusión de esta investigación.

1.10.1. Publicación en revistas indexadas y capítulos de libro

Piedrahita, A. Montoya, D. Ovalle. "Modelo de integración de agentes móviles en redes de sensores inalámbricos". Tendencias en Ingeniería de Software e Inteligencia Artificial. v.3. ISBN 978-958-44-5820-9. Octubre 2009.

Alberto Piedrahita Ospina, Alcides Montoya Cañola and Demetrio Ovalle Carranza. Performance Evaluation of an Intelligent Agents Based Model within Irregular WSN Topologies. Innovations in Computing Sciences and Software Engineering, 2010. Springer Netherlands. ISBN: 978-90-481-9112-3. pag. 571-576. 2010.

1.10.2. Publicaciones en eventos internacionales

Piedrahita, A.; Montoya, A.; Ovalle, D.; "Integration model of mobile intelligent agents within Wireless Sensor Networks,". LATINCOM '09, IEEE Latin-American Conference on Communications, 2009. 10-11 Sept. 2009.

Alberto Piedrahita Ospina, Alcides Montoya Cañola and Demetrio Ovalle Carranza. Performance Evaluation of an Intelligent Agents Based Model within Irregular WSN

Topologies. International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering, Cisse 2009. December 4-12, 2009.

1.10.3. Publicaciones en eventos nacionales

Alberto Alejandro Piedrahita Ospina, Alcides Montoya Cañola, Demetrio Arturo Ovalle Carranza. Modelo de Integración de Agentes Móviles Inteligentes en Redes de Sensores Inalámbricos. Tercer Encuentro de Investigadores en Inteligencia Artificial e Ingeniería de Software. Universidad Nacional de Colombia-Sede Manizales, Campus La nubia. Octubre 22 y 23. 2009.

Alberto A. Piedrahita O., Demetrio A. Ovalle C., Alcides J. Montoya C., Evaluación del desempeño de un modelo basado en Agentes Inteligentes en WSN con topologías irregulares. Quinto Congreso Colombiano de Computación, 5CCC. Universidad de Cartagena, Cartagena, Colombia. Abril 14-16, 2010.

1.10.4. Artículos en revisión

Alberto Alejandro Piedrahita O., Demetrio A. Ovalle C. y Alcides J. Montoya C. Evaluación del desempeño de la integración de Agentes Móviles en Redes de Sensores Inalámbricos. Revista de Ingeniería e Investigación, Universidad Nacional de Colombia – Sede Bogotá, ISSN 0120-5609.

2. AGENTES MÓVILES INTELIGENTES Y WSN

En este capítulo se realizará un recorrido por los conceptos teóricos que intervienen en esta tesis de maestría. Se presentan los dos temas centrales de este trabajo: las WSN y los Sistemas Multi-Agente. En cuanto a las WSN se realiza una caracterización de los dispositivos de WSN y los conceptos de básicos de las WSN. Adicionalmente se presentan las definiciones más importantes en cuanto a Sistemas Multi-Agente y Agentes Móviles Inteligentes. También se incluye una sección acerca del estándar FIPA para el Diseño y Desarrollo de Sistemas Multi-Agente. Finalmente se presentan las necesidades de integrar las tecnologías de programación multi-agente con WSN y trabajos relacionados en la comunidad científica.

2.1. Inteligencia Artificial Distribuida y Sistemas Multi-Agente

La Inteligencia Artificial (Russell & Norvig, 1995) es la rama de las Ciencias de la Computación que busca emular habilidades humanas en procesos computacionales, algunas de las más reconocidas áreas de la Inteligencia Artificial son la Visión Artificial, el Procesamiento de Lenguaje Natural, el Razonamiento Lógico, entre otras.

Actualmente la expansión de las Redes Informáticas alrededor del mundo ha hecho necesario que en todas las áreas de la Ciencias de la Computación sea considerado el aspecto distribuido para los procesos computacionales. La Inteligencia Artificial Distribuida (IAD) (O'Hare & Jennings, 1996) es una propuesta de la Inteligencia Artificial, la cual es usada fundamentalmente en la resolución de problemas complejos, que se caracterizan por tener una distribución funcional intrínseca y requieren de una solución distribuida a través de procesos de colaboración, negociación, movilidad, entre otros.

2.1.1. Agentes de Software

Un agente es una entidad o proceso de software (Wooldridge & Jennings, 1995) (Wooldridge, 1999) (Wooldridge, 2009) el cual es ubicado en algún entorno y allí es capaz de ejecutar acciones autónomas con el fin de lograr los objetivos o metas que se le han asignado. Para interactuar con su entorno un agente posee sensores (input) y actuadores (output) que le permiten percibir y actuar sobre su entorno, tal como se aprecia en la Figura 2.1.

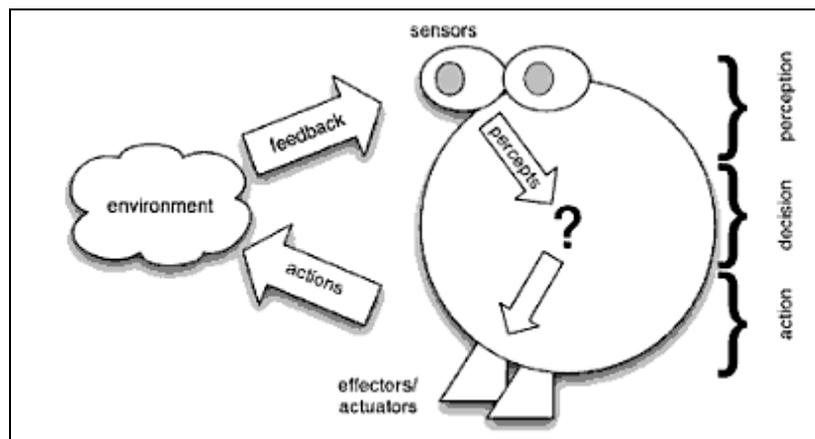


Figura 2.1. Ilustración de un agente en su entorno, tomada de (Wooldridge, 2009)

Fuera de las capacidades de percepción y actuación que posee un agente de software, se hacen necesarias nuevas capacidades y características, para que el agente sea inteligente y autónomo. Características tales como: deliberativo, reactivo, proactivo, comunicativo, social, móvil, negociador, cooperador, entre otras, son características deseables en agentes de software con el fin de tener agentes inteligentes y autónomos. A continuación se describen dichas características.

Agente deliberativo: es un agente que puede de manera autónoma tomar decisiones de acuerdo a la información o conocimientos que tiene del entorno. La deliberación puede ser conformada por distintas técnicas de la Inteligencia Artificial o incluso modelos híbridos de ellas, tales como la Lógica difusa, Redes Neuronales, Motores de Inferencia, Estadística Bayesiana, entre otras.

Agente reactivo: es aquel agente capaz de reaccionar de manera autónoma a estímulos en el entorno, dichos cambios son percibidos a través de sus sensores (input), lo cual dispara comportamientos o procesos internos en el agente para efectuar acciones.

Agente proactivo: es un agente capaz de proponer cambios en el entorno de forma autónoma después de un proceso de deliberación, un agente proactivo suele adaptarse a las situaciones que puedan surgir y presentar soluciones a dichas situaciones.

Agente comunicativo: es aquel agente que puede establecer actos de comunicación con otros agentes, con el fin de informar, proponer, aceptar o rechazar información hacia y desde otros agentes.

Agente social: es un agente que necesita interactuar con otros agentes para lograr sus objetivos, esta característica suele ser emulada del comportamiento social de muchas especies naturales, especialmente la humana.

Agente negociador: es un agente que interactúa con otros agentes proponiendo situaciones o planes de acción a otros agentes y adaptándose a las situaciones o planes que le son propuestos, siempre buscando optimizar sus resultados para lograr sus objetivos propios.

Agente cooperador: es un agente que interactúa con otros agentes buscando cooperar para lograr objetivos comunes entre los agentes involucrados.

2.1.2. Agentes Móviles

Los agentes móviles (Zhou & Gao, 2010) son procesos capaces de moverse a través de una red informática, ya sea LAN (Local Área Network) o WAN (Wide Area Network), migrando o clonando su código y estado de una máquina a otra. Interactuando con dispositivos extraños, recopilando información para luego volver a su origen con los datos obtenidos. Los agentes móviles han tenido una gran aceptación en sistemas distribuidos gracias a su movilidad, puesto que es mucho más eficiente que un agente se movilice hasta una ubicación remota y haga una búsqueda, a traer toda la información para luego ser procesada y filtrada.

Dos de las características más importantes en cuanto a Agentes Móviles son la Clonación y la Migración:

Migración: es el proceso por el cual un agente es capaz de desplazarse a través de una red informática. Existen dos tipos de migraciones: Una migración fuerte en la cual el agente puede cargar consigo su código y estado de una máquina a otra, y reanudar su labor en el máquina destino; el segundo tipo es una migración débil en la que el agente sólo lleva su código, por tal motivo el agente debe reiniciar nuevamente su ejecución en la máquina destino.

Clonación: en este proceso el agente tiene la habilidad de hacer una copia de sí mismo. El agente puede clonarse en la máquina actual o clonarse en una máquina remota de la red. Al igual que la migración, la clonación puede ser fuerte ó débil. La clonación fuerte es aquella en la cual el agente clon posee el mismo código y estado de su padre reanudando su ejecución. En la clonación débil el clon sólo hereda de su padre el código, por lo tanto empieza su ejecución desde el inicio.

2.1.3. Modelo de razonamiento BDI (Belief, Desires and Intentions)

Un agente (Wooldridge, 2009) se encuentra ubicado en un espacio geográfico caracterizado en un entorno cercano. Este entorno será el ambiente de trabajo que tendrá un agente para interactuar (Collier et al, 2000). Dicha interacción que se muestra en la Figura 2.2, la cual está concebida como la manera como el agente percibe y luego afecta el entorno (input-output).

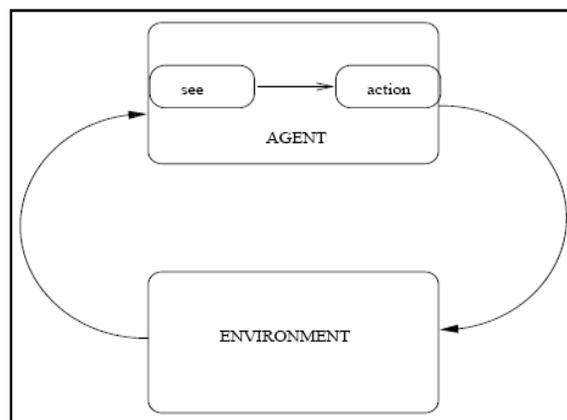


Figura 2.2. Un agente en su entorno. Tomada de (Wooldridge, 1999)

Para permitir percibir y afectar su entorno este modelo introduce dos tipos de extensiones al agente, los perceptores con los cuales percibe su entorno (sensores) y actuadores con los cuales afecta su entorno.

El modelo BDI (Rao & Georgeff, 1991)(Georgeff et al, 1999) establece que cada agente posee Creencias (Beliefs), Deseos (Desires) e Intenciones (Intentions). Un agente posee un conjunto de creencias las cuales serán percibidas a través de sus actuadores o derivadas de otras creencias (inferencia). El conjunto de deseos de un agente serán

aquellos objetivos o metas que el agente considere que debe llevar a cabo. Por último las intenciones son un subconjunto de los deseos a los que el agente se compromete a cumplir basándose en sus recursos, estos recursos por los cuales un agente podría o no comprometerse a realizar un deseo son: tiempo, dinero, procesamiento, espacio geográfico, consumo de energía, o cualquier ítem que el agente considere necesario para cumplir un deseo. A continuación se detalla un poco más en este proceso inspirado del razonamiento humano.

La arquitectura BDI tiene sus bases en el razonamiento práctico: el proceso por el cual un individuo decide momento a momento, cuáles acciones ejecutar en busca de sus metas. Este razonamiento práctico tiene dos procesos sumamente importantes: primero, decidir qué metas se quieren lograr y segundo, cómo lograrlas.

