



UNIVERSITY CARLOS III OF MADRID
DEPARTMENT OF TELEMATICS ENGINEERING

**SERVICE ARCHITECTING AND DYNAMIC COMPOSITION IN
PERVASIVE SMART ECOSYSTEMS FOR THE INTERNET OF THINGS
BASED ON SENSOR NETWORK TECHNOLOGY**

Author: Miguel Santos Familiar Cabero, MSc

Advisor: José Fernán Martínez Ortega, PhD

Co-advisor: Carlos García Rubio, PhD

Madrid, Spain

March 2014

A DISSERTATION SUBMITTED TO THE DEPARTMENT OF TELEMATICS ENGINEERING OF
UNIVERSITY CARLOS III OF MADRID IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY

TESIS DOCTORAL (DOCTORAL DISSERTATION)

Título (Title): Service Architecting and Dynamic Composition in Pervasive Smart Ecosystems for the Internet of Things based on Sensor Network Technology

Autor (Author): Miguel Santos Familiar Cabero, MSc

Director (Advisor): José Fernán Martínez Ortega, PhD

Codirector (Co-advisor): Carlos García Rubio, PhD

Firma del Tribunal Calificador (Review Committee):

Firma (Sign)

Presidente (Chairman): _____

Vocal (Member): _____

Secretario (Secretary): _____

Calificación (Mark): _____

Leganés, Madrid, _____ de _____, 2014

To Isabel

Πάντες ἄνθρωποι τοῦ εἰδέναι ὀρέγονται φύσει

- All men by nature desire knowledge

Aristotle (384 BC – 322 BC), Metaphysics Book Alpha

Phantasie ist wichtiger als Wissen, denn Wissen ist begrenzt

- The true sign of intelligence is not knowledge but imagination

Albert Einstein (1879 - 1955)

Service Architecting and Dynamic Composition in Pervasive Smart Ecosystems for the Internet of Things based on Sensor Network Technology

Miguel Santos Familiar Cabero

Abstract: Why pervasive awareness and Ambient Intelligence are perceived by a great part of the academia and industry as a massive revolution in the short-term? In our best knowledge, a cornerstone of this thought is based on the fact that the ultimate nature of the smart environment paradigm is not in the technology itself, but on a people-centered approach. Perhaps, is in this apparently simple conception where precisely lies the boldness of this promising vision, which has been consolidated in recent years with the emerging proliferation of mobile, personal, portable, wearable and sensory computing: to reach everyone and everywhere. On the one hand, it touches our daily lives in a close manner, minimizing the required attention from the users, anticipating to their needs with the main intention of redefining our idea of Quality of Experience. On the other hand, this new wave impacts everywhere at both global and personal scales allowing expanded connectivity between devices and smart objects, in a dynamic and ubiquitous manner, as a natural extension of the physical world around us.

According to the above, this doctoral dissertation focuses on contributing to the integration of software and networking engineering advances in the field of pervasive smart spaces and environment using sensor networks. This is founded on the convergence of some information technology and computer science paradigms, such as service and agent orientation, semantic technologies and knowledge management in the framework of pervasive computing and the Internet of Things. To this end, the nSOM (nano Service-Oriented Middleware) and nSOL (nano Semantics-Oriented Language) approaches are presented. Firstly, the nSOM proposal defines a service-oriented platform for the implementation, deployment and exposure of agent-based in-network services to the Internet cloud on heterogeneous sensor devices. Secondly, the nSOL solution enables an abstraction for supporting ubiquitous service composition based on semantic knowledge management. The integration of both contributions leads to the formal modelling and practical development of adaptive virtual sensor services for pervasive Ambient Intelligence ecosystems. This work includes also the related performance characterization of the resulting prototype according to several metrics such as code size, volatile memory footprint, CPU overhead, service time delay and battery lifetime. Main foundations and outcomes presented in this essay are contextualized in the following European Research Projects: μ SWN (FP6 code: IST-034642), DiYSE (ITEA2 code: 08005) and LifeWear (ITEA2 code: 09026).

Keywords: pervasive computing, dynamic service composition, semantic knowledge management, ambient intelligence, internet of things, wireless sensor network.

Service Architecting and Dynamic Composition in Pervasive Smart Ecosystems for the Internet of Things based on Sensor Network Technology

Miguel Santos Familiar Cabero

Extended Summary in Spanish: ¿Por qué la sensibilidad ubicua y la inteligencia ambiental son percibidas por una gran parte de las comunidades académica e industrial como una revolución masiva en el corto plazo? En nuestra opinión, una piedra angular de este pensamiento es el hecho de que la naturaleza última del paradigma de entornos inteligentes no reside en la tecnología en sí misma, sino en una aproximación centrada en las personas. Y es quizá en esta aparente simple concepción donde se halla precisamente el atrevimiento de esta prometedora visión, consolidada en los últimos años con la emergente proliferación de la computación móvil, personal, portable, llevable y sensorial: llegar a todos y a todas partes. Por un lado, esta alcanza nuestras vidas de una manera cercana, minimizando la atención requerida por los usuarios, anticipándose a sus necesidades con el objetivo de redefinir nuestra idea de calidad de experiencia. Por otro lado, esta impacta en todas partes tanto a escala global como personal, con una conectividad expandida entre dispositivos y objetos inteligentes, de un modo ubicuo y dinámico, como una extensión natural del mundo que nos rodea.

Conforme a lo anterior, esta tesis doctoral se centra en contribuir en la integración de los avances de ingeniería de redes y software en el ámbito de los espacios y entornos inteligentes ubicuos basados en redes de sensores. Esto se fundamenta en la convergencia de diversos paradigmas de las tecnologías de la información y ciencia de la computación, tales como orientación a servicios y agentes, tecnologías semánticas y de gestión del conocimiento en el contenido de la computación ubicua en la Internet de las Cosas. Para este fin, se presentan las aproximaciones nSOM (nano Service-Oriented Middleware) y nSOL (nano Semantics-Oriented Language). En primer lugar, nSOM define una plataforma orientada a servicios para la implementación, despliegue y exposición a la nube de servicios basados en agentes e implementados en red sobre dispositivos heterogéneos de sensores. En segundo lugar, nSOL habilita una abstracción para proporcionar composición ubicua de servicios basada en gestión semántica del conocimiento. La integración de ambas contribuciones conduce a un modelado formal y de implementación práctica de servicios de sensor virtual adaptativos para ecosistemas de inteligencia ambiental. Este trabajo incluye la caracterización del rendimiento del prototipo resultante, basándonos para ello en métricas tales como tamaño de código, tamaño de memoria volátil, sobrecarga de procesamiento, retardo en tiempo de servicio y autonomía de baterías. Los principales fundamentos y resultados discutidos en este ensayo están contextualizados en los siguientes Proyectos de Investigación Europeos: μ SWN (FP6 código: IST-034642), DiYSE (ITEA2 código: 08005) y LifeWear (ITEA2 código: 09026).

Este trabajo profundiza en los aspectos de estudio previos de manera más específica. Los párrafos siguientes, que han sido extraídos del capítulo de conclusiones finales, presentan un resumen pormenorizado acerca de las principales contribuciones realizadas en esta tesis.

En el análisis del estado del arte respecto a tendencias de investigación y desarrollo alrededor de la tecnología de Redes Inalámbricas de Sensores, se hace especial énfasis en aspectos de ingeniería de protocolos y arquitectura, computación orientada a servicios, ingeniería software orientada a agentes, gestión semántica del conocimiento y la Internet de las Cosas.

La discusión de lo anterior se plantea del siguiente modo: Primero, y respecto a aproximaciones en la pila de protocolos de las redes de sensores, las propuestas más recientes apuntan a diseños sensibles al consumo energético, soportando integración de datos entre capas con el fin de promover colaboración. En este caso, el verdadero desafío no está en el desarrollo de soluciones altamente optimizadas de propósito específico, sino en explotar el potencial de las mismas para alcanzar aplicaciones heterogéneas adaptativas y ubicuas que optimicen sus recursos. Segundo, el mapa de la investigación de la computación orientada a servicios incluye fundamentos de servicios, composición de servicios y gestión de servicios. Los fundamentos de servicios están centrados en el middleware orientado a servicios, el cual incluye enfoque a negocio, independencia de fabricante y funcionalidades centrales heterogéneas. Aprovechando la composición de servicios pueden crearse abstracciones virtuales de servicios basados en sensores mediante composiciones inteligentes y dinámicas. Adicionalmente, la gestión de servicios puede ayudar a proporcionar propiedades adaptativas y de resiliencia de red en computación ubicua. Tercero, la ingeniería de software orientada a agentes crea interacciones automáticas de procesos de negocio con bajo acoplamiento y basados en sensores de un modo proactivo, permitiendo así la descentralización de conocimiento y computación al mismo tiempo que potencia la coordinación entre instancias de agentes. Cuarto, la ingeniería de gestión semántica de conocimiento está motivada por la integración de diversas áreas de investigación, tales como gestión de ontología, descubrimiento de conocimiento y tecnologías de lenguaje humano. La aplicación de estos paradigmas en el desarrollo de plataformas de provisión y composición de servicios distribuidos habilita el acceso y gestión inteligente de la información. Finalmente, la Internet de las Cosas es el marco donde las aproximaciones anteriores se entrelazan para construir ecosistemas inteligentes y ubicuos. Esto permite disponer de comunicaciones en cualquier momento, lugar y medio para un amplio rango de dominios de aplicación incluyendo distribución y logística, cuidado de la salud, entorno social y personal.

Para la discusión de antecedentes en el área de tecnologías inalámbricas de sensores para el desarrollo de plataformas de servicios distribuidos, consultamos trabajos relacionados en middleware orientado a servicios y tecnologías dinámicas de composición de servicios.

La discusión de lo anterior se plantea del siguiente modo: Por un lado, la tecnología middleware ha sido identificada como un aspecto crítico en áreas emergentes como la computación en malla y en la nube, así como redes móviles y ubicuas. Las arquitecturas middleware tradicionales han sido generalmente ideadas para entornos de red distribuidos con grandes recursos, poniendo de manifiesto la necesidad de diseñar soluciones específicas que afronten la complejidad y desafíos en computación ubicua y embebida. Sin embargo, la mayoría de las propuestas middleware bien conocidas en el ámbito de las redes de sensores han sido concebidas como soluciones específicas ad hoc. Para tratar con estos aspectos, se propone una arquitectura middleware basada en el concepto de computación orientada a servicios. De

acuerdo a este paradigma, las nuevas soluciones middleware orientadas a servicios deben ofrecer independencia de plataforma y sistema operativo, funcionalidades centrales de propósito general, soporte para la composición de servicios heterogéneos, facilidad en el desarrollo de aplicaciones e interconectividad con nubes externas de servicios. De acuerdo a lo anterior, algunas de las propuestas relacionadas más significativas han sido discutidas. Por otro lado, la composición de servicios es una propiedad comúnmente asociada a los sistemas de cómputo ubicuos, la cual ha sido identificada como un aspecto clave de los mismos. Dicha composición de servicios ubicuos debe abordar el carácter inherentemente dinámico del acceso a la información en cualquier momento, lugar y medio, la heterogeneidad de dispositivos y las comunicaciones no fiables y sujetas a errores entre nodos, donde los servicios basados en sensor aparecen y se desvanecen con relativa frecuencia. Tal naturaleza dinámica demanda soluciones de entrega de servicios capaces de adaptar su composición en función de los servicios disponibles en la red. Conforme a este paradigma se han identificado los principales requisitos para plataformas futuras de composición dinámica basadas en sensor en soluciones orientadas a servicio. Esto incluye lenguajes ligeros de composición de servicios, composición de servicios adaptativa, integración de servicios dirigida por semántica y basada en ontologías, abstracciones de servicio de sensor virtual e integración basada en servicios con la Internet de las Cosas. A partir de los aspectos anteriores se ha llevado a cabo una revisión de algunas de las propuestas más significativas en este campo.

Basándonos en las contribuciones previas de análisis del estado del arte y trabajos relacionados, se define la correspondiente contribución de formalización del presente trabajo de investigación. Esta proporciona un modelo de formalización abstracto para la creación y composición dinámica y orientada a servicios en redes ubicuas de sensores, atendiendo para ello a los desafíos de diseño que han sido presentados con anterioridad.

La discusión de lo anterior se plantea del siguiente modo: Por un lado, se propone un modelo arquitectónico de referencia orientado a servicios para dispositivos heterogéneos basados en tecnología de sensores y de computación embebida. Esta aproximación de referencia define cinco niveles de abstracción, denominadas plataformas: Plataforma de Dispositivo Físico, Plataforma Middleware Orientado a Servicios, Plataforma de Servicios Ubicuos, Plataforma de Interconexión de Servicios y Plataforma de Soporte al Desarrollo. La Plataforma de Dispositivo Físico está enfocada a proporcionar independencia de fabricante, utilizando para ello encapsulación de detalles específicos tales como hardware, sistema operativo o pila de protocolos de red. La implementación de una abstracción middleware reusable con provisión de funcionalidades genéricas reside en la Plataforma Middleware Orientado a Servicios. El soporte para composición multi-servicio se define en los nodos mediante la Plataforma de Servicios Ubicuos, soportando así composición dinámica de servicios de sensor virtual entre nodos, mediante la Plataforma de Interconexión de Servicios, la cual además provee operación con nubes de servicios en redes externas. Finalmente, la Plataforma de Soporte al Desarrollo contempla la definición y adopción de metodologías y aproximaciones que, aplicadas tanto al middleware como a las entidades de servicios basados en sensor, permitan el desarrollo masivo de software de un modo rápido, con bajo coste y fácilmente extensible. Por otro lado se

contempla la formalización abstracta de un modelo de composición de servicio virtual basado en tecnología de sensores y computación embebida, materializado en el paradigma de integración de servicios que implementa el marco de referencia orientado a servicios. Este propone, de un modo conforme a los conceptos de perfil de servicio, modelo de proceso y despliegue de servicios de la especificación OWL-S (Web Ontology Language for Services), la noción de servicio ubicuo en red PS_i . Adicionalmente, se incluye la definición de capacidades en términos de lógica de negocio discreta, función de transición, resultado de servicio esperado, operaciones de entrada y salida y precondiciones de servicio. La propuesta anterior se extiende con la definición formal de servicio de sensor virtual $PVSS_i$, como soporte a la composición de procesos atómicos de servicios ubicuos en red, al mismo tiempo que incluye información de acceso al servicio. Finalmente, se especifica la definición formal de Plataforma de Servicios Ubicuos PSP dentro del modelo de referencia orientado a servicios en términos de lógica de negocio de composición, servicios ubicuos en red y servicios de sensor virtual, en un escenario de despliegue e instante de tiempo específicos, con un conjunto de interfaces entre plataformas adyacentes de la arquitectura propuesta.

Como la primera de las contribuciones de aplicación práctica, se discute la aproximación nSOM. Esta representa la realización del modelo marco de referencia orientado a servicios, abordando el diseño e implementación del middleware, así como el flujo de trabajo basado en tecnología de agentes para exposición y descubrimiento de servicios definido para los modelos de operación e interconexión de servicios internos al nodo y entre nodos.

La discusión de lo anterior se plantea del siguiente modo: Nuestra aproximación nSOM adopta una arquitectura basada en componentes software en la Plataforma Middleware Orientada a Servicios e implementa una solución de servicios ubicuos en red, utilizando para ello agentes software en la Plataforma de Servicios Ubicuos y la Plataforma de Interconexión de Servicios. Primero, el middleware orientado a servicios define un conjunto de funcionalidades básicas de propósito general incluyendo servicios de control, servicios de bajo nivel, servicios de alto nivel y servicios ortogonales. Estos proporcionan a las capas superiores un marco de arquitectura orientada a servicios para el desarrollo rápido de procesos de aplicación, con garantías de interoperabilidad y fácil ensamblaje de los mismos. Así mismo, este enfoque enfatiza en la propuesta de servicios basados en tecnología de agentes software, los cuales se definen como entidades autónomas, reusables, con alta portabilidad y no propietarias. Segundo, se han implementado mecanismos de comunicación e interacción entre componentes software y entidades de aplicación locales al nodo, así como procedimientos de interacción entre nodos de la red. Por lo que respecta a la interacción local al nodo, la Plataforma de Servicios Ubicuos constituye una abstracción de contenedor orientada a agentes para el despliegue y ejecución de servicios ubicuos en red, conforme al modelo de programación orientada a aspectos. Esta capa ofrece además capacidad de descubrimiento dinámico de componentes del middleware en tiempo de carga, así como un procedimiento uniforme de interacción entre servicios basado en información sensible a contexto. Finalmente, la contribución afronta composición multi-servicio e interoperabilidad con redes externas de servicios utilizando para ello el soporte de interacción entre nodos. Este se implementa en la Plataforma de Interconexión de Servicios

mediante las entidades de red Orquestador e Intermediador modeladas como procesos basados en agentes software. Para la comunicación entre los anteriores y los nodos de dispositivo inalámbricos, así como con la Plataforma orientada a servicio de Servidor de Aplicaciones vía nodo sumidero/pasarela, se introduce un protocolo de exposición y descubrimiento de servicios para computación de dispositivos embebidos, el cual se apoya en la definición de una Unidad de Datos de Protocolo optimizada para tal efecto, denominada nSOM-PDU.

Lo anterior se completa con la segunda contribución de aplicación práctica, la cual se ha denominado nSOL. Esta define una solución centrada en composición para instanciar servicios de sensor virtual abordando la definición de una ontología para computación ubicua dirigida por semántica basada en la definición concreta de un lenguaje de definición de interfaz, así como una sintaxis de notación ligera de servicios.

La discusión de lo anterior se plantea del siguiente modo: La contribución nSOL propone una implementación para permitir la composición y provisión de servicios ubicuos de un modo dinámico y adaptativo. Se aborda la formalización concreta para la representación de servicios ubicuos en red y su agregación en servicios de sensor virtual mediante una propuesta que aplica gestión semántica del conocimiento basada en ontología. Esto permite la concepción de anotaciones orientadas a potenciar la composición, el descubrimiento y la invocación de servicios en computación ubicua basada en sensores. Con el fin de minimizar la sobrecarga en la transmisión y procesamiento de datos se propone el uso del estándar JSON (JavaScript Object Notation) como formato ligero, independiente de lenguaje y auto-descriptivo de representación e intercambio de datos. Adicionalmente, se ha aplicado la solución SMD (Service Mapping Description) para la definición de descripciones de servicios e interfaces de invocaciones en entornos de computación ubicua. Las descripciones SMD de interfaz de servicio están anotadas de acuerdo al modelo de datos de grafo basado en ontología con el estándar RDF N3 (Resource Description Framework Notation3). Este permite definir una sintaxis compacta, proporcionando un modo simple de expresar triplas al mismo tiempo que contiene la sobrecarga del lenguaje. De acuerdo a lo anterior, se define el flujo de trabajo de composición de servicios. Este se ejecuta en los agentes de orquestación, los cuales instancian en tiempo de ejecución los servicios de sensor virtual de acuerdo a la plantilla de composición, los anuncios propagados en la red inalámbrica de sensores al agente de intermediación designado, y los candidatos de servicio obtenidos del criterio de selección. Estas abstracciones de sensor virtual son registradas en la nube a través de la Plataforma orientada a servicio de Servidor de Aplicaciones, la cual proporciona una integración efectiva, dinámica, ágil y fácil de procesos de negocio en redes distribuidas escalables apoyándose en una conectividad de servicios abierta, normalizada y multi-interfaz.

Adicionalmente, se presenta un estudio de caracterización del rendimiento de las contribuciones previas. Este análisis aborda desde la definición del escenario y configuración del prototipo hasta las métricas de ingeniería de software aplicadas durante la validación experimental. Dichas métricas de rendimiento incluyen tamaño de código, tamaño de memoria volátil, sobrecarga de procesamiento, retardo en tiempo de servicio y autonomía de baterías de los nodos inalámbricos.

La discusión de lo anterior se plantea del siguiente modo: Se ha implementado un prototipo de servicio personal ambiental de sensor virtual (PEVSS), utilizando para ello la plataforma programable Sun SPOT (Small Programmable Object Technology) de Oracle conforme a los objetivos definidos para el demostrador del proyecto de investigación DiYSE. El servicio compuesto resultante está basado en la integración de servicios ubicuos en red implementados mediante tecnología de agentes: temperatura, luz y ruido. Basándonos en el análisis de los resultados obtenidos podemos concluir que se ha obtenido una implementación ligera con un buen rendimiento para la creación de servicios y la provisión dinámica de servicios de sensor virtual, incluyendo una definición de interfaz compacta con un grado de respuesta de servicios basados en sensor y latencias de servicio aceptables. En lo referente a tamaño de código, la ocupación de memoria Flash es menor del 1,5% del tamaño disponible en los nodos, mientras que en el caso peor la ocupación de memoria volátil es del 20,6% de la RAM de los nodos inalámbricos de sensores y del 28,5% en la caso de nodo Intermediador/Orquestador. En lo que respecta a la sobrecarga de procesamiento, a una tasa de una transacción por segundo en los nodos inalámbricos de sensores la carga de procesador fue del 4,6% y del 15,5% en el caso del nodo Intermediador/Orquestador con tres transacciones por segundo, con una gestión contenida y acotada de los hilos de usuario. Como parte del estudio del retardo en tiempo de servicio se ha incluido una comparación entre SMD/JSON y los formatos tradicionales WSDL (Web Service Description Language) y XML, el cual muestra una optimización de 2,6:1 y 1,4:1 para la definición de interfaz de servicio y el intercambio de solicitud/respuesta de datos, respectivamente. Para completar el estudio de esta métrica de rendimiento, se presentan medidas específicas adicionales. En retardo promedio de registro de servicio en el nodo Intermediador/Orquestador es de $105,7 \pm 30,1$ ms, mientras que el tiempo promedio de ida y vuelta para un servicio ubicuo en red es de $171,2 \pm 46,8$ ms. El tiempo promedio para instanciación virtual en el nodo Intermediador/Orquestador es de $41,9 \pm 16,5$ ms y el tiempo de adquisición de datos de sensor virtual observado es de $326,3 \pm 80,4$ ms. Finalmente, el tiempo de autonomía de las baterías para los nodos de sensores mostraba una autonomía limitada, operando con un ciclo de trabajo del 100% en esta plataforma Java orientada al desarrollo rápido de prototipos. Dicha autonomía varía entre las 6,2 y las 10,6 horas aproximadamente, en función del número de transacciones por segundo cursadas.

Para concluir, se esbozan las principales líneas de investigación futuras como posible continuación del trabajo presentado en esta disertación. El eje central de las mismas reside en un aspecto abierto en computación ubicua: la gestión autónoma de servicios. En este sentido, nuestra propuesta va orientada a la investigación de mecanismos de resiliencia ubicuos y autónomos para proporcionar planos de control inteligentes, permitiendo así capacidad de auto-gestión de planos de datos con orientación a servicios e interoperabilidad entre dominios, que permita un balanceo eficiente y justo entre los requerimientos heterogéneos demandados por los servicios y los recursos disponibles en la red ubicua. En este sentido, se propone la aplicación de ontologías sensibles a Calidad de Servicio para el modelado de requisitos de servicios y recursos de red, así como de anotaciones semánticas embebidas en los mismos para describirlos de manera formal y de algoritmos de encaminamiento multi-restricción para la construcción de rutas en tiempo de ejecución que satisfagan los acuerdos de nivel de servicio específicos.

Contents

Abstract	IX
Extended Summary in Spanish.....	XI
Contents.....	XVII
List of Figures	XXIII
List of Tables.....	XXVII
Abbreviations	XXIX
INTRODUCTION AND OBJECTIVES.....	1
Chapter 1: Introduction	3
1.1 Wireless Sensor Networks.....	3
1.1.1 Design Principles.....	4
1.1.2 Emerging Research	5
1.2 Objectives and Thesis Contributions	6
1.3 Research Framework	8
1.4 Dissertation Outline	9
STATE OF THE ART AND BACKGROUND.....	11
Chapter 2: State of the Art.....	13
2.1 Architecting Wireless Sensor Networks	13
2.1.1 Physical Layer.....	14
2.1.2 Data Link Layer	15
2.1.3 Network Layer.....	15

2.1.4	Transport Layer	16
2.1.5	Application Layer.....	17
2.2	Service-Oriented Computing.....	18
2.2.1	Service-Oriented Architecture as a Paradigm	18
2.2.2	Service-Oriented Engineering.....	19
2.2.2.1	Service Foundations	20
2.2.2.2	Service Composition.....	20
2.2.2.3	Service Management	21
2.2.3	Research Challenges in Wireless Sensor Networks.....	21
2.3	Agent-Oriented Software Engineering	22
2.3.1	Agent Theory	23
2.3.2	Multi-Agent Systems and Standards.....	25
2.3.3	Research Challenges in Wireless Sensor Networks.....	27
2.4	Semantic Knowledge Management	28
2.4.1	Knowledge Foundations.....	28
2.4.2	Towards Semantic Knowledge Engineering.....	30
2.4.2.1	Ontology Management	30
2.4.2.2	Knowledge Discovery	31
2.4.2.3	Human Language Technologies	32
2.4.3	Research Challenges in Wireless Sensor Networks.....	33
2.5	Internet of Things	34
2.5.1	Enabling Technologies for Building the Internet of Things	34
2.5.2	Applications Domains based on Internet of Things Ecosystems	36
2.5.2.1	Transportation and Logistics Domain	36
2.5.2.2	Healthcare Domain	37
2.5.2.3	Personal and Social Domain.....	37
2.5.3	Research Challenges in Wireless Sensor Networks.....	38
2.6	Conclusions.....	39
Chapter 3:	Related Work.....	43
3.1	Related Work in Middleware Technologies	43
3.1.1	Middleware Foundations.....	44
3.1.2	Limitations of Traditional Middleware Frameworks	44

3.1.3	Mainstream Middleware for Sensor Networks	45
3.1.4	Service-Oriented Middleware for Wireless Sensor Networks	48
3.1.4.1	MiSense	48
3.1.4.2	SStreaMWare.....	49
3.1.4.3	TinySOA.....	50
3.1.4.4	USEME.....	52
3.1.4.5	SOMM	53
3.1.4.6	ESOA.....	54
3.1.4.7	MidCASE	55
3.1.4.8	UbiSOAP	56
3.1.5	Discussion	57
3.2	Related Work in Service Composition	59
3.2.1	Services and Composition Methodologies.....	59
3.2.2	Service Integration and Composition in Pervasive Computing	60
3.2.3	Dynamic Service Composition in Wireless Sensor Networks	61
3.2.3.1	SCPQ	61
3.2.3.2	SDCBAN	62
3.2.3.3	SWSN	63
3.2.3.4	DSIDMSS	64
3.2.3.5	MRSWSN	66
3.2.3.6	MORE-WSBPEL.....	67
3.2.3.7	Orchestrator	68
3.2.3.8	SNMAIW.....	69
3.2.4	Discussion	70
3.3	Conclusions.....	72
RESEARCH CONTRIBUTIONS FORMALIZATION, DESIGN, IMPLEMENTATION AND ANALYSIS.....		75
Chapter 4: Formalization Model		77
4.1	Problem Statement.....	77
4.2	Formalization of the Service-Oriented Framework.....	79
4.2.1	Physical Device Platform	79
4.2.2	Service-Oriented Middleware Platform	80
4.2.3	Pervasive Service Platform	80

4.2.4	Service Internetworking Platform	80
4.2.5	Development Support Platform.....	80
4.3	Formalization of the Service Composition Model.....	81
4.3.1	Conceptualization of a Pervasive In-Network Service	81
4.3.2	Composition Approach for Pervasive Virtual Sensor Services	81
4.3.3	Service Model of the Service-Oriented Framework	82
4.4	Conclusions.....	82
Chapter 5: nSOM: A Service-Oriented Middleware for Pervasive Smart Environments.....		85
5.1	Overview of the nSOM Approach.....	85
5.2	Middleware Architecture	87
5.2.1	Control Services Layer.....	87
5.2.2	Cross Services Layer.....	88
5.2.3	Low-Level Services Layer	89
5.2.4	High-Level Services Layer	89
5.3	Middleware Workflow	90
5.3.1	Intra-node Model.....	90
5.3.2	Inter-node Model.....	92
5.4	Conclusions.....	98
Chapter 6: nSOL: A Semantics-Oriented Language for Ubiquitous Knowledge Management.....		101
6.1	Overview of the nSOL Approach	101
6.2	Semantic-Driven Pervasive Service Composition.....	102
6.2.1	Ontology-based Graph Data Model	103
6.2.2	Composition-centric Service Engineering in nSOL / nSOM	105
6.3	Interface Language and Semantic Notation Syntax	106
6.3.1	Interface Definition using Service Mapping Description	106
6.3.2	Semantic Annotation for Service Architecting and Composition.....	108
6.3.3	Pervasive Service-Oriented Integration into the Internet Cloud	111
6.4	Conclusions.....	111
Chapter 7: Performance Characterization		115
7.1	Testbed Definition.....	115
7.1.1	Prototype Modelling.....	116
7.1.2	Development Platform	117

7.2 Code Size.....	119
7.3 Volatile Memory Footprint.....	121
7.4 CPU Overhead.....	122
7.5 Service Time Delay.....	124
7.6 Battery Lifetime.....	125
7.7 Conclusions.....	127
CONCLUSIONS AND FURTHER RESEARCH.....	129
Chapter 8: Conclusions.....	131
8.1 Conclusions.....	131
8.2 Further Research and Directions.....	136
8.3 List of Publications.....	137
References.....	139