En este proceso de razonamiento práctico, un individuo tiene un conjunto de posibles opciones, las cuales son consideradas deseos que puede elegir para lograr metas. Las elecciones que tome el individuo de este conjunto de posibles serán las intenciones (Cohen et al, 1990), y estas serán escogidas por una serie de razones. Las razones por las cuales un individuo adopta una intención están íntimamente ligadas a las metas del individuo y a los recursos de los que dispone, tales como tiempo, dinero, inteligencia, espacio, energía, ánimo, amor, miedo, entre otros.

Si una intención es adoptada, la idea es que el individuo que la adoptó se comprometa a lograrla. Para ello se deben proporcionar recursos que sean destinados a este fin, hasta que la intención se satisfaga. Si bien se les denomina intenciones persistentes a aquellas intenciones que se mantienen vigentes a lo largo de la vida de un individuo, la secuencia normal de un razonamiento práctico es que las intenciones varíen a lo largo del tiempo. Un ejemplo de esto, en el caso de los agentes de software, sería una intención que fue adoptada desde hace mucho tiempo y no se ha cumplido, a pesar de la consistencia y tenacidad del agente, esta intención puede y deberá ser removida, para dar paso a otra intención que se pueda lograr. Otra razón por la cual una intención puede salir de vigencia, es cuando cambian las razones por las cuales la intención fue adoptada. Esta constante rotación de intenciones permite a un agente de software razonar basándose en sus capacidades, para tomar decisiones y ejecutar acciones para finalmente satisfacer sus objetivos.

2.1.4. Sistemas Multi-Agente

Un Sistema Multi-Agente (SMA) es una sociedad organizada compuesta por agentes semiautónomos que interactúan entre sí, ya sea para colaborar en la solución de un conjunto de problemas o en la consecución de una serie de objetivos individuales o colectivos. Estos agentes informáticos pueden ser homogéneos o heterogéneos y pueden tener metas comunes o no, pero siempre involucrarán algún grado de comunicación entre ellos (Wooldridge, 1999).

La comunidad científica actualmente cuenta con un estándar para la construcción de Agentes de Software, denominado FIPA (Foundation for Intelligent Physical Agents) (FIPA, 2011) (Ver sección 2.1.5), el cual consolida un conjunto de especificaciones para el desarrollo y la implementación de Agentes de Software. El estándar FIPA aborda desde distintos dominios de aplicación, el desarrollo de SMAs. Las áreas de aplicación de SMA son variadas, tales como: Planificación, Robótica, Comercio Electrónico, Minería de Datos, Sistemas tutoriales Inteligentes, Simulación de campos de batalla, Interpretación de Imágenes, entre otros.

Comunicación

Los Lenguajes de Comunicación de Agentes (ACL, acrónimo inglés para Agent Communication Languages) son los lenguajes a través de los cuales los agentes pueden establecer actos de comunicación y entender dicha comunicación. Siguiendo los lineamientos de (Labrou et al, 1999) un ACL brinda a los agentes la posibilidad de intercambiar información y conocimiento entre ellos. Genesereth y Fikes en (Genesereth & Fikes, 1992) han comparado la agencia con la capacidad de un sistema para intercambiar conocimiento con un ACL. Lo que diferencia ACL de esfuerzos pasados para comunicación como: invocación de métodos remotos, llamadas a procedimientos remotos o CORBA (Common Object Request Broker Architecture) son los objetos de discurso y la complejidad semántica. ACL se encuentra arriba de CORBA por dos razones:

- ACLs manejan proposiciones, reglas y acciones en lugar de simples objetos sin una semántica asociada.
- Un mensaje ACL describe un estado de deseo o petición en un lenguaje declarativo, en lugar de un procedimiento o método.

ACL sin embargo no cubre todo el espectro de información que podrían compartir las aplicaciones. Los agentes deberán intercambiar objetos complejos como: planes y metas, las cuales necesitarán compartir con otros agentes, o incluso experiencias y estrategias a largo plazo.

El ACL define los tipos de mensaje que los agentes pueden intercambiar y su significado. Los agentes sin embargo, no solo se comunican a través de mensajes simples, ellos tienen conversaciones orientadas a tareas como secuencias de mensajes compartidos que ellos seguirán, tales como una negociación o subasta. De la misma manera el alto nivel de conceptualización de las estrategias y comportamientos del agente conlleva a un formalismo suficiente para lograr el comportamiento comunicativo requerido en sociedades de agentes.

Cooperación y Negociación

Según (Rosenschein & Zlotkin, 1994) y (Guttman & Maes, 1998) la negociación es la toma de decisiones entre dos o más partes, las cuales de manera conjunta buscan un espacio de posibles soluciones con el objetivo de alcanzar un consenso. La economía y la teoría de juegos describen la negociación como una interacción en términos de protocolos y estrategias. Los protocolos de una negociación comprenden reglas del juego. Un ejemplo de un protocolo de negociación es una Subasta Inglesa No discriminatoria, donde la única jugada es publicar una oferta más alta que la oferta actual más alta por una cantidad mínima antes de que cierre la subasta.

2.1.5. Estándar para Tecnologías Basadas en Agentes

FIPA (acrónimo inglés de The Foundation for Intelligent Physical Agents) (FIPA, 2011) (O'Brien & Nicol, 1998) es un comité del IEEE (acrónimo inglés de Institute of Electrical and Electronic Engineers) para la promoción de tecnologías basadas en agentes y la interoperabilidad de sus estándares con otras tecnologías.

FIPA está compuesto por un conjunto de estándares ó especificaciones que tienen como objetivo promover la interacción entre agentes heterogéneos y los servicios entre ellos. Entre las principales especificaciones de FIPA se encuentran:

- **FIPA Abstract Architecture Specification:** En este documento y sus especificaciones derivadas se definen la arquitectura abstracta propuesta por FIPA para la construcción de Tecnologías Basadas en Agentes.
- **FIPA Agent Management Specification:** Dicho documento contiene especificaciones para la gestión de agentes, incluyendo servicios de gestión de agentes, ontologías, y transporte de mensajes dentro de la plataforma de agentes.
- **FIPA Nomadic Application Support Specification:** Este documento contiene las especificaciones para agentes móviles incluyendo conceptos propios de la tecnología informática actual. En el documento se especifican ambientes móviles con características distintas a sistemas distribuidos tradicionales, tales como: ancho de banda, latencia, retardo, tasa de error, interferencia, interoperabilidad, poder de cómputo, entre otras.

2.1.6. Ontologías y SMA

Una Ontología es la "formalización de una conceptualización", la descripción de los conceptos y relaciones entre ellos (Gruber, 1993), que pueden formar parte del conocimiento de un agente o una sociedad de agentes.

Las ontologías nacen de la necesidad de representar el conocimiento de un agente a través de un método formal. En el contexto de los Sistemas Multi-Agente las ontologías son fundamentales para el ambiente distribuido que surge entre las entidades (Agentes) que componen el sistema, ya que gracias a la ontología se establecen actos de comunicación complejos entre los agentes. Las dificultades a continuación se benefician con el uso de ontologías:

- Abundancia de comunicación entre agentes.
- Interoperabilidad de sistemas y plataformas.
- Problemas semánticos.

Para que los agentes logren establecer una comunicación entre ellos, deben conocer el mismo idioma, vocabulario y protocolos (McGuinness & Van Harmelen, 2004). Es aquí donde entra a jugar un papel importante la ontología en agentes de software, ya que es

gracias a una ontología que se representa el idioma, el vocabulario y los protocolos de comunicación.

Una ontología habitualmente se representa a través de tres partes fundamentales Conceptos, Acciones y Predicados, es a través de estos tres ítems que se elaboran complejos arboles de conocimiento, con los cuales los agentes construyen su conocimiento.

- **Los conceptos:** son entidades simples a través de las cuales se representan sustantivos: objetos, actores, entre otros., los cuales tienen características o propiedades particulares. A través de conceptos los agentes almacenan el conocimiento de su entorno. Algunos ejemplos de conceptos son: Casa, Fruta, Libro, Vendedor, Comprador, Policía, Ladrón, entre otros.
- **Las acciones:** son el ejercicio de cambiar el estado actual de un concepto a otro estado. Una acción siempre se ejecuta sobre, hacia o desde un concepto particular para cambiar el estado de este. Algunos ejemplos de acciones son: Vender, Comprar, Lanzar, Arrestar, Estacionar, entre otros.
- **Los predicados:** son conceptos que han sufrido un cambio, los cuales han sido afectados por una acción determinada. Los predicados siempre tienen asociado un concepto el cual ha sido afectado. Algunos ejemplos de predicado son: Fruta vendida, Libro comprado, Ladrón Arrestado, Pelota Lanzada, entre otros. Nótese el carácter de adjetivo que tiene un predicado sobre un concepto.

Según (Preuveneers et al, 2004), un ejemplo para el uso de una ontología es el entorno en aplicaciones para Inteligencia Ambiental, donde dichas aplicaciones pueden disponer de información acerca del contexto en el cual ellas operan. Para lograr esto diversa información deberá ser ensamblada para construir una representación del contexto en donde se desempeñará la aplicación. Para lograr interoperabilidad entre ambientes inteligentes es necesario que la terminología del contexto sea común a todos los entes involucrados, y de esta manera la comunicación sea transparente entre ellos.

2.2. Redes de Sensores Inalámbricos ó WSN

Debido a la necesidad de medir y controlar diferentes variables físicas en entornos que pueden ser hostiles, inaccesibles o extensos, nacen las Redes de Sensores Inalámbricos

ó WSN (acrónimo inglés para Wireless Sensor Networks) (Mainwaring et al, 2002)(Yick et al, 2008). Las WSN se definen como sistemas distribuidos, y al igual que las redes de cómputo tradicionales están compuestas por nodos de red. Ver Figura 2.3. Cada nodo es un pequeño dispositivo electrónico embebido estratégicamente en un entorno físico, el cual posee capacidades de procesamiento, almacenamiento y comunicación inalámbrica. Adicionalmente, cuentan con puertos de entrada y salida para interactuar con su medio ambiente cercano; los puertos de entrada a menudo son utilizados para instalar sensores con el fin de sentir cambios en su entorno o monitorear variables ambientales (Perceptores); los puertos de salida, por su parte, son conversores análogo-digitales que pueden controlar periféricos con el fin de manipular el medio ambiente cercano (Actuadores).

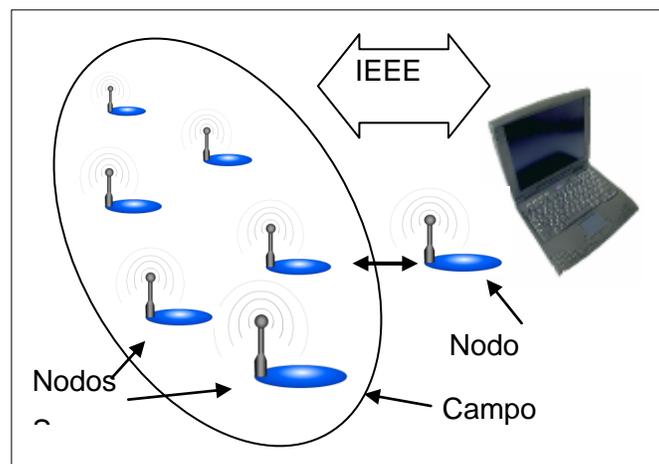


Figura 2.3. Representación gráfica de una WSN

Cada uno de los nodos de una WSN es ubicado en un espacio geográfico para interactuar con el entorno. También se conoce con el nombre de "mote" proveniente del inglés, como una mota de polvo invisible que tiene las capacidades de recolectar y procesar datos, para posteriormente enrutarlos hacia el nodo "Sink" o "Base", el cual actúa como gateway o nodo recolector de la red para la extracción de datos. Cabe señalar que la WSN actúa a través de una arquitectura ad-hoc de múltiples saltos, por lo cual todos los nodos tienen características similares y no existe una jerarquía entre ellos.

Ahora bien el diseño de una WSN depende de varios factores tales como el entorno, la topología, el protocolo de comunicación, el consumo energético, las capacidades de los dispositivos, entre otros factores. Cada factor mencionado influye en la implementación de una WSN, haciendo su programación compleja. Otra característica relevante en las WSN es su espacio de trabajo, al cual se encuentran expuestos los dispositivos durante su funcionamiento, por el cual existe una alta probabilidad de fallos en los dispositivos, debido al clima, la fatiga mecánica y al medio ambiente agreste al que se encuentran sometidos.

2.2.1. Factores que influyen en una WSN

Una WSN tiene una colección de limitaciones y factores relevantes (Doherty et al, 2001), los cuales influyen en su desarrollo e instalación, y de las cuales depende su vida útil en un entorno de trabajo.

- **Entorno:** es el espacio físico de interés donde interactúa la WSN, es un ambiente dinámico y propenso a cambios donde ocurren eventos que deben ser monitoreados. Entre los entornos más comunes para una WSN están: plantas industriales, bloques forestales, campos de agricultura, centros comerciales, lugares turísticos e incluso volcanes activos.
- **Topología:** es la distribución geográfica de nodos WSN en el entorno de interés. La topología depende principalmente de la forma del entorno, la cual puede ir desde una topología plana en un espacio de cultivo de hortalizas llano, hasta una topología tridimensional en un edificio de varios pisos apilados. La topología también dependerá de la densidad requerida de dispositivos en el entorno, con el fin de monitorear un área del entorno con una mayor resolución que otras áreas.
- **El protocolo de comunicación:** es el método a través del cual los dispositivos de WSN se comunican entre sí. El protocolo de comunicación varía entre distintas tecnologías de WSN, los tipos van desde Bluetooth con una diminuta tasa de pérdida de paquetes y alto costo en energía hasta zigbee con una tasa más alta de pérdida de paquetes pero con un ahorro considerable en energía.
- **El consumo energético:** es el gasto en energía que realizan los dispositivos durante su operación. Dispositivos WSN con mayores capacidades de cómputo consumirán más energía que aquellos con menos capacidades. También en el

consumo energético interviene considerablemente la comunicación con otros dispositivos, se debe reducir al mínimo la comunicación entre los dispositivos lo cual extenderá la vida útil de los mismos.

- **Las capacidades de los dispositivos:** son las propiedades computacionales con las cuales cuentan los dispositivos WSN, tales como Procesamiento, Memoria y Almacenamiento.

2.2.2. Programación de una WSN

La programación de una WSN es quizá la característica menos atractiva de las WSN, puesto que están compuestas por dispositivos con recursos limitados. La arquitectura interna de un dispositivo WSN se basa en la unión de componentes electrónicos de bajo nivel, los cuales son a menudo programables a través de eventos de hardware y mediante una lógica de programación tradicional. Conceptos como la programación orientada a objetos son ajenos a la programación de una WSN, lo cual conlleva a una limitación importante a la hora de decidir utilizar una WSN para monitorear o actuar sobre un espacio físico, puesto que los desarrolladores se ven obligados a aprender lenguajes de bajo nivel lo cual retrasa el desarrollo de aplicaciones.

Las WSN son sistemas altamente concurrentes y esta propiedad puede ser vista desde diferentes niveles. Según (Cheong, 2007), tales niveles son: sistema operativo, nodo, middleware y macro-programación. Estos niveles corresponden a la forma de programación de un sistema WSN y en cada uno de estos niveles se centran esfuerzos de diferentes grupos de investigación en el mundo.

- **Nodo:** Es el nivel más básico de una WSN y corresponde al dispositivo físico que compone una WSN. Este nivel de programación se caracteriza por tener un alto factor de electrónica e instrumentación.
- **Sistema Operativo:** Corresponde al componente de software que podrían tener los dispositivos de la WSN con el cual se pueden estandarizar procesos y funcionalidades. Este nivel debe estar íntimamente ligado con el nivel de nodo.

- **Middleware:** Este nivel corresponde a una capa de programación intermedia que realiza el puente o la integración entre la programación tradicional de WSN con programación de alto nivel usada comúnmente por desarrolladores.
- **Macro-Programming:** Por último, el nivel de macroprogramación consiste en intervenir en cada uno de los niveles anteriores de la programación de una WSN, para obtener paquetes de soluciones WSN, con los cuales intervenir problemas específicos muy puntuales, como Invernaderos, entornos industriales, entre otros.

El nivel de programación en el cual se centran los mayores esfuerzos por grupos de investigación en el mundo es en el nivel de Middleware. Esto quizás se debe a que es en este nivel que es posible integrar las WSN con los enfoques y modelos tradicionales de las Ciencias de la Computación. A través de un Middleware (Ver sección 2.5) es posible hacer transparentes funcionalidades complejas proporcionando dichas funcionalidades complejas a través de métodos o procedimientos simples.

Las WSN siempre se han caracterizado por ser sistemas con un alto componente electrónico y de instrumentación física, lo cual ha dificultado su integración con los enfoques de programación tradicionales. Por tal motivo se han ido desarrollando Middlewares o Capas de Programación Intermedia por parte de la comunidad científica para abstraer dichas dificultades y permitir el desarrollo de aplicaciones que utilicen las funcionalidades de una WSN.

En lugar de requerir que el desarrollador aprenda un lenguaje de programación, en esta tesis de maestría se propone un modelo middleware modular basado en capas para integrar las capacidades de una WSN con Sistemas Multi-Agentes.

2.3. Estrategias para la Integración de Agentes Móviles y WSN: Estado del arte

Un nodo de una WSN es solo un dispositivo que puede percibir y actuar sobre un entorno determinado, sin embargo hasta el momento carecen de una programación que permita la observación inteligente del entorno y la conservación eficiente de su energía. En este punto es donde se potencia la utilización de Agentes Inteligentes que optimicen los recursos de los dispositivos WSN, de modo que sea un agente quien determine el momento adecuado para tomar medidas del entorno y actuar sobre el entorno. En otros

contextos los dispositivos son programados para que tomen muestras de las variables ambientales de manera automatizada durante una arbitraria frecuencia de muestreo. Lo que se pretende es lograr que los agentes de forma inteligente tomen control sobre los dispositivos WSN y decidan en qué momento desean tomar las medidas de las variables ambientales.

De esta forma, la integración de Agentes con WSN se ha convertido en una oportunidad de explotar al máximo el carácter intrínseco y distribuido de la WSN. Por medio de Agentes de software es posible favorecer el rendimiento de la WSN con las habilidades de cooperación y negociación de los agentes, eso sin mencionar la utilización de agentes móviles entre plataformas a través de la Internet (Wang et al, 2006) (Piedrahita et al, 2010).

A continuación se presenta una revisión en orden cronológico de los trabajos de investigación sobre las estrategias de integración entre Agentes y WSN, los trabajos presentados cuentan con sus bondades y limitaciones y permitieron identificar nuevos rumbos para construcción de esta tesis de maestría.

2.3.1. Agilla Framework

Agilla (Fok et al, 2005) es una plataforma para WSN que provee un estilo de programación basado en agentes móviles. Una aplicación de Agilla consiste de agentes móviles que pueden de forma proactiva migrar su código y estado a través de la red. Esta plataforma posee una interfaz gráfica para la interacción con la red, en cuanto a manipulación e inyección de agentes móviles. Sin embargo, Agilla presenta limitaciones en cuanto a la teoría formal de agencia vigente en la comunidad científica. Adicionalmente, posee una sintaxis y gramática de programación de bajo nivel y en consecuencia no ofrece una abstracción suficiente para dotar con las características de inteligencia artificial, propias de un Sistema Multi-Agente.

2.3.2. Agent Factory Framework

Agent Factory (Collier, 2001) es una plataforma modular y extensible que provee soporte para el desarrollo y el despliegue de aplicaciones orientadas a agentes. Este ha sido desarrollado por investigadores del Laboratorio PRISM en la escuela de Ciencias de la Computación en la UCD (University College de Dublin) de Irlanda. Es parte de una

iniciativa de investigación que pretende entender cómo las tecnologías de agentes pueden ser usadas para construir sistemas distribuidos complejos en un diverso rango de dominios de aplicación, incluyendo computación móvil y ubicua, redes de sensores distribuidos, robótica social, y realidad aumentada. Esta plataforma fue diseñado y desarrollado conforme a los estándares FIPA, para la interacción con otros Sistemas Multi-Agente. Sin embargo, hace falta investigación en cuanto a arquitecturas orientadas a componentes y su desarrollo y aplicación para dispositivos embebidos, propios de una WSN.

2.3.3. Agent Factory Micro Edition (AFME)

Agent Factory Micro Edition o AFME (Muldoon, 2008) (Muldoon et al, 2007) es una versión reducida de la plataforma de desarrollo y despliegue de Sistemas Multi-Agente Agent Factory. AFME busca desarrollar Sistemas multi-agente que soporten aplicaciones para dispositivos pequeños que poseen recursos limitados.

AFME es la plataforma de agentes deliberativos con menos carga computacional en el mundo (Collier et. al, 2006a) (Bordini, 2009). El factor más importante en la reducción de carga computacional de AFME fue el estilo de programación. En el cual el código fue reescrito sin borrar funcionalidades, sino reduciendo redundancia en las instrucciones, y optimizando las mismas. No obstante, existen ciertas características de la plataforma original de Agent Factory que no son soportadas en AFME.

2.3.4. ActorNet Platform

ActorNet (Kwon et al. 2006) es una plataforma de actores que provee un ambiente para código móvil, liviano, orientado a objetos y concurrente en WSN. ActorNet permite un amplio rango de nuevas aplicaciones dinámicas en WSN, incluyendo soporte para consultas propias y funciones de agregación, facilidades de depuración interactiva en la red, y un alto nivel de programación concurrente en la plataforma de redes de sensores. Además, ActorNet integra limpiamente todas estas características dentro de un interpretador bien ajustado con un esquema Multi-hilos embebido, lo cual soporta programas compactos y de fácil mantenimiento. Sin embargo, esta plataforma para la programación de WSN, carece de integración con otros sistemas de cómputo y un soporte para integrar mecanismos de inteligencia artificial.

2.3.5. SoSAA framework

SoSAA (Socially Situated Agents Architecture) (Dragone, 2007) es una plataforma para la integración de arquitecturas basadas en componentes y software orientado a agentes. Es motivado por la necesidad de integrar herramientas de desarrollo de agentes con funcionalidades de bajo nivel sofisticadas, en campos como la robótica, donde se han adoptado ampliamente la ingeniería de software orientada a objetos (OO, acrónimo inglés para Oriented-Objects) y componentes (CBSE, acrónimo inglés para Component-based software engineering). SoSAA integra entonces el paradigma de agentes móviles con una arquitectura de componentes. Sin embargo, a pesar de abarcar gran cantidad de aplicaciones, SoSAA se encuentra muy inclinado hacia entornos robóticos, dejando de lado otros sistemas tales como las WSN, los cuales poseen otro tipo de características.

2.3.6. Famiware

FamiWare (Gámez & Fuentes, 2010) es un middleware para Aml (Inteligencia Ambiental) que busca la heterogeneidad de los dispositivos. En el momento permite dispositivos TinyOS y Android. La adquisición de datos es conducida a eventos. Su programación es flexible gracias al enfoque SPL (Software Product Line). A pesar que Famiware esta orientado hacia la inteligencia ambiental carece de soporte para Sistemas Multi-Agente, adicionalmente no posee la modularidad necesaria debido a la gran cantidad de dispositivos existentes.

2.4. Limitaciones encontradas en los trabajos relacionados

En los trabajos relacionados revisados y presentados anteriormente se encontraron varias limitaciones para la integración de Agentes con WSN. De las limitaciones encontradas, la limitación más relevante es la carencia de un capa intermedia o Middleware enfocada a integrar Sistemas Multi-Agentes con WSN. Aunque algunos de los enfoques mencionados indican que existen acercamientos entre Agentes y dispositivos de WSN, no existe un soporte correcto conforme a los especificados por FIPA. Por otro lado aquellos enfoques robustos que relacionan dispositivos con arquitectura de componentes con Sistemas Multi-Agente no están destinados a dispositivos de WSN sino que se inclinan por dispositivos robóticos los cuales poseen características diferentes a las características encontradas en dispositivos WSN.

De esta manera en esta Tesis de Maestría se propone una capa intermedia Middleware para la integración de Sistemas Multi-Agente con dispositivos WSN, y que a su vez sea conforme con los estándares del diseño y desarrollo de Sistemas Multi-Agente, tales como los estándares de FIPA.