List of Figures

Figure 2.1:	<i>Wireless Sensor Network protocol stack</i> (Akyildiz et al. [12]).	14
Figure 2.2:	<i>Service-Oriented Computing research roadmap</i> (Papazoglou et al. [224]).	20
Figure 2.3:	<i>Primary agent typology</i> (Nwana [212]).	24
Figure 2.4:	<i>Framework of the FIPA specifications</i> (IEEE FIPA [135]).	26
Figure 2.5:	<i>Knowledge dimensions: (a) explicit knowledge and tacit knowledge</i> (Nonaka [209]); <i>(b) declarative knowledge and procedural knowledge</i> (Nickols [73]).	29
Figure 2.6:	<i>Perspectives of the Internet of Things paradigm</i> (Atrozi et al. [25]).	35
Figure 3.1:	<i>Mainstream middleware for Wireless Sensor Networks</i> (Hadim and Mohamed [121]).	45
Figure 3.2:	<i>The MiSense overall framework</i> (Khedo and Subramanian [152]).	49
Figure 3.3:	<i>The SStreaMWare middleware</i> (Gurlen et al. [120]).	50
Figure 3.4:	<i>Overview of the TinySOA approach</i> (Avilés-López and García-Macías [27]).	51
Figure 3.5:	<i>The USEME service-oriented middleware</i> (Cañete et al. [55]).	52
Figure 3.6:	<i>The SOMM service-oriented architecture</i> (Faghih and Moghaddam [97]).	53
Figure 3.7:	<i>The ESOA middleware platform</i> (Vanitha et al. [282]).	54
Figure 3.8:	<i>The architecture of the MidCASE framework</i> (Bai et al. [28]).	55
Figure 3.9:	<i>The ubiSOAP middleware contribution</i> (Caporuscio et al. [60]).	56
Figure 3.10:	<i>The SCPQ service composition approach</i> (Wang et al. [293]): <i>(a) example of persistent query in a service-oriented sensor network; (b) overlay system model.</i>	62

Figure 3.11: <i>Overview of the SDCBAN service framework (Coloberti et al. [70]): (a) network architecture; (b) TGD based on multi-agent FSM keeping track of service availability.</i>	63
Figure 3.12: <i>Service-Centric Solution for Wireless Sensor Networks (Zhou et al. [315]): (a) EWSN sensor-based architecture; (b) SWSN dynamic service platform.</i>	64
Figure 3.13: <i>The DSIDMSS proposal for redundant service provisioning and binding in rescue systems (Liu et al. [175]).</i>	65
Figure 3.14: <i>The MRSWSN contribution (del Cid et al. [85]): (a) middleware architecture; (b) configuration semantics for components.</i>	66
Figure 3.15: <i>The MORE-WSBPEL middleware (Timm et al. [274]): (a) MORE design; (b) mapping, design and deployment environment.</i>	67
Figure 3.16: <i>The overview of the Orchestrator service infrastructure (Kang et al. [151]): (a) architecture on a mobile device; (b) architecture on a sensor device.</i>	68
Figure 3.17: <i>The architecture of the SNMAIW platform (Sardis et al. [249]).</i>	69
Figure 4.1: <i>UML Package Diagram of the reference service-oriented framework.</i>	79
Figure 5.1: <i>Overview of the nSOM architecture proposal.</i>	86
Figure 5.2: <i>UML Component Diagram of the Service-Oriented Middleware Platform.</i>	88
Figure 5.3: <i>UML Sequence Diagram of the intra-node communication workflow.</i>	91
Figure 5.4: <i>Inter-node internetworking model: proposed network architecture.</i>	93
Figure 5.5: <i>nSOM-PDU format: (a) common header; (b) service object.</i>	93
Figure 5.6: <i>UML Sequence Diagram of the inter-node communication workflow.</i>	96
Figure 6.1: <i>Semantic digraph of pervasive in-network service modelling and composition.</i>	103
Figure 6.2: <i>Semantic annotation example for service announcement.</i>	109
Figure 6.3: <i>Semantic annotation example for service data invocation.</i>	109
Figure 6.4: <i>Semantic annotation example for service data response.</i>	110
Figure 7.1: <i>Characterization scenario for prototype analysis.</i>	116
Figure 7.2: <i>Oracle Sun SPOT hardware platform (Oracle Corporation).</i>	118
Figure 7.3: <i>CPU overhead analysis in wireless sensor node: (a) processor usage; (b) threads management.</i>	122

Figure 7.4: <i>CPU overhead analysis in Broker/Orchestrator node: (a) processor usage;</i> <i>(b) threads management.</i>	123
Figure 7.5: <i>Battery lifetime in wireless sensor and Broker/Orchestrator nodes.</i>	126

List of Tables

TABLE 3.1: <i>Comparative analysis of service oriented middleware architectures for Wireless Sensor Networks.</i>	58
TABLE 3.2: <i>Comparative analysis of dynamic service composition architectures for Wireless Sensor Networks.</i>	71
TABLE 5.1: <i>nSOM service discovery protocol: Messages specification</i>	94
TABLE 7.1: <i>Code size for Service-Oriented Middleware Platform</i>	119
TABLE 7.2: <i>Code size for Pervasive Service Platform</i>	120
TABLE 7.3: <i>Code size for Service Internetworking Platform</i>	120
TABLE 7.4: <i>Volatile Memory Footprint</i>	121

Abbreviations

ACL	Agent Communication Language
AmI	Ambient Intelligence
AOP	Aspect-Oriented Programming
AOSE	Agent-Oriented Software Engineering
API	Application Programming Interface
BAN	Body Area Network
BDI	Believe, Desire, and Intention
BPEL	Business Process Execution Language
CLDC	Connected Limited Device Configuration
CORBA	Common Object Request Broker Architecture
CSMA-CA	Carrier Sense Multiple Access-Collision Avoidance
DCOM	Distributed Component Object Model
DPWS	Device Profile for Web Services
EAI	Enterprise Application Integration
EJB	Enterprise JavaBeans
ESB	Enterprise Service Bus
FIPA	Foundation for Intelligent Physical Agents
FSM	Finite State Machine
GUI	Graphical User Interface

IDE	Integrated Development Environment
IDL	Interface Description Language
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IoT	Internet of Things
IP	Internet Protocol
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
KM	Knowledge Management
KSE	Knowledge Sharing Effort
MAC	Medium Access Control
MAS	Multi-Agent System
MASIF	Mobile Agent System Interoperability Facilities
MIME	Multipurpose Internet Mail Extensions
MOM	Message-Oriented Middleware
M2M	Machine to Machine
NFC	Near Field Communications
nSOL	nano Semantics-Oriented Language
nSOM	nano Service-Oriented Middleware
OMG	Object Management Group
OSGi	Open Services Gateway initiative
OWL-S	Web Ontology Language for Services
PDU	Protocol Data Unit
QoE	Quality of Experience
QoS	Quality of Service
RDF	Resource Description Framework
RDF N3	RDF Notation3
REST	Representational State Transfer
RFC	Request for Comments
RFID	Radio Frequency Identification

RMI	Remote Method Invocation
RPC	Remote Procedure Call
SMD	Service Mapping Description
SNMP	Simple Network Management Protocol
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SoC	Separation of Concerns
SOC	Service-Oriented Computing
SPARQL	SPARQL Protocol and RDF Query Language
SQL	Structured Query Language
TCP	Transmission Control Protocol
TPS	Transaction Per Second
UDP	User Datagram Protocol
UI	User Interface
UML	Unified Modelling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UWB	Ultra Wideband
W3C	World Wide Web Consortium
WMSN	Wireless Multimedia Sensor Network
WS	Web Service
WSDL	Web Service Description Language
WSN	Wireless Sensor Network
XML	Extensible Markup Language
6LoWPAN	IPv6 over Low-Power Wireless Personal Area Network

PART I

INTRODUCTION AND OBJECTIVES

Introduction

*P*ERVASIVE computing, also called ubiquitous computing, is becoming a reality nowadays. This paradigm, as initially imagined by Mark Weiser in the early 90's, defines a vision of ecosystems where digital technology and wireless communication devices are highly integrated into the objects of the physical world, thus unnoticeable to the human eyes. The development of personal services on these devices leads to the creation of smart spaces and ambient intelligence environments which are conceived as an overlay, as a digital infrastructure that proactively, but sensibly, supports people in their every daily lives.

In this way, the scientific and research communities have done a significant effort during the last decade to empower next-generation ubiquitous services, with the aim of their dissemination at large-scale. The features of smart environments present important research open issues and challenges for the development of embedded architectures in devices of reduced functionality, such as wireless ad hoc and sensor networks.

The rest of this chapter is organized as follows. First, in Section 1.1 we overview the fundamentals of Wireless Sensor Networks. Second, the objectives and contributions of this doctoral thesis are described in Section 1.2. Third, in Section 1.3 the research framework where this work has been carried out is presented. Finally, in Section 1.4 we review the outline of this dissertation followed to achieve the expected objectives.

1.1 Wireless Sensor Networks

Wireless Sensor Network (WSN) technology is at the convergence point of communication networks, software engineering and distributed systems in reduced functionality devices. As a key piece of ubiquitous computing [295], this type of ad hoc networks are composed by tiny nodes with the ability to model the environment through sensors, processing the gathered

information and influencing by means of actuator devices, providing a significant number of features that make them an interesting solution for a wide range of fields. These cover from military forces and battlefield supervision [162] to target control and location [173], including security and surveillance [66], healthcare infrastructures [154], inventory management [187], monitoring of ambient parameters and disaster areas [77], tracking and handling of goods and foods [77, 313] or underwater environments [96], among many others.

Previous application domains are enhanced by ubiquitous features of Wireless Sensor Networks. These include small node size, wireless networking capabilities and battery powered autonomy. Additionally, sensor nodes can be deployed in inaccessible geographical areas in a flexible way, without relying on a fixed network topology and using network protocols and algorithms which provide them with self-organizing capacity. However, it is the own pervasive nature of these networks that defines its main limitations. A wireless sensor node is inherently resource constrained mainly by its memory capacity, computing power, bandwidth and battery lifetime. This is the main reason why most software approaches to such networks are ad hoc solutions, where software development is usually done from scratch [51].

1.1.1 Design Principles

The design of a WSN is influenced by several design factors [12] such as fault tolerance, scalability, production costs, operating environment, network topology, hardware constraints, transmission media and energy consumption. Understanding these factors is a basic issue in order to enhance the software and networking engineering processes in such embedded and reduced functionality devices.

- **Fault tolerance:** Sensor nodes can break down for several reasons, including blocking or battery depletion. The failure of a node should not affect the overall operation of the sensor network. The fault tolerance of a Wireless Sensor Network is defined, therefore, as the ability of the network to continue providing its functionality without interruption, even when failure situations involving one or more wireless sensor nodes.
- **Scalability:** The set of nodes that compose a sensor network can be of the order of hundreds or even thousands, depending on the specific application. Therefore, new mechanisms to manage this high number of sensor devices efficiently are required, where the range of node density can be from a few sensor nodes to hundreds of them in places with a diameter of tens of meters.
- **Production costs:** Due to the large number of nodes that can form a Wireless Sensor Network, the cost of each node is very important to justify the overall cost of the network. In order to assure a viable deployment and operation of the WSN in economic terms, the cost of each of the nodes has to be less than the deployment of other related radio technologies, such as Bluetooth or Radio Frequency Identification (RFID).
- **Operating environment:** The nodes making up a Wireless Sensor Network are deployed near to the observation perimeter, or even in the same area. Due to the small device size,

wireless communication support and being powered by batteries without wires, this technology is suitable for its deployment in remote or inaccessible geographical areas.

- **Network topology:** Massive deployment of nodes in a WSN requires a proper management of the topology maintenance. This issue covers aspects such as pre-deployment and deployment issues (e.g., particular or massive node installation), post-deployment issues (e.g., node replacement or location update), and re-deployment issues (e.g., addition of new nodes to the embedded network).
- **Resource constraints:** Wireless sensor nodes have important constraints in terms of computation, memory, bandwidth and available power. Besides this, additional restrictions imposed by such ad hoc networks should be considered. These include the need of extremely low power consumption, the ability to operate in high node density areas with autonomy and unattended operation, as well as adaptability to the specific environment conditions.
- **Transmission medium:** Wireless links between nodes can be radio, infrared, or optical. Most wireless sensor hardware platforms are based on radio-frequency circuits, such as Bluetooth transceivers in the 2.4GHz band, or single-channel radio transceivers operating at 916MHz. Other common communication systems are infrared (free license communication and robust to interference) or optical systems, but these have the limitation of requiring direct sight line between terminals.
- **Energy consumption:** Wireless sensor nodes use power supplies with limited autonomy, such as batteries. In some scenarios, the replacement of these power sources could not be possible, where in most cases the node lifetime is highly dependent on the autonomy of its batteries. Due to the nature of the sensor nodes, energy consumption is divided into phases of data capture, data processing and communication.

1.1.2 Emerging Research

The recent advances in processing and memory technologies, as well as novel developments in power supplies and batteries autonomy, have led in that sensor networks are no longer a basic input device responsible for gathering environmental data and transmit them to an external management centre where they are processed. The real challenge of this computing paradigm is not to develop highly-efficient single-purpose applications, but to exploit its potential for the development of pervasive service provisioning while optimizing the physical resources.

In this regard, there are several topics that illustrate the state-of-the-art research and development in Wireless Sensor Networks. These includes, among others, cross-layer design techniques [192], pervasive security [185], bio-inspired networking [93], context-awareness [71], fuzzy logic systems [255], body area networks [307], knowledge-aware computing [72], service discovery architectures [196], information theory and network coding [101], opportunistic routing [170] and wireless multimedia sensor networks [18]. Moreover, and within the context of this dissertation, some other novel research directions in Wireless Sensor

Networks can be highlighted. Next paragraphs introduce these topics, which will be surveyed in detail in Section 2.

- **Service-Oriented Computing:** This software engineering approach encourages the notion of building dynamic service networks with application entities in a loosely coupled and interoperable way. In this regard, Service-Oriented Computing enables the development of rapid, evolvable, interoperable and easy assembly business application processes, where services are defined as autonomous, self-described, reusable, highly-portable and vendor-neutral entities [224].
- **Agent-Oriented Software Engineering:** Agents are commonly identified as autonomous, adaptive software entities with a goal-driven behaviour, managing resources and being able to interact with the environment, communicating and cooperating with other agents [46]. Taking into account the autonomy, flexibility and robustness provided by the agent technology, Agent-Oriented Software Engineering is perceived as a very promising approach for building service architectures over Wireless Sensor Networks [286].
- **Semantic Knowledge Management:** Knowledge Management is a research area which comprises a set of methods, practices and strategies defined in order to simplify the processes related to represent, create, extract, organize, transfer and retrieve knowledge in information systems [89]. The combination of this approach with semantic technologies provides a powerful framework for dynamic and interoperable management of knowledge assets [83].
- **Internet of Things:** Next-generation computing remarks the vision of future digital ecosystems composed by interconnected devices. The combination of the Internet with the ubiquitous networking technologies lets to convert everyday use objects into smart objects [156]. This is the basis for building pervasive smart environments while offering end-users a rich set of facilities for improving their daily life experiences, integrating seamlessly with the real-world environment where they are deployed.

1.2 Objectives and Thesis Contributions

The main motivation of this dissertation is *to contribute to the research of distributed service architecting and delivery platforms for dynamic provisioning of virtual services based on sensor technology in pervasive smart spaces and environments for the Internet of Things*. To accomplish the previous, the work presented in this essay is centred on the software and networking engineering advances in the domain of wireless ad hoc and sensor networks for the Information and Networked Society, provided by means of the convergence of several paradigms in the field of information technology and computer science. These include service and agent orientation, semantic technologies and knowledge management in the framework of pervasive computing and the Internet of Things. The application and integration of such approaches establishes a promising research direction in next-generation sensor networks.

In this context, the core objectives of this dissertation are summarized as follows. Around these starting point requirements, main contributions of this thesis are provided.

- To formalize, design, implement and validate a full service-oriented middleware for pervasive smart environments based on Wireless Sensor Network technology. This reference architecture accomplishes the following requirements. First, it enables discovery, access and sharing of the network resources, data and services, while providing interoperability, low-coupling and flexibility in heterogeneous networking infrastructures. Second, it implements lightweight and efficient procedures for service internetworking with heterogeneous and external networks, allowing more advanced and complex services. Third, it offers features and tools for the development of sensor-based services by third parties without requiring technical skills in embedded computing.
- To formalize, design, implement and validate a dynamic, semantically-driven service composition schema for pervasive smart environments based on Wireless Sensor Network technology. To accomplish the previous, this proposal defines the following requirements. First, pervasive in-network services are modelled using software agents with the separation of concerns approach, in order to minimize the functionality overlap between them. Second, a service composition approach using semantic knowledge management techniques based on a service ontology definition in the orchestration process is defined. Third, a service discovery protocol is implemented including procedures for dynamic service advertisement and invocation both in the wireless sensor domain and on the Internet cloud.

In order to achieve the previous objectives, the following analyses will be conducted as the main *state-of-the-art evaluation contributions* of this dissertation:

1. Analysis of the state-of-the-art protocol foundations for Wireless Sensor Networks and software research and development in Service-Oriented Computing, Agent-Oriented Software Engineering, Semantic Knowledge Management and the Internet of Things paradigm. This study will enable us to understand the background of these approaches, as well as how to apply and integrate those according to the resource-constrained nature of such embedded and ad hoc networks.
2. Analysis of the related work regarding middleware technologies and approaches for pervasive computing. This includes theoretical foundations, legacy proposals for traditional networks and related solutions specifically defined for Wireless Sensor Networks. This study is centred on surveying the main research trends on architecting ubiquitous middleware platforms, with the aim of discussing our contributions according to the state of the art in service-oriented sensor networks.
3. Analysis of the related work regarding ubiquitous service composition methodologies and proposals, from the general design issues in composition-centric architectures to the specific approaches for dynamic service integration in Wireless Sensor Networks. This study focuses on distilling the major research lines and trends on architecting pervasive service composition platforms, with the aim of placing our contribution within the state of the art in adaptive sensor-based service composition.

4. Analysis of the engineering requirements in pervasive smart spaces and environments covered with service orientation and composition architectures. This study will allow us to identify the required design principles and challenges for modelling a reference framework for heterogeneous reduced functionality devices. This will enable full service-oriented functionalities in the middleware and semantically-driven dynamic virtual service composition, supporting integration with external network clouds.

From the conclusions obtained in these analyses, main *formalization and practical application contributions* provided by this dissertation can be classified as follows:

1. Abstract formalization of a reference service-oriented framework for pervasive networks based on sensing and embedded computing. In this specification, a requirement definition process for a Smart Environment will be performed. This will lead to provide a common abstraction to design and implement Service-Oriented Architectures for pervasive smart infrastructures and ecosystems based on heterogeneous wireless sensor technology.
2. Abstract formalization of a virtual sensor service composition model for pervasive networks based on sensing and embedded computing. In this specification, both pervasive service fundamentals and composition approaches for pervasive services are covered. This theoretical study will be oriented to define a virtual sensor composition-centric paradigm for pervasive in-network services in ubiquitous smart spaces and environments.
3. Practical approach for a service-oriented middleware for Wireless Sensor Networks. This contribution is called nSOM (nano Service-Oriented Middleware). It represents our practical implementation of the reference service-oriented framework, including the middleware design, implementation and validation, as well as the definition of the service discovery model for both intra-node and inter-node interactions.
4. Practical approach for a semantically-driven virtual service composition model for Wireless Sensor Networks. This contribution is called nSOL (nano Semantics-Oriented Language). It represents our proposed design, including related implementation and validation, of the pervasive service composition model which defines the ontology-based orchestration scheme of sensor-based, pervasive in-network services developed using agent orientation.

Finally, the main outcome of this doctoral thesis has been on *knowledge dissemination contributions*, where the main aspects of the previous work have been presented to the research community. This issue covers the publication of the main results discussed in this dissertation, focusing on both service and agent oriented architectures for ubiquitous and sensor computing, as well as dynamic service composition on the Internet of Things.

1.3 Research Framework

The work presented in this dissertation has been developed within the framework of several projects funded by the European Commission in the field of pervasive computing, embedded networking and the Internet of Things. In order to illustrate the research context of this thesis, next paragraphs summarize the main objectives of these European Research Projects.

- LifeWear “*Mobilized Lifestyle with Wearables*” (ITEA2 code: 09026, start date: 03/2010, end date: 03/2013) [169]: The LifeWear project focuses on integrating wearable and wireless sensor-based computing into the daily living environments based on the smart home notion, in order to simplify and enhance the use of common electronic devices. In the context of wireless sensor network computing, the main objective of this project is to research in the composition and management mechanism of heterogeneous services with IP-based connectivity and explicit Quality of Service (QoS) requirements.
- DiYSE “*Do-it-Yourself Smart Experiences*” (ITEA2 code: 08005, start date: 10/2008, end date: 02/2012) [90]: The DiYSE project is focused on exploit the Internet of Things notion enabling end-users without technical skills to create their own Smart Spaces with personalized services and applications. Regarding pervasive computing in the sensor network technological domain, the main objective of this project is to define semantic knowledge management procedures for virtual services instantiation, including their integration into the cloud using Web-based technologies.
- μ SWN “*Solving major problems in MicroSensorial Wireless Network*” (FP6 code: IST-034642, start date: 10/2006, end date: 11/2009) [278]: The μ SWN project focuses on researching and developing a generic and reusable hardware and software solutions based on Wireless Sensor Networks. This is an antecedent of the thesis and a core part of its motivations. The main objectives of this project include to research and develop an agent-based middleware for sensor networks, a configurable and energy-efficient inter-node communication protocol, and a mote platform with sensor plug-and-play support.

1.4 Dissertation Outline

The work presented in this thesis is organized in four differentiated parts. The Part I represents the dissertation introduction and objectives, and it consists of Chapter 1. The Part II is composed by the state-of-the-art survey and related background, and it consists of Chapters 2 and 3. The Part III contains the formalization, design, implementation and validation of the research contributions, and it consists of Chapters 4, 5, 6 and 7. Finally, there is the Part IV. It includes the conclusions and further work discussion, and it consists of Chapter 8. The main contents of the previous chapters are depicted in the following paragraphs.

Chapter 1 introduces the foundations of sensor network technology, including its design principles and emerging research. Additionally, the objectives and main contributions of this thesis are presented, remarking the research framework where this work has been carried out. Finally, the structure followed in the rest of the essay is outlined.

Chapter 2 analyzes the research areas of interest to this work within the framework of the Wireless Sensor Network paradigm. This includes, based on the protocol engineering foundations for ad hoc and sensor networks, the study of Service-Oriented Computing, Agent-Oriented Software Engineering, Semantic Knowledge Management and the embedded Internet of Things.

Chapter 3 surveys background and related work regarding middleware architectures and service composition, as a key issue for creating Smart Environments based on sensor network

technology. This chapter covers both theoretical foundations and legacy approaches, as well as middleware solutions and pervasive service composition technologies for ubiquitous sensor networks.

Chapter 4 contextualizes the problem statement and the requirements definition, formalizing the proposed service-oriented reference model. From the analysis of the design principles and challenges in pervasive smart environments, this chapter formulates the proposed service-oriented framework and service composition model in such ubiquitous infrastructures.

Chapter 5 presents nSOM (nano Service-Oriented Middleware), which is a service-oriented middleware proposal for service provisioning in pervasive smart environments based on heterogeneous Wireless Sensor Networks. This chapter reviews in detail the architectonic foundations and internetworking model approach of nSOM.

Chapter 6 discusses nSOL (nano Semantics-Oriented Language), which is a virtual sensor service composition proposal for sensor networks based on semantic knowledge management technology. This chapter analyzes the proposed underlying ontology for service composition, its interface language, notation syntax definition and service description format.

Chapter 7 illustrates the experimental analysis and the consequent discussion of the related results for this research work. These include the testbed and platform description, as well as the evaluation of the service-oriented middleware architecture and composition approach according to several specific performance metrics for embedded computing and ad hoc sensor networks.

Chapter 8 concludes this dissertation remarking the final conclusions and providing an overview of the future directions for our research work. Additionally, this chapter summarizes and presents the main scientific dissemination results which are related to the work presented and discussed in this doctoral thesis, which justifies the novelty and impact of our contribution.

PART II

STATE OF THE ART AND
BACKGROUND

State of the Art

ENGINEERING potential of Wireless Sensor Networks is almost endless, currently being a cutting-edge research area. Initially encouraged by strategic and military applications, the significant developments from both the academia and industry corporations around this paradigm that have taken place during the last decade have led to an exponential growth in terms of the capabilities and potentials of these ad hoc and embedded networks. In this regard, this technology is one of the fundamentals of pervasive computing, and presumably it will play a crucial role in the future Information and Networked Society.

This chapter performs a state-of-the-art evaluation regarding technological research directions in Wireless Sensor Networks, as well as some telecommunication and computer science paradigms which are part of the background for this dissertation. Section 2.1 presents an overview of architecture foundations concerning novel protocol stack engineering trends for sensor networks. Section 2.2 introduces the Service-Oriented Computing paradigm. Agent-Oriented Software Engineering is presented in Section 2.3. Semantic Knowledge Management is discussed in Section 2.4 and the Internet of Things vision is analyzed in Section 2.5. Finally, Section 2.6 summarizes main conclusions of this chapter.

2.1 Architecting Wireless Sensor Networks

The protocol stack defined in sensor nodes is energy consumption sensitive, integrating data with routing protocols, enabling efficient communication through the wireless radio interface, and promoting collaboration among wireless sensor nodes in the ad hoc network. According to the previous work of Akyildiz et al. [12], and as illustrated in Figure 2.1, horizontal layers that make up the protocol stack for sensor networks are the following: Physical Layer, Data Link Layer, Network Layer, Transport Layer and Application Layer.

Additionally, and as shown in the figure, a set of management orthogonal planes have been defined. These are responsible for allowing nodes to operate in an energy-efficient routing mode in mobile environments, as well as sharing resources between nodes. These transverse planes are the following: Power Management, Mobility Management and Task Management. On the one hand, the Power Management Plane defines mechanisms for managing the energy consumption in sensor nodes. On the other hand, the Mobility Management Plane is responsible for detecting and recording the movements of the nodes, in order to know which nodes are their neighbours. Finally, the Task Management Plane is responsible for balancing and planning tasks related to the data capture through the physical sensors.

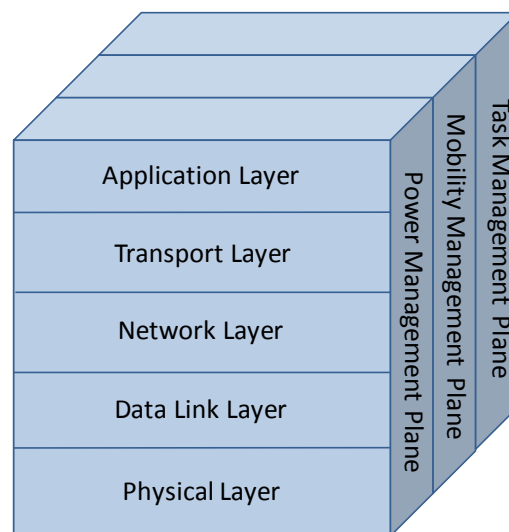


Figure 2.1: *Wireless Sensor Network protocol stack* (Akyildiz et al. [12]).

2.1.1 Physical Layer

The Physical Layer is responsible for the selection of frequency, carrier frequency generation, signal detection, modulation and data encryption at hardware level. In designing the Physical Layer of wireless sensor nodes, peculiarities related with the resource-constrained features of these devices should be taken into account. In this regard, sensitive designs for minimizing power consumption are required, in order to maximize the node lifetime.

Ultra Wideband (UWB) technology has been strengthened as a very interesting solution for wireless networks of embedded devices in recent years due to its ability to allow low-power consumption communications, high-transmission rates and radio coverage on the order of tens of meters. Around this approach, and with the aim of optimizing the management of the node resources, several related technologies have emerged. Among them, we highlight on Impulse Radio Time-Hopping UWB (TH-IR-UWB) [312], which is a solution based on the transmission of very short duration pulses (i.e., order of hundreds of picoseconds). According to this proposal, time is divided into frames, each of which is divided into chips of very short duration. Each source sends a pulse on a chip per frame while multiple accesses are provided by temporal sequences of pseudo-random jumps. The main advantage of TH-IR-UWB is its low energy

consumption, as well as the generation of high data transmission rates with a very simple circuitry design, thus allowing low-cost radio modules. Further advantages are the high-processing gain in the presence of interferences, as well as its flexibility since data rates can be adjusted depending on the power spectral density and multi-path performance [13].

Besides the previous, spread-spectrum techniques are an adequate option for sensor networks, because they meet the requirements of such low-capacity devices more efficiently than the narrow-band techniques [306]. Narrow-band communications allow optimizing the bandwidth efficiency while spread-spectrum technology and UWB are appropriate to balance between bandwidth and energy savings. Additionally, narrow-band systems are less robust to interference compared to spread-spectrum systems.

2.1.2 Data Link Layer

The Data Link Layer is responsible for data stream multiplexing, data frame transmission and reception, shared Medium Access Control (MAC) and error detection. The objective of a MAC protocol in a multi-hop sensor network is twofold. First, it seeks the creation of the network infrastructure, allowing the establishment of communication links between nodes. Second, the MAC layer management should provide fair and efficient resource sharing. In order to achieve the previous, new protocols and algorithms for Wireless Sensor Networks are required.

Several proposals regarding MAC protocols based on duty-cycle techniques for sensor networks can be found in the related literature. Duty-cycle MAC protocols [129] allow to minimize energy consumption by periodically switching between sleep and wake-up states. Main duty-cycle based MAC protocols for sensor networks can be classified into three categories: Asynchronous (e.g., X-MAC [52]), synchronous (e.g., D-MAC [178]) and hybrid (e.g., Z-MAC [241]). Asynchronous MAC protocols are based on preamble sampling. Synchronous MAC protocols adjust the listening active period (i.e., wake-up) using a visual synchronous method between groups where the communication takes place. Hybrid MAC protocols use both asynchronous and synchronous procedures depending on the transmission patterns.

Additionally, a widely used standard in most commercial sensor platforms is IEEE 802.15.4 [137]. This specification defines both the Physical and Data Link Layers for wireless communications in the radio bands of 868MHz (11 radio channels, 20kbps), 915MHz (11 radio channels, 40kbps) and 2.4GHz (16 channels radio, 16kbps) with a transmission range up to 100 meters. The IEEE 802.15.4 Data Link Layer uses the CSMA-CA (Carrier Sense Multiple Access-Collision Avoidance) algorithm for managing media access and collision detections, providing a guaranteed time slot mechanism as an option for communications with low-latency requirements.