2.5. Middleware

Un Middleware (Ibrahim, 2009) es una tecnología de apoyo para el desarrollo, despliegue, ejecución y interacción de aplicaciones. Un Middleware se puede definir como una capa intermedia entre dos tecnologías aisladas para lograr la interacción entre ellas. Este modelo de programación consigue la integración puesto que logra hacer transparente las funcionalidades de una tecnología con respecto a la otra.

Las capas intermedias middleware han evolucionado desde ofrecer funcionalidades básicas, hasta ocultar y envolver minuciosas capacidades en sofisticados sistemas, estos sistemas se encuentran equipados para administrar recursos y ofrecer al programador una Interfaz de Programación(API) para el desarrollo de nuevas aplicaciones.

3. MODELO PROPUESTO

El objetivo principal de esta tesis de maestría es proponer un modelo de interoperabilidad middleware para facilitar la integración entre la tecnología de programación basada en agentes y la tecnología basada en dispositivos de WSN. De esta manera los agentes sienten lo que está ocurriendo en su entorno y por consiguiente adquieren una consciencia del contexto que les permite tomar mejores decisiones.

En este capítulo se propone de esta manera un modelo para lograr la integración entre Agentes y WSN. En primer lugar se enumeran varias de las consideraciones que se tuvieron en cuenta al momento de diseñar el modelo, luego se presentan los componentes de la arquitectura multi-capas del middleware

3.1. Consideraciones del modelo

Con el fin de diseñar un modelo integral que integre las diferentes tecnologías de WSN con el diseño y el desarrollo de Sistemas Multi-Agente, es necesario cumplir con algunas consideraciones y exigencias que surgen de dicha integración.

Las consideraciones presentadas a continuación surgen de la revisión del estado del arte en cuanto a la integración de WSN con Sistemas Multi-Agente. Dichas consideraciones pretenden enfrentar las limitaciones encontradas, las cuales fueron presentadas en el capítulo 2.

3.1.1. Heterogeneidad de dispositivos

Actualmente existe una gran cantidad de dispositivos comerciales de WSN, así como también dispositivos diseñados y desarrollados para una situación particular. Debido a esta oferta de dispositivos de WSN, es necesario proveer un modelo escalable que sea acorde a la dinámica del mercado de WSN. Se espera que en un espacio geográfico la infraestructura de WSN sea muy colorida, o sea conformada por distintas tecnologías de WSN que poseen a su vez distintos sensores y actuadores, pero no por esto se deben desarrollar diferentes sistemas, sino que por el contrario el sistema o modelo planteado debe integrar las tecnologías en una única WSN y manejar interoperabilidad de componentes heterogéneos.

Esto implica que el modelo propuesto debe integrar sensores y actuadores de los distintos dispositivos, debe ser consciente de las capacidades de cómputo, memoria y almacenamiento para explotar el rendimiento de los dispositivos, y debe implementar múltiples protocolos de comunicación simultáneos en una única capa de comunicación con los medios.

3.1.2. Interoperabilidad

La interoperabilidad es la característica de un sistema de lograr que todas sus partes interactúen entre sí, en este caso se busca que las partes en el modelo se integren a través de capas de abstracción, las cuales de manera transparente provean los mecanismos necesarios para su integración.

3.1.3. Escalabilidad

La escalabilidad es la propiedad que puede tener un sistema para crecer o disminuir su tamaño sin que éste altere su funcionamiento, con esta premisa surgen dos subconsideraciones con respecto a la escalabilidad.

Escalabilidad a nuevas tecnologías: En primer lugar y volviendo al tema de la diversidad de dispositivos, es de esperar que surjan nuevos dispositivos de WSN, que mejoren las tecnologías existentes en todos sus aspectos actuales. Los nuevos dispositivos sobrepasarían a los actuales en un sin número de características de interés, tales como capacidad de cómputo, almacenamiento, sensores más precisos, actuadores más invasivos, mayores tasas de transferencia de datos con menos fallos durante la comunicación, así como también una importante reducción en el tamaño físico del dispositivo.

De acuerdo a lo anterior se espera que el modelo middleware propuesto sea capaz de integrar próximas tecnologías de WSN, por medio de un diseño e implementación basado en módulos lo cual implica proveer al modelo de un entorno modular. Para lograr esto es necesario asociar esta consideración al concepto de Modularidad.

Escalabilidad a tecnologías soportadas: Una WSN es un sistema distribuido que consta de nodos geográficamente espaciados, que se encuentran expuestos a condiciones propias del entorno y por ende se ven deteriorados con el paso del tiempo. Por ser inalámbricos carecen de una fuente permanente de electricidad y

por tanto están sujetos a duración de la batería. En este caso aparece la necesidad de añadir nuevos dispositivos a la WSN para reemplazar los defectuosos o aquellos sin energía sin que esto afecte el sistema que se encuentra en funcionamiento, ni las operaciones que en un momento dado se puedan estar ejecutando. De esta forma, el sistema automáticamente debe detectar, integrar y transferir las operaciones de los dispositivos defectuosos a aquellos que están conectados.

3.1.4. Modularidad de programación

Para conseguir aspectos claves como la escalabilidad es muy importante considerar una programación modular, que facilite el acople y desacople de nuevos componentes de software. Así, el modelo debe poseer una arquitectura modular de modo que integre nuevas funcionalidades y/o puedan ser actualizadas las actuales con facilidad.

La modularidad se logra a través de módulos o bundles, los cuales encapsulan mecanismos aislados que se acoplan a través de servicios ofrecidos. Una de las principales ventajas de este tipo de programación es la facilidad cuando se tienen sistemas muy extensos, o grandes equipos de desarrollo.

A través de módulos el sistema puede encapsular la implementación necesaria para la comunicación con un determinado conjunto de dispositivos de WSN, de manera que sea transparente al resto de componentes.

3.1.5. Transparencia

La transparencia es la función fundamental de un middleware, es la capacidad de mapear funcionalidades y mecanismos complejos en interfaces cómodas para el desarrollador. En este caso se refiere a la posibilidad de que los agentes de software, puedan utilizar dichas interfaces para acceder a dispositivos complejos de bajo nivel con capacidades de percepción y actuación en un entorno determinado.

3.1.6. Formalización del entorno

El entorno está definido como el espacio geográfico de interés, en el cual son distribuidos los nodos de la WSN con el fin de percibir y actuar sobre dicho entorno. Ahora bien, el entorno es un ambiente dinámico y altamente cambiante lo cual lo convierte en una

increíble fábrica de información, que a su vez se convertirá en conocimiento para aquellos agentes que soliciten conocer su entorno. Sin embargo, no todo es color de rosa, es necesario dar un formato a todo el conocimiento del entorno y para ello el modelo debe implementar una Ontología de modo que se puedan establecer actos comunicativos de alto nivel.

3.2. Modelo Planteado

Después de realizar una revisión de las consideraciones que se tuvieron en cuenta en la sección anterior, se presenta un modelo mediante una arquitectura en capas altamente modular que busca cumplir con las necesidades planteadas.

3.2.1. Arquitectura en Capas

El modelo de interoperabilidad middleware propuesto en esta tesis de maestría para la integración entre el paradigma de programación con agentes y una WSN se plantea a través de una arquitectura basada en capas la cual se encuentra presentada en la Figura 3.1. En este sentido el modelo está compuesto por una primera capa física la cual provee los mecanismos para la comunicación con los distintos dispositivos; la segunda capa tiene por objetivo estandarizar las plataformas que conforman los nodos de la WSN, así como también clasificar sus distintas capacidades de percepción y actuación; por último aparece una tercera capa que ofrece una representación de la WSN y expone interfaces a través de las cuales es posible interactuar con la WSN, dicha interacción permite a los agentes de software realizar consultas y ejecutar actuadores, entre otros.

Capa Física

Esta capa está encargada de la comunicación con los nodos de la WSN. La capa física permite comunicarse con una WSN sin importar cuál es el medio de comunicación (Serial, TCP, Bluetooth, WiFi, entre otros.), ésto se debe a que cada uno de estos medios es soportado a través de un módulo o bundle específico que actúa como un driver para integrar una tecnología de WSN al Middleware. De esta forma se ofrece modularidad al Middleware y se hace escalable a nuevos dispositivos de WSN, para los cuales sólo será necesario desarrollar un módulo driver con el fin de integrar dichos dispositivos al Middleware.

Otro aspecto importante de la capa física es la posibilidad de operar varias plataformas de WSN simultáneamente. Aunque las plataformas de WSN varíen en cuanto a la velocidad de transmisión serial o el protocolo de comunicación inalámbrico que usan, el modelo está en capacidad de operar los distintos dispositivos de WSN al mismo tiempo, ya que son aislados los protocolos de comunicación a través de puertos e interfaces. Una vez que la capa física tiene acceso a las tecnologías de WSN posteriormente integran y estandarizan las capacidades de varias WSNs en la capa superior.

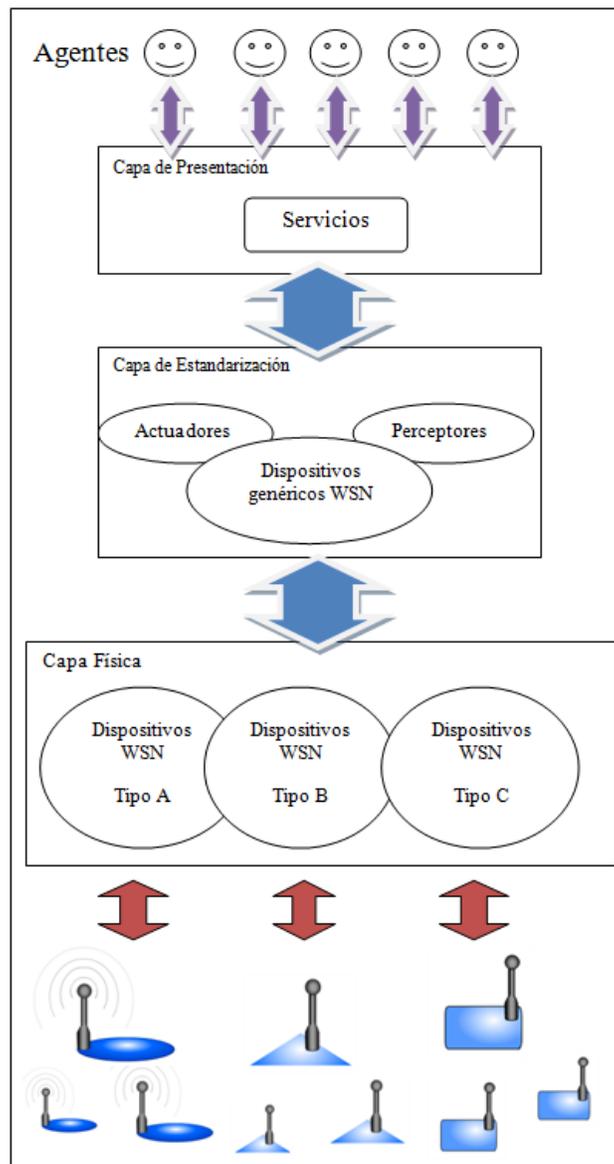


Figura 3.1. Arquitectura multi-capas para el modelo middleware propuesto***Capa de Estandarización***

La capa de estandarización como su nombre lo indica estandariza las Tecnologías de WSNs y construye una plataforma WSN integrada (WSNI), con todas las capacidades que determina la capa física de las WSNs que se encuentran operativas en el sistema.

La WSNI que se construye en esta capa cuenta con todos los nodos operativos de las WSNs activas. Dichos nodos son representados como un nodo genérico que puede o no tener diferentes capacidades de percepción ó actuación sobre su entorno. Las capacidades del nodo dependen del dispositivo físico que este posee, es decir, la instrumentación o electrónica que tenga añadida a él. Para ilustrar un poco, suponga un dispositivo Mica2 con una tarjeta de sensores MTS310, la cual cuenta con un acelerómetro que le permite percibir sus movimientos; por otro lado suponga un dispositivo TmoteSky que en lugar de aceleración, puede percibir temperatura, humedad y luz. Para este caso aunque los dispositivos poseen capacidades diferentes son abstraídos por el modelo como nodos genéricos en la capa de estandarización y sus capacidades son representadas como variables booleanas, donde se especifica si tiene o no tiene una propiedad particular (ej. Temperatura, Aceleración, entre otros). De esta manera las capacidades de la WSN se pueden ver de manera transparente sin importar el dispositivo que se tenga.

Capa de Presentación

La última capa del modelo tiene como función ofrecer la WSNI ante instancias y paradigmas de programación más funcionales como: Servicios Web, Sistemas Multi-Agente, Computación Móvil, entre otros. Sin embargo, para el propósito de esta Tesis de Maestría solo se trabajarán Sistemas Multi-Agente, puesto que los demás paradigmas de programación se salen del enfoque de este trabajo.

La capa de presentación consigue mediante interfaces proveer métodos y funciones para la interacción del modelo con otros sistemas. De un modo general, un SMA que requiera el conocimiento proveniente del entorno sólo debe acceder a dichas interfaces y preguntar por los servicios ofrecidos. En un caso particular es el agente quien accede a los métodos para realizar consultas sobre el entorno, y construir su base de conocimiento.

Es importante señalar que para representar el conocimiento del entorno la capa de presentación cuenta con una ontología del entorno, la cual permite establecer la comunicación desde y hacia los agentes, así como también formalizar la comunicación entre los mismos agentes que interactúen con el Middleware. Esta Ontología hace parte fundamental del modelo planteado en esta Tesis de Maestría ya que estandariza el conocimiento permitiendo la interacción no solo con Sistemas Multi-Agente sino también con otros paradigmas de programación que son conformes al uso de ontologías.

3.2.2. Modularidad del Modelo Propuesto

Con el fin de conseguir que el modelo fuera escalable, se propuso una arquitectura modular para el Middleware soportada sobre un marco de trabajo o plataforma para modularidad denominado OSGi (acrónimo inglés para Open Services Gateway Initiative). Este marco de trabajo modular permite dividir sistemas complejos y monolíticos en sistemas flexibles y modulares para facilitar tanto su diseño e implementación como su posterior mantenimiento y actualización.

Antes de comenzar a describir los módulos diseñados en este modelo es importante introducir el concepto de *Bundle*, el cual se puede definir como un módulo o componente de software que envuelve ciertas funcionalidades de manera oculta, conformando sistemas altamente desacoplados. Un sistema OSGi se encuentra dividido en *bundles* los cuales publican y suscriben servicios para la interacción con otros bundles. Cada Bundle puede ser idealizado como una pequeña caja negra cuya implementación es transparente a otros bundles del sistema, y sólo se dedican a consumir y proveer servicios.

A través de este enfoque modular el modelo de Middleware planteado en esta Tesis de Maestría puede ser fácilmente escalable a nuevas tecnologías y dispositivos WSN, puesto que sólo basta con construir el respectivo Bundle y acoplarlo sin necesidad de replantear todo el sistema. Se puede observar, por ejemplo, en la Figura 3.2 el diagrama de módulos o bundles de la capa física. En esta capa existen dos bundles principales: el Comunicador de dispositivos ó CD quien se encarga de realizar todas las comunicaciones hacia los dispositivos de una tecnología en particular; y el Administrador de dispositivos ó AD quien tiene como tarea integrar, conocer y gestionar toda la infraestructura de

dispositivos WSN existente y ofreciendo transparencia ante entidades o módulos de mayor nivel en el modelo propuesto.

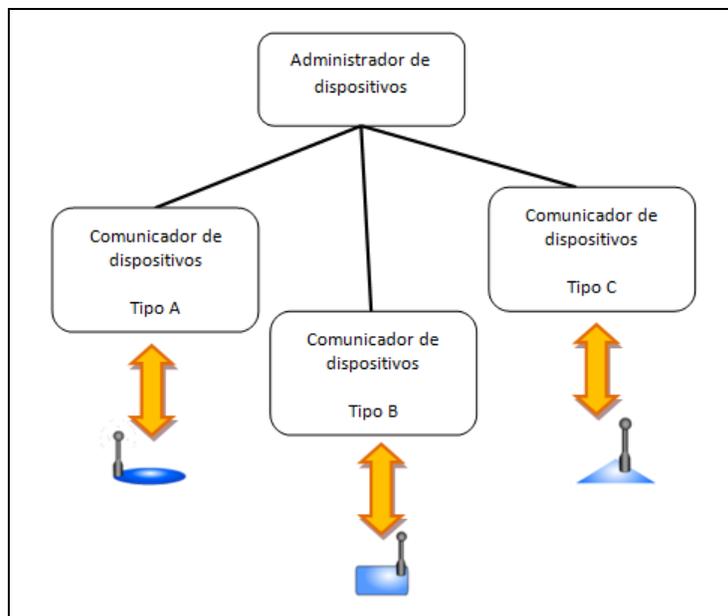


Figura 3.2. Diagrama de Módulos: Capa Física

A continuación se describen en detalle los módulos utilizados en el modelo planteado para la integración entre Agentes y tecnologías de WSN.

Comunicador de dispositivos (CD)

Este módulo permite agrupar todos los mecanismos de comunicación de una tecnología particular, por ejemplo si tenemos dispositivos de tipo “A” que se comunican a través de un protocolo de comunicación determinado, dichos dispositivos tendrán su propio módulo de comunicación asociado que le permitirá interactuar con el sistema de modo transparente. En el interior de este módulo no sólo se implementan los protocolos de comunicación sino también las capacidades de percepción y actuación de este tipo de dispositivo en particular, y todo lo estrictamente relacionado con la Tecnología A de WSN, logrando así la modularidad requerida.

Administrador de dispositivos (AD)

El sistema de integración de SMA y WSN puede estar conformado por una variedad de dispositivos con diferentes características, para lograr la interoperabilidad fué necesario

incluir este módulo Administrador de Dispositivos (AD) y como su nombre lo indica cumple con la función de administrar y manejar los dispositivos de la WSN. Este módulo proporciona una vista general de los dispositivos existentes, así como también las interfaces para ofrecer funcionalidades genéricas de WSN en el sistema. El AD representa los distintos dispositivos de WSN en un único dispositivo genérico que cuenta con capacidades de percepción y actuación.

Este módulo también tiene la tarea de conocer toda la infraestructura existente de WSN, y el estado de cada uno de los dispositivos: Activo, Inactivo, Ocupado, Hibernando, entre otros. Esto es de vital importancia puesto que al conocer el estado de los dispositivos, este módulo puede informar en que lugares o puntos del entorno puede ser adquirida información por los Agentes de software.

De acuerdo a la heterogeneidad de dispositivos considerada en la Sección 3.1.1, se espera que un entorno particular contenga diversidad en cuanto a los dispositivos instalados allí, y por lo tanto también una diversidad en cuanto a las capacidades de percepción y actuación sobre el entorno. Por tal motivo el AD conoce y almacena la información que le permita saber cuáles son los sensores y actuadores de cada tipo de dispositivo. Adicionalmente, el AD tiene en cuenta que la información proveniente de los dispositivos se encuentra sujeta a la ubicación geográfica del dispositivo en el entorno y por lo tanto dicha ubicación también debe ser almacenada a modo de información. Considere por ejemplo un dispositivo que señala una temperatura de 45°C en un entorno industrial con el fin de monitorear maquinaria, sería muy importante saber a qué distancia se encuentra el dispositivo de la máquina objetivo y así decidir si es una condición alarmante o no lo es.

Interfaz Gráfica del AD

En la Figura 3.3 se puede apreciar la interfaz gráfica de usuario (GUI, acrónimo inglés para Graphics User Interface) del módulo AD por medio de la cual se puede observar la infraestructura de dispositivos existente en el entorno. Nótese en dicha GUI cómo conviven dos tipos de dispositivos simultáneamente en la misma infraestructura de WSN. Los dispositivos que se aprecian en la figura son: TmoteSky y Mica2, los cuales se encuentran discriminados en colores, azul y naranja respectivamente. El módulo AD

permanece constantemente monitoreando la inserción y deserción de dispositivos en la WSN, anotando un registro de dichos eventos en el *Panel Consola* ubicado en la parte de derecha de la GUI.

El modelo planteado propone un conjunto de estados para los dispositivos en el entorno. No es suficiente considerar sólo dispositivos como activos o inactivos, también es necesario considerar estados como disponible o no disponible, dado que un dispositivo podría estar activo pero no disponible quizás porque ya está siendo usado en algún servicio o porque su batería se ha comenzado a extinguir, en este último caso el AD mantiene durante un periodo de tiempo el dispositivo en su inventario esperando recibir respuesta a mensajes enviados, en caso de no recibir respuesta el dispositivo es removido del inventario de dispositivos existente.

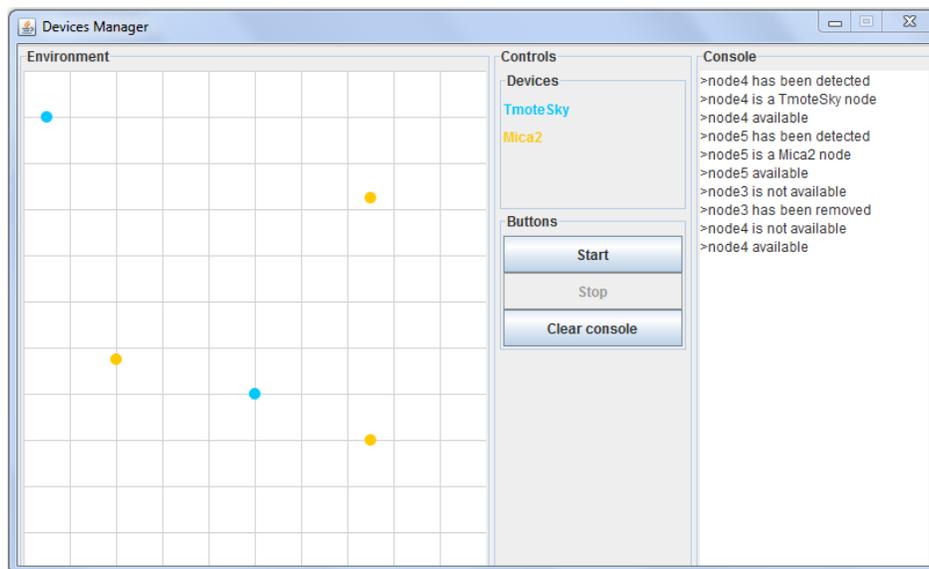


Figura 3.3. Interfaz gráfica del Administrador de Dispositivos

En el panel izquierdo de la GUI del módulo AD se visualizan los dispositivos existentes en el entorno. Es importante señalar que los dispositivos allí se encuentran estáticos y no se encuentran en movimiento como se podría considerar en un entorno dinámico, su ubicación en el panel debe ser determinada explícitamente por un usuario del sistema ya que la capacidad de localización no se encuentra considerada dentro del alcance de esta Tesis de Maestría y corresponde a trabajo futuro.

Distribuidor de servicios (DS)

El DS es el módulo encargado de ofrecer las funcionalidades de la WSN a través de interfaces y posteriormente servicios, que pueden ser consumidos posteriormente por entidades de más alto nivel, como Agentes de Software con el fin de percibir e interactuar con el entorno. La función del DS es publicar todos los servicios que pueden ser ofrecidos por la WSN, esto incluye servicios para obtener mediciones de variables ambientales del entorno, tales como temperatura, luminosidad, entre otros, y ejecutar actuadores sobre el entorno, como encender fuentes de luminosidad ó calor, activar alarmas, entre otros actuadores. Los servicios ofrecidos por el DS dependen de las características de los dispositivos encontrados por el AD, lo cual implica que los servicios ofrecidos son dinámicos de acuerdo a las características encontradas por el modulo anterior AD y al estado de los dispositivos en el sistema.

Algunos de los servicios ofrecidos por el DS, a manera de ejemplo, son presentados en la Figura 3.4, sin embargo, dichos servicios resultan de la información provista por el AD y por los dispositivos activos en el entorno.