2.1.3 Network Layer

Most of traditional routing protocols for ad hoc networks do not fulfil the energy efficiency, data-centric routing and data aggregation requirements of sensor networks [104]. In this way, several novel networking schemas optimized for ad hoc and sensor networks have arisen in the last years. Among them, we can emphasize on bio-inspired routing algorithms, opportunistic routing and IP-compliant routing protocols, such as 6LoWPAN.

Bio-inspired routing approaches [93] offer significant adaptation, reliability and robustness patterns in heterogeneous environments, providing a useful mechanism for designing networking schemas for dynamic and adaptive sensor networks [141]. In essence, these routing algorithms model the behaviour of insect colonies, which are inherently self-organized and self-adapted. Examples of sensor networks routing protocols based on biologically-inspired algorithms are Ant Colony Optimization [92], which is based on the behaviour of ants, and BEES [2] that is based on bee swarms.

Opportunistic routing [170] allows exploiting the broadcast nature of multi-hop packet sensor networks using sets of forwarding nodes between the neighbouring nodes of the sender. In this schema, each node selects from among its neighbouring nodes a set of forwarding candidate nodes, establishing a priority that will determine which node or nodes retransmit the packet to the next hop. The main advantage of the opportunistic routing approaches over the ones based on single-path routing is the increase of reliability and end-to-end network throughput, but introducing an additional overhead due to the synchronization procedures between nodes. Examples of novel contributions based on the opportunistic routing paradigm are CORE [303] and SOAR [247].

6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) [253] is an IETF standard that provides IPv6 connectivity to energy-limited devices based on IEEE 802.15.4 by means of IP packets compression. This technology offers several advantages [180]. First, popularization and applicability, since the IP technology is widely used being IPv6 the future connectivity technology. Second, a larger address space, being suitable for mass deployment of computing systems based on sensor networks. Third, auto-configuration address support, so nodes working with IPv6 can read their MAC addresses at start-up, allowing self-establishment of the IPv6 address according to a given policy. Finally, ease of access according to IP networks is achieved, enabling developers to take advantage of IP technology in application programming.

2.1.4 Transport Layer

This layer is especially important when the system is focused on accessing the Internet or other external data networks. The TCP and UDP protocols of traditional wired IP networks do not fit the unique aspects of wireless networks [122]. On the one hand, TCP is a connection-oriented transport layer protocol, which has been designed to assume that packets loss is due to congestion in the buffers of intermediate nodes. For this reason, sliding-window mechanisms and TCP congestion control reduce the transmission rate, whereas packet loss in wireless networks is usually caused by signal interference or fading. On the other hand, UDP is a connectionless transport protocol which does not implement flow and congestion control, thus without providing reliable transmission.

In this regard, contributions in the Transport Layer are mainly based on introducing changes and improvements to the TCP protocol, in order to adapt its behaviour to the specific requirements of wireless networks. Based on the classification provided by Rathnayake et al. [237], protocols designed for the Transport Layer of Wireless Sensor Networks can be classified into three main

groups: First, proposals only focused on improving reliability. Second, solutions only designed to enhance congestion control. Finally, approaches aimed to empower both reliability and congestion control.

PHTCCP [201] is an example of Transport Layer protocol for sensor networks focused only on improving the congestion control. This proposal is based on a priority scheduler that assigns different levels to the sensors. Congestion detection is based on packet service rate, implicit congestion notification and collision avoidance using rate adjustment. Le et al. present E RTP [160] which is a transport protocol for sensor networks optimized to handle data streams. This approach belongs to the category of Transport Layer protocols for sensor networks that only improves reliability. E RTP provides packet-level reliability in the upstream with hop-by-hop recovery mechanisms, using ACK and iACK (i.e., implicit acknowledgment) mechanism for loss detection and notification. TRCCIT [252] is a transport protocol focused on implementing both congestion control and improved reliability in WSNs. This contribution develops congestion detection when incoming packet rate is greater than the rate of outgoing packets, using proactive avoidance by means of opportunistic transmission through several network paths. Regarding the improvement on reliability, it performs probabilistic broadcasting, retransmission timer management, hop-by-hop packet recovery and a hybrid acknowledgment schema.

2.1.5 Application Layer

As discussed in Section 1.1, Wireless Sensor Networks have a huge number of potential application scenarios. In this regard, the underlying diversity of practical use cases imposes to the Application Layer the need for providing protocols dealing with sensor management, task assignment and data advertisement, as well as sensor query and data dissemination [111].

A novel approach for protocol design providing sensor management support in the Application Layer is based on SNMP (Simple Network Management Protocol), which is the most widely used management protocol for remote and network management in wired environments. In this research line, several contributions have been proposed such as [134, 142]. On the one hand, LiveNCM [142] enables management procedures in an energy-efficient way. This SNMP-based management solution implements a non-invasive, context-aware diagnosis for data management as well as an estimation model for value prediction. On the other hand, in [134] an approach that integrates the SNMP protocol with the TEDS (Transducer Electronic Data Sheet) [136] standard is presented, which is responsible for defining the appropriate properties of sensors and actuators to automate the information addition process with the aim to support plug-and-play features.

In the case of Application Layer protocols focused on task assignments, Abdelsalam et al. [3] propose a task management protocol for sensor networks. The authors present a lightweight protocol optimized for Wireless Sensor Networks which assigns tasks to the sensors based on their energy consumption, in order to distribute the energy consumption impact among all nodes in the network uniformly. Also, this proposal defines mechanisms for data aggregation to optimize the transmitted information, including exact evaluation aggregation functions,

minimum and maximum value, and OR logic function. On the other hand, in [148] an Application Layer protocol that implements a cluster-based, self-organizing schema for wireless sensor and ad hoc networks is analyzed. In this proposal, the node that receives the information of interest from the sink node is self-selected as clusterhead, managing the advertisement to its neighbouring cluster nodes. When the cluster has been consolidated, the clusterhead forwards the information of interest to another intermediate node, which will forward again the information of interest. The first node that receives this is self-selected as a new clusterhead. Iteratively, this process is repeated until a state of network convergence is finally reached.

Regarding the sensor query and data dissemination protocols in the Application Layer, Sun et al. define MQDD [269], which is a data dissemination protocol based on multicast queries suitable for environments with multiple sources, multiple sinks supporting network mobility. MQDD uses a multicast protocol for disseminating queries and a unicast protocol for transmitting data. Additionally, this proposal makes use of geographical multicast routing in order to minimize the energy consumption. Another proposed Application Layer protocol for data dissemination supporting multiple mobile sinks is illustrated in [95]. The authors propose a two-dimension hexagonal mesh structure in contrast of classical square mesh models, in order to exploit the geographic routing capability. This architecture defines virtual areas called highways, which are used for data caching. This type of honeycomb-shaped structure allows the use of geographic routing, minimizing the energy consumption and accelerating data access to sink nodes.

2.2 Service-Oriented Computing

The Service-Oriented Computing (SOC) paradigm [132] is an emerging wave for distributed computing that promises to change the way in which applications are architected, implemented, provided and consumed. In this regard, SOC offers support for the development of rapid, evolvable and interoperable application entities, which can be easily integrated into a loosely-coupled manner with the aim of building dynamic business service platforms [6]. As shown, the core of this vision is the notion of service. In SOC, a service is perceived as a building block that is by definition autonomous, self-described, reusable and highly-portable, being significantly different from its traditional approach as a software artefact [294]. In this regard, the promising advances offered by the Service-Oriented Computing are paving the way for the technologies involved in the Future Internet, from semantics to cloud computing, including pervasive computing and the Internet of Things, among others [16].

2.2.1 Service-Oriented Architecture as a Paradigm

A major cornerstone in realizing Service-Oriented Computing is the concept of Service-Oriented Architecture (SOA), which is the natural evolution of the object-oriented and component-based software engineering [280]. Although Web Services are currently the most well-known example of Service-Oriented Architecture, it is important to noting that SOA is not limited to Web Services. SOA is a design paradigm, rather than a concrete and specific software solution, and Web Services are a way to implement it [146].

Services defined according to the SOA approach follow a set of design principles, which allow exploiting the advantages of this service engineering approach. Among these principles [257] the followings can be emphasised: cohesion (i.e., degree in which elements of the system belongs together), coupling (i.e., strength of associations or dependencies between systems), reusability (i.e., used for more than one service consumer), composability (i.e., aggregated into an advanced, added-value functionality) and autonomy (i.e., self-control over its execution environment). This collection of architectural practices is completed by Rosen [245] who also highlights separation of concerns (i.e., isolate business logic avoiding overlapping), accommodation of change (i.e., flexible architecture) and abstraction (i.e., hiding heterogeneity for providing platform-independent solutions).

As discussed by Erl [94], the design principles and practices previously reviewed can be synthesized in three basic features for Service-Oriented Architectures, which reflect and summarize the nature of such service platforms. These features are the following: business-driven, vendor-neutral and composition-centric. First, a framework for implementing business processes based on services is provided in a rapid and efficient manner, with low development cost. Second, it defines an abstraction for decoupling the middleware of a single platform or specific hardware device, with the aim of facilitating their development regardless of the hardware manufacturer. Third, strengthening of this distributed computing paradigm founded on service dynamically is the support for combining a set of simple, independent services into more complex, added-value business services.

2.2.2 Service-Oriented Engineering

Service-Oriented Computing involves several notions, concepts and technologies from a wide range of disciplines. These include telecommunications, software engineering, computer architectures, distributed systems, network security and knowledge management [39]. Due to the underlying heterogeneity of these knowledge areas, advances in the research roadmap of Service-Oriented Computing were very fragmented during the last decade, but where a significant number of interdisciplinary contributions have been proposed.

Based on the previous work of Papazoglou et al. [224], the research agenda of Service-Oriented Computing can be illustrated as in Figure 2.2. This has been modelled as an extended SOA framework. This differentiates three logical planes, where service foundations or middleware are at the bottom, service composition is built over the service foundations, and service management is on top of such approach. The separation between these layers is motivated by the need to isolate service capabilities within this conceptual framework. On the one hand, the service foundations plane provides to the service composition plane a platform independent, internetworking backbone with a common set of general-purpose features for service development. On the other hand, the service composition plane is focused on developing the required business logic for coordinating service architecting in a flexible and dynamic way. Finally, the service management plane implements the precise monitoring and decision-making procedures for building a resilient network infrastructure with autonomic and self-management computing procedures.

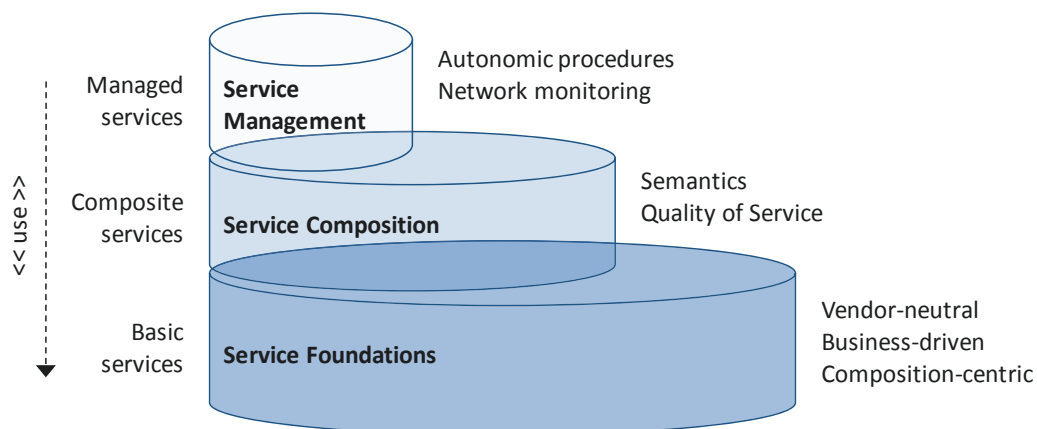


Figure 2.2: *Service-Oriented Computing research roadmap* (Papazoglou et al. [224]).

2.2.2.1 Service Foundations

This plane is the middleware backbone with service-orientation, which implements the SOA runtime environment. As a middleware, it abstracts the network heterogeneity regarding device platforms, while providing an integrated and distributed environment for application development with a set of core, general-purpose services [184]. Additionally, this plane follows a service-orientation approach. It fulfils the ideal features of SOA as previously discussed in Section 2.2.1, focusing on vendor-neutral, business-driven and composition-centric issues [94]. Identified research challenges for this layer to be covered in the next years include, among others, dynamically reconfigurable frameworks (in terms of service components and resources according to both user and application requirements), on-the-fly connectivity capabilities (without requiring the use of a separate piece of code for each service), end-to-end secure frameworks (both at network and application levels) and enhanced service discovery support (enabling fine-grained discovery features while minimizing the required user attention).

2.2.2.2 Service Composition

Service aggregation is proposed in this plane for building more complex application entities, while automating a particular task or business process [40]. This procedure can be applied to both simple and composite services, thus being a very flexible mechanism for generating additional functionality. In service engineering, novel interaction protocols enabling service composition can be described in terms of orchestration or choreography approaches [88]. On the one hand, service orchestration is associated with the interaction at message and business logic level under the control of a single party. On the other hand, service choreography describes message and rule interactions between multiple business processes and end-points. Future directions in service composition research include, among others, adaptive service composition (providing dynamic service aggregation in an autonomic manner), semantic service composition (based on knowledge management techniques), QoS-aware service composition (regarding Quality of Service policies and Service Level Agreements), and business-driven automated composition (enabling integration of distributed business processes and transactions).

2.2.2.3 Service Management

The mechanisms implemented in this plane allow ensuring the health of the system when composing services, by means of status monitoring and action-taking procedures [223]. The importance of service management in service-orientation lies on the composition-centric nature of this approach: a failure in one service can lead from the interruption of a specific service provisioning to the malfunction of several composed business processes, or even the whole service network. Several motivations [46] lead to define these management procedures in next-generation service-oriented networks using autonomic computing [47], with the aim of enhancing the system administration tasks without human intervention. These represent the future challenges for autonomic service management, among which the following ones can be highlighted: self-configuration (setting up themselves according to the available resources), self-healing (resilient procedures for diagnosis, self-repair and fault-correction), self-optimization (reconfiguration mechanisms for optimizing its performance) and self-protection (proactive detection, identification and protection against attacks and security threads).

2.2.3 Research Challenges in Wireless Sensor Networks

As discussed, Service-Oriented Computing is an emerging cross-disciplinary methodology with several implications in disciplines such as telecommunications engineering and computer science, including wireless ad hoc and sensor networks. In this regard, there are an important number of open issues within the SOC roadmap. The following paragraphs review these main future directions in the context of this dissertation.

Service foundations, also called service-oriented middleware, is the area where more scientific contributions have been provided in the field of sensor network computing during recent years. In this regard, recent studies [199, 200] analyze the specific requirements of service-oriented middleware approaches in Wireless Sensor Networks. Based on the conclusions drawn in these background works, several challenges in pervasive sensor networks should be tackled using middleware technologies for enabling service-oriented solutions in a global manner. These include the following open issues: First, platform and OS independence, providing software abstractions for specific hardware solutions that enhance interoperability for the middleware infrastructure and the application business processes. Second, provisioning support for general-purpose middleware backbone functionalities which includes service discovery and configuration capabilities, real-time requirements, secure communications and interoperability with external networks and service clouds. Third, support for heterogeneous multi-service composition, allowing service internetworking and provisioning to the end-users in a seamless manner. Fourth, enhance software application development, making programming tasks ideally accessible to third parties with little or no technical skills in reduced functionality devices. Fifth, internetworking support with external service clouds and platforms. These challenges for service foundations in sensor networks are used as an input for the further state-of-the-art evaluation contribution regarding service-oriented middleware trends in Chapter 3. This will be used as a basis for the formalization and practical application contributions in service-oriented middleware, as discussed in Chapters 4 and 5, respectively.

Service composition is a natural property of pervasive computing systems [244, 270]. Emerging research on composition procedures and methodologies remarks several future directions in ubiquitous sensor networks that should be addressed for developing sensor-based service integration into pervasive smart spaces. These include the following open issues: First, definition of lightweight service composition languages for specifying service interfaces and invocation mechanisms in integration procedures [31, 32]. Second, adaptive service composition is required, where services can be integrated dynamically according to their availability in the network [31, 32]. Third, Semantic Knowledge Management based on ontology enhances expressiveness and service selection, thus improving service discovery and integration processes [277]. Fourth, virtual sensor services [43, 147, 302] allow composition schemas for supporting intelligent decisions that would be impossible to create with isolated physical sensors. Fifth, service-based integration with the Internet of Things (IoT) [25] paradigm is required, developing advanced services to external clouds and networks for ubiquitous smart spaces [79]. The previous is used as an input for the further state-of-the-art evaluation contribution regarding pervasive service composition in Chapter 3. This will be used as a basis for the formalization and practical application contributions in service composition, as discussed in Chapters 4 and 5, respectively.

Service management in WSNs is the most novel research area within the Service-Oriented Computing framework. The importance of this plane is based on the high-scalability degree of these ad hoc networks (i.e., magnitude orders of 10^2 , 10^3 or even higher) and their potential unattended operation in inaccessible areas [12]. In this regard, autonomic service management procedures for enabling network survivability without human intervention are needed. This fact is even more remarkable in future, next-generation digital ecosystems. In this large scale smart spaces the coexistence of services with heterogeneous nature (i.e., modelled with different QoS parameters and requirements) will impose new challenges for network and service engineers. Among these, we can appoint on service failure detection (reactive and proactive service path protection and recovery mechanism), inter-domain interoperability (handling interactions between networks with different administrative policies and authorities) or security models for resilient control planes (in order to guaranty a proper function of the data plane). In this regard, a brief outlook illustrating a cutting-edge research proposal regarding pervasive service resilience for Wireless Sensor Networks is presented in Chapter 8, as part of the future directions of the work discussed in this essay.

2.3 Agent-Oriented Software Engineering

To date, computer science has been strongly influenced among others by five trends: ubiquity, interconnection, intelligence, delegation and human-orientation. The convergence of these has led to the emergence of Multi-Agent Systems [300], which are defined as a federation of two or more agents where each one defines its own objectives based on its autonomous nature [191]. In this regard, a wide range of application areas can benefit from Agent-Oriented Software Engineering (AOSE) [202]. These include e-commerce [203], automation [194], healthcare [305], human aid support [273], robotics [5], e-learning [105], entertainment [161], traffic control management [23] and smart grid [22].

2.3.1 Agent Theory

In the early 90's, and motivated mainly by the initial works carried out during the 70's in the field of distributed artificial intelligence and specially on the seminal concurrent actor model [127], there was an explosion related to the agent-related technology research [212]. However, there is no clear consensus regarding what a software agent is [296]. In this sense, a significant number of definitions for the agent notion, including [50, 123, 183, 248, 260, 301], can be found in [102] where this concept is discussed from a computer science perspective. As identified by the authors, most of the previous works agree on identifying the same basic properties related to the agent concept, which are autonomy, environmental awareness and objectives. In this regard, a software agent can be perceived as an autonomous entity with a goal-driven behaviour according to which it interacts with the environment where is placed.

Nevertheless, and as remarked by Jennings and Wooldridge [145], traditional software agents do not show an intelligent behaviour because they lack on flexible autonomy in order to meet their design objectives. The notion of flexible autonomy includes responsiveness (i.e., agents are environmental-aware for responding to changes that occur on it), proactive (i.e., agents are opportunistic, taking the initiative where appropriate) and social (i.e., agents are able to interact with others in order to complete their tasks). As above, intelligent agents enable automatic interactions, reacting autonomously according to the changes produced in the environment, but also acting proactively, thus originating actions that affect to their surrounding environment.

On the one hand, previous definition adheres to the weak notion of agent [301]. On the other hand, the strong notion of agent that is provided by the authors offers further enhancements by means of human-like attributes such as the Believe, Desire and Intention (BDI) architecture [234, 235], which is a very elegant model for formalizing the internal representation of an agent. The BDI model combines three distinct components [281]: a philosophical foundation, a software architecture, and a logical formalization. Believes represent the agent knowledge and expectations regarding its environment, regarding other agents in the society and their role or regarding himself. Desires correspond to the preferences over future states in the framework of a specific believe. Intentions are a subset of the desires in which the agent is concentrated at a given time interval. One of the main advantages of the BDI paradigm applied to the internal representation of agents is that it is well suited for complex and dynamic environments. This makes easier the use of declarative representations for modelling the concepts of believes, desires and intentions [35, 227].

Extending the previous discussion, and with the aim of defining an agent typology, Nwana [212] identified three ideal attributes that agents should exhibit. These include, in addition to the autonomy capability identified in prior studies, the learning (i.e., act and/or interact with the external environment) and cooperation (i.e., ability to interact with other agents and humans) properties. The combination of these attributes leads to define four types of primary agents as illustrated in Figure 2.3, which are smart agents, collaborative agents, collaborative learning agents and interface agents. This classification has been extended taking into account additional issues such as the static/mobile and reactive/deliberative behaviours of the software agents, their role (preferably, if the roles are major ones) and their hybrid/native nature (i.e., if the

agent combines or not two or more agent philosophies). The resulting typology of software agents identifies up to six practical approaches of software agents: collaborative agents, interface agents, mobile agents, information agents, reactive agents and hybrid agents. Several works in agent technology research have been made based on the assumption of this typology, including [7, 45, 106, 204, 205, 276].

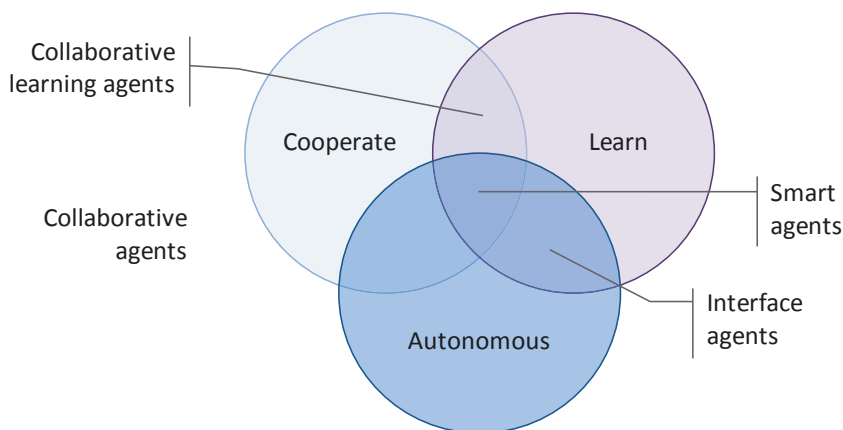


Figure 2.3: *Primary agent typology* (Nwana [212]).

The discussion of the previous categories is as follows. Firstly, collaborative agents highlight on autonomy and cooperation as primary attributes, remarking also in responsiveness and proactiveness [211]. These agents are able to operate autonomously, rationally and sociably in open and time-constrained environments. Secondly, interface agents emphasise on autonomy and learning while playing a personal assistant role, collaborating with the user in the environment [159]. In order to provide human-computer interface support, agents develop an evolutionary learning of the actions taken by the user for providing future suggestions. Thirdly, and in case of mobile agents, these are able to roam data networks, enabling both remote execution (i.e., transferring code and data to the target node before executing a concrete task) and agent migration (i.e., agent transfer occurs during task execution, migrating also the state related to the agent) [265]. In such a way, mobile agents provide a flexible approach with low dependability on the environment. Fourthly, information agents access to one or multiple distributed, heterogeneous information sources acquiring, managing and correlating relevant information on behalf of its owner or other agents in a proactive manner [251]. These agents are mainly motivated by the existing information overload, providing a way to manage such information explosion. Fifthly, reactive agents operate in a stimulus-response manner according to their internal state, without having an environmental representation model [262]. These agents are characterized by emergent functionality (i.e., design relatively simple and basic interaction support), task decomposition (i.e., modular architecture of the agent) and raw data operation (i.e., closer to raw data from sensors instead of high-level abstractions). Finally, hybrid agents are those that combine two or more basic approaches into a single instance [10], maximizing the strengths and minimizing the weaknesses of each agent philosophies for its concrete purpose.

2.3.2 Multi-Agent Systems and Standards

In Agent-Oriented Software Engineering, a Multi-Agent System (MAS) is viewed as a loosely-coupled network of software agents interacting in a shared environment in order to achieve their own objectives [144, 195]. Agents in a Multi-Agent System develop asynchronous computation with decentralized data [272], representing on behalf of users with very different motivations. It is important to remark that there is not an overall system objective, but the own goals of each separate software agent [300]. From a general point of view, a MAS allows decentralization of knowledge and computing, making easier the modelling of complex components in a dynamic way while enhancing and maintaining coordination with others [258]. In a more precise way, benefits of Multi-Agent Systems can be classified in two main categories: MAS for developing robust, flexible and extensible systems, and MAS as a modelling approach [191]. On the one hand, systems built around multi-agent technology can provide fault tolerance using agent redundancy, as well as flexibility for selecting the most appropriate action while making easier the addition of new functionalities. On the other hand, Multi-Agent Systems can be used as a modelling approach for complex systems generated by the interaction of simple entities in order to define, simulate and analyze how complex behaviours may emerge.

Multi-Agent Systems provide one of the best approaches for architecting distributed systems because information processing is inherently ubiquitous [296]. In practice, agent distribution is performed using platforms. An agent platform, also known as agent execution platform, defines the required network infrastructure for providing the environment where software agents can exist and operate to achieve their objectives, supporting and managing the life cycle of agents that includes their creation, registration, location, removing and communication procedures [119, 193, 226]. Several implementations of agent platforms raised in last years. Among those we can highlight some well-known examples, such as FIPA-OS [230], Grasshopper [34], Aglets [158], JACK Intelligent Agents [299], JADE [36] and SPRINGS [140], among others.

As in most of telecommunication and computer science related fields, the large number of existing proprietary solutions remarks the importance, even the need, for adopting standards that ensure compatibility between different agent platforms. During the last decade several significant initiatives were performed for enhancing interoperable agent platforms, where Knowledge Sharing Effort (KSE) [206, 225], Mobile Agent System Interoperability Facilities (MASIF) [198, 215] and the Foundation for Intelligent Physical Agents (FIPA) [135, 231] led these efforts. Pioneer proposals in the 90's for defining an Agent Communication Language (ACL) were based on the work of DARPA KSE, which was highly active during roughly five years [115]. During this period, the Knowledge Sharing Effort was involved in the research and development of procedures, methodologies and tools for enabling knowledge sharing between knowledge-based systems, including intelligent agents. MASIF is a specification of the OMG (Object Management Group) proposed in 1998 for providing a set of interoperable interfaces for mobile agent systems. These works were mainly focused on migrating mobile agents between Multi-Agent Systems based on the same agent profile using Common Object Request Broker Architecture Interface Description Language (CORBA IDL). To this end, MASIF normalizes four aspects of agents, which are the following: agent management, agent transfer,

agent system names, and agent system type and location syntax. Nowadays, MASIF has not activity being FIPA the leading, non-profit organization in agent standardization [54, 74], which has absorbed a significant part of the concepts and trends of the previous initiatives [74].

The specifications generated by the Foundation for Intelligent Physical Agents have become the de facto standards in software agent technology used by the academia and corporate professional [82], being accepted in 2005 as a standards committee of the IEEE Computer Society. In this regard, FIPA standardizes the inter-agent communication and core middleware service support in the framework of an abstract architecture definition for agent platforms [36]. It includes approved applications, agent communication, agent management and agent message transport. The high-level overview of the specifications developed by the Foundation for Intelligent Physical Agents is illustrated in Figure 2.4.

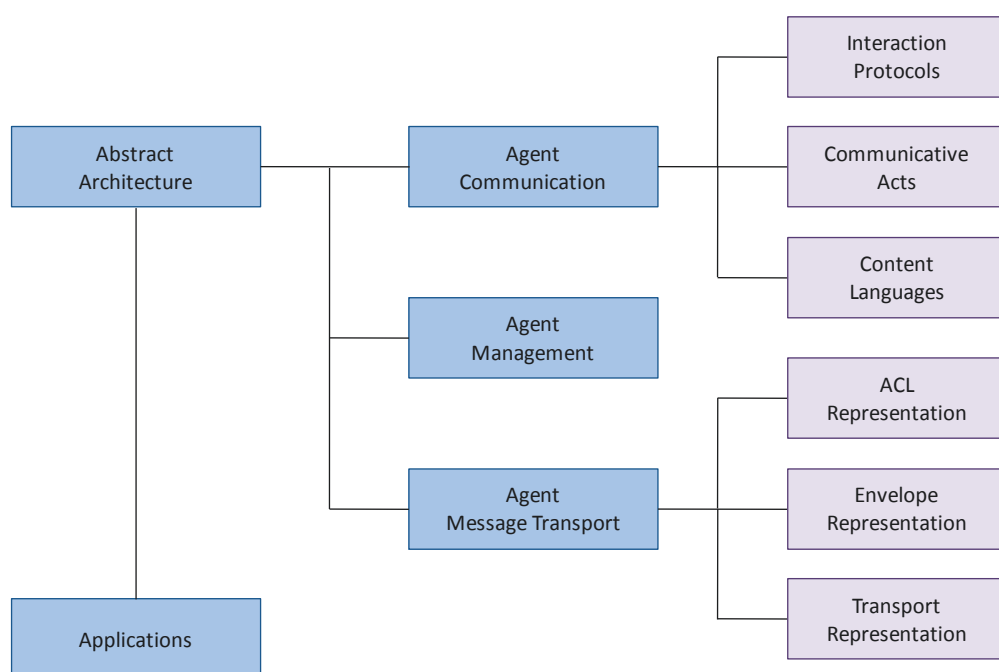


Figure 2.4: *Framework of the FIPA specifications* (IEEE FIPA [135]).