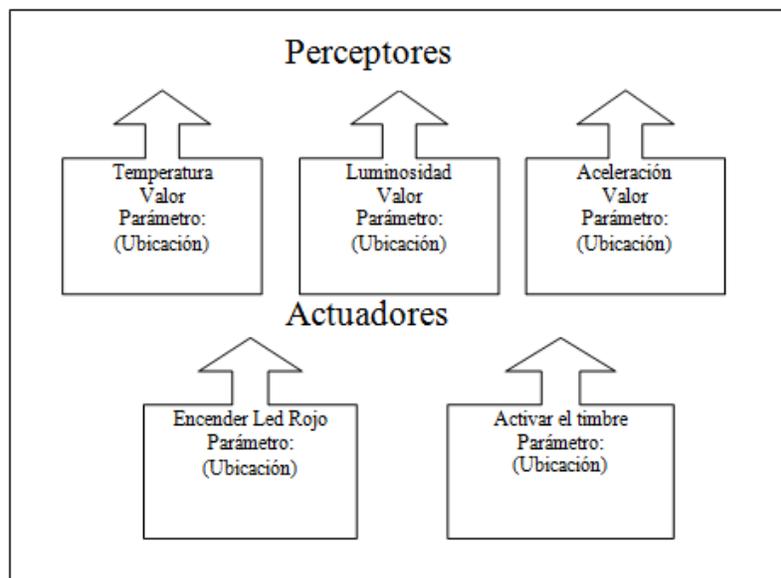


Figura 3.4. Servicios ofrecidos por el módulo DS

3.3. Representación del entorno

El entorno está definido como el espacio físico ubicado geográficamente en el cual dinámicamente ocurren eventos y situaciones de interés. Cabe señalar que el propósito de esta Tesis de Maestría es proveer a través del middleware propuesto a Agentes de Software Móviles e Inteligentes la oportunidad de ser conscientes del entorno y tomar las mejores decisiones con la información obtenida.

Un entorno es un ambiente naturalmente dinámico, lo cual lo convierte en una fuente constante de información desordenada. La organización de la información de un entorno es muy importante para que sea procesada por los agentes, quienes necesitan representar la información de manera formal a través de ontologías, de modo que se exprese el conocimiento del entorno en términos de Conceptos, Predicados y Acciones.

Así, en esta Tesis de Maestría se diseñó una *Ontología del Entorno* la cual permite una comunicación más clara y formal entre el modelo planteado y los agentes que requieran información del entorno. A través de una *Ontología del Entorno* se formaliza la información y se convierte en conocimiento, que puede ser fácilmente comprendido por agentes de software de acuerdo a los estándares FIPA.

Existen un sinnúmero de características en un entorno determinado, desde las formas de los objetos, las texturas de los objetos, los sonidos, la luz, la presión, la visibilidad, el dióxido de carbono, la visibilidad, la humedad, la temperatura, entre muchas otras características que distinguen un entorno de otro. Sin embargo muchas de las características mencionadas no son fácilmente medibles o cuantificables, se necesitarían sensores de precisión con características complejas que se comparen a los sentidos humanos (vista, tacto, oído, entre otros), sin embargo, cabe resaltar la cantidad de información que puede surgir de un entorno geográfico.

En la Figura 3.5 se puede apreciar la Ontología genérica del entorno propuesto en esta Tesis de Maestría para la comunicación formal con agentes de software. Nótese que existen dos acciones básicas que el agente puede realizar “get-measure” (Obtener medida) y “execute-actuator” (Ejecutar Actuador). Del diagrama mostrado se puede notar que el conocimiento que el agente puede almacenar está relacionado con los valores medidos de las variables físicas incluyendo el tiempo en el cual el valor fue medido.

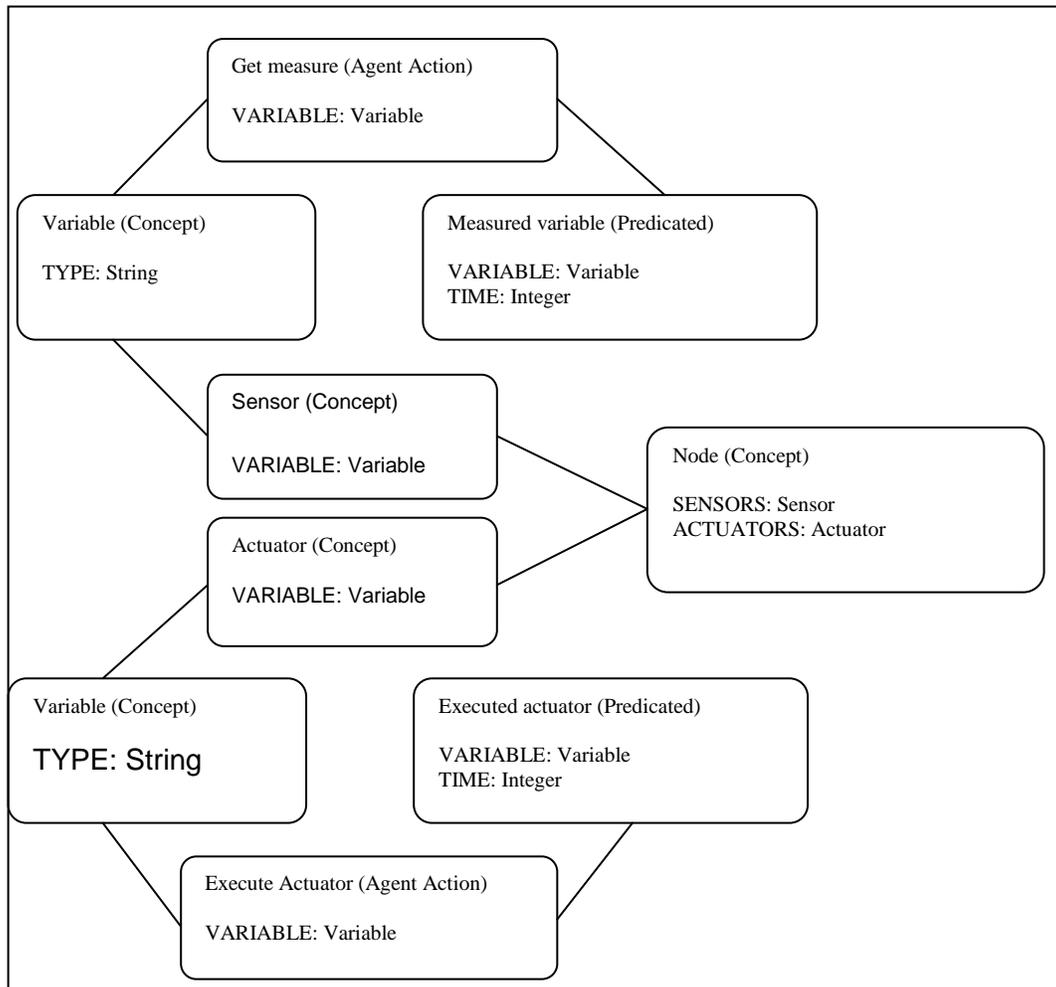


Figura 3.5. Ontología genérica del entorno

4. IMPLEMENTACIÓN DEL PROTOTIPO Y RESULTADOS

La validación del modelo propuesto se realizó a través de la implementación de un prototipo de Sistema Multi-Agente que busca satisfacer varios usuarios simultáneamente en cuanto a sus preferencias de temperatura y luminosidad en un entorno cerrado. Los agentes buscarán maximizar su utilidad negociando con un agente coordinador quien tomará las decisiones que satisfagan a la gran mayoría de usuarios.

EL objetivo de este SMA es probar la utilidad del modelo propuesto de interoperabilidad middleware para la integración entre Agentes Inteligentes y WSN. Este caso de estudio pretende evaluar la pertinencia del modelo como una herramienta para la construcción de aplicaciones en el área de la Inteligencia Ambiental, así como también, la capacidad de los agentes de extraer información del entorno para la toma de sus decisiones.

4.1. Caso de Estudio

En la Figura 4.1 se aprecia un entorno cerrado similar a un restaurante donde se implementa el prototipo de esta tesis de maestría. Suponga por ejemplo, un usuario del restaurante, quien llega a disfrutar de una cena en compañía de su pareja, el usuario se encuentra en la indecisión de cuál mesa deberá escoger para su cena, y su elección dependerá de factores tales como iluminación, temperatura, ruido, humo de cigarrillo, entre otros. Estos factores aunque muy simples y cotidianos logran influir en la sencilla decisión acerca de cuál mesa elegir en el restaurante.

Mediante el modelo propuesto se pretende ayudar al usuario a tomar la decisión de cuál mesa elegir y además solicitar cambios en el entorno que le permitan mejorar las condiciones para su cena.

Un restaurante resulta un espacio dinámico en el cual no sólo participa un único usuario sino un espacio social donde concurren múltiples usuarios, esto hace un poco más difícil actuar sobre el entorno en busca de condiciones ambientales particulares. En este punto

interviene la habilidad de agentes de software móviles para la negociación y la cooperación con otros agentes.



Figura 4.1. Entorno para el caso de estudio: Restaurante

4.2. Herramientas utilizadas

Para la construcción del prototipo en esta tesis de maestría se utilizó en primer lugar una plataforma denominada OSGi (2011) para la construcción modular del sistema. Para la construcción de los agentes de software se utilizó JADE (Bellifemine et al., 2005), un marco de trabajo en Java para el desarrollo y despliegue de agentes móviles. Por último se implementó la ontología en Protegé (Grosso et al., 1999), una herramienta para el diseño de ontologías, dicha herramienta se ajusta bien al formato necesario para los agentes de software de JADE.

4.3. Sistema Multi-Agente

El Sistema Multi-Agente propuesto ha sido nombrado como Sistema de Satisfacción de Clientes (SSC), puesto que permite mejorar los entornos sociales donde el usuario participa para mejorar su bienestar. Una característica adicional es registrar las preferencias del usuario con propósitos estadísticos para mejorar las condiciones de los entornos.

El prototipo implementado en esta Tesis de Maestría está compuesto de los siguientes agentes proporcionados por JADE:

Agent Management System, AMS: Es el agente encargado de la supervisión y control sobre el acceso y uso de la plataforma. Además, presta el servicio de páginas blancas, es decir, es el responsable de la autenticación de los agentes y el control de registros, asegurando que cada agente en la plataforma disponga de un nombre único.

Directory Facilitator, DF: Es el agente que presta el servicio de páginas amarillas. Aquí es donde cada agente del sistema puede registrar los servicios que ofrece. Gracias al agente DF, un agente puede encontrar otros agentes en el sistema que proporcionen los servicios necesarios para lograr sus objetivos.

Remote Monitoring Agent, RMA: Es el agente que permite controlar al resto de agentes en una plataforma. Provee una interfaz gráfica que facilita las funciones de monitorización y control.

Adicional a los agentes provistos por JADE, el prototipo cuenta con algunos agentes propios del negocio:

Agente Usuario: El agente usuario tiene la tarea de representar a un usuario buscando satisfacer la preferencias de éste. El agente usuario tiene consigo las preferencias del usuario previamente configuradas por este último. Su misión es observar el entorno a través del modelo propuesto y comparar los valores medidos con los valores objetivo que satisfacen al usuario. En caso de que los valores medidos varíen más de cierto valor (umbral) el agente tiene la obligación de realizar las solicitudes al agente coordinador para alterar el entorno buscando favorecer las preferencias de su usuario.

Agente Entorno: Es el agente que se comunica directamente con el entorno y construye una representación del ambiente. La función de este agente es tener conocimiento del entorno para informar a otros Agentes en el SMA.

Agente Coordinador: Este agente actúa como moderador o jurado durante las negociaciones entre los agentes usuario. El agente recibe las solicitudes de los agentes usuario y calcula una función de optimización en la cual se busca satisfacer la mayor cantidad de agentes usuario. Una vez el agente coordinador ha calculado el punto óptimo propone a los agentes usuario la acción escogida, los cuales podrán aceptar o rechazar la propuesta. Cuando los agentes usuario han llegado a un concilio el agente coordinador ejecuta la acción adecuada sobre el entorno.

En la Figura 4.2 se muestra el diagrama de secuencias provisto por el agente sniffer (Olfateador) de JADE que describe la secuencia principal del sistema, donde se tienen dos agentes usuario que realizan solicitudes para cambiar el entorno al Agente Coordinador. Una vez que el agente coordinador recibe la solicitudes envía como respuesta una propuesta para cambiar el entorno a la cual los agentes aceptan y se procede a ejecutar la acción o las acciones solicitadas e informa el éxito o fracaso de la acción a los agentes usuario involucrados.

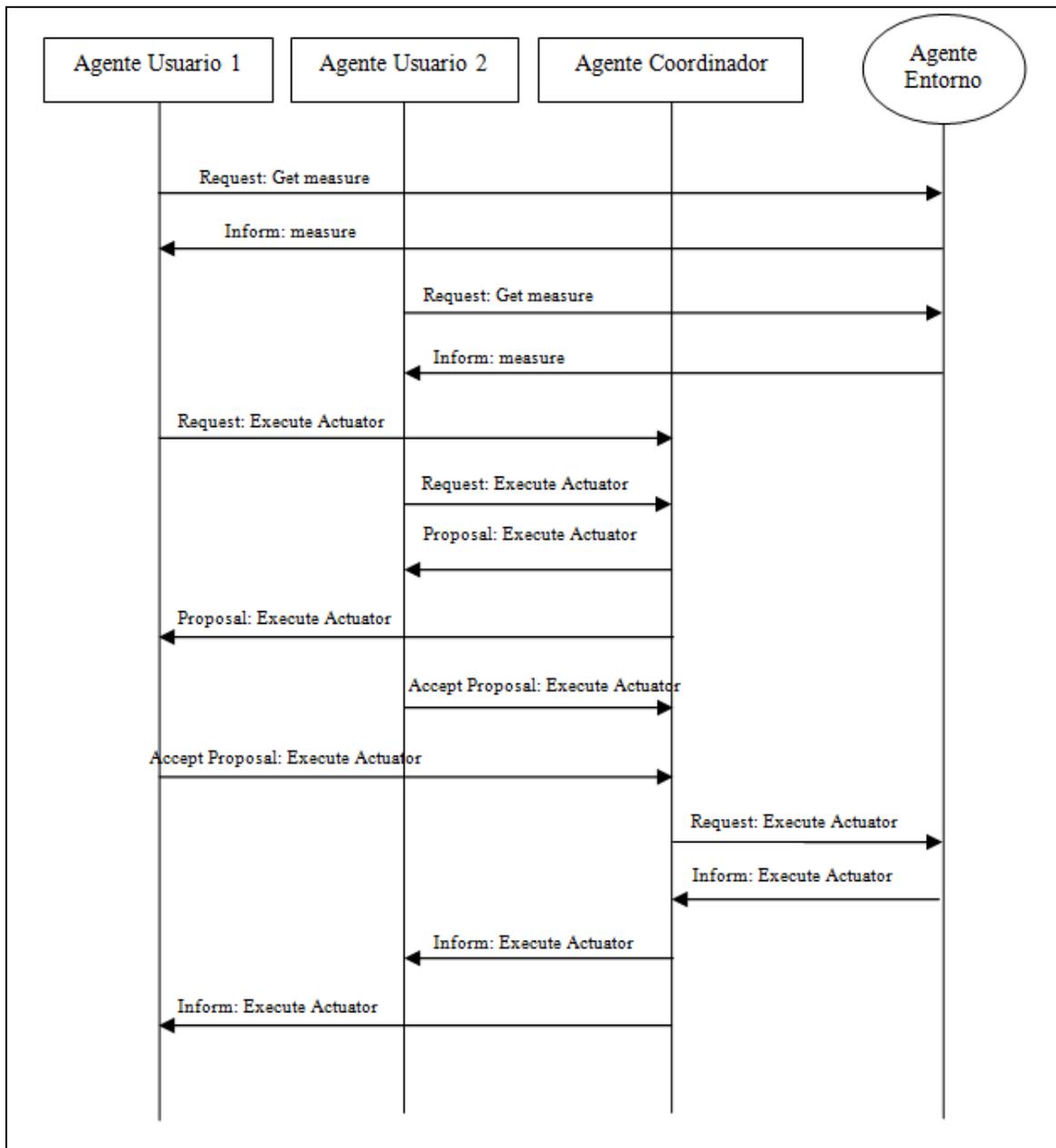


Figura 4.2. Diagrama de Secuencia: Negociación entre agentes

4.4. Escenario de Validación

El escenario planteado para este prototipo fue un espacio cerrado de aproximadamente 21mts² simulando el salón de un restaurante. En cuanto a las condiciones iniciales planteadas se iluminaron algunas áreas del espacio y se calentaron algunas otras con el

fin de proponer una temperatura e iluminación inicial que permitiera crear la necesidad de los agentes de observar el entorno a través de la infraestructura de WSN y efectuar acciones para modificar dicho entorno.

En la Figura 4.3 se presenta un diagrama del escenario utilizado para validar los resultados del modelo empleado, en la misma figura se muestra el espacio utilizado y la ubicación de cada dispositivo en el entorno. También fueron añadidas dos fuentes de iluminación y calor a través de dos bombillos de 100Watt en el entorno para variar condiciones físicas del entorno.

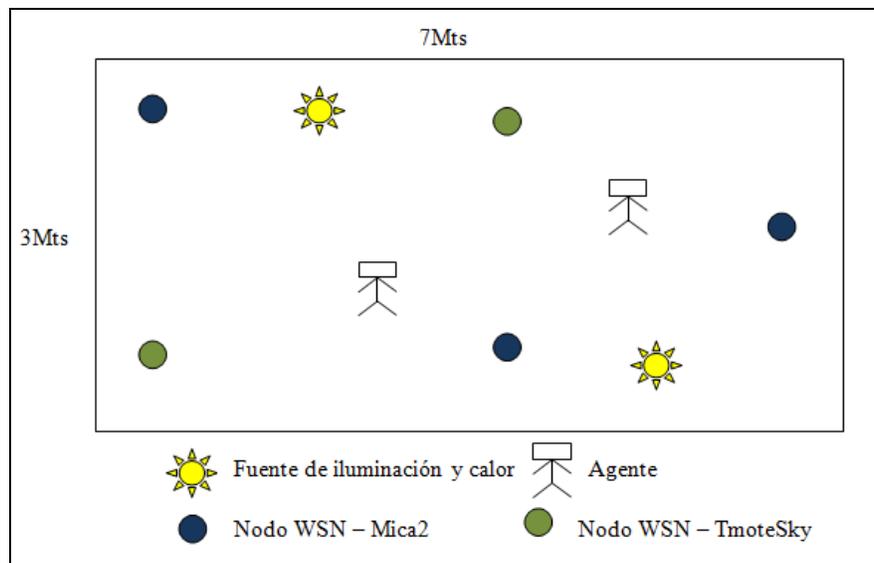


Figura 4.3. Escenario utilizado en las pruebas realizadas

En el escenario fueron insertados dos agentes usuario para iniciar una negociación, cada uno de los agentes insertados representa las preferencias de un usuario particular. Dichas preferencias son presentadas en la Tabla 4-1.

Tabla 4-1. Preferencias de los usuarios

Preferencias de usuario		
Variable	Usuario 1	Usuario 2
Temperatura:	25°C	17°C
Tolerancia a Temperatura:	±3	±5

Luminosidad:	1000 luxes	400 luxes
Tolerancia a Luminosidad:	±500	±200

La Figura 4.1 muestra las preferencias para dos usuarios: usuario1 y usuario2. En las preferencias de usuario sólo se encuentran contempladas las variables Temperatura y Luminosidad; adicionalmente, las variables van acompañadas de una tolerancia al valor deseado. La tolerancia mencionada funciona como un coeficiente o peso de modo que el valor deseado no es obligatorio y es posible que el agente ceda sus preferencias ante las tolerancias de otros usuarios.

4.5. Resultados

A través del prototipo implementado se validó el modelo propuesto para la integración de Sistemas Multi-Agentes con dispositivos de WSN.

Del prototipo implementado se buscó que los agentes entraran en un conflicto de intereses en cuanto a la Temperatura y Luminosidad requerida, una vez que los agentes entraran en conflicto debían resolver por sí mismos el conflicto y llegar a un resultado que los dejara satisfechos a ambos para actuar sobre su entorno.

En este caso el acuerdo al que llegaron ambos agentes usuario en cuanto al valor de luminosidad y temperatura se muestra a través de las 2 rectas numéricas presentadas en la Figura 4.4.

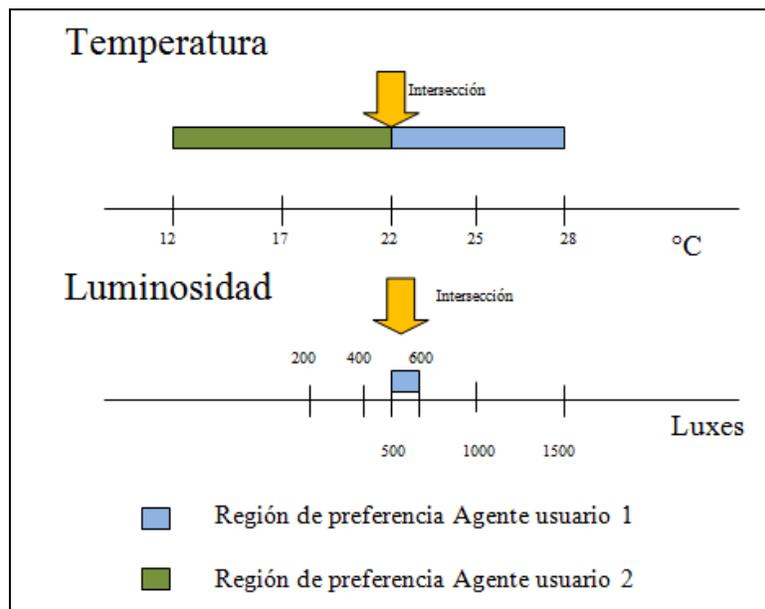


Figura 4.4. Resultado de negociación entre agentes: intersección de regiones

En el caso de la temperatura el valor acordado es de 22°C dado que el agente1 tiene un valor esperado de 25°C y una tolerancia de ± 3 ; y el agente 2 tiene un valor esperado 17°C y una tolerancia de ± 5 .

Para el caso de la luminosidad el valor acordado surge de una franja entre 500 lx y 600 lx, dado que el agente1 tiene un valor esperado de 1000 lx y una tolerancia de ± 500 ; y el agente 2 tiene un valor esperado 400 lx y una tolerancia de ± 200 . Sin embargo, para resolver un valor específico de luminosidad y no una franja, simplemente se eligió el punto medio de la franja, 550 lx.

Cuando se han acordado los valores apropiados de las variables físicas, el agente coordinador procede a ejecutar los actuadores del MD de manera que se logre una temperatura y luminosidad apropiada.

4.6. Comparación con trabajos relacionados

Realizando una comparación con otras estrategias para la integración de agentes y WSN, es posible construir una tabla comparativa con algunas de las características más relevantes en cuanto a la integración realizada.

Los tópicos en los cuales se comparan las estrategias son:

- Paradigma de programación: corresponde al enfoque utilizado para la programación de dicha integración. Suele ser una característica importante debido al rendimiento que podría representar el uso de paradigmas distribuidos u orientados a servicios.
- Conformidad con FIPA: En cuanto a paradigmas de programación, la conformidad con FIPA se refiere a la posible integración del sistema con otros sistemas externos que también sean conformes con FIPA. Es un aspecto importante cuando se piensa en la escalabilidad de sistema.
- Dispositivos/Plataformas WSN: Se refiere a los dispositivos o plataformas para los cuales el modelo tiene soporte. Cabe destacar que existen actualmente una gran cantidad de dispositivos en el mercado, y se buscan sistemas que no se encuentren amarrados a una tecnología particular.
- Tecnologías de red: Corresponde a la comunicación de red que maneja el sistema. Esta comunicación dependerá del medio por el cual se transmite así como también los protocolos por medio de los cuales es posible la comunicación entre los nodos.
- Modularidad: Constituye quizás uno de los aspectos fundamentales en la integración. Es la posibilidad de dividir las capacidades de la arquitectura en pequeñas unidades con funciones y/o tareas muy específicas, esto con el fin de adaptarse a cambios en el modelo, nuevas tecnologías en comunicación, almacenamiento, procesamiento, entre otras.