Among the most significant achievements of FIPA, and in the context of the previous standards, we can emphasise on the following. First, an abstract architecture modelling functional blocks for enabling the development of agent platforms seamlessly integrated with the computing and networking environment has been normalized, while improving agent reusability and interoperability. This standard includes a service and discovery model for agents, specifying message transport, ACL representation formats, content languages, and directory services for both agents and services. Second, several agent communication standards have been provided. These include the definition of an Agent Communication Language (i.e., FIPA-ACL) and a set of content languages (e.g., FIPA-RDF). On the one hand, ACL consist of a set of communicative acts messages and a set of interaction protocols. On the other hand, the content languages define the encoding syntax for the content of the messages according to a normalized ontology. Third, an agent management specification has been developed. This

standard defines the reference framework where the FIPA-compliant agents exist and operate including its instantiation, location, communication, migration, and termination. The agent management specification defines an agent runtime environment, an agent platform for deploying agents, an agent management system acting as a white-pages service and a message transport service for communication between registered agents. Fourth, a message exchange approach between interoperation agents is defined in the agent message transport, which provides a reference model for message transport service as well as the expression of message transport information. In this regard, it covers the standardization of the representation proposal used in the transport protocol, the envelope and the ACL. Finally, several application specifications where a set of possible use cases in which FIPA agents can be deployed are also covered representing ontology and service description definitions. Examples of these include specifications for nomadic application support, personal travel assistance, audio-visual entertainment and broadcasting, as well as personal assistant, among others.

2.3.3 Research Challenges in Wireless Sensor Networks

The agent paradigm has a natural application in pervasive computing, particularly in the case of sensor networks, at least for its fundamental and core notions. In fact, Agent-Oriented Software Engineering has been identified as a very interesting and promising approach for building autonomous, flexible and robust sensor networks [286].

In the context of the contributions of this dissertation, a promising research is on the use of the agent-orientation paradigm for developing intelligent pervasive in-network services based on the separation of concerns (i.e., aspect-oriented computing) paradigm. This approach focuses on enhancing the service composition procedure in resource-constrained and embedded devices according to the virtual sensor service paradigm, which was presented in Section 2.2.3. In this regard, one primary aspect is the definition of our service processing model. In contrast to legacy strategies in WSNs based on surrogate architectures (i.e., dynamically augmentation of the wireless network capabilities by exploiting wired infrastructure by means of resource-rich devices acting as servers) [250], the application of in-network processing sensor-based services is proposed. The service processing model based on in-network services (i.e., services implemented on-sensor using computation in the wireless network) [11] allows enhancing the network scalability, unlike other back-to-base based communication topologies because decisions are taken based on locally sensed data [291].

Based on the theoretical survey performed in Section 2.3.1, and as discussed in [103], the following agent orientation features can be exploited for enhancing service provisioning in Wireless Sensor Networks. First, agents are autonomous entities executing self-controlled procedures for problem detection and action-taking without external aid, thus improving the agent-based service provisioning. Second, agents have objective orientation. In this way, they are able to handle sensor information in order to perform the necessary procedures to reach their specific objectives. Third, they show reactive and proactive behaviour. Agents react autonomously according to the data read from sensors, but also acting proactively, influencing in the node operation. Fourth, agents develop communication and social support, interacting with

other local and remote sensor-based agents and building a distributed intelligent environment for addressing the resolution of complex problems.

Nevertheless, research approaches in the Agent-Oriented Software Engineering domain usually does not address the constrained computational and communication resources of sensor networks [242]. In this regard, embedded features of WSNs impose the need for developing lightweight agent-oriented solutions while minimizing the footprint size, consumption of physical resources and related overhead in the embedded nodes. In order to fit with the resource-constrained nature of wireless ad hoc and sensor networks, the agent development in our work has been performed from the perspective of separation of concerns (SoC) approach, also known as Aspect-Oriented Programming (AOP). The essence of this approach is that crosscutting concerns are isolated into different abstractions or aspects, which are responsible for a specific crosscutting concern [21]. The separation of concerns approach applied to the service engineering process enables to design the code so that application unit blocks (i.e., classes and modules) have clear, well-defined and isolated business logics [133]. This leads to improve the system modularity, readability and simplicity while minimizing the functional overlap with the rest of software pieces, thus decreasing the codification overhead [17, 182].

2.4 Semantic Knowledge Management

Knowledge Management (KM) is a research area that comprises a set of strategies, methodologies and practices for improving the processes related to the acquisition, extraction, organization, transfer, retrieval, maintenance and generation of knowledge by means of using a combination of information technologies and business processes [89, 165, 232]. In our economy and society, handling and enabling knowledge is a critical aspect for improving organizational performance through the better use of intellectual assets [287]. In this regard, Knowledge Management systems offer a set of specific techniques and tools for knowledge elicitation, modelling and integration as well as a collection of formalisms for representing and reasoning knowledge in an automated way [14, 232]. This fact has led to the application of semantic technologies to improve legacy knowledge management approaches, as well as to meet emerging requirements of knowledge-based services in a trend known as semantic knowledge management [83, 84].

2.4.1 Knowledge Foundations

The history of knowledge is the history of humanity, being the accumulation of knowledge the basis for the human evolution. Knowledge is a broad and abstract idea that has led significant epistemological debates since the ancient Greece [15]. From the significant number of contributions in such discussion during last decade of the 20th century, including [8, 81, 113, 155, 210, 298], the notion of knowledge can be tackled from three different perspectives [20, 73]. These are the following. First, knowledge referred as a knowing state. Davenport and Long [81] perceive it as information combined with personal experience, context, interpretation and reflection, while Kock and Queen [155] characterize it as a new or modified insight or predictive understanding. Second, knowledge stated as an accumulative process. According to this approach, for Grant [113] the knowledge is that which is known. In the case of Ackoff [8],

the knowledge is described as authenticated information. Third, knowledge identified as a capacity for action. In this regard, Nonaka and Takeuchi [210] identify knowledge as a factor of production, while Wiig [298] states that it consists of truths, believes, perspectives, concepts, methodologies and know-how.

However, and unlike the previous discussions regarding what knowledge is, there is a major consensus regarding its main forms. Based on the previous work of Polanyi [228], Nonaka [209] defines two main dimensions for knowledge: explicit knowledge and tacit knowledge. On the one hand, explicit knowledge is defined as knowledge that can be easily modelled, interpreted and thus codified. On the other hand, tacit knowledge is defined as knowledge that cannot be codified, articulated or verbalized. Thus, it represents an important challenge because knowledge management can handle and represent only codifiable knowledge [128]. This notion of knowledge types was extended by Nickols [73] from a cognitive-psychological perspective, who introduced two additional types: declarative knowledge and procedural knowledge. On the one hand, declarative knowledge is explicit knowledge that consists of descriptions, methods or procedures. On the other hand, procedural knowledge is stated as knowledge that manifests itself in the doing of something. These perceptions of knowledge are illustrated in Figure 2.5.

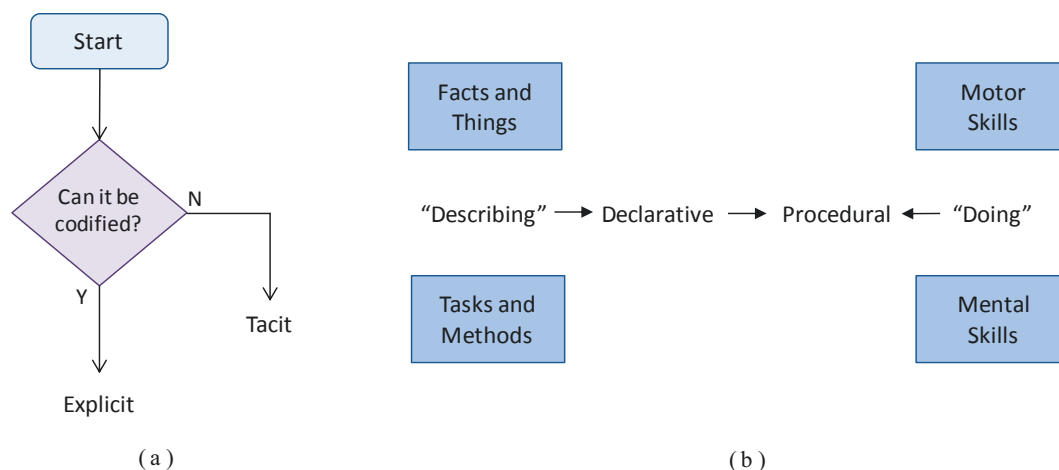


Figure 2.5: Knowledge dimensions: (a) explicit knowledge and tacit knowledge (Nonaka [209]); (b) declarative knowledge and procedural knowledge (Nickols [73]).

Although seminal works on knowledge management are at the late 70's by the hand of Horton [130], it was not until the decade of the 90's when knowledge engineering emerges as a mature field of study, closely related to computer science [188, 304]. In this regard, and as discussed by Nissen et al. [208] and Swan et al. [271], the roots of Knowledge Management in telecommunications engineering and information technologies can be found in several disciplines. These include artificial intelligence, information management, expert and decision support systems, data mining and warehousing, and business process engineering. However, it is important to appreciate that Knowledge Management has a multidisciplinary nature [78]. This includes organizational and cognitive sciences, linguistics and computational linguistics, technical writing and journalism, anthropology and sociology, education and training, and collaborative technologies, among others.

This multidisciplinary dimension of knowledge management science leads to several different classifications regarding its own nature and potential application domains. As surveyed by An and Wang [19], up to five possible classifications for knowledge management technologies and applications have been identified in the related literature. First, knowledge flow processes oriented classification. In this case, approaches are classified according to the process where they are applied. This includes from knowledge generation, knowledge storing, codification and representation, knowledge transformation and use, and knowledge transfer, sharing, retrieval, access and searching. Second, knowledge conversion and transfer processes oriented classification. According to it, solutions are categorized based on the conversion and transfer processes of explicit and tacit knowledge. This addresses the modelling of socialization technologies (i.e., tacit to tacit), externalization technologies (i.e., tacit to explicit), combination technologies (i.e., explicit to explicit) and internalization technologies (i.e., explicit to tacit). Third, knowledge processing modes oriented classification. According to that, knowledge management systems can be segmented as integrative applications (i.e., enabling the interaction of producers and consumers only with an intermediate knowledge repository) and interactive applications (i.e., supporting interaction between peers holding tacit knowledge). Fourth, knowledge characteristic oriented classification. Depending on the knowledge characteristics, knowledge management systems are divided according two axes: knowledge locus (i.e., artefact or individual) and the level of the *a*-priority structure (structured or unstructured), determining four dimensions in which knowledge management applications can be catalogued. Fifth, business strategies oriented classification. It states that knowledge management solutions can be classified in three main types, which are component technologies (i.e., basic blocks for building knowledge management architectures), generic knowledge management applications (i.e., integrate several component technologies with well-defined functionality) and business-driven knowledge management applications.

2.4.2 Towards Semantic Knowledge Engineering

One critical issue remarks the importance of architecting knowledge management frameworks according to semantic technologies in corporate environments: the high degree of unstructured information resources in the enterprise, which is around the 80% [229]. This fact, as well as the existing information overload, leads to the application of semantic technologies for improved management of the information assets, which can be highly cost-effective for medium and large-size corporations. Semantic knowledge engineering theory and practice is based on the integration of several research areas including ontology management, knowledge discovery and human language technologies. Next sections review each one of these domains in the context of knowledge management.

2.4.2.1 Ontology Management

Being usually understood as the specification of a conceptualization (i.e., formal definition of a set of terms and relations between them defined in an explicit knowledge domain) [126], the notion of ontology plays a crucial role in semantics engineering. As known, semantic graphs built using ontologies characterize as nodes the concepts or individual objects while edges or

arcs represent properties or rules correlating such concepts. This semantic network is augmented by properties and attributes, constraints and functions, which model the behaviour of the concepts. In this regard, ontology management covers the procedures and tools for managing knowledge bases defined around ontologies and their meta-data, producing and maintaining consistency between formal and real-world semantics [100].

Research and development on ontology management have several open issues and future directions [83]. These include ontology mapping, ontology learning, processing of inconsistent ontologies and querying of ontologies. Ontology mapping is mainly focused on providing interoperability across data sources based on heterogeneous ontologies. This is done based on a mapping process by identifying the correspondences between these inter-domain ontologies, as well as modelling the correspondences using an appropriate mapping formalism. Ontology learning comprises the methods for autonomic ontology generation from unstructured or semi-structure data sources. These methodologies are based on machine learning or natural processing language techniques, providing a framework for dynamic modelling of learning generation from the underlying data sources and the represented knowledge. Processing of inconsistent ontologies deals with inconsistencies detected in knowledge bases. To address this issue, two methodologies are proposed: debugging inconsistent ontologies (i.e., detect and repair inconsistencies) and reasoning with inconsistent ontologies (i.e., apply heuristic reasoning to obtain meaningful information in the presence of inconsistencies). Ontology-based query answering is based on an underlying ontology to drive and give meaning to the queries provided by the users against a knowledge base. A widely-known example of this is SPARQL Protocol and RDF Query Language (SPARQL) [289], which is a standardized ontology-based query language for the Resource Description Framework (RDF) [288].

2.4.2.2 Knowledge Discovery

Knowledge discovery process is focused on seeking for new knowledge in a specific area or application domain. This research discipline focuses on finding and extracting interesting pieces of information from raw data while concerning the whole knowledge extraction process [69]. This covers from data extraction to data access, including datasets storage and analysis, results interpretation and visualization, as well as machine-to-human interaction support. Main motivations to structure knowledge discovery as a well-defined, formal development process are to ensure that the final product will be useful to the end user, the understanding of the process itself and the provisioning of management procedures.

While semantic technologies allow integrating existing notions with the deal of enhancing the descriptive nature of the entities under study, the knowledge discovery approach has shown a great potential to address and support specific problems of semantic-oriented communication systems [114]. As discussed in [83], it covers from ontology construction to human language technologies (see Section 2.4.2.3 for further information) including domain knowledge, handling dynamic data and capturing semantics of multi-modal and multilingual data. Ontology construction can be partially implemented in a semi-automatic way based on knowledge discovery procedures. In this regard, semi-automatic ontology construction consists on domain and data understanding, task description, ontology learning, ontology validation and refinement

with human in the loop. Domain knowledge can be captured and incorporated in information systems based on semantic support. In this regard, knowledge discovery approaches can be used to deal with user preferences via contextual information and behavioural pattern analysis. Data of interest can be presented to the end-user in the context of a specific domain applying knowledge discovery solutions. Handling dynamic data in semantic applications is an open issue that can be addressed using knowledge discovery approaches. Information systems usually have component-application dependence, where data changes over the time can be connected to different applications even this dynamic component can be also associated to data models. The capture of semantics of multi-modal and multilingual data can be enhanced using knowledge discovery. The primary objective of this task is related to the data pre-processing and its visualization for further analysis.

2.4.2.3 Human Language Technologies

The importance of the textual language is very significant in current information systems, being estimated that more than the 95% of the human-to-computer information inputs will be based on human language technologies during this decade [168]. In this regard, there is a clearly defined parallelism between the semantic evolution of knowledge management on the one hand and the increasing of resources based on human language on the other [75]. The adoption of technologies related with human language improves both the information access and the human interaction, while amplifying our awareness for knowledge activities that fall in our interests.

Knowledge management can be applied for enhancing a wide range of human language technologies. In this regard, and as discussed in [264], key aspects of this approach include knowledge mapping, discovering expertise and new knowledge discovery. Additionally, human language technologies offer significant potentials that can be used by knowledge management. These cover from input analysis to information retrieval, including extraction, question answering, translation, dialogue management, agent and user modelling, summarization, presentation and awareness and collaboration [190]. Input analysis provides a wide support for knowledge access to users through multi-modal, multiplatform natural language interfaces. Information retrieval takes advance of input processing and content-based access, enhancing the richness of accessible material as well as improving retrieval precision and recall. Extraction offers object and event identification in multimedia sources, leading to direct access to knowledge items as well as to reuse support for media elements. Question answering can be enhanced in terms of human-to-machine interaction, allowing source selection and segmentation, extraction and semantic inter-domain integration. Human-assisted machine translation presents new developments around fast construction of multilingual corpora, as well as efficient retrieval, summarization and translation. Dialogue management in knowledge management systems incorporates adaptation procedures for natural, mixed initiative human-computer interaction flows. Agent and user modelling enable knowledge management features, including expert databases for improving corporate awareness and efficiency, or track user context information. Summarization allows to choice and resume contents for its presentation where multimedia, multilingual and cross-document summarization are open issues. Effective presentation comprises knowledge discovery and visualization requiring mixed media and

mixed mode displays adapted to the users and its context, leading to new representation models for massive and multimedia repositories. Finally, awareness and collaboration are performed both asynchronously and synchronously. These enhance aspect such as awareness and expertise, allowing inter-domain knowledge discovery and connection.

2.4.3 Research Challenges in Wireless Sensor Networks

Traditionally, most sensor networks are data-centric [221]. This fact, along with other issues such as the growing availability of sensor devices and the large amount of data generated in ad hoc networks, highlights the need for implementing efficient knowledge management in such environments [308]. In this regard, efforts on knowledge management for Wireless Sensor Networks have been largely on sensor fusion and data aggregation with the aim of minimizing network load and energy consumption [179].

In the context of the contributions presented in this dissertation, a promising research area is on knowledge management models based on domain ontologies applied to pervasive service composition using annotations. In our study case, and according to the application of ontology-based composition procedures, annotations with well-defined semantics are bound to the service descriptions that are provided by the embedded devices in the ubiquitous network. This approach is focused on building an orchestration schema for agent-based pervasive services in resource-constrained and reduced functionality devices according to the virtual sensor service paradigm, as presented in Section 2.2.3. The application of semantic technology in the development of distributed service platforms and service composition procedures is based on the fact that this approach provides more intelligent information access and management by defining semantic meta-information on services [83]. In this regard, semantic annotations identify concepts and relationships between them in a formal and precise manner, being focused primarily on automatic processing by machines. In this sense, this process relies on a knowledge management approach based on semantic information from domain ontologies.

The main motivation of this proposal emerges with the aim of developing a very flexible framework for generating advance service functionality in pervasive environments. These are highly distributed, being characterized by the heterogeneous and dynamic nature of their ubiquitous computing devices. In this regard, our approach has been designed to fulfil some of the most remarkable interoperability challenges in service composition for pervasive computing, as stated in [220]. These future directions include the following aspects. First, interoperability support based on dynamic functionality for locating and integrating business processes from heterogeneous sources. Second, interface definition for providing service functionality without specific knowledge about how it will be used. Third, interoperability separation defined between application entities and internetworking mechanisms.

To address the previous, a semantically driven approach for virtual sensor services in pervasive computing is defined, which is called nSOL (nano Semantics-Oriented Language). It provides an interoperable framework for ubiquitous service composition using domain ontologies, which integrates the functionality of agent-based pervasive in-network services into complex virtual sensor services. To this end, semantic annotations are embedded into the interface descriptors,

being used as an input by the service composition process. In addition, this proposal has been built over nSOM (nano service-Oriented Middleware), which represents a service-oriented architecture for embedded computing devices, isolating the business logic of the virtual sensor services to the underlying intermediation and communication support. Chapter 4 and beyond are devoted to present these approaches, which are the formalization and practical application contributions of this dissertation.

2.5 Internet of Things

Although the original vision of the Internet of Things (IoT) notion is dated at the late 90's in the seminal work of Kevin Ashton, the technological explosion of this paradigm is being a cutting-edge research topic nowadays [243]. The elementary foundation behind the Internet of Things idea makes computing truly pervasive, implementing a global infrastructure around the physical world saturated with a wide variety of internetworked physical objects, which have the ability to communicate, interact and cooperate between them and with other external networks with the aim to reach common goals [138, 156, 157].

The horizon depicted by this emerging and promising approach assumes several technical steps in technology development and integration creating new directions and challenges. This comprises, among other issues, standardization efforts for the IoT paradigm, networking schemas for accessing and operating with these distributed clouds of networked devices, congestion control and data buffering designs for transport protocols, patterns definition for traffic modelling, characterization and QoS support, as well as network security and privacy, including authentication, data integrity and confidentiality.

2.5.1 Enabling Technologies for Building the Internet of Things

In the Future Internet, things are connected and seamlessly integrated between them and with the physical surrounding environment according to the ubiquitous computing notion, providing anytime/anywhere/anymedia access and communication support. The way in which these different solutions and technologies coexist leads to the IoT paradigm, which is the result of the convergence of three main different visions [25]: Internet-oriented, Semantic-oriented, and Things-oriented perspectives. The integration of such approaches in the framework of the Internet of Things is illustrated in Figure 2.6.

According to the Internet-oriented vision of the Internet of Things, a global infrastructure that interconnects virtual and physical objects extending the current Internet and external networks is perceived, while enabling the integration and provisioning of federated services and applications. According to this approach, IP-based technologies for embedded internetworking (e.g., 6LoWPAN [253], InternetØ [107], uIP [283]) play a key role in integrating pervasive services into external service clouds. Besides the previous, the Web of Things [117] appears as an approach closely correlated with the Internet-oriented perspective of the IoT. In this regard, traditional Web Service standards are applied to build, expose and consume services that are implemented with everyday life objects and embedded devices.

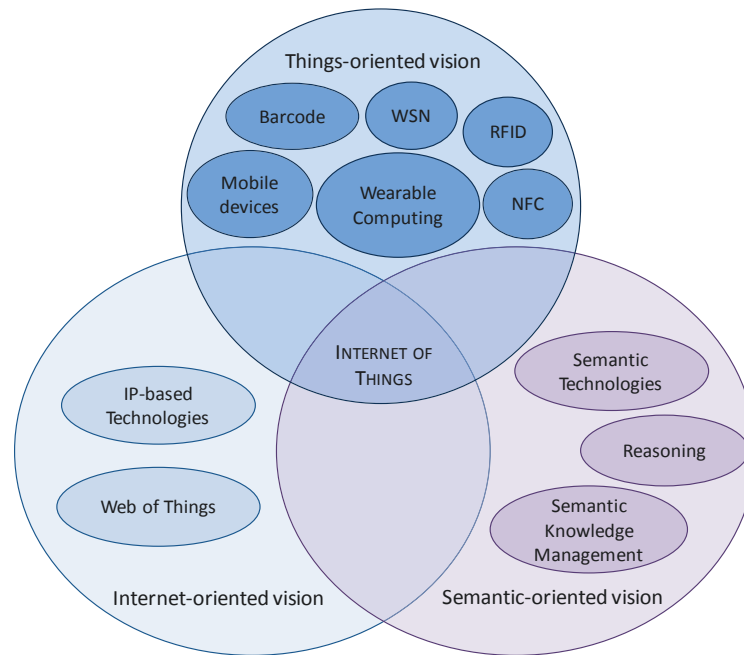


Figure 2.6: *Perspectives of the Internet of Things paradigm* (Atrozi et al. [25]).

Concerning the Semantic-oriented perspective of the Internet of Things, one significant challenge for the Future Internet is related to the high number of involved distributed objects. This fact, along with the existing significant information overload, depicts a very challenging horizon for information management on the Internet of Things. As discussed in Section 2.4.2, the application of semantic technologies with knowledge management techniques provides support for exploiting and improving modelling solutions for enabling reasoning over data generated in the IoT in terms of ontology management, knowledge discovery and human language technologies, as discussed before.

Regarding the Things-oriented approach of the Internet of Things, this leads to a loosely-coupled, distributed network of smart objects, which provides embedded interaction support for autonomous physical/digital objects and things powered with sensing, actuating and networking capabilities. As remarked in [64] some underlying technologies provide support for materializing this vision. These include, among others, the following: barcodes (one or two dimensional barcodes used to code an identification number for an item), wireless ad hoc and sensor networks (using sensors for modelling the environment where they are deployed and exerting and influence over it by means of actuators), Radio Frequency Identification (RFID, both passive tags operating without batteries and active tags using batteries to power sensors and wireless communication modules), mobile and personal devices (including smartphones, PDAs, pads, tabs and netbooks), Near Field Communications (NFC, short range radio frequency technology providing communication support for devices in very close proximity) and wearable computing (non-intrusive input, output and personal data-processing equipment that become an extension of our bodies, giving us extra ubiquitous sensing and computational capabilities).

Besides the previous, and as remarked by Mattern and Floerkemeier [189], several technological challenges arise for building the Internet of Things. First, scalability in IoT is expected to be a critical factor, operating in environments with a potentially high range of device density. Second, it enables ready-to-go operation support according to which mobile smart objects can configure itself according to the conditions of the visited environment. Third, interoperability between heterogeneous smart objects and devices is a critical issue, so standards and common practices for enabling communication and cooperation are required. Fourth, automatic discovery capabilities for suitable services are mandatory in dynamic environments. Fifth, software complexity providing an appropriate balance between minimal embedded software in resource-constrained smart devices and a more extensive networking and software infrastructure to manage such objects is demanded. Sixth, increasing data volumes from heterogeneous sources in specific application scenarios and with different nature and generation rates have to be managed efficiently. Seventh, data interpretation can be enhanced by means of context-aware information processing providing support to the users of smart things. Eighth, security and personal privacy in the IoT extend the existing research directions in the classical Internet with new challenges, including selective access to objects, preventing communications depending on the location, time or interaction manner, and protecting business transactions involving smart objects. Ninth, fault tolerance has to be defined in order to create a world of smart object in a robust and trustworthy manner, by means of implementing resilient mechanisms for building data planes with autonomic recovery mechanisms. Tenth, power supply for ubiquitous objects is an open issue where energy management modules need to be designed with low-energy consumption requirements and where portable and long-term energy sources should be researched.

2.5.2 Applications Domains based on Internet of Things Ecosystems

The paradigm promised by the Internet of Things envisages the development and empowerment of many application scenarios of our daily life, where some of them are drawn in our current reality and others are proposed in a not too far future. The radical change claimed by this model is centred on integrating into everyday objects the required embedded hardware and software for creating around us a pervasive world of smart objects. In this regard, these intelligent items have the possibility to interact with others, as well as to elaborate the data perceived from the surrounding, thus leading to a wide range of heterogeneous application domains. Among these, and according to the classification provided in [25, 254], we can highlight on transportation and logistics domain, healthcare domain and personal and social domain.

2.5.2.1 Transportation and Logistics Domain

Recent advances in embedded computing and systems, mainly applied to the automotive engineering (e.g., developing cars, trains or buses) as well as to the building engineering (e.g., construction of roads, bridges or rails), are paving the way for the deployment and exploitation of smart infrastructures. Additionally, goods and foods are also saturated with radio frequency tags and sensors, which send and report information to the corresponding control points. This will lead to improve traffic routing, response to contingencies in less time, provide appropriate transport information, as well as to monitor the status of goods and foods.

These are several identified applications on the Internet of Things regarding transportation and logistics domain [48, 239, 313, 314]. These include logistics (radio frequency technologies can be applied to monitor in real-time almost every part of the supply chain, including product design, purchase raw material, production, transportation, storage, delivery, sales, and after-sale service), assisted driving (incorporation of sensor, actuator and data processing power in both vehicles and transport infrastructures for improving the navigation and safety, providing statistical information as well as with planning purposes), mobile ticketing (use of short-range radio technologies in mobile and personal devices with the aim to retrieve information from multimedia panels and get access to ticketing services), monitoring environmental parameters (sensor technologies offer an important background for enhancing the efficiency of food supply chain, monitoring the ambient and conservation conditions) and augmented maps (tour guide service is provided using maps equipped with short-range radio frequency tags, allowing access through mobile devices to external information services).

2.5.2.2 Healthcare Domain

According to the significant increase of the elderly population in the occidental countries during last decades, the scientific community has put forth much effort to finding efficient solutions for improving the quality of care services while reducing the associated monetary cost [80]. In this sense, related technologies to the Internet of Things paradigm are perceived as a powerful approach to create advanced, non-intrusive healthcare management solutions based on sensor and mobile computing while minimizing deployment costs of networking infrastructure.

Novel research in healthcare domain of Internet of Things leads to several application scenarios [143, 166, 177]. These can be classified in the following groups: tracking (providing support for the identification and location monitoring of a person or object in motion, including real-time location-tracking of patients and medical staff), identification and authentication (including patient identification, medical record maintenance and infant identification, as well as grant access procedures for medical staff according to their profile permissions), data collection (automatic mechanism for data gathering and transfer, as well as procedures for generating reports for automated care, medical intake and inventory management), and sensing (sensor devices enable computation centred on patients, providing telemedicine solutions which include real-time supervision, data delivery to the medical staff and alert systems, leading to continuous bio-signal monitoring even in case of patient mobility conditions).

2.5.2.3 Personal and Social Domain

With the explosion of the Web 2.0, social computing (i.e., multi-disciplinary approach for modelling community behaviours on different media and platforms in order to produce interactive and collaborative applications) [153] is rising rapidly. The consolidation of this paradigm is due, in large part, to the technological ubiquity promised by the Internet of Things vision, where the proliferation of personal communication and mobile devices is a crucial issue. In this context, use cases related to personal and social domain enable users to interact with other people, as well as for building and maintaining social relationships based on automatic messaging exchange for event detections [297].

Major applications in the personal and social domains of the Internet of Things include social networking (performing real-time, automatic updates in social networking websites based on embedded technology interactions, generating events about people and places that are displayed to the rest or a part of the community), historical queries (tracking user activities building historical logs with information about objects and data events, with the aim to improve and support applications and processes related to business projects and collaborations), losses (apply sensor-based computing to build integrated search engines for smart objects in the environment, bringing support for enabling location registering and discovering locations for tagged objects), and thefts (in a similar way to the previous one, the use of embedded computing can be focused on presence monitoring in restricted areas, notifying the user that a specific object has been removed from its prefixed location).

2.5.3 Research Challenges in Wireless Sensor Networks

The Internet of Things paradigm continues on the road behind the smart environment idea, paving the way for next-generation digital ecosystems and application scenarios based on sensor network technology in the Future Internet [138]. In this regard, IoT has been recognized as a way for building smart spaces, which offers the necessary infrastructure for the realization of the Ambient Intelligence (AmI) vision [33, 86]. Proactively, but sensibly, Ambient Intelligence supports people in their daily lives in a non-intrusive way, focusing on improving the end-user experience with additional added-value services while minimizing the explicit human-computing interaction [26].