A continuación se presenta una tabla en la cual se comparan las estrategias presentadas en la sección 2.3 de este documento bajo los tópicos presentados anteriormente. Adicionalmente se presenta dentro de los mismos tópicos el modelo propuesto en esta tesis de maestría. De esta manera se puede tener una idea del estado del modelo propuesto entre las estrategias existentes de integración entre Agentes y WSN.

Tabla 4-2. Tabla de comparación entre estrategias de integración de agentes con WSN

	Paradigma de programación	Conformidad con FIPA	Dispositivos/Plataformas	Tecnologías de Red	Modularidad
Agilla	Agentes móviles	No	Mica2	Zigbee	No
Agent Factory Micro Edition	Agentes móviles	Si	Basados en Java (e.g PDA, celulares, entre otros)	De acuerdo al dispositivo	No
SOSSA	Agentes móviles	Si	Componentes electrónicos - Robótica	-	No
Famiware	Servicios	No	TinyOS y Android. Extensible a otros	Todas	SPL approach
Modelo Propuesto	Agentes	Si	Mica2 y Tmote Sky Extensible a otros	Todas	OSGi Framework

5. CONCLUSIONES Y TRABAJO FUTURO

Nos encontramos en una época de coyuntura tecnológica, con nuevas tecnologías que se orientan a facilitar el día cotidiano del hombre común a través de servicios y disposición. Ya la tecnología no lucha por ser más eficiente y rápida, el nuevo orden exige que la tecnología sea más útil y amigable con el usuario. De esta manera se busca en los nuevos albores del siglo sistemas amigables y colaboradores que proyecten al usuario confort. Se pretende construir entornos basados en Interfaces gráficas cómodas, inteligentes y en lo posible transparentes e inteligentes que faciliten tareas complejas y se transformen en ayudantes ejemplares en cada paso del usuario.

Los principales aportes de esta Tesis de Maestría son: un modelo de Integración Middleware para la integración entre Sistemas Multi-Agente y Tecnologías de WSN. Para ello se buscó por un lado proporcionar a los agentes con el conocimiento directamente recibido de sensores distribuidos en el entorno con el fin de adquirir una consciencia del contexto para la toma de decisiones. Por otro lado en este trabajo se propuso proveer de inteligencia a los dispositivos WSN por medio de Agentes de software inteligentes que obtengan mediciones de las variables físicas en el entorno, contrario al enfoque tradicional de WSN donde se trata de dispositivos automatizados capturando datos de manera redundantes, gracias a agentes los dispositivos podrían tener un mayor ciclo de vida y por consiguiente la capacidad de monitorear los espacios durante mayor tiempo.

Uno de los aportes adicionales en esta Tesis de Maestría es el diseño y la construcción de una Representación formal del entorno, mediante una Ontología. Dicha Ontología permite establecer una comunicación con Sistemas Multi-Agentes propia de los estándares del diseño y la construcción de Sistemas Multi-Agentes. A través de la ontología el agente consigue convertir la información proveniente de la WSN en conocimiento que puede utilizar para realizar inferencias y afectar su entorno.

En esta tesis de maestría cabe resaltar la implementación utilizada y el proceso de desarrollo para la construcción de dicho modelo. En el modelo se utilizó un marco de trabajo modular que provee escalabilidad, interoperabilidad y heterogeneidad en los dispositivos de WSN existentes y próximos. Mediante el enfoque modular empleado

resulta más fácil continuar el desarrollo de nuevos módulos o bundles para proveer nuevas funcionalidades al modelo, así como también facilitar su mantenimiento y actualización.

Como trabajo futuro de esta tesis de maestría se pretende añadir el soporte para nodos móviles de WSN. Para lograr esto es necesario realizar avances en el campo de localización de dispositivos, puesto que los nodos son pequeños y se podrían extraviar. Con la utilización de dispositivos WSN móviles se podría ahorrar en dispositivos WSN, puesto que con un sólo dispositivo desplazándose en un entorno se podría cubrir mas espacio sin la necesidad de dispositivos extras.

También se tiene previsto analizar, estudiar y cuantificar el beneficio de un Sistema Multi-Agente en el rendimiento de una WSN, principalmente en lo relacionado con su ciclo de vida. Se busca a través de métricas observar el rendimiento de una WSN que es operada mediante un SMA. Se espera que como resultado se evidencie una mejoría en la longevidad de la WSN, disminuyendo los muestreos de variables físicas únicamente a muestreos planeados por agentes en el Sistema Multi-Agente.

El modelo propuesto solo es aplicable al paradigma de programación de agentes. Como otro trabajo futuro a esta Tesis de Maestría se busca extender este modelo a nuevos paradigmas de programación, tales como los orientados a servicios o los orientados a contextos; de esta manera se podría extender el alcance a aplicaciones Web o dispositivos móviles (celulares, PDA, Tablet, entre otros).

REFERENCIAS BIBLIOGRÁFICAS

Bordini, R. (2009). Multi-Agent Programming: Languages, Tools and Applications. Springer-Verlag New York Inc.

Bellifemine, F., Rimessa, G., Trucco, T, Caire, G. (2005) "JADE PROGRAMMER'S GUIDE". Marzo.

Cheong, E. (2007). Actor oriented programming for wireless sensor networks. Doctor of Philosophy in Engineering Electrical Engineering and Computer Sciences Dissertation.

Cohen, P. R., Levesque, H. J., (1990) "Intention Is Choice with Commitment." *Artificial Intelligence, Vol. 42, No. 2-3*, pp. 213-261.

Collier, R., Rooney, C.F.B., O'Donoghue, R.P.S., O'Hare, G.M.P., (2000) Mobile BDI Agents. Proceedings of the 11th Irish Conference (AICS 2000) on Artificial Intelligence & Cognitive Science, Galway, Ireland.

Collier, R., (2001) "Agent Factory: A Framework for the Engineering of Agent-Oriented Applications". Ph.D. Thesis.

Collier, R., Muldoon, C., O'Hare, G.M.P., O'Grady M.J., (2006a) Agent factory micro edition: A framework for ambient applications. Computational Science ICCS 2006, Volume 3993/2006:727–734.

Doherty, L., Warneke, B., Boser, B., Pister, K. (2001). Energy and performance considerations for smart dust. *International Journal of Parallel and Distributed Systems and Networks*, 4(3):121-133.

Dragone, M., (2007) "SoSAA: An Agent-Based Robot Software Framework", *PhD thesis, University College Dublin*.

Estrin, D., Culler, D., Pister, K., Sukhatme, G., (2002), "Connecting the Physical World with Pervasive Networks", *IEEE Pervasive Computing*, pages 59-69.

FIPA (2011), <http://www.fipa.org>. [Última visita 24/07/2011]

Fok, Ch-L., Roman, G-C., Lu, Ch., (2005) "Rapid Development and Flexible Deployment of Adaptive Wireless Sensor Network Applications." *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*.

Gámez, N., Fuentes, L., (2010) "FamiWare: a family of event-based middleware for ambient intelligence." *Pers Ubiquit Comput*. Springer. Septiembre.

Genesereth, M., Fikes R.E., (1992) Knowledge Interchange Format, Version 3.0 Reference Manual, tech. report, Computer Science Department, Stanford Univ., Stanford, California.

Georgeff, M., Pell, B., Pollack, M., Tambe, M., and Wooldridge, M. (1999). The belief-desire-intention model of agency. *Intelligent Agents V: Agents Theories, Architectures, and Languages*, pages 1-10.

Grosso, E., Eriksson, H., Ferguson, R., Tu, S., and Musen, M. (1999) "Knowledge modeling at the millennium the design and evolution of Protégé 2000". In *Proceedings of KAW-99*, Banff, Canada.

Gruber, T.R., (1993) "A Translation Approach to Portable Ontology Specifications," Technical Report, Knowledge Systems Laboratory, Computer Science Department, Stanford University, April.

Guttman, R.H., Maes, P., (1998) Cooperative vs. competitive multi-agent negotiation in retail electronic commerce, in: M. Klusch, G. Weiß (Eds.), *Cooperative Information Agents II*, in: *Lecture Notes in Artificial Intelligence*, Vol. 1435, Springer, Berlin, pp. 135–147.

Ibrahim, N., (2009) "Orthogonal Classification of Middleware Technologies", *Mobile Ubiquitous Computing, Systems, Services and Technologies, 2009. UBIKOMM '09. Third International Conference on*, pages 46 -51.

Kwon, Y., Sundresh, S., Mechitov, K., Agha, G., (2006) "ActorNet: an actor platform for wireless sensor networks." In *Proceedings of the Fifth international Joint Conference on*

Autonomous Agents and Multiagent Systems (Hakodate, Japan, May 08 - 12, 2006). AAMAS '06. ACM, New York, NY, pp. 1297-1300.

Labrou, Y., Finin, T., Peng, Y., (1999) "Agent communication languages: the current landscape," *Intelligent Systems and their Applications*, IEEE , vol.14, no.2, pp.45-52, Mar/Apr.

Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., Anderson, J., (2002) "Wireless sensor networks for habitat monitoring", *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, WSNA '02, Atlanta, Georgia, USA, pages 88-97.

McGuinness, D.L., Van Harmelen F., (2004) "OWL Web Ontology Language Overview," World Wide Web Consortium (W3C) recommendation, February.

Muldoon, C. (2008). *An Agent Framework for Ubiquitous Services*. PhD thesis, School of Computer Science and Informatics, University College Dublin.

Muldoon, C., O'Hare, G., and Bradley, J. (2007). Towards reactive mobile agents for resource-constrained mobile devices. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1-3. ACM.

O'Brien, P. D. and Nicol, R.C. (1998) "FIPA — towards a standard for software agents".

O'Hare, G., Jennings, N., (1996) "Foundations of Distributed Artificial Intelligence". Wiley-Interscience: New York, 1996.

OSGi alliance (2011), <http://www.osgi.org>. [Última visita 22/09/2011]

Piedrahita, A., Montoya, A., and Ovalle, D. (2010). Evaluación del desempeño de un modelo basado en Agentes Inteligentes en WSN con topologías irregulares. *Fifth Colombian Computing Conference*.

Preuveneers, D., Van den Bergh, J., Wagelaar, D., Georges, A., Rigole, P., Clerckx, T., Berbers, Y., Coninx, K., Jonckers, V., De Bosschere, K., (2004). "Towards an Extensible Context Ontology for Ambient Intelligence", *Department of Computer Science*, pages 148-159.

Rao, A.S, Georgeff, M.P., (1991) "Modeling rational agents within a BDI-architecture." In: J. Allen, R. Fikes, & E. Sandewall (Eds.): *Proceedings of the International Workshop on Knowledge Representation (KR91)*. pp. 473-484, Morgan Kaufmann, San Mateo CA.

Rosenschein, J., Zlotkin, G., (1994) *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. MIT Press.

Russell, S., Norvig, P., (1995) *Artificial Intelligence: A Modern Approach*. Prentice-Hall.

Sharp, H., Rogers, Y., Preece, J. (2007) "Interaction design : beyond human-computer interaction". Wiley. Pag 800.

Wang, X., Wang, S., and Jiang, A. (2006). Optimized deployment strategy of mobile agents in wireless sensor networks. In *Intelligent Systems Design and Applications, 2006. ISDA'06. Sixth International Conference on*, volume 2.

Weiser, M., (2002), "The computer for the 21st Century", *Pervasive Computing, IEEE*, jan.-march , vol 99, pages 19 -25.

Wooldridge, M., Jennings N. (1995) "Agent Theories, Architectures and Languages: A Survey". En *Intelligent Agents*, Vol. 890 de LNAI. pp. 1-39. Springer-Verlag.

Wooldridge, M., (1999) "An Introduction to Multiagent Systems", J.Wiley, England, First Edition.

Wooldridge, M., (2009) "An Introduction to Multiagent Systems", J.Wiley, England, Second Edition.

Yick, J., Mukherjee, B., and Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks*, 52(12):2292-2330.

Zhou, D., Gao, J., (2010) "Maintaining Approximate Minimum Steiner Tree and k-center for Mobile Agents in a Sensor Network," *INFOCOM, 2010 Proceedings IEEE* , vol., no., pp.1-5, 14-19 March.