In the context of the proposed contributions in this dissertation, a promising research area is on Wireless Sensor Network technology for enabling and building dynamic and adaptive Ambient Intelligence ecosystems. Regarding ubiquitous computing, the effective use of pervasive smart environments represents a well-known challenge [250]. This issue focuses on providing an overlay infrastructure where users can interact and communicate in a natural way, improving the use of the real-world regions where they are deployed. Nevertheless, it is the own nature of such future infrastructures which impose some open issues [42, 87, 311]. These challenges include interoperability of heterogeneous entities, spontaneous communications and environmental dynamicity, ease of application development, support for different application requirements and user-centric service provisioning.

In a more precise way, the analysis of the previous is as follows. First, pervasive smart spaces are developed by interconnecting different computing entities such as sensor networks, smartphones, handholds, netbooks and wearable devices. This imposes the needed for specific approaches dealing with hardware and software heterogeneity and interoperability. Second, in these environments the availability of services may change during the system operation depending on several issues such as radio-channel conditions, device mobility and runtime failures. In this regard, and with the aim of enhancing the user experience, dynamicity in service provisioning should be guaranteed. Third, service development in smart infrastructures should be an open task, ideally accessible by third parties with little or even without concrete experience on embedded computing development. According to the previous issue, the provisioning of software approaches that facilitate the application implementation and

service integration into the environment is mandatory. Fourth, in such smart spaces is also necessary to provide support for different application domains and scenarios due to the underlying heterogeneity in existing service requirements. In this sense, service foundations should provide an independent internetworking platform with a set of general-purpose features for the business processes. Finally, service provisioning should follow a user-centric approach, where end-users are not interested in the internal work of devices and services. In this regard, encapsulation of complex control interfaces of individual devices into advanced functionalities while combining several pervasive in-network services in the smart environment is demanded.

To address the previous our approach focuses on distributed service platforms for providing dynamic composition of virtual sensor services in pervasive smart spaces and environments, while integrating the sensor network into external service clouds that will be accessed by the end-users using their own personal devices. This is done using middleware architectures based on service orientation, in-network service development with agent orientation and separation of concerns, as well as semantically-driven service composition based on ontology and knowledge management. This enables sensor technology to evolve from the perspective of traditional input devices to backbone platforms for deploying and integrating ubiquitous services.

2.6 Conclusions

In this chapter, a state-of-the-art analysis regarding research and development trends in Wireless Sensor Networks, as well as other information and computer science paradigms of interest for this doctoral thesis have been surveyed. These include architecture engineering for sensor networks, service and agent orientation, semantic technology and knowledge management in the framework of pervasive computing and the Internet of Things. As justified before, the application and integration of such service science and networking engineering approaches represents a cutting-edge research direction in next-generation wireless ad hoc and sensor networks. The rest of this section is devoted to provide a summary regarding main aspects of these technologies in the context of this dissertation.

Research engineering applied for architecting and developing Wireless Sensor Networks is a very active area. The protocol stack defined on sensor nodes is energy-consumption sensitive, integrating data with routing protocols, enabling efficient communication through the wireless radio interface and promoting collaboration between the wireless sensor nodes in the ad hoc network. The architecture defined for sensor networks is stratified in several levels, including Physical Layer, Data Link Layer, Network Layer, Transport Layer and Application Layer. In addition to the previous, a set of management orthogonal planes have been defined. These are responsible for allowing nodes to operate in an energy-efficient routing mode in mobile environments, as well as sharing resources between nodes. The transverse planes that have been defined are Power Management, Mobility Management and Task Management. From the analysis of the contributions made by the scientific and research communities during last years, it is observed that the real challenge of this ubiquitous computing paradigm does not rely on developing highly-efficient, single-purpose applications but to exploit their potential in order to build pervasive service provisioning while optimizing their physical resources.

Service-Oriented Computing offers support for the efficient development of application processes and their integration into dynamic business service platforms. Research agenda in service orientation differentiates three logical planes: service foundations at the bottom, service composition built over service foundations and service management on top. On the one hand, service foundations or service-oriented middleware for sensor networks present several open issues that have to be addressed. This remarks on vendor neutral and OS independent abstractions, provisioning of generic middleware functionalities, heterogeneous multi-service composition and programming support for application development. On the other hand, service composition in sensor networks can take advantage of sensor service virtualization. This can be combined with agent technology and semantic knowledge management in order to create a dynamic pervasive composition schema for providing intelligent decisions that would be impossible to create with isolated physical sensors. Finally, service management in sensor networks focussed on implementing autonomic service management procedures to assure network survivability without human intervention is a novel research area. In this regard, identified challenges in pervasive service include service failure detection, inter-domain interoperability and security models for resilient control planes, among others.

Agent-Oriented Software Engineering has been identified as a promising approach for developing sensor networks in an autonomous, flexible and robust manner. According to the agent theory, intelligent agents enable automatic interactions reacting autonomously to the changes in their environment, but also acting proactively, originating actions that affect to their environment. In this regard, Multi-Agent Systems are viewed as a loosely-coupled network of software agents interacting in a shared environment in order to achieve their own objectives. This paradigm allows decentralization of knowledge and computing making easier the modelling of complex components in a dynamic way, while maintaining coordination with others. In the context of this dissertation, the use of agent orientation technology for developing intelligent in-network services focuses on empower service composition in resource-constrained and embedded devices, according to the virtual sensor service paradigm. The service processing model based on in-network services allows enhancing the network scalability, unlike other back-to-base based communication topologies where decisions are taken according to locally sensed data. The separation of concerns approach applied to the service engineering leads to improve system modularity, readability and simplicity while minimizing the functional overlap with the rest of software pieces, thus decreasing the codification overhead.

Semantic Knowledge Management improves well-known, legacy knowledge management approaches. This enhances emerging requirements of knowledge-based services while providing a powerful framework for dynamic and interoperable management of knowledge assets. Semantic knowledge engineering is based on the integration of several research areas, including ontology management, knowledge discovery and human language technologies. On the one hand, ontology management covers the procedures and tools for managing knowledge bases defined around ontologies and their meta-data, producing and maintaining consistency between formal and real-world semantics. On the other hand, knowledge discovery research focuses on finding and extracting pieces of information of interest from raw data while concerning the whole knowledge extraction process. Finally, human language technologies improve information access and human interaction while amplifying our awareness for knowledge activities that fall in our interests. In

the context of our work, a knowledge management model based on domain ontologies applied for pervasive service composition using annotations is proposed. The application of semantic technology in the development of distributed service platforms and service composition procedures is based on the fact that it provides more intelligent information access and management by annotating service description interfaces with semantic meta-information.

The Internet of Things notion makes computing truly pervasive, implementing a global infrastructure saturated with a wide variety of internetworked physical objects. These have the ability to communicate, interact and cooperate between them and with other external networks with the aim of reaching common goals. This model enables anytime/anywhere/anymedia access and communication, being the result of the convergence of three main visions: Internet-oriented, Semantic-oriented, and Things-oriented perspectives. The Internet of Things enables smart objects with the possibility to interact with others as well as to elaborate the data perceived from the surrounding. This leads to a wide range of application domains including transportation and logistics domain, healthcare domain, and personal and social domain. In the context of this doctoral thesis, the application of wireless sensor and ad hoc networks for implementing Ambient Intelligence ecosystems is investigated. The effective use of smart environments provides an overlay infrastructure where users can interact and communicate in a natural way. However, it is important to appreciate that such paradigm also leads to several challenges. This covers from interoperability of heterogeneous entities to spontaneous communications and environmental dynamicity, including ease of application development, support for different application requirements and user-centric service provisioning, among others.

Related Work

CONTINUOUS advances and emerging opportunities in Wireless Sensor Networks point out that this ad hoc technology is presumably destined to change and redefine the way we perceive and interact with the physical world around us. People live connected with other in digital environments that are intended to enhance our everyday life, extending our interaction and communication capabilities through personal, mobile and sensory technology. In this regard, a cornerstone issue for enabling this vision is the remarkable investment on embedded and pervasive computing with funding from both national and international research programs performed by both academia and corporations.

This chapter visits the state-of-the-art related work regarding wireless sensor technologies for developing distributed service platforms based on service-oriented middleware and dynamic service composition. Section 3.1 reviews middleware technology from the foundations and legacy solutions to specific contributions for sensor networks, taking into account both traditional and service-oriented solutions. Section 3.2 surveys service composition, including foundations of service internetworking research and dynamic service composition in pervasive sensor networks. Finally, main conclusions of this chapter are summarized in Section 3.3.

3.1 Related Work in Middleware Technologies

Distributed computing has been significantly enhanced during last decade by the technological and engineering advances in both telecommunications and computer science. These achieved breakthroughs provide a unique opportunity for several domains of outstanding importance in the Information and Networked Society. In this regard, middleware technology has been identified as a cornerstone in distributed systems, especially in emerging areas such as grid and cloud computing as well as in mobile and pervasive networks.

3.1.1 Middleware Foundations

But, which is the real essence of the middleware? The answer to this question is based mainly on the fact introduced in the previous paragraph: as Information Technologies evolve so fast in terms of sophistication and popularity, computing systems and networks become more complex, diverse and heterogeneous [285]. In this regard, distributed communication systems currently operates with a high number of business processes, networking protocols, operating systems, microprocessors and diverse architectures. These include wired, wireless and hybrid environments with different types of hardware, from enterprise nodes to personal devices, fixed or mobile [284].

Several inner features of the middleware are focused on addressing the question of what the middleware is, concentrating in two main goals. First, middleware can hide the underlying network heterogeneity [112]. Second, middleware enables the development of applications [9]. These notions are merged in a more complete one [68, 184]. According to this, the middleware is perceived as a software layer for hiding the network distribution that helps to solve interoperability problems between heterogeneous hardware platforms, while offering a generic abstraction for simplifying the development and management of distributed applications. In this regard, middleware technology offers the capacity for integrating and reusing software components, providing support for masking and managing the inherent complexity of the computing environment and services available. This software abstraction supports the addition of a broad range of non-functional properties. These cover from scalability to usability, including reliability, extensibility, reusability, QoS assurance, efficiency and security [200].

3.1.2 Limitations of Traditional Middleware Frameworks

Legacy middleware architectures, such as Java RMI [116], CORBA [217], Jini [24], EJB [268] and DCOM [197], are normally heavyweight requiring significant computational, memory and network capabilities, not being suitable, in most cases, for the resource-constraint nature of pervasive computing devices such as wireless sensor nodes [103, 236, 310]. These well-known, classical middleware architectures have been usually devised for distributed networking environments with enriched resources, like enterprise systems, in contrast of the low resources available in ad hoc and embedded sensor networks [222].

Some practical examples can be visited to highlight the previous assessment. On the one hand, CORBA is a well-known solution that provides an architecture for object-based middleware frameworks, but most existing full-compliant implementations are typically too resource-consuming. The OMG's Minimum CORBA specification [216] defines a simple, compact profile optimized for reduced functionality systems. Nevertheless, with a footprint of about 100KB, the use of CORBA is not adequate for most wireless sensor hardware platforms [51]. On the other hand, Jini defines an object-oriented middleware architecture for Java-enabled equipment. However, the high memory overhead of Jini (i.e., the smallest implementation has a footprint of over 1MB), mainly due to the use of Java RMI, makes it unsuitable for highly resource-constraint devices such as wireless and ad hoc sensor nodes [131, 184]. In this regard, and in contrast of legacy middleware solutions, new approaches with lightweight and efficient

designs for sensing-based applications providing optimized abstractions for dealing with the resource-constrained nature of wireless sensor devices have to be proposed [176].

3.1.3 Mainstream Middleware for Sensor Networks

As justified before, new specific middleware developments and contributions for addressing the complexity and challenges in sensor networks are required. In this regard, and based on the previous work of Hadim and Mohamed [121], traditional middleware approaches for WSNs can be classified according to their programming model. This is illustrated in Figure 3.1. As shown, two main categories are defined: programming abstractions and programming support. Each of these is described in the next paragraphs.

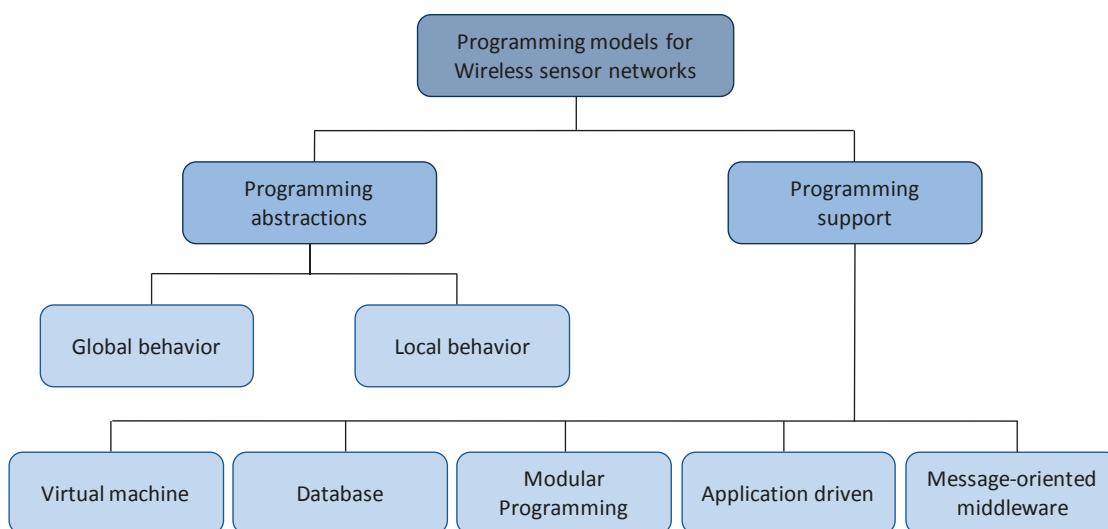


Figure 3.1: *Mainstream middleware for Wireless Sensor Networks* (Hadim and Mohamed [121]).

On the one hand, programming abstractions are related to the manner in which the sensor network is perceived, providing concepts and notions related to the nodes and sensor data. This includes two approaches: global behaviour and local behaviour. These are analyzed in the following paragraphs.

The first approach defined in the category of programming abstractions turns around to the global behaviour of a distributed sensor network, which is also called macro-programming. According to it, the sensor network is programmed as a whole, instead of writing low-level software to manage individual nodes. An example of middleware approach for sensor networks with a programming abstraction adopting a global behaviour model is Kairos [118]. This contribution establishes the necessary notions for allowing the design and implementation of a macro-programming abstraction for sensor-based computing. In this context, developers can model the system behaviour in the form of Java Applets, which can be executed locally on the nodes. This process involves both compile-time and runtime, where the developer is in charge of handling only a small number of primitives. The Kairos middleware makes it transparent to the programmer, especially the low-level details such as the distributed code generation and

access/remote data management, as well as the coordination tasks of program flows between network nodes. In addition, this proposal offers programming language abstractions that are mainly focused on node management, monitoring and data gathering.

The second approach is related to the behaviour of network nodes from a local perspective in distributed computing. The proposal of local behaviour focuses on the nature of the information generated by the sensors, and particularly for those on a specific network location. An example of middleware proposal for sensor networks providing a programming abstraction based on the local behaviour model is EnviroTrack [4]. This data-centric contribution is built on top of the TinyOS [164] operating system, which adopts an attribute-based naming system using context labels. The proposed routing mechanism is based on the content of the requested data, where the system is able to detect and report the presence of mobile targets. These issues confirm the suitability of this solution for many traditional application domains of ad hoc sensor computing, object location and tracking, battlefield surveillance and environmental monitoring. The EnviroTrack middleware is based on different services developed at the University of Virginia according to an initiative called VigilNet [124], which is a project developed around sensor network technology for surveillance missions.

On the other hand, the programming support approach manages the provisioning of systems, services and execution procedures as well as reliable code dissemination, secure code execution and other application-specific services. This includes five proposals: virtual machine, database, modular (agents), application driven and message-oriented middleware. These are analyzed in the following paragraphs.

The first approach defined according to the programming support is the virtual machine. This model includes virtualization software, code interpreters and mobile agents. Its main feature is flexibility. This allows developers to deploy applications fragmented into small modules, which are injected and distributed throughout the network using specific algorithms with the aim of optimizing the resource consumption. An example of middleware contribution for sensor networks with programming support based on the virtual machine paradigm is Mate [163]. This project focuses on the need for new programming paradigms, in order to overcome the resource-constrained features of sensor nodes. In this way, Mate provides a wide range of facilities to reprogram the system, offering a virtual machine system that runs over TinyOS. This abstraction enables support for a specific routing schema and a reprogramming model for each type of content. The key components of this architecture are the virtualization platform, the network, the logger or data recorder, the boot manager and the scheduler. This middleware contribution uses a synchronous execution model triggered in response to a received event, thus avoiding in-memory data storage.

The second approach included in the programming support is the database. This model looks at the network as a virtual database, offering an easy-to-use interface that allows the extraction of relevant data through queries in the sensor network. An example of middleware proposal for sensor networks with programming support adopting a database paradigm is TinyDB [181]. It is a query-processing system to extract information from a Wireless Sensor Network. This middleware uses TinyOS, which forces the developer to write code embedded in C Language for data from the sensors. This middleware approach simplifies the extraction of data using an SQL-

based interface. To specify the type of readings and the nodes of interest, simple data queries are applied. TinyDB stores a virtual database where its columns contain information about the sensor type, an identifier of the node and its remaining battery level.

The third approach is the modular programming (mobile agents). According to it, the use of mobile code facilitates the injection and distribution through the network leading to the modularity of applications. It takes less energy to broadcast small modules instead of a complete application. In this regard, Impala [174] is a mainstream middleware solution for sensor networks with programming support based on modular programming. This proposal leads to application adaptation at runtime, providing security against programming bugs. Mobility, transparency and scalability are some of the main services offered, alternating between different protocols and operating modes depending on the applications and network conditions. Furthermore, the Impala middleware provides self-adaptation regarding the appropriate protocols. Its high modularity aims to a significant good efficiency for energy consumption in sensor-based applications supporting small updates on mobile agents, which also generate a small overhead in the transmission and, therefore, high-battery lifetime.

The fourth approach is the application-driven paradigm. This provides a framework with a networking protocol stack, which allows developers to perform a fine tuning of the network in accordance with the requirements of the applications. According to the previous, an example of middleware proposal for sensor networks with programming support based on the application-driven paradigm is MiLAN [125]. This solution focuses on high-level details using an interface characterized by both applications and network. This contribution allows specifying the constraints of business processes in terms of QoS requirements, thus leading to setting up the routing features with the aim of increasing the lifetime of the applications based on their needs. To perform this task, MiLAN receives the QoS requirements of each application, the relative priority of each individual application for the overall operation of the system and the available resources in the network, as well as the remaining battery of the nodes and the bandwidth of the channel.

The fifth proposed approach is the Message-Oriented Middleware (MOM). This is basically a communication model for distributed sensor networks that enables the information exchange between nodes by means of messages (e.g., using a publish/subscribe mechanism). It supports asynchronous communication decoupling between producers and consumers of information. Mires [263] is an example of middleware solution for sensor networks with programming support following a Message-Oriented Middleware approach. This implements a component-based programming solution developed by using a publish/subscribe infrastructure built on top of TinyOS and embedded in C code used for their modules. Moreover, data aggregation service is provided for optimizing transmission over the radio interface. This contribution is able to provide to the users an asynchronous interworking service through publish/subscribe interactions in multi-hop environments, where applications are able to subscribe to the data of interest using a Graphical User Interface (GUI).

A major conclusion obtained from the review of traditional development platforms for Wireless Sensor Networks is that most of them have adopted an ad hoc approach based on application-specific designs, which makes difficult their reuse in heterogeneous application domains [186,

199, 240, 282]. Early contributions in this field were very sensible about the low capacity and resources in sensor networks, focusing on maximizing the performance of the applications in terms of memory usage, computing power, bandwidth and battery lifetime. This fact prompted that programming of initial sensor-based applications was a hard task, minimizing the code reuse and leading to a point where future software developments have to be undertaken from scratch [51].

3.1.4 Service-Oriented Middleware for Wireless Sensor Networks

As discussed, legacy middleware architectures for Wireless Sensor Networks have been specifically designed for addressing the complexity related to the resource-aware nature of this ubiquitous technology. However, due to the need for a high degree of optimization, most of these traditional contributions have been conceived as ad hoc solutions, being devised for addressing and solving particular problems in a very concrete application domain. This issue represents a real challenge for large-scale exploitation of sensor networks applied to the development of dynamic business service platforms in heterogeneous and multidisciplinary pervasive smart spaces.

In this regard, and with the aim to tackle the previous, a cutting-edge research direction is on translating the Service-Oriented Computing principles into the world of Wireless Sensor Networks. As discussed in Section 2.2, the SOC paradigm is promised to redefine the manner in which business applications are architected, implemented, provided and consumed. This provides support for the development of rapid, evolvable and interoperable application entities, which can be easily integrated into a loosely-coupled manner. Besides the previous, Service-Oriented Computing applied to middleware development provides advanced functions that are useful for reducing the complexity of designing and implementing software for different Wireless Sensor Network applications [200].

As one of its major goals, the application of this paradigm in the software engineering process for building Service Oriented Architectures over sensor and ad hoc networks is especially interesting since it creates non-specific design solutions, which is a key factor in the initial project development stages. In addition, it enhances code reuse, system scalability, parallel service development, unit testing and subsystem integration phases of software architecture [261]. With the aim of placing our contribution within the state of the art in service-oriented middleware trends for Wireless Sensor Networks, some of the most remarkable and novel approaches in this field are surveyed.

3.1.4.1 MiSense

MiSense [152] is a generic service-oriented, component-based middleware approach for Wireless Sensor Networks developed over the TinyOS [164] operating system. This framework has been designed to reduce the complexity existing in sensor-based applications with heterogeneous requirements. In this regard, this focuses on enhancing the optimization for specific scenarios, as well as the software flexibility and reusability degrees for implementing energy-aware applications in distributed sensor networks.

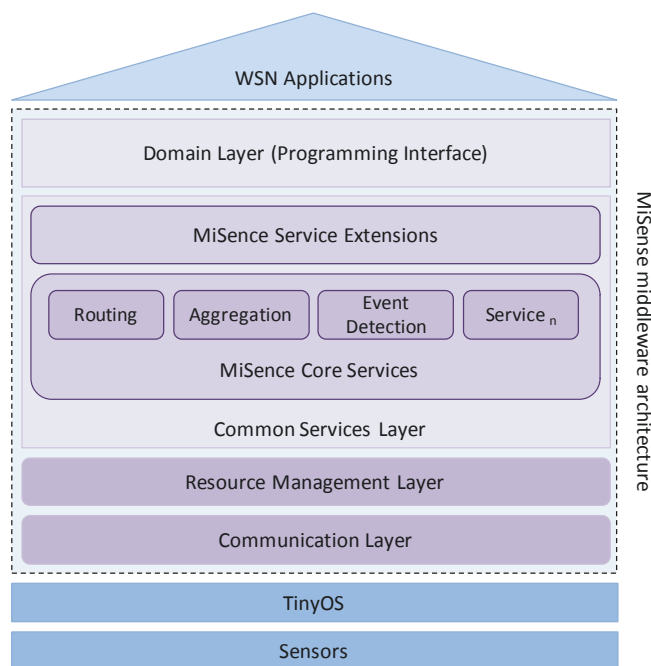


Figure 3.2: *The MiSense overall framework* (Khedo and Subramanian [152]).

As illustrated in Figure 3.2, this architecture defines a middleware approach that is divided in four layers. The communication layer is at the bottom, which implements a content-based publish/subscribe communication model called MiPSCoM. It allows application developers to adapt the behaviour of the services performing orthogonal choices in the components of the communication protocol to deliver subscriptions and notifications. This layer separates those service sub-tasks that are expected to have a large impact on the resource usage. Above this is the resource management layer, which performs and coordinates tasks for resource sharing in sensor nodes. This layer dialogues with the communication layer allowing access control handling to the required resources while commanding their allocation and adaptation. Over this is the common services layer. It enables a set of core services that are demanded by the applications including aggregation, event detection, topology management and routing. Additionally, this defines service extensions to grant and optimize existing services, providing plug-in support without modifying the current code. Some functionalities related to network configuration and dynamic adaptation for increasing network lifetime have been considered. Finally, on top is the programming interface. This enables procedures for allowing users to manage and program sensors with independence to the hardware and networking protocol.

3.1.4.2 SStreaMWare

Gurlen et al. analyze SStreaMWare [120], which is a middleware architecture based on the Open Services Gateway initiative (OSGi) [219] specification for information management tasks in pervasive sensor networks. This framework allows representation of data gathered from sensors in a generic manner. It is based on a service supporting a declarative query language that has been defined to access data streams while handling sensor heterogeneity and dynamicity. The proposed architecture is illustrated in Figure 3.3.

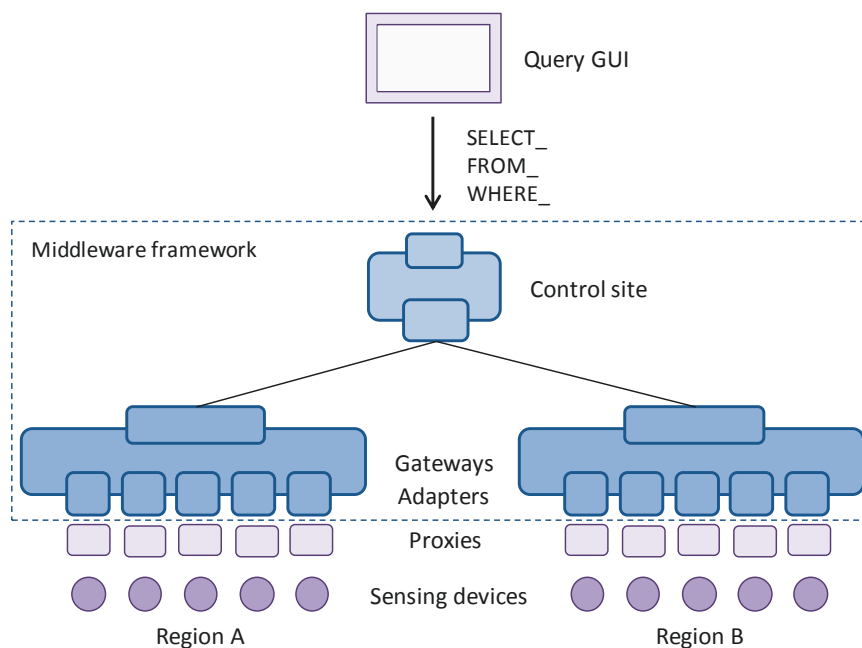


Figure 3.3: *The SStreamWare middleware* (Gurlen et al. [120]).

As discussed by its authors, this proposal has been designed as a hierarchical service-oriented framework based on the natural hierarchy of geographic locations and distributed query evaluation performed at different levels. The main elements considered in the SStreamWare contribution are the sensors, the proxies and adapters, the gateways and the control sites. The sensors, which are located at the bottom of the hierarchy, are distributed in the surrounding environment generating and sending the obtained physical measures periodically. The proxies and adapters support hardware heterogeneity. On the one hand, the proxy is a proprietary software API to access and manipulate a specific sensor. On the other hand, the adapters support the integration of the sensor specific proxy into the middleware, performing the translation between method call and data between the sensor platforms and the SStreamWare middleware. In this regard, and as stated by its authors, one of the main objectives of this contribution is to reuse deployed sensors by providing a unified manner to access them. The gateways are software platforms that manage sensors in a specific region, hosting different types of proxies and/or adapters. Finally, control sites are at the top of the hierarchy architecture. These are responsible for handling the queries produced in the environment, which is composed of several regions. In this framework, query evaluation is distributed among these levels being formulated according to a notation based on Structured Query Language (SQL), and evaluated continuously in data streams.

3.1.4.3 TinySOA

The TinySOA [27] middleware proposal provides to application developers and programmers a service-oriented API for accessing the Wireless Sensor Network in a homogeneous way via the language of their choice. Its design follows the Service Oriented Architecture paradigm, being implemented over the TinyOS operating system. The main objective of this contribution is to

provide adequate abstractions to developers of ubiquitous applications and services, as well as for making easier the integration of embedded sensor and ad hoc networks into the cloud allowing Internet-based applications to gather data from distributed sensors. The overview of this service-oriented framework is illustrated in Figure 3.4.

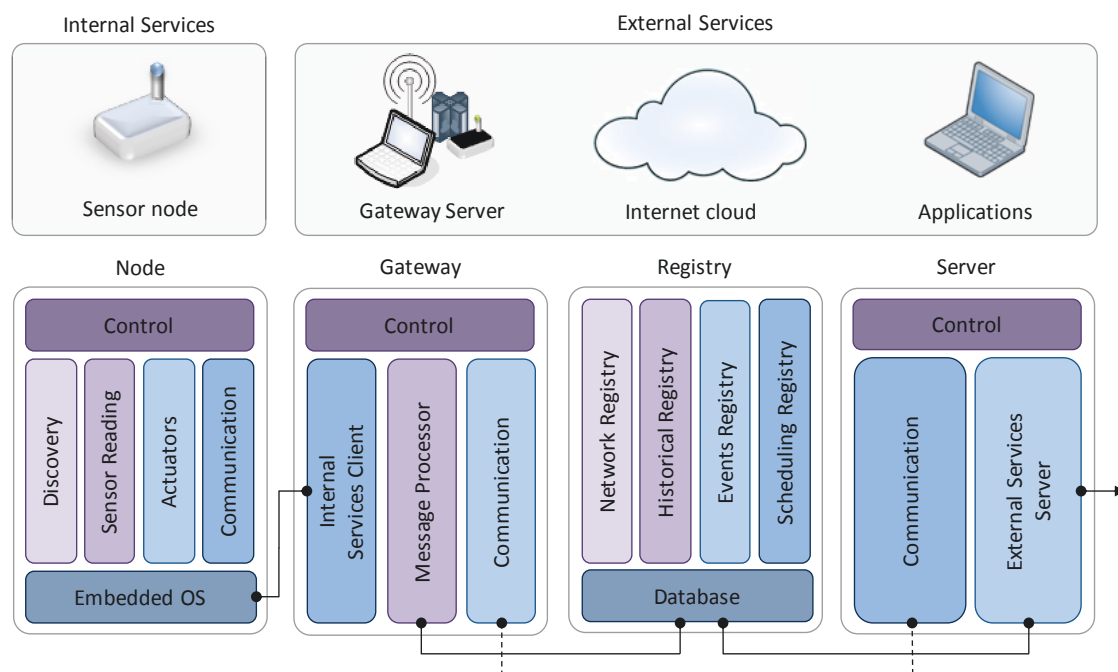


Figure 3.4: Overview of the TinySOA approach (Avilés-López and García-Macias [27]).

The operations performed in this approach take place at three sections, which are capture, concentration and application. As shown, the proposed architecture provides both internal and external services with the intervention of four components: node, gateway, registry and server. First, the node component encapsulates the functionality corresponding to a sensor device. This is composed of several entities such as service discovery, sensor reading, actuators and communication. These are deployed over the embedded operating system and managed by the control function. Second, the gateway component performs internetworking functions between the Wireless Sensor Network domain and the external service cloud. In this regard, several entities implement the gateway function including the internal services client, message processor and communication. This also includes a control entity, which is the responsible for initializing the gateway procedures, as well as for handling the rest of components. Third, the registry component stores the infrastructure information. There are several registry entities, saving network, historical, event and scheduling data. All of these rely on a database manager system. Finally, the server component acts as a Web Services platform for the provisioning of services based on the WSN. It implements an interface to query the offered services, check the registry, consult and register events, as well as for maintenance operations. In such a case, the control entity is used to initialize the server logic and to manage the communication, both in the sensor network and with external service servers.

3.1.4.4 USEME

The service-oriented middleware platform used for supporting the USEME [56] framework, which has been devised in order to simplify the development and use of lightweight services in ubiquitous Wireless Sensor Networks, is presented in [55]. As illustrated in Figure 3.5, this contribution identifies two views. On the one hand, there is the platform-independent part where an abstract programming language is provided in order to define the business processes, interactions and system nodes. On the other hand, there is the implementation part, which provides developers hardware-dependent skeletons to program the services in the declarative part. The code obtained when developers complete the skeleton files is the USEME-based application. This abstraction interacts with the USEME middleware through a given Application Programming Interface.

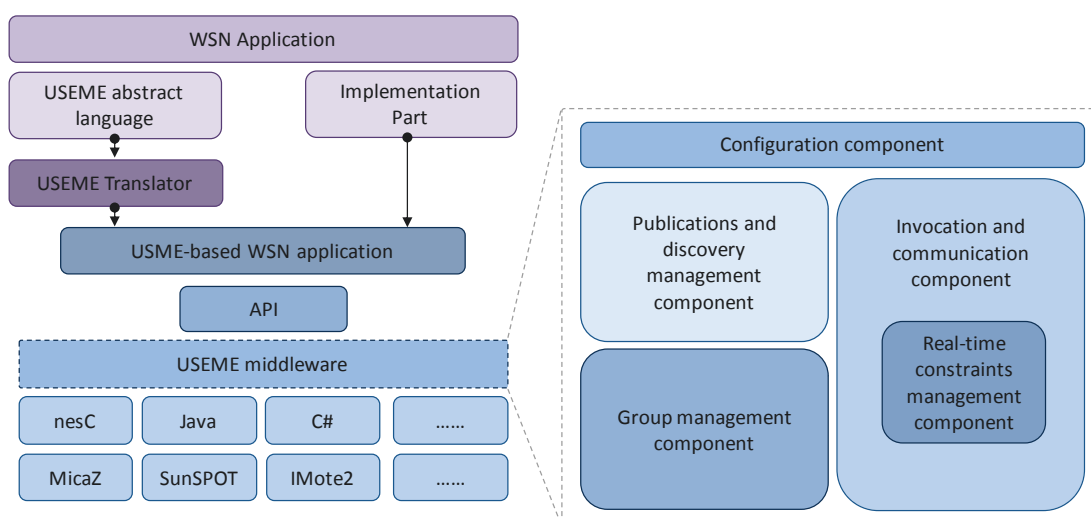


Figure 3.5: *The USEME service-oriented middleware* (Cañete et al. [55]).

The proposed middleware enables dynamic relationships for service interworking in the wireless sensor domain. It also provides support for an easier deployment of applications in the network while allowing a proper use of the network resources. As illustrated in the previous figure, the identified components in the service-oriented middleware are the following: configuration, publication and discovery, invocation and communication, real-time constraint, and group management. First, the configuration component allows the applications to setting-up some parameters of the middleware including time deadlines, discovery frequency or maximum packet size. Second, the publication and discovery component performs tasks related to distributed service announcement and registering, as well as network monitoring for detecting topology variations. Third, the invocation and communication component deals with remote command execution in a transparent way to the user applications, translating commands invocations into communication packets between peer sensor nodes. Fourth, the real-time constraints management component enables handling of real-time requirements. These can be defined as both events and commands through the USEME abstract language in terms of priority, deadline and periodicity constraints. Finally, there is the group management

component. It focuses on improving the scalability and efficiency degrees by means of sharing common behaviours, constraints and service management aspects inside the different logical groups of the sensor network.

3.1.4.5 SOMM

Faghih and Moghaddam define in [97] the SOMM approach. This represents a service-oriented middleware approach devised for supporting the application development process in Wireless Multimedia Sensor Networks (WMSN) according to an agent-based service model. The main objective of this SOA contribution is to enhance the programming tasks of modifiable and scalable multimedia applications in ad hoc and sensor networks, as well as providing multiple services to multiple end-users with an agreed level of Quality of Service. The overview of this middleware solution is illustrated in Figure 3.6.

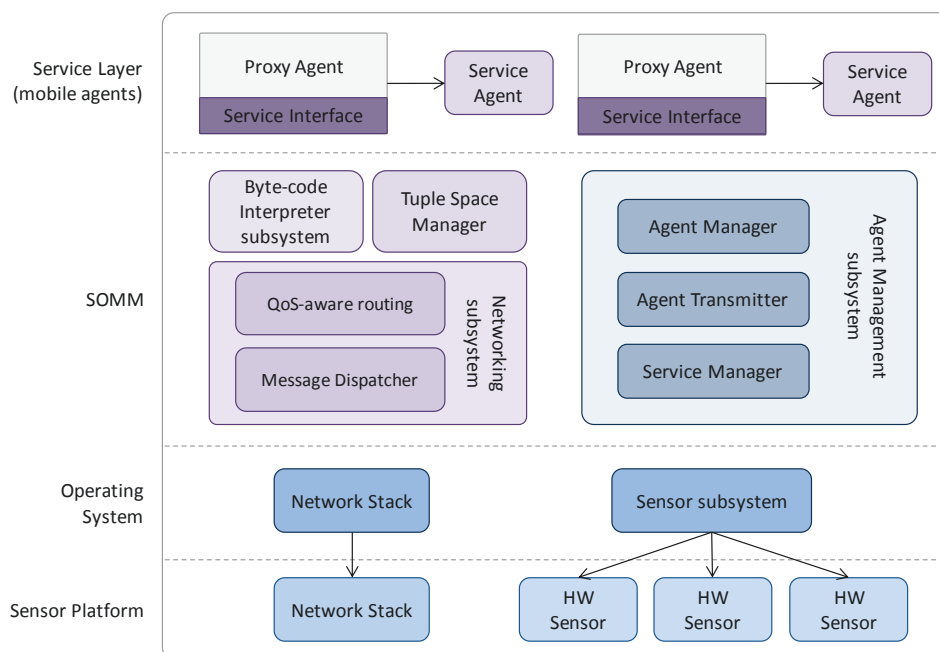


Figure 3.6: *The SOMM service-oriented architecture* (Faghih and Moghaddam [97]).

The SOMM platform is based on virtual machine and mobile agent technologies. On the one hand, the virtual machine approach provides independence of the operating system while increasing network flexibility. In this regard, this paradigm improves the portability of the software framework. On the other hand, mobile agents are applied for addressing the high size of multimedia applications, modifying the required business logic in each sensor node using mobile code. The previous figure shows the architecture of the SOMM middleware in the server node, over the sensor platform and the operating system layers. The agent management subsystem handles the agent mobility in this approach, being supported by the agent transmitter in the origin node and the agent manager in the target node. The networking subsystem supports QoS routing and networking setting-up for mobile code repositories and service registries using a XML-based syntax for service interactions. On top of the networking

subsystem there are the byte-code interpreter subsystem and the tuple space manager. On the one hand, the byte-code interpreter performs the agent code execution for application tasks. On the other hand, the tuple space manager provides a memory pool in each node for communication activities between distributed processes using read and write operations. Finally, over the SOMM abstraction and in the service layer, two types of agents are defined: proxy agents and service agents. Proxy agents provide an interface for the corresponding service, while the implementation of the service himself is in the service agent.

3.1.4.6 ESOA

The Extended Service Oriented Architecture (ESOA) [282] puts forward a layered framework over the classical SOA model for providing service consolidation and management. This contribution, which is built on top of the LiteOS [59] operating system, supports software development in C Language while providing UNIX-like abstractions for WSNs. A specific design for enhancing compatibility with other hardware platforms while making easier the programming of tasks on reduced functionality devices has been considered in this proposal. The resulting architecture of the ESOA middleware is illustrated in Figure 3.7.

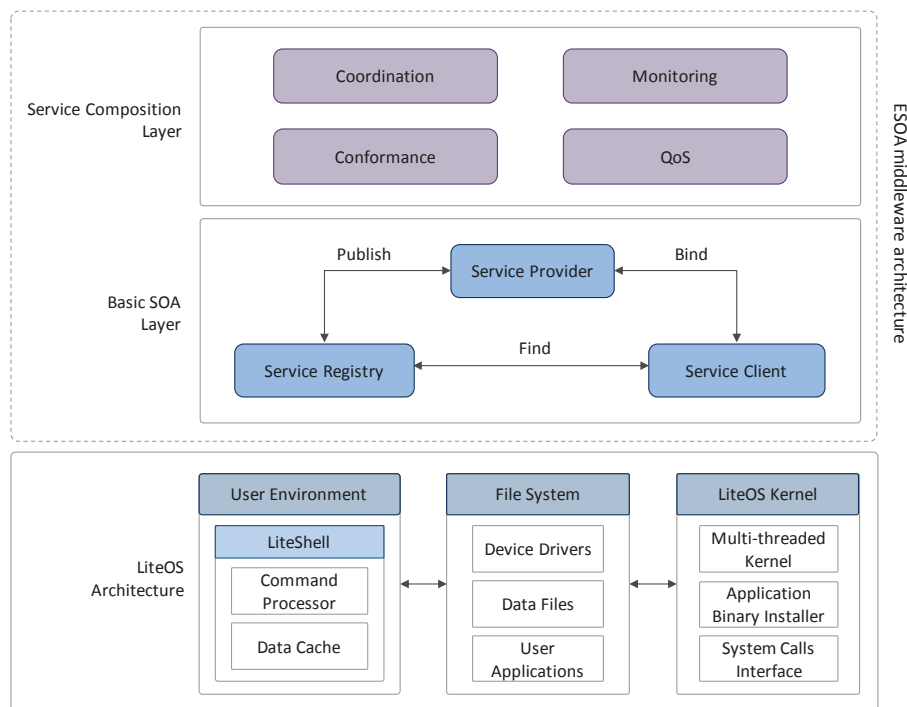


Figure 3.7: *The ESOA middleware platform* (Vanitha et al. [282]).

As shown in the previous figure, the ESOA service-oriented framework is deployed over LiteOS. This embedded operating system is divided in three subsystems: LiteShell, LiteOS File System and LiteOS Kernel. The LiteShell is implemented on a base station device and interacts with existing sensor nodes deployed within the radio coverage area. The mounting mechanism in LiteOS is provided by the LiteOS File System. It supports that nodes near to the base station can be auto-mounted wirelessly. Once mounted, the node appears as a file directory from the base station. At

this point, the LiteShell supports the execution of UNIX commands. The LiteOS operating systems is an interactive and reliable model providing the environment for object-oriented languages while enhancing software reusability, modularity and extensibility. On top of LiteOS there are the two layers of the ESOA service-oriented middleware: basic SOA layer and service composition layer. On the one hand, the basic SOA layer implements an interaction framework for message exchange between service requestors (i.e., clients) and service providers, relying on a service discovery agency. The interactions involve publish, find and bind operations. On the other hand, there is the service composition layer. It is implemented by the service aggregators and used as a business process by the service requestors. The procedures defined by the service aggregators include coordination (controlling the execution workflow of the service composition process), monitoring (allows subscribing to status information events), conformance (ensuring the integrity of the composite service) and Quality of Service composition (manages the QoS of the components to derive the composite one).

3.1.4.7 MidCASE

Bai et al. [28, 29] present the Middleware enabling Context-Awareness for Smart Environment (MidCASE) architecture. This service-oriented approach has been proposed with the aim to enable application programming of context-aware wireless ad hoc and sensor networks using reasoning techniques, focusing on distributed extensibility and efficiency. In this regard, the MidCASE proposal has been devised to bridge the gap between the service layer of heterogeneous application domains and the hardware and platform layer of resource-constrained and reduced functionality devices, supporting synchronization awareness in order to assure sensitivity to context-switching procedures.

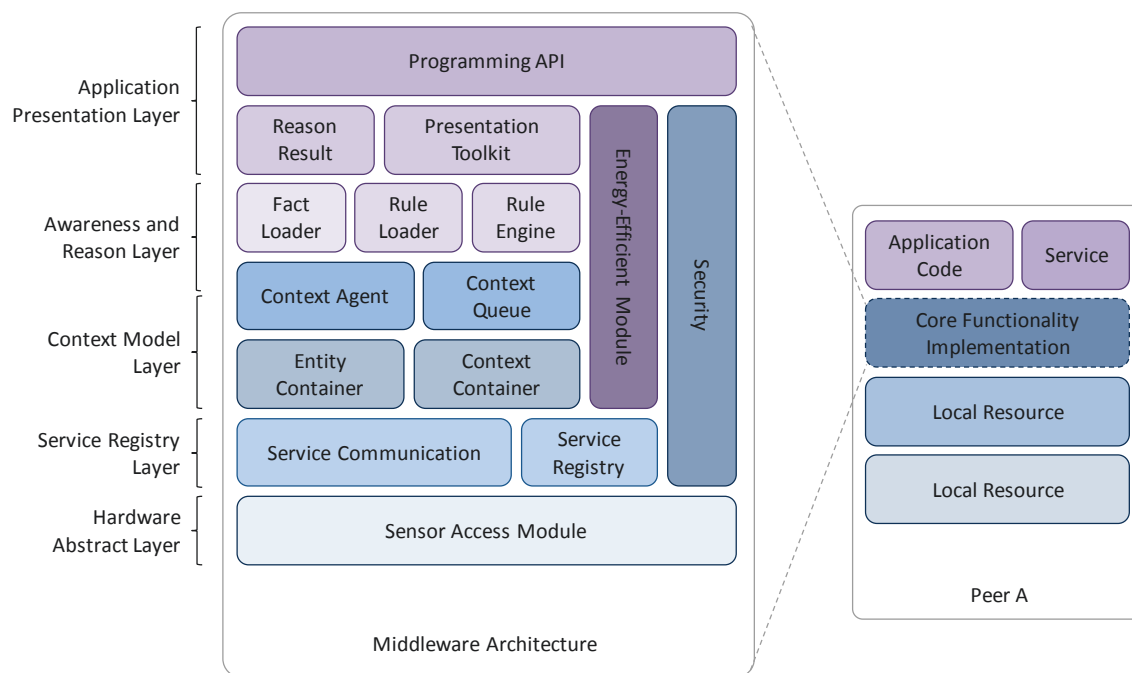


Figure 3.8: The architecture of the MidCASE framework (Bai et al. [28]).

The design principles of the MidCASE framework include service orientation, smart context-awareness, energy efficiency and rule-based reasoning, and middleware extensibility. As previously illustrated in Figure 3.8, this contribution is composed by five layers. These are the following: hardware abstract, service registry, context model, awareness and reason, and application presentation. First, the hardware abstract layer comprises heterogeneous sensor-based data sources to gather context-related information. Second, the service registry layer implements procedures for service registration and communication between services. Third, the context model layer makes easier the process of awareness and reasoning. The entity container models the environment while the context container connects contextual data provided by the hardware abstract layer. Fourth, the awareness and reason layer implements a rule engine embedded in the middleware, which supplies facts and rule loaders. Finally, the application presentation layer characterizes how the result of the reasoning process is used, being transferred using context change events. Besides the previous, the MidCASE framework defines two orthogonal layers, which are called modules in this approach. These are the energy-efficient and security modules. On the one hand, the energy-efficient module controls network energy consumption according to specific network conditions and application values. On the other hand, the security module refers to hardware authentication, as part of the context acquisition process, and access priority management of context-aware data.

3.1.4.8 UbiSOAP

The ubiSOAP middleware, which is discussed in [60], constitutes a framework for service-oriented communications in pervasive environments based on network-agnostic connectivity. To this end, functionalities based on the Simple Object Access Protocol (SOAP) standard are exploited, supporting legacy Web Services in ubiquitous computing ecosystems while focusing on connectivity between heterogeneous radio interface technologies for realizing a multi-network service overlay with seamless networking capabilities. According to the previous, the proposed architecture is illustrated in Figure 3.9.

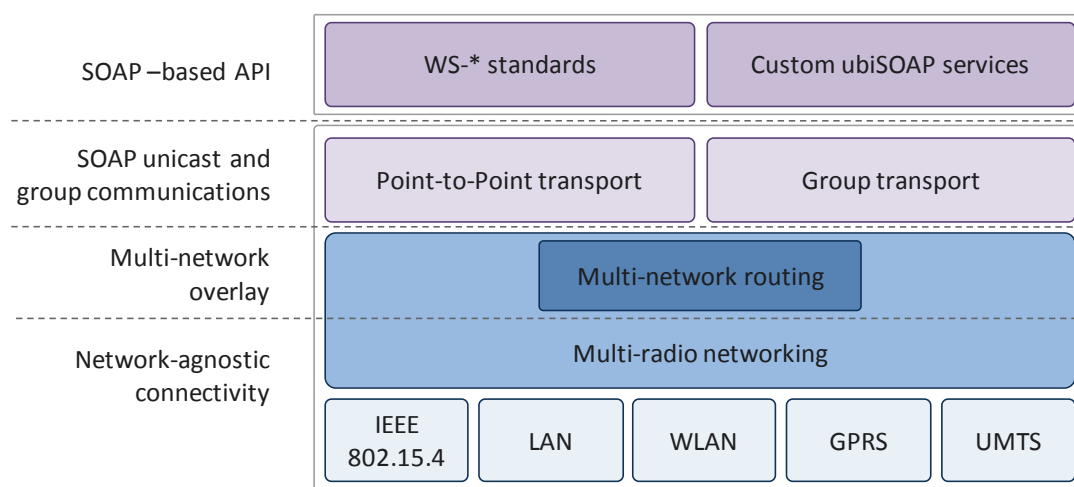


Figure 3.9: *The ubiSOAP middleware contribution* (Caporuscio et al. [60])

As shown, the architecture of the ubiSOAP middleware relies on a two-layered design. First, multi-radio technology supports network-agnostic connectivity. This layer deals with network and device heterogeneity, providing multi-network routing based on a multi-network overlay. Second, ubiSOAP communications define point-to-point and group-based transport protocols, offering to the upper layers a SOAP-based Application Programming Interface composed of interworking primitives oriented to ubiquitous computing systems. On the one hand, the network-agnostic connectivity layer implements a set of functionalities for handling the multi-radio heterogeneity in an effective manner. This is achieved by means of a network-agnostic addressing schema, a QoS-aware network link selection mechanism, and a unicast/multicast routing procedure. On the other hand, session mobility, efficient messages routing and both point-to-point and group-based interactions are supported using ubiSOAP communications. To achieve the previous, a custom SOAP transport layer is introduced. It leverages network-agnostic connectivity while providing an API based on SOAP for developers in the application layer. In addition, this contribution defines a service discovery protocol. This covers service providers, service consumers and service repositories where interactions between the previous follow a directory-based schema. This proposal, which is called Ubiquitous Service-Discovery Service (ubiSD-S), has been devised for providing dynamic, interoperable and context-aware service discovery functionalities. In this regard, an ubiSD-S plug-in is provided. This enables the clients to discover services in the multi-network environment.

3.1.5 Discussion

As introduced in Section 2.2.3, and based on prior studies [199, 200], a set of challenges to be addressed with Wireless Sensor Networks by means of middleware architectures for providing service-oriented solutions are defined. These open issues are the following:

- Platform and operating system independence are required using software abstractions for specific hardware solutions. This issue leads to enhance the interoperability of the middleware infrastructure and the business processes at the Application Layer.
- Middleware backbone functionalities. These include, among others, general-purpose facilities such as service discovery and configuration, real-time requirements and secure communications, as well as interoperability with external networks and service clouds.
- Support for heterogeneous multi-service composition. This feature is mainly focused on enhancing seamless service interworking and provisioning to the end-users, enabling service orchestration and discovery at the middleware level.
- Enable software application development. In this regard, programming tasks should be accessible to third parties with little or no technical background on software development in reduced functionality devices.
- Internetworking support with external service clouds and platforms. This feature provides capabilities for plug-in pervasive wireless sensor devices into advanced smart environments, while requiring open interfaces that enable integration and validation of data exchange between heterogeneous networking domains.

TABLE 3.1: *Comparative analysis of service oriented middleware architectures for Wireless Sensor Networks.*

SOA approach	Platform and operating system independence	General purpose middleware functionalities	Multi-service composition architecture	Software application development	Internetworking with external service platforms
MiSense [152]	Built on top of TinyOS operating system	Resource management, data aggregation, event detection and topology management	Feature not supported	Service extensions and programming interface	Feature not supported
SStreaMWare [120]	Independence with gateway-side middleware	Information management based on data queries	Feature not supported	Based on OSGi bundles	Feature supported
TinySOA [27]	Independence with in-network middleware	Discovery, data readings, actuators management, and network communications	Feature not supported	Feature not supported	Feature supported
USEME [55, 56]	Independence with in-network middleware	Configuration management, group management, and invocation and communication management	Feature supported	Abstract programming language	Feature not supported
SOMM [97]	Independence with in-gateway middleware	HW resource management, security and QoS	Feature not supported	Programming tasks based on byte-code	Feature supported
ESOA [282]	Built on top of LiteOS operating system	Coordination, monitoring, conformance, QoS and service discovery	Feature supported	Feature not supported	Feature not supported
MidCASE [28, 29]	Independence with in-network middleware	HW resource management, service registry, context-awareness, reasoning and security	Feature not supported	Feature not supported	Feature not supported
ubiSOAP [60]	Independence with in-network middleware	SOAP-based network communication and service discovery	Feature not supported	Feature not supported	Feature supported

In Table 3.1, a summary of the surveyed service-oriented middleware approaches regarding the open issues previously identified in Wireless Sensor Networks is illustrated. As main conclusion of this study, none of the reviewed approaches fulfils all the identified requirements in a global manner. First, approaches such as MiSense [152] or ESOA [282] have dependencies with specific operating systems and do not implement mechanisms to provide independence to the middleware. The MiSense architecture has been developed over TinyOS [164] while the ESOA framework has been built on top of LiteOS [59]. Second,

some of the reviewed contributions do not provide generic middleware functionalities for developing application services in a comprehensive way. In this regard, the SStreaMWare [120] architecture is focused on data management based on queries, while ubiSOAP [60] has been designed mainly for enabling SOAP-based communications and service discovery. Also, some proposals that apparently provide general-scope middleware services, like USEME [55, 56] or ESOA [282], do not address hardware resource management, which is a basic engineering issue in reduced functionality and embedded devices. In the case of security services, these only have been taken into account in SOMM [97] and MidCASE [28, 29]. Third, provisioning of multi-service composition support has not been defined in most of the proposals under study. Only the USEME [55, 56] and ESOA [282] approaches have included this feature in their design. On the one hand, the service composition schema defined in USEME is based on ports, in terms of defining an event-based communication mechanism between services. On the other hand, ESOA includes procedures for coordination, monitoring, conformance and management of Quality of Service requirements in service composition. Nevertheless, its authors do not provide specific implementation details. Fourth, development support for software development is not covered by all the surveyed service-oriented middleware contributions, only being implemented in MiSense [152], SStreaMWare, USEME [55, 56] and SOMM [97]. MiSense provides service extensions for plug-in optimized and redefined services. SStreaMWare is based on the OSGi [219] standard framework, thus offering an interface based on bundles for software development and deployment. In the case of USEME, its authors define an abstract programming language used for specifying nodes, groups and services. The programming of agents in SOMM is supported using byte-code. Fifth, the internetworking with external service clouds and platforms is a feature not considered in most of the reviewed approaches. Only SStreaMWare [120], TinySOA [27], SOMM [97] and ubiSOAP [60] include mechanisms based on Web Services for internetworking between pervasive networks and external systems.

3.2 Related Work in Service Composition

Ambient Intelligence ecosystems are fully saturated by autonomous and internetworked devices providing heterogeneous functionality to the end-users in a flexible, integrated and almost transparent manner. The singular nature of the ubiquitous computing environments has put in the spotlight some further research directions. In this regard, the development of dynamic and autonomic procedures for service composition and delivery is perceived as a particularly important research line for pervasive smart space engineering.

3.2.1 Services and Composition Methodologies

Service composition can be understood as a coordinated aggregation process of multiple service entities into higher-level business applications according to existing service patterns with the aim of providing an advanced, added-value functionality to the end-users [41, 76]. Identified coordination methods among simple services (i.e., atomic service units used in the composition procedure) include orchestration and choreography processes, which were discussed in Section 2.2.2.2. The integration and assembly mechanisms implemented by the service composition

process focus on combining two or more instances of service units, or even composite services, from one or more service providers over heterogeneous networks. In this regard, service composition provides a natural way to extend the ability of the services enabling rapid application development and service reuse [167]. Nevertheless, and although there is a significant current interest in service composition, it has been a research topic since the annals of distributed computing. Considering well-known legacy composition schemas, we highlight on web service orchestration, traditional agent-based interaction and service mashups [49]. These approaches are the basis for pervasive computing service composition, which will be surveyed in next section.

From a general point of view, service composition can be modelled according to several design aspects. In this regard, a characteristic taxonomy is proposed by Zhou et al. [316]. It considers reliability, adaptability, flexibility, representability, semantics, reusability, availability, stochastics, security, visualability, atomicity and coordination as potential features for service composition. Reliability comprises the capability of a service composition to develop its functions under stated conditions for a specified time interval. Adaptability represents the support of a service composition to adapt itself to the environmental conditions. The ability of a service composition to act in response to inner or external changes is expressed as flexibility. Representability is defined as the capability of describing all the composition relationships, while semantics can be applied to service composition using knowledge management techniques. The capability of using several times a composite service with minimal or no modification is referred as reusability. Availability is related to non-disruption support for service provisioning in the environment. Stochastic represents the non-deterministic nature of services in dynamic composition processes focusing on achieving the expected results in uncertain environments. Robust service composition in adverse environments is afforded by means of security. The property which enables visual composition of services is visualability, while atomicity provides consistency of applications during the composition process. Finally, coordination involves all the necessary methodologies for managing, organizing and synchronizing service operation in composition-based architectures.

3.2.2 Service Integration and Composition in Pervasive Computing

The ad hoc, seamless service composition is a commonly highlighted property of pervasive networking systems, being identified as a critical research issue for the success of ubiquitous computing [58]. This is mainly motivated by the own nature of such smart environments, which are fully saturated with heterogeneous devices, thus generating a highly diverse characterization [309]. On the one hand, pervasive spaces embody a significant number of heterogeneous devices with disparate capabilities including full-function and resource-constrained computing and networking equipment. By means of service composition techniques, heterogeneous devices are able to provide advanced services that would otherwise not be available [53]. On the other hand, pervasive computing is inherently dynamic. In ubiquitous environments, services should evolve showing proactiveness and adaptability to current conditions [266]. This dynamic behaviour is focused on supporting the effective use of smart spaces. In this regard, aspects such as invisibility of technology and masking of uneven conditions play a key role in pervasive computing [250]. These previous issues motivate the dynamic combination of service

units in order to create composite services with the aim of providing support for complex tasks in service-oriented environments [150].

In contrast to traditional distributed networks, service composition in ubiquitous computing imposes new requirements. According to the dynamicity of anytime/anywhere/anymedia information access and the heterogeneous nature of ubiquitous devices, the design and implementation of service composition is a complex and challenging task [62, 65]. In this regard, and based on the prior work of Brønsted et al. [49], several goals have been recognized for modelling and developing service composition. These include context awareness (i.e., the composition process or its constituent services are sensitive to context changes), managing contingencies (i.e., support for handling variations in the unpredictable availability of services), leveraging heterogeneous devices (i.e., middleware providing interoperability between devices and information models) and empowering users (i.e., new interaction models based on user-centric service composition).

3.2.3 Dynamic Service Composition in Wireless Sensor Networks

The resource-constrained nature of Wireless Sensor Networks leads to an error-prone and unreliable communication between ad hoc nodes, where sensor-based services are coming up and going down frequently. This fact, together with the composition-centric nature that characterizes ubiquitous computing, has a deep impact on the process and outcome of service composition engineering. Such environmental dynamicity demands service architectures able to adapt their composition schemas by taking advantage of the currently available services in the media [30]. In this situation, one major objective is to increase the availability of the new, composite services in the pervasive smart infrastructure.

In our best knowledge, the number of practical approaches with significant contributions to this topic in sensor networks is quite limited, as stated in [108, 109]. This is mainly due to the novel facet of this scientific area in embedded sensing devices. As discussed in Section 2.2.3, and according to the provided overview regarding Service-Oriented Computing in ad hoc and sensor networks, most of the current works are on service foundations or service-oriented middleware while service composition and management remains relatively unexplored. In this respect, and with the aim of placing our contribution in the state-of-the-art research regarding dynamic service composition in wireless ad hoc and sensor networks, next sections survey some of the most remarkable approaches in this area of study.

3.2.3.1 SCPQ

Service Composition with Persistent Queries (SCPQ) in service-oriented Wireless Sensor Networks is discussed in [293]. According this work, Wang et al. analyze the impact of service composition in terms of persistent query tasks and energy consumption. As the main outcome of this research, the authors provide two algorithms. These focus on minimizing the number of composition solutions during the lifetime of a query and the transmission cost for executing a persistent query, respectively. The optimality of both converges on a service composition solution with persistent queries that minimizes the energy consumption.

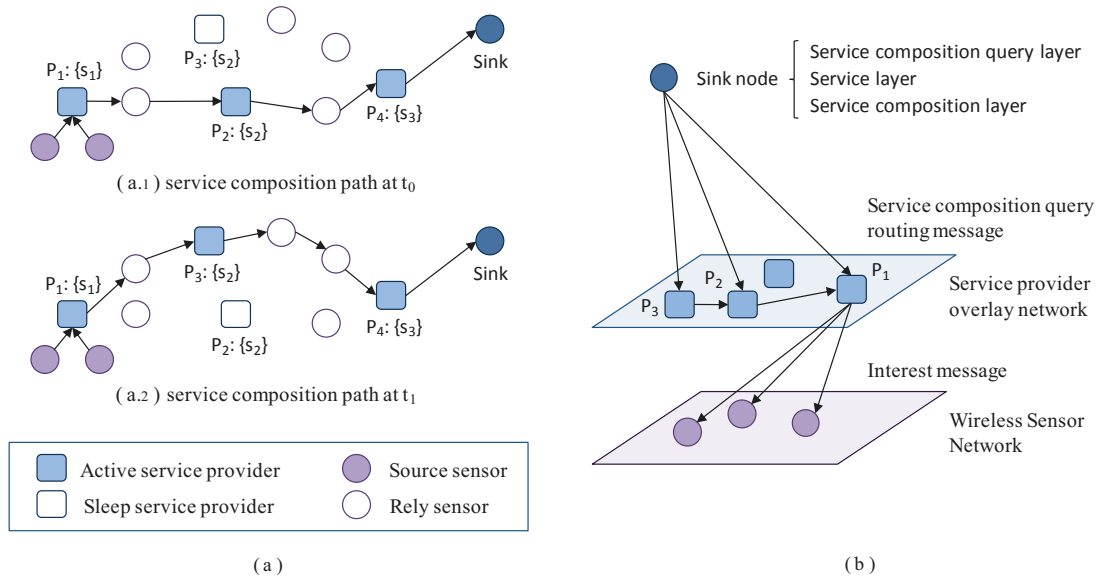


Figure 3.10: The SCPQ service composition approach (Wang et al. [293]): (a) example of persistent query in a service-oriented sensor network; (b) overlay system model.

This contribution addresses the problem of routing persistent queries from service clients to service providers in WSNs while supporting sleep-aware operation modes. For a persistent query, the routing schema may need to be executed several times due to the sleep mechanisms implemented at MAC level. This can lead to a path switch during the completion of a persistent query for a specific service resulting in an unavailability of the composition route between several simple services (see Figure 3.10.a). According to the discussed solution, the proposed overlay framework is illustrated in Figure 3.10.b. This SOA defines three layers at the sink node: service composition query layer, service layer and service composition layer. According to this model, the SCPQ solution develops a service-oriented query protocol based on advertisement flooding from the sink node to the service providers. It includes service composition solution, duration and interest. On the one hand, the interest will be broadcast in the network in an ordered manner by first service providers, in order that sensor devices can send the gathered data to the service provider. On the other hand, service providers that are in the composition but not at first place will use a general-purpose routing protocol to find a route to their upstream service provider. This approach enables self-managed service composition to the wireless sensor nodes in a transparent manner.

3.2.3.2 SDCBAN

Coloberti et al. [70] distilled a framework for Service Discovery and Composition in Body Area Networks (SDCBAN), which has been developed as a two-layered architecture based on a distributed service directory. This addresses the mobility of the sensor nodes using a clustering approach based on network connectivity patterns. A multi-agent Finite State Machine (FSM) has been defined with the aim of detecting the availability of complex services while providing them. As main identified advantages, its authors claim for a dynamic behaviour and lightweight implementation, also remarking its context and QoS-aware, service-oriented design.

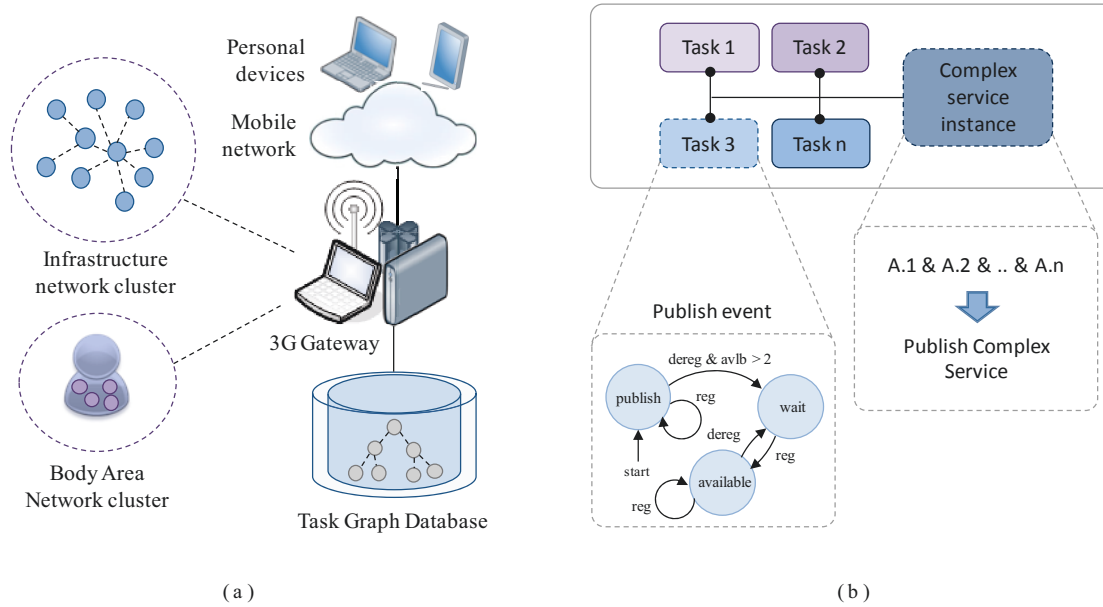


Figure 3.11: Overview of the SDCBAN service framework (Coloberti et al. [70]): (a) network architecture; (b) TGD based on multi-agent FSM keeping track of service availability.

According to the SDCBAN architecture, complex services are stored in a Task Graph Database (TGD) in a server side. This is connected to the Body Area Network (BAN) using a 3G mobile communication system and a gateway device. There is a cluster topology for the wireless sensor nodes composing the BAN, where a service directory is defined for collecting and dynamically updating the service information of their neighbours. The information in the distributed directory is modelled using a TGD, which receives service advertisements making them available in the wireless domain. The overview of the SDCBAN architecture is illustrated in Figure 3.11.a. As shown, a key piece of this framework is the TGD. It is perceived as a mesh of simple service interconnections. This database paradigm abstracts from the particular design and implementation details of each hardware sensor platform, offering support for identifying complex services that are executed in the WSN. As illustrated in Figure 3.11.b, the Task Graph Database is based on a multi-agent Finite State Machine, which follows a producer-consumer paradigm where each agent is involved in the composition process. This multi-agent approach is adapted to the dynamic interactions of the TGD by using high and low capability profiles depending on the resources of the devices in the sensor network and BAN.

3.2.3.3 SWSN

Focusing on exploiting wireless ad hoc sensor and RFID networks to address the industry requirements as stated in [315], the Service-Centric Solution for Wireless Sensor Networks (SWSN) is overviewed. This work defines a service platform that integrates pervasive sensor-based services into the Internet of Things world for industrial environments and applications. Being designed according to the foundations of SOA, this architecture uses XML and SOAP standards to describe, discover and invoke services while Semantic Web Services have been applied for enabling self-service integration. According to the previous, the proposed framework approach is illustrated in Figure 3.12.

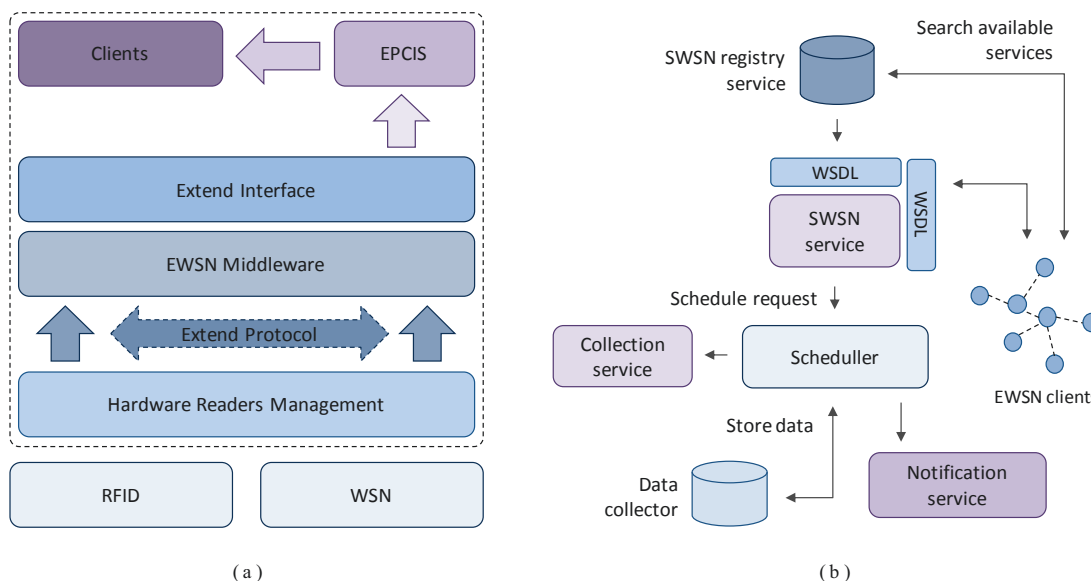


Figure 3.12: *Service-Centric Solution for Wireless Sensor Networks* (Zhou et al. [315]): (a) *EWSN sensor-based architecture*; (b) *SWSN dynamic service platform*.

The software system deployed in sensor nodes and RFID devices is called Extended WSN (EWSN), which is illustrated in Figure 3.12.a. In this regard, the EWSN middleware has been devised to gather and distribute data from hardware sensors based on WSN and RFID events, by using rules for supporting event monitoring and in-network data processing. According to this contribution, EWSN clients implemented over this approach lie on the Electronic Product Code Information Services (EPCIS) specification. This defines a standard for identifying, naming and sharing information of smart objects within and across the enterprise domain. Besides the previous, the authors propose the integration of the EWSN approach into the Internet of Things paradigm. This is achieved in a smooth manner by taking advantage of SOA design principles. This leads to the Service-Centric EWSN (SWSN) contribution, which overview is illustrated in Figure 3.12.b. The SWSN framework defines service registries for supporting service publication and discovery functions for EWSN clients in the network. According to a data-centric approach, SWSN services are based on tasks for managing service collection and notification by means of service scheduling. Moreover, Semantic Web Services are included for supporting autonomic service composition based on semantic knowledge management. It focuses on providing a common understanding in the wireless sensor domain for service discovery, composition, invocation and monitoring. To this end, SWSN services are described using Web Service Description Language (WSDL) and XML schema with semantic annotations applying ontologies.

3.2.3.4 DSIDMSS

Liu et al. discuss [175] the Distributed Service Integration for Disaster Monitoring Sensor Systems (DSIDMSS) middleware. It represents an architectural model enabling distributed, sensor-based service integration for allowing flexible and reliable provisioning of rescue capabilities in high-dynamic environments. This contribution proposes the implementation of

proactive self-adaptation and service integration procedures for wireless sensor systems and networks using the foundations of the Service Oriented Architecture paradigm. As stated by its authors, this solution leads to a loosely-coupled interworking proposal between producers and consumers, thus minimizing the impact of potential service-related dependencies on the rescue composition workflow. According to the previous, the service provisioning framework in the DSIDMSS architecture is illustrated in Figure 3.13.

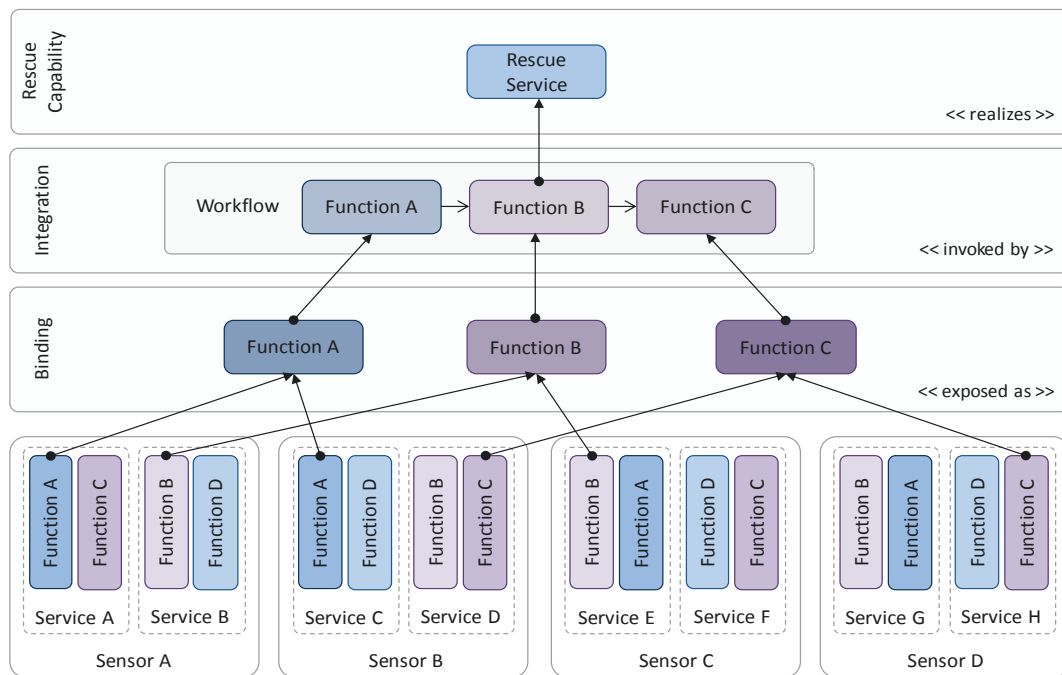


Figure 3.13: *The DSIDMSS proposal for redundant service provisioning and binding in rescue systems* (Liu et al. [175]).

In this contribution, each network node provides a set of services that can be integrated into a more complex functionality for providing search-and-rescue capability. This is done by means of dynamic binding, where services are integrated into a workflow to fulfil a specific objective. The DSIDMSS solution addresses both service discovery and redundant service binding. On the one hand, distributed service discovery is implemented with an adaptive peer-to-peer search (AEPS) algorithm [172] based on social behavioural patterns. On the other hand, service provisioning is enhanced by using redundant service binding. In addition, and in order to provide robust rescue support, required functions of services are allocated in different nodes defining a proactive self-adaptable and reconfigurable algorithm for service switching in case of service disruption or failure. This feature has a strong impact on the flexible and reliable integration of services, where the rescue capability evolves according to the environmental changes. The core of this service composition approach includes rich information services, evolving ontology and service interoperability, as well as on-the-fly optimisation and planning. As stated by its authors, major advantages of the DSIDMSS proposal are on timeliness, reliability and fault tolerance. These issues provide dynamic discovery support and service composition capability to this system based on sensor technology.

3.2.3.5 MRSWSN

The Middleware for Resource Sharing in multi-purpose Wireless Sensor Networks (MRSWSN) architecture is analyzed in [85]. This constitutes a sensor network framework for efficient management of shared resources supporting context-aware operations. The authors claim for the importance of WSNs in multi-purpose deployments with the aim of developing lightweight service delivery platforms for distributed applications. In this regard, the main contribution of this research is on defining a dynamic composition model based on shareable components with heterogeneous parameters about quality of data. This is supplemented with a planner for enabling effective management of the physical hardware resources in sensor-based wireless infrastructures.

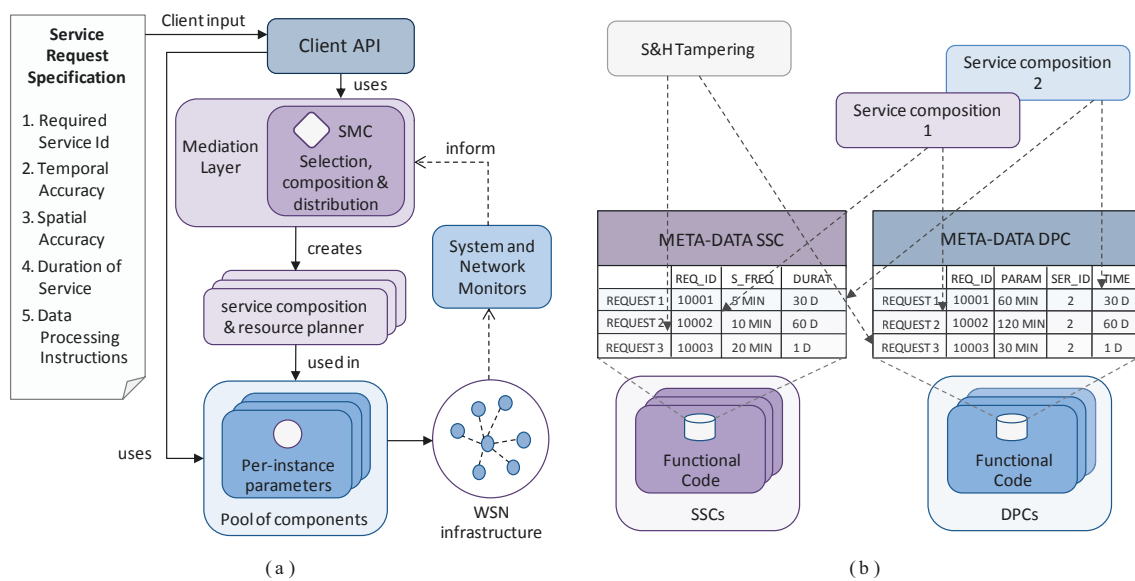


Figure 3.14: The MRSWSN contribution (del Cid et al. [85]): (a) middleware architecture; (b) configuration semantics for components.

As illustrated in Figure 3.14.a, this approach considers simple compositions and per-service instance parameterization. This is enabled by using a service request specification to describe the expected functionality and data qualities. These service requests are managed by the mediation layer running on the clusterhead/back-end of the sensor network, which is implemented by the Service Management Component (SMC). This entity is responsible for decoding the client input queries, selecting the appropriate service and instantiating a service composition involving specific services from a shared pool of components. The service request specification is supported at composition level, specifying how the pool is used and managed. According to the associated client input, the requirements about quality of data are expressed in a per-service instance manner along with the configuration semantics for each composition, as illustrated in Figure 3.14.b. This semantic specification supports the definition of the composition strategy in the SMC for the requested service. This parses the service parameterization, which will be defined in terms of Sensing Service Components (SSCs) and Data Processing Components (DPCs). The specialized components are deployed

in the sensor nodes, performing service related procedures and submitting these configuration related parameters to the SSC and DPC by using a client Application Programming Interface. This issue allows sharing a single instance of the components for being used across multiple service composition processes.

3.2.3.6 MORE-WSBPEL

The MORE-WSBPEL service composition approach for sensor networks and embedded devices is presented in [274]. This focuses on implementing orchestration procedures for services on top of the proposed MORE middleware, which is based on Device Profile for Web Services (DPWS) [213] and Open Services Gateway initiative (OSGi) [219]. The resulting contribution develops service integration on central servers and network gateways where Web Services are created at runtime. In addition, it allows high-level applications for accessing the service chain or performing service orchestrations based on Web Services Business Process Execution Language (WS-BPEL) [214] processing.

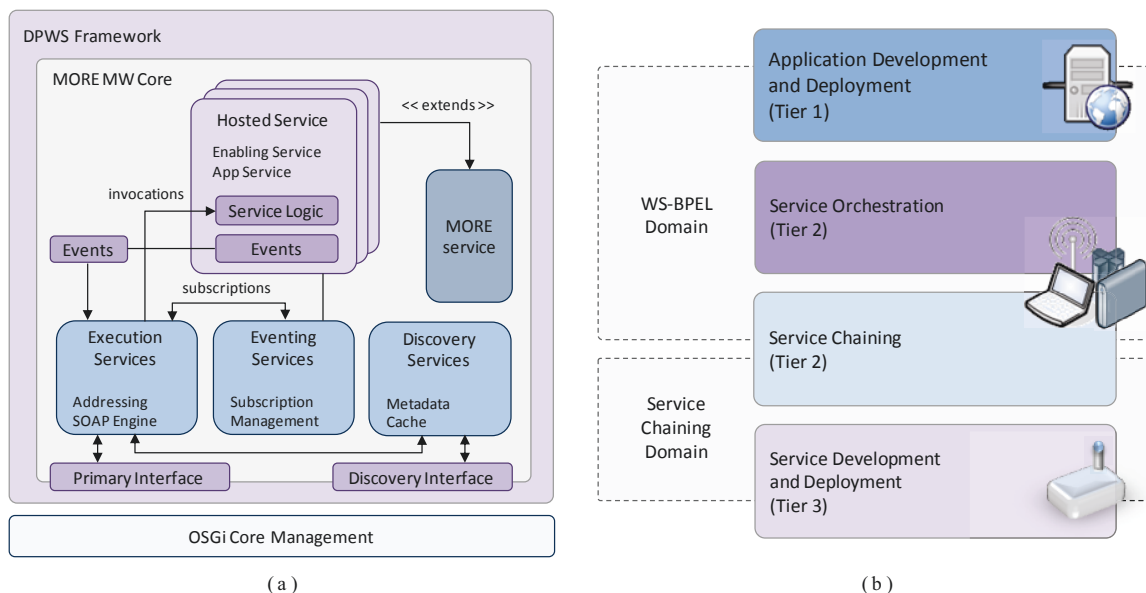


Figure 3.15: *The MORE-WSBPEL middleware (Timm et al. [274]): (a) MORE design; (b) mapping, design and deployment environment.*

As stated by its authors, the MORE platform provides a SOC-based architecture for general-purpose middleware functionalities. It includes data, group and resource management services, as well as remote service invocation based on SOAP and HTTP bindings. As illustrated in Figure 3.15.a, this contribution takes advantage of the OSGi framework for service component development and life cycle management, as well as of the DPWS approach for enabling connectivity compliant with Web Service technology but optimized for reduced functionality and embedded devices. As shown, hosted services are nested inside the MORE MW core. It allows discovery, eventing and executing support according to the publish/subscribe paradigm. As an extension of the MORE middleware, WS-BPEL orchestration has been considered in this

contribution. This solution defines a language for business processes integration, which imports the functionalities of a set of Web Services and exports them as a new composed service where the interaction model is based on Web Service Description Language interfaces. In this regard, service chains in MORE-WSBPPEL are proposed as a service composition technique for hierarchical wireless sensor and ad hoc networks, as illustrated in Figure 3.15.b. As shown, and considering a classical three-tier network deployment, the service chains provide a framework for connectivity compliant with DPWS in resource-constrained devices while full WS-BPEL processing for service integration and composition is executed on full-function devices such as gateways and network servers.

3.2.3.7 Orchestrator

The Orchestrator [151] architecture has been proposed as a service integration approach based on active resources for mobile and sensor-based context monitoring by using personal devices, such as smartphones or PDAs. This solution enables elastic service orchestration of multiple context-aware business processes in smart environments. It focuses on persistent context monitoring for supporting dynamic share of scarce embedded resources. The composition engine performs an active search looking for the best combination of physical resources according to their current status and availability in the network, as well as the identified monitoring requirements. The contribution is illustrated in Figure 3.16, including the proposed architecture for both mobile devices and sensor nodes.

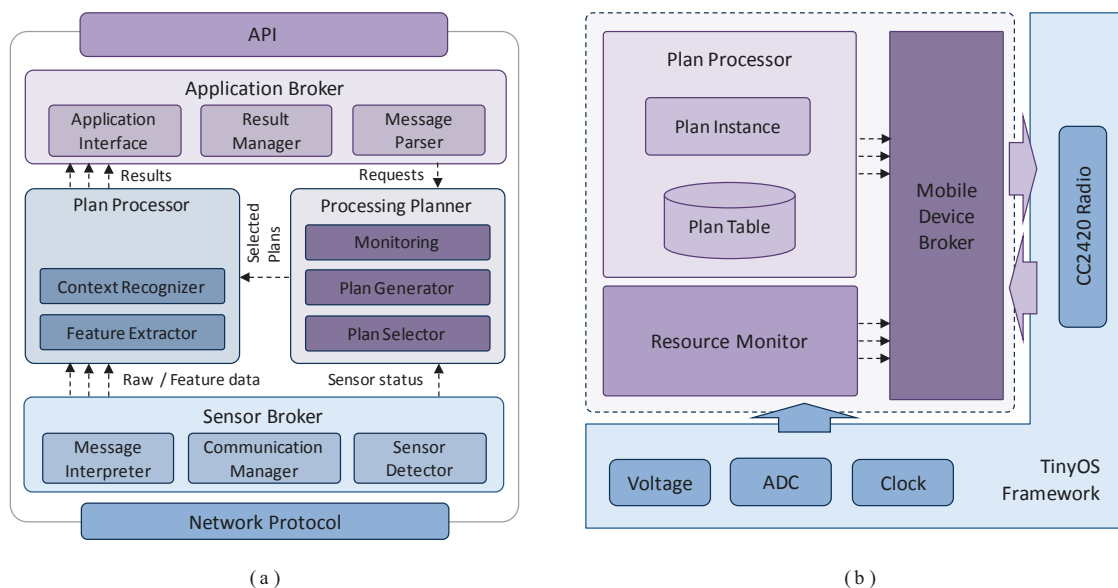


Figure 3.16: The overview of the Orchestrator service infrastructure (Kang et al. [151]): (a) architecture on a mobile device; (b) architecture on a sensor device.

According to this framework, multiple plans for resource usage are defined. Each one of these workflows is able to support the management of a high-level application context, where this can be processed depending on several sensing modalities and classification methods. This processing plan is adapted according to the resource status and availability at runtime. In this

regard, flexibility and adaptation of multiple context-aware applications that are deployed in environments with dynamic and highly limited resources is enhanced. On the one hand, and concerning the mobile device architecture, the processing planner entity implements the resource orchestration. It covers the plan generator and selector, while the plan processor entity is responsible for executing each concrete plan. The selection changes adaptively according to the dynamic nature of the hardware resources. On the other hand, the plan processor entity also executes sensing plans in the sensor nodes, while the resource monitoring entity keeps track of the status of the embedded hardware. This includes monitoring of parameters such as processing power, memory, energy and bandwidth. These are implemented on top of TinyOS [164] operating system. The effective interworking developed in the Orchestrator approach between mobile devices and sensor nodes is allowed by using communication protocols with support of automatic sensor discovery and heartbeat procedures, as well as status reporting and resource availability.

3.2.3.8 SNMAIW

With the aim of managing the workflow execution in corporate production lines, Sardis et al. devise the Sensor Networks and Multi-Agents in Industrial Workflows (SNMAIW) [249] approach. This research work is mainly focused on exploiting Multi-Agent System (MAS) architectures for controlling the composition tasks of sensor-based services in production processes. Being the main motivation of this work to address the challenges of collaborative production lines with dynamic resources and distributed knowledge in the corporate environment, its authors deal with issues such as interoperability and cooperation support between heterogeneous Wireless Sensor Networks. This leads to reach adaptive communication, effective orchestration and monitoring capability. According to the previous, the architecture of the SNMAIW proposal is illustrated in Figure 3.17.

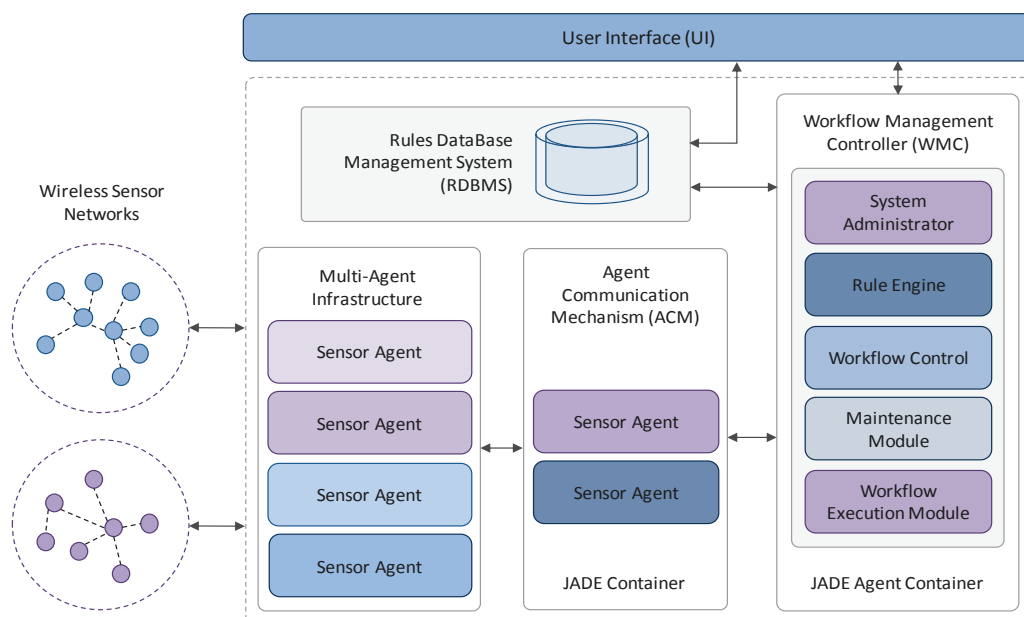


Figure 3.17: The architecture of the SNMAIW platform (Sardis et al. [249]).

This contribution defines a software infrastructure for workflow orchestration, manipulation and reconfiguration through simple end-user interfaces. It applies the Resource Description Framework standard for developing an interoperable framework for sensor information management based on semantics, where messages in agent-based workflows are modelled using the RDF/XML schema. The main software modules in the SNMAIW proposal are the following: the Workflow Management Controller (WMC), the Agent Communication Mechanism (ACM) and the User Interface (UI). On the one hand, the WMC module provides FIPA-compliant communication support for wireless sensor and ad hoc nodes by means of sensor-based agents deployed in the WSN. There are several sub-modules including system administrator, rule engine, workflow control and execution, and maintenance. Additionally, connectivity with an output processing module is supported. On the other hand, the ACM module adopts a Multi-Agent System approach. It covers those signals received from human agents using sensor identification and recognition algorithms, as well as those received from object agents using object recognition patterns and tracking algorithms. Finally, the UI module enables sensor configuration tasks and the construction of the workflow sequence.

3.2.4 Discussion

As introduced in Section 2.2.3, several novel research contributions regarding sensor networks and ubiquitous computing [31, 32, 43, 79, 277] have been analyzed as an input for our further formalization and practical application contributions. It allows us to define a set of challenges for developing future sensor-based dynamic composition platforms in service-oriented solutions. These open issues are the following:

- Service composition languages, including lightweight syntaxes for generating and processing composite services, are of interest. This comprises languages for specifying and describing service interfaces, invocation mechanisms and composition procedures.
- Adaptive service composition is required. This focuses on flexible composition methods improving scalability, robustness and resilience, where services can be integrated dynamically according to their availability in the network.
- Semantic Knowledge Management based on ontologies is considered one of the best approaches for implementing service composition. This approach provides a high, fine-grained expressiveness with better service selection, as well as improved service discovery and integration functions.
- Virtual sensor services offer a key potential in dynamic pervasive composition. This novel schema provides support for intelligent decisions that would be impossible to create with isolated physical sensors.
- Service-based integration with the Internet of Things (IoT) should be investigated. This internetworking proposal focuses on bridging the gap between sensor networks and Ambient Intelligence, with the aim of providing advanced services to external networks and clouds for building next-generation smart environments using ubiquitous computing.

TABLE 3.2: Comparative analysis of dynamic service composition architectures for Wireless Sensor Networks.

SOA approach	Service composition language	Adaptive service composition	SKM based on ontologies	Virtual sensor services	Service-based integration with the IoT
SCPQ [293]	Feature not supported	Service and provider selection based on a greedy optimal algorithm	Feature not supported	Feature not supported	Feature not supported
SDCBAN [70]	Feature not supported	Dynamic service composition using an agent-based Finite State Machine	Feature not supported	Feature supported	Interoperability with 3G mobile networks
SWSN [315]	Proprietary semantic annotations for WSDL and XML schema	Based on Semantic Web Services	Ontological processing	Feature not supported	Interoperability using WS-* specifications
DSIDMSS [175]	Proprietary language for service interfaces and composition templates	Service discovery with AEPS and proactive self-adaptability and diagnosis	Evolving Ontology	Feature not supported	Interoperability using WS-* specifications
MRSWSN [85]	Proprietary language for service request specification	Configuration semantics for resource-aware composition	Feature not supported	Feature not supported	Integration between the enterprise and the Internet domains
MORE-WSBPEL [274]	XML-based using WS-BPEL for service composition	Web Service orchestration	Feature not supported	Feature supported	Interoperability using OSGi initiative and WS-* specifications
Orchestrator [151]	Proprietary language for service request and composition specification	Holistic resource-aware plan selection and dynamic plan adaptation	Feature not supported	Feature not supported	Feature not supported
SNMAIW [249]	RDF/XML for workflow description	Agent-based workflows with rule-based reasoning	Workflow ontology model	Feature not supported	Feature not supported

According to the Table 3.2, a review of the service-oriented dynamic composition platforms previously discussed focusing on the highlighted research challenges for Wireless Sensor Networks is provided. The main conclusion concerning this analysis is that there is no one reviewed solution that fulfils all the identified requirements in a global manner. First, only the MORE-WSBPEL [274] and SNMAIW [249] contributions tackle the use of service composition languages in an open and standardized way. On the one hand, MORE-WSBPEL adopts the WS-BPEL solution for service orchestration with WSDL and XML schema while SNMAIW applies

RDF/XML for the definition of the service workflow. The rest of the discussed approaches define service composition languages based on proprietary or semi-proprietary solutions, while SCPQ [293] does not address this issue on its design. Second, all the surveyed frameworks include active service composition performing dynamic integration into pervasive sensor-based services. While some of the reviewed contributions use well-known techniques, such as web service orchestration (e.g., SWSN [315] and MORE-WSBPEL [274]), semantic composition (e.g., MRSWSN [85]) and agent-based integration approaches (e.g., SDCBAN [70] and SNMAIW [249]), other works focus on more specific-purpose methodologies (e.g., SCPQ [293], DSIDMSS [175] and Orchestrator [151]). Third, Semantic Knowledge Management based on ontologies has been explored for developing service composition frameworks in some of the previous contributions. This includes SWSN [315], DSIDMSS [175] and SNMAIW [249]. In the case of SWSN, an ontological model is applied for service annotation and processing, as well as for describing sensing data. In the case of DSIDMSS, an evolving ontology for dependability, capability and system assessment has been developed. Additionally, and with the aim of building larger processes and systems, the SNMAIW proposal investigates the notion of workflow modelling using ontologies. Fourth, virtual sensors as service-oriented abstractions for creating advanced composition schemas are explored in SDCBAN [70] and MORE-WSBPEL [274]. In both cases, virtual sensor services are proposed as the composition of simple services (i.e., atomic building blocks) in complex in-network sensor instances, with the aim of expanding the capacity of isolated physical sensors. In the case of SDCBAN it is developed with a Task Graph Database of Finite State Machines, and MORE-WSBPEL performs this by using WS-* standards. Fifth, most of the analyzed architectures address the service-based integration with the Internet of Things paradigm. In most of cases, approaches addressing this issue are based on Web Services (e.g., SWSN [315], DSIDMSS [175] and MORE-WSBPEL [274]). However, solutions involving the OSGi specification (e.g., MORE-WSBPEL) and 3G mobile communication networking (e.g., SDCBAN [70]) have been also considered.

3.3 Conclusions

In this chapter, a state-of-the-art analysis of the foundations and related work regarding service-oriented middleware architectures and pervasive service composition engineering in Wireless Sensor Networks has been performed. This issue covers from the initial research motivations to the scientific contributions surveyed in the related literature, where these challenges have been discussed and contrasted with the reviewed proposals. The rest of this section comprises a summary of the main issues previously discussed concerning these paradigms in the framework of this dissertation.

On the one hand, middleware technologies are perceived as a cornerstone of modern and advanced computing systems. These enhance the interworking between heterogeneous systems while offering support for the development of distributed applications. The resource-constrained nature of embedded sensor networks does not make suitable, in most of cases, legacy middleware frameworks, initially designed for the corporate environment. This issue led early research efforts on pervasive computing to devise new approaches, more lightweight and efficient, constituting the mainstream middleware solutions for WSN. Nevertheless, most of

these architectures where focused on addressing the complexity related to the limited resources of sensor networks, being conceived as ad hoc solutions. In this regard, the development of distributed, dynamic service platforms in pervasive smart environments represents a major challenge. With the aim of tackling this issue, the application of Service-Oriented Computing theory and practice for designing middleware abstractions for Wireless Sensor Networks is proposed and argued.

This service-oriented paradigm supports the development of rapid, evolvable and interoperable application entities. It contributes to provide non-specific design solutions, enhancing code reuse, system scalability, parallel service development, unit testing and subsystem integration phases. Based on the study of the related literature, main challenges to be tackled with service-oriented middleware architectures in Wireless Sensor Networks have been identified. These are the starting point of the formalization and practical application contributions in this research line of this dissertation. These features are the following: platform and OS independence, provisioning of general-purpose middleware backbone functionalities, support for heterogeneous multi-service composition, enable software application development and internetworking support with external service cloud and platforms. With the aim of analyzing the impact of such topics, as well as to place our contribution in the state-of-the-art research in service-oriented middleware for sensor networks, some of the most remarkable proposals in this field have been studied and contrasted with these challenges.

On the other hand, service composition comprises an integration mechanism of component service entities into higher-level application processes in a coordinated manner. This focuses on extending the functionality of simple services into added-value business services, while enabling rapid application development and service reuse. Although service integration methodologies have been explored since the annals of distributed systems, service composition remains as a natural property of ubiquitous computing systems. This is mainly based on the fact that distributed ad hoc networks are usually saturated with devices with heterogeneous resources and capabilities, where the integration of services makes possible to provide advanced services that would otherwise not be available. Additionally, ubiquitous services can evolve showing dynamic adaptability for supporting the effective use of pervasive smart spaces. In this regard, composition procedures are a key piece for building AmI ecosystems for the Internet of Things based on Wireless Sensor Networks.

The inherent dynamicity of anytime/anywhere/anymedia information access of ubiquitous networks imposes specific constraints for designing smart environments. In this regard, service delivery platforms can adapt their composition mechanisms by taking maximum advantage of the available services in the WSN according to the specific end-user request, with the aim of increasing the service availability and improving the Quality of Experience (QoE). Based on the related literature, main challenges for developing dynamic composition platforms using service-oriented solutions in Wireless Sensor Networks have been identified. These are the starting point of the formalization and practical application contributions in this research line of this dissertation. These features are the following: service composition languages for generating and processing composite services, adaptive service composition for flexible composition methods, composition driven by Semantic Knowledge Management based on ontologies,

dynamic virtual sensor services and service-based integration with the Internet of Things. With the aim of analyzing the impact of such topics as well as for placing our contribution in the state-of-the-art research regarding dynamic service composition for sensor networks, some of the most remarkable proposals in this field have been studied and discussed according to these challenges.

The main extracted conclusion from these previous surveys is that none of the reviewed solutions addresses all the previous challenges in a global manner. With the aim of contributing to this research line, next chapter includes the requirement analysis and the formalization model for our approach, considering both service-oriented middleware and dynamic service composition for pervasive smart environments based on Wireless Sensor Networks.

PART III

RESEARCH CONTRIBUTIONS
FORMALIZATION, DESIGN,
IMPLEMENTATION AND ANALYSIS

Formalization Model

DEVELOPING the future Internet of Things comprises Wireless Sensor Networks as one of the most remarkable core technologies. From its initial beginnings in the 90's and being promoted by DARPA-funded projects in top US universities to create autonomous sensor-based computing devices, there are currently remarkable worldwide research efforts and initiatives around this novel computing paradigm. However, further steps to materialize this promised vision are required. In this regard, the definition of standardized models based on formalization proposals for achieving explicit representations of how pervasive technology can be formulated and exploited plays a key role.

This chapter proposes our formalization model for defining dynamic service platforms based on pervasive sensor networking in the context of this dissertation. Section 4.1 points out the problem statement, which is addressed in our research. Section 4.2 presents the formalization model for the service-oriented framework, while the formalization model for the dynamic virtual sensor-based service composition schema is defined in Section 4.3. Finally, main conclusions of this chapter are summarized in Section 4.4.

4.1 Problem Statement

Pervasive smart spaces and environments are conceived as an overlay approach for enabling end-user experiences and interactions in an improved manner, enhancing the regions of the physical world where they are implemented. The development of these next-generation digital ecosystems includes several challenges [42, 87, 311]. As presented in Section 2.5.3, these open issues cover from interoperability of heterogeneous entities to spontaneous communications and environmental dynamicity including ease of application development, support for disparate application requirements and user-centric service provisioning, among others. In this regard, the

particular nature of ubiquitous infrastructures has significant open issues for the development of embedded architectures in reduced functionality devices, such as wireless sensor and ad hoc networks. Most of smart environments are not readily supported by current wireless sensor systems [171], where more sophisticated applications and services should be provided in a global manner as a critical issue [42].

The previous discussion points out the problem statement to be addressed with the formalization and practical application contributions of this dissertation. In this regard, and as defined in Section 1.2, our approach is focused on the research and development of distributed platforms for dynamic provisioning of virtual sensor services in pervasive smart environments, while integrating sensor networks into the Internet cloud for the Information and Networked Society. To this end, several specific design challenges to be covered with future wireless ad hoc and sensor networks were introduced in Section 2.2.3. These identified requirements comprise two main knowledge areas in computer science. These are service-oriented middleware and dynamic service composition in pervasive computing. On the one hand, future work regarding service-oriented middleware engineering in sensor networks should address platform and OS independence, provisioning for general-purpose middleware backbone functionalities, support for heterogeneous multi-service composition, enabling software application development and internetworking support with external service cloud and platforms. On the other hand, and concerning dynamic service composition architectures, issues such as lightweight service composition languages, adaptive service composition, semantically-driven service integration based on ontologies, virtual sensor services and service-based integration with the Internet of Things should be considered in further developments for WSNs. These issues were discussed in detail in Sections 3.1.5 and 3.2.4, respectively. Additionally, these were contrasted with some of the most remarkable state-of-the-art approaches in this field in Tables 3.1 and 3.2, respectively. As main conclusion of these conducted analyses, and in our best knowledge, most existing SOA approaches based on sensor network technology for service provisioning do not address the design of heterogeneous pervasive smart ecosystems in a global way [98, 99]. To accomplish this issue, several emerging research approaches have been considered for pervasive computing with sensor networks in our contribution. This includes Service-Oriented Computing, Agent-Oriented Software Engineering, Semantic Knowledge Management and the Internet of Things. These novel paradigms in ad hoc and sensor networks are presented in Chapter 2, where their application in our research work is discussed and justified in detail.

Next sections contribute on defining a conceptual framework in terms of service-oriented architecting and dynamic composition for pervasive networks based on distributed and sensor-based embedded computing. To this end, a formalization model is proposed. This will support the formulation of the theoretical basis for achieving completeness, consistency and unambiguity in both architecting and verification phases regarding the software engineering process. This approach consists of two main areas. On the one hand, an abstract formalization of a reference service-oriented framework is proposed. This will support the definition of a common abstraction to design and implement Service Oriented Architectures for addressing pervasive smart spaces and environments based on sensor technology. On the other hand, an abstract formalization of a virtual sensor service composition approach is provided. This model

specifies a composition-centric paradigm for building sensor-based pervasive in-network services. This solution is implemented in the proposed service-oriented framework, developing dynamic and adaptive service composition using ubiquitous computing.

4.2 Formalization of the Service-Oriented Framework

As previously introduced, this contribution defines a reference model for developing Service-Oriented Architectures over reduced functionality and sensor-based devices focusing on the design requirements summarized in previous sections. As a reference framework, a common model for service-oriented computation in heterogeneous smart environments is provided, extending our previous research work [98]. This proposal, which is illustrated in Figure 4.1 by using Unified Modelling Language (UML) notation, determines how the practical software architecture is structured and which functional aspects have to be defined in each subsystem, as well as the explicit interrelations that have to be fulfilled between them. As shown, this approach includes five abstraction levels that are represented as UML subsystems, and which have been called platforms in this proposal. These are the following: Physical Device Platform, Service-Oriented Middleware Platform, Pervasive Service Platform, Service Internetworking Platform and Development Support Platform. Their descriptions and motivations are presented in the next sections.

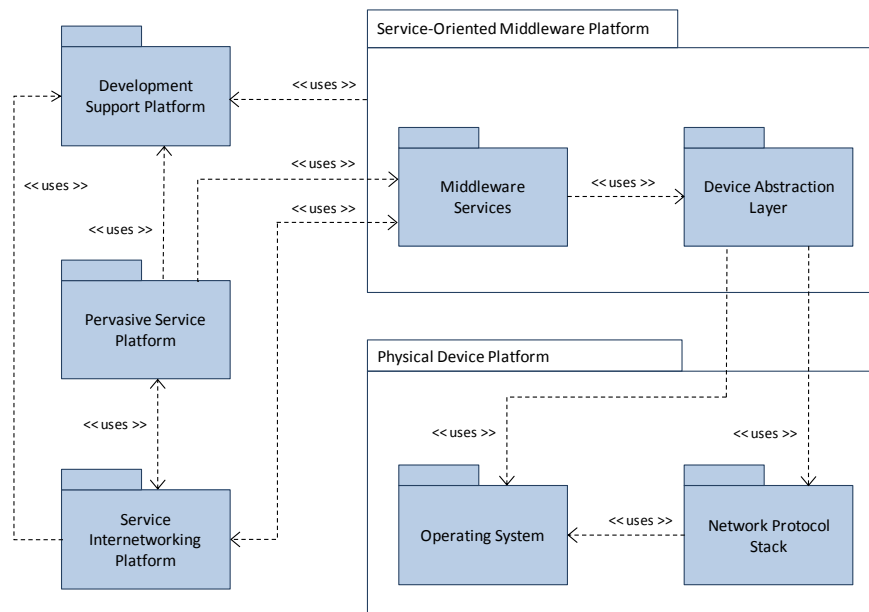


Figure 4.1: UML Package Diagram of the reference service-oriented framework.

4.2.1 Physical Device Platform

This horizontal abstraction layer is a virtual conceptualization of the physical hardware device and the underlying language of the specific platform. It includes the hardware support and the required, minimum low-level software for deploying the rest of the architecture. Therefore, it encapsulates the hardware device, the operating system and the networking protocol stack.

The main motivation of this platform is to provide transparency and independence of the peculiarities of the operating system and the routing protocol stack to the higher levels of the Service Oriented Architecture.

4.2.2 Service-Oriented Middleware Platform

This horizontal abstraction layer comprises the middleware service infrastructure and the Device Abstraction Layer (DAL). Middleware support encompasses procedures for managing the life cycle of the components and the interaction between them. In addition, it provides the required set of supplementary core services for the pervasive application processes. As shown, the middleware is located above the DAL abstraction, which provides independence from the specific hardware platform. The main motivation of this platform is to define a generic and reusable Service-Oriented Middleware loosely coupled with the physical equipment and strongly focused on the application process development.

4.2.3 Pervasive Service Platform

This horizontal abstraction layer identifies service entities as lightweight business processes with well-defined functionality based on the identification and encapsulation of the system properties into separate unit blocks. These application instances are modelled as pervasive in-network services using aggregation mechanisms for developing composed virtual sensor services based on the service-oriented middleware support. The main motivation of this platform is to enable the deployment of pervasive services in embedded devices optimizing the use of physical resources, also enabling the provisioning of complex sensor-oriented virtual services based on composition schemas.

4.2.4 Service Internetworking Platform

This orthogonal abstraction layer is responsible for developing a composition-centric model for the service-oriented framework. On the one hand, this includes the required mechanisms for implementing service integration of pervasive in-network services in the Pervasive Service Platform into aggregated services, in an adaptive and flexible manner. On the other hand, procedures for addressing communication and internetworking with external data networks and cloud infrastructures are also supported, for both service publications and invocations. The main motivation of this platform is to implement the required support for dynamic service composition, as well as integrating the pervasive device into the Internet of Things.

4.2.5 Development Support Platform

This orthogonal abstraction layer incorporates interface models, specific development tools and environments, as well as collaborative initiatives for promoting the programming of middleware and application services in embedded devices. The main motivation of this platform is to ease the implementation tasks in and around this framework. This includes aspects such as code reuse and sharing efforts between programmers for enhancing the development of open, fast, low-cost, evolvable and massively distributed ubiquitous services.

4.3 Formalization of the Service Composition Model

As stated before, this contribution focuses on describing a service composition model for virtual sensors in the service-oriented framework modelled in the previous section. It is based on the foundations of OWL-S (Web Ontology Language for Services) [290] in terms of service profile, process model and service grounding definition. This specification accomplishes an ontology approach for Web Services, which enables automation of business processes in terms of discovery, execution, composition and interoperation. Our service model approach addresses some main design requirements for composition-centric pervasive computing on the Internet of Things in an optimized and lightweight manner, as identified in previous sections. In addition, this approach represents the reference formalization of our practical research contribution [99] in this area.

4.3.1 Conceptualization of a Pervasive In-Network Service

Our contribution defines a compact set of objects and statements to model ubiquitous service computing in the proposed service-oriented framework. In this regard, a pervasive in-network service can be formulated as an abstract resource $PS_i \equiv L_{PS_i} \{SR(O_i^{out})\} : f(O_i^{in}, SPre_i, AInf_i)$. It comprises a discrete business logic L_{PS_i} to get a service result SR based on a finite set of output operations $O_i^{out}, \forall i \in (1, n)$ that is a function f of a finite set of input operations $O_i^{in}, \forall i \in (1, n)$, a finite set of service preconditions $SPre_i, \forall i \in (1, n)$ and specific access information $AInf_i, \forall i \in (1, n)$. This transition function f is implemented in terms of L_{PS_i} that constitutes the service workflow across a finite set of states $S = \{S_1, S_2, S_3 \dots S_{n-1}, S_n\}$ according to a state-transition approach. Both PS_i input and output finite set of operations are represented in terms of a 5-tuple $O_i = \{Pa_i^{in}, Pa_i^{out}, S_i, S_{i+1}, OPre_i\}$, where both $Pa_i^{in}, \forall i \in (1, n)$ and $Pa_i^{out}, \forall i \in (1, n)$ represent the finite set of concrete input and output operation parameters, respectively; $S_i, \forall i \in (1, n) \in S$ and $S_{i+1}, \forall i \in (1, n) \in S$ denote the initial and final states for the operation, and $OPre_i, \forall i \in (1, n)$ is the optional finite set of operation preconditions that has to be satisfied for the invocation of the service. Thus, the service profile of a pervasive network service PS_i is defined in terms of a programmatic interface $I_i = \{O_i^{in}, O_i^{out}, SPre_i, SR\}$ composed by a finite set of input operations, a finite set of output operations, an optional finite set of service preconditions and the final service result of the execution workflow. Additionally, the service grounding of a pervasive in-network service instance PS_i defines the specific information regarding how to access and to invoke a concrete service realization according to its service unique identifier, as well as the device and application endpoint references for locating it.

4.3.2 Composition Approach for Pervasive Virtual Sensor Services

Extending the previous approach, a pervasive virtual sensor service can be represented as an abstract resource $PVSS_i \equiv L_{PVSS_i} \cdot \sum_{i=1}^n PS_i$. This is modelled as a discrete business logic L_{PVSS_i} implemented for solving a computational problem based on the orchestration of a finite set of in-network pervasive services $PS_i, \forall i \in (1, n), |i| \geq 2$. Being a concrete service instance $PVSS_i$ of a

state-transition system, our service approach for describing the process model is formulated as a graph $CPro_i = (APro, Exp)$, where $APro$ is the set of graph vertex and Exp is the set of graph edges. On the one hand, $APro_i, \forall i \in (1, n)$ represents the finite set of atomic processes that integrates the composite process, being defined by a finite set of programmatic interfaces I_i . On the other hand, $Exp_i, \forall i \in (1, n)$ represents the finite set of execution paths between each pair of atomic processes in the composition. This is described as the union between the finite sets of input and output operations $O^{in, out}$ and the interaction protocols $P^{in, out}$ of the atomic processes. According to it, two concrete atomic services $\{APro_i, APro_{i+1}\}$ of two concrete pervasive in-network services $\{PS_i, PS_{i+1}\}$ can be integrated into a pervasive virtual sensor service $PVSS_i$ if and only if there is an execution path $Exp_i = \{O_i^{out} \times P_i^{out}, O_{i+1}^{in} \times P_{i+1}^{in}\}$ that can be established between them. Considering a transition function f' defined by $L_{P_{CS_i}}$, if $\exists Exp_i \{PS_i, PS_{i+1}\} \rightarrow \exists f': (O_i^{out} \times P_i^{out}) \in PS_i, (O_{i+1}^{in} \times P_{i+1}^{in}) \in PS_{i+1}, O_i^{out} = O_{i+1}^{in}$, and P_i^{out} corresponds P_{i+1}^{in} .

4.3.3 Service Model of the Service-Oriented Framework

Based on the previous concepts of pervasive in-network service PS_i and pervasive virtual sensor service $PVSS_i$, the service approach of the service-oriented framework presented in Section 4.2 is provided. In this regard, the Pervasive Service Platform can be modelled as an abstract resource $PSP \equiv \langle L_{PSP} \cdot (\sum_{i=1}^n PS_i + \sum_{i=1}^m PVSS_i) \mid Env(t), \{I_{SOMP}, I_{SIP}, I_{DSP}\} \rangle$. This defines a discrete business logic L_{PSP} for managing a finite set of pervasive in-network services $PS_i, \forall i \in (1, n)$ and a finite set of pervasive virtual sensor services $PVSS_j, \forall j \in (1, m)$ in a specific environment Env and in a given instant of time t with three interfaces I_{SOMP}, I_{SIP} and I_{DSP} to the rest of the architecture. On the one hand, this discrete business logic $L_{PSP} \rightarrow \sum_{i=1}^{n+m} SC_i$ implies a finite set of service containers $SC_i, \forall i \in (1, n+m)$ that represents the concrete application model of the execution environments used by the service instances, for both pervasive in-network and virtual sensor services, where specific scheduling and execution policies are defined for each concrete service realization. On the other hand, the interfaces considered in the Pervasive Service Platform PSP are the following. The I_{SOMP} interface provides general-purpose core functionalities implemented in the Service-Oriented Middleware Platform using the service containers SC_i . The I_{SIP} interface defines service integration into the Internet cloud, supporting both service advertisement and service invocation through the Service Internetworking Platform. Finally, there is the I_{DSP} interface. It enables procedures and mechanisms for the design, implementation, testing and deployment of service entities based on the Development Support Platform.

4.4 Conclusions

In this chapter, our formalization contribution for dynamic service platforms based on ubiquitous sensor computing has been presented. As previously stated, it represents the theoretical foundations of the related practical application contributions of our work, in terms of service-oriented and composition-centric architecture, which are discussed in the next chapters. In this regard, the proposed theoretical model has been defined according to the

engineering challenges to be addressed for building pervasive smart spaces and environments with dynamic composition frameworks in service-oriented solutions with Wireless Sensor Networks. The rest of this section comprises a summary of the main issues regarding these aspects in the framework of this dissertation.

As discussed, one of the major open issues in the definition of the problem statement for the development of pervasive service delivery platforms lies on the inherent heterogeneity and dynamicity of both user personal devices and communication networks for supporting anytime/anywhere/anymedia interactions. These aspects highlight the need for advanced sensor-based computing and networking solutions with the aim of building effective pervasive smart infrastructures. According to the previous, the review of the related literature and the state-of-the-art contributions in this field led us to identify a set of key design challenges for service-oriented computing in pervasive and sensor networks, which were discussed and justified in the previous chapter. These requirements have been used, in the current chapter, as an input for the formal modelling of both service-oriented middleware abstractions and dynamic composition-centric service approaches. The main objective of the above is to address the modelling and development of practical solutions for building ubiquitous smart ecosystems in a global manner. As the main outcome, the practical implementation contributions of the discussed formal models are in the following chapters.

On the one hand, a formalization approach regarding a reference service-oriented framework for ubiquitous networks based on heterogeneous sensor devices and embedded computing has been proposed. It defines a common abstraction to design and implement Service Oriented Architectures for pervasive smart infrastructures covering from the hardware and low-level software to the integration and service composition, also including support for the software development phase. This reference model defines five abstraction levels or platforms, which are the following: Physical Device Platform, Service-Oriented Middleware Platform, Pervasive Service Platform, Service Internetworking Platform, and Development Support Platform. First, the Physical Device Platform provides transparency and independence of the specific peculiarities of the operating system and routing protocol to the higher levels of the architecture. Second, the Service-Oriented Middleware Platform defines a generic and reusable service-oriented middleware that focuses on the applications and business process development. Third, the Pervasive Service Platform enables pervasive in-network services based on the identification and encapsulation of the system properties into separate unit blocks, also providing support for implementing composed virtual sensor services. Fourth, the Service Internetworking Platform develops the service composition-centric model of the reference framework, which enables dynamic service composition while supporting the integration of the pervasive network into external service clouds. Finally, the Development Support Platform contains procedures and tools for enhancing the development of open, fast, low-cost, evolvable and massively distributed services, thus minimizing the required technical skills in embedded software programming and promoting the development by third parties.

On the other hand, a formalization approach of a virtual sensor composition-centric model for pervasive networks comprising sensor-based and embedded computing has been proposed. It represents the integration paradigm that has been provided for developing pervasive virtual

sensor services in the reference service-oriented framework. This approach extends the semantics of the OWL-S specification, with the aim to define and compose lightweight pervasive sensor-based services in a compliant manner with its foundations in terms of service profile and process model. First, and regarding the service profile, the formal notion of pervasive in-network service PS_i has been proposed. This includes the definition of its capabilities in terms of discrete business logic, transition function, expected result, input and output operations, and service preconditions. Second, and concerning the process model, the formal definition of pervasive virtual sensor service $PVSS_i$ has been also provided. This notion includes the formulation of a service composition-centric approach according to a directed graph, representing both the set of related atomic processes and the execution paths between them defined by the transition function. Third, service access information is also considered in this service modelling approach. It makes sense for the service grounding in terms of the service identifier, as well as the device and application endpoint references. Finally, the formal definition of the Pervasive Service Platform PSP of the proposed reference service-oriented framework has been performed. This comprises its modelling in terms of discrete business logic, set of pervasive in-network services and pervasive virtual sensor services, the concrete execution environment in a given time and the set of interfaces with the adjacent platforms of the service-oriented framework.

nSOM: A Service-Oriented Middleware for Pervasive Smart Environments

*R*ESearch opportunities concerning middleware architecting have emerged during last years with the rapid evolution of personal and mobile computing, enhancing the technical and economical feasibility of the vision promised by the pervasive paradigm. This approach highlights the need for combining heterogeneous information sources, devices and technologies. Although almost all of the necessary hardware and software components already exist, their seamless and effective integration remains as a cutting-edge innovation point. In this regard, middleware abstractions are perceived as a cornerstone to address the previous challenge in environments that are highly sensitive, adaptive and responsible to human needs.

This chapter presents our approach of service-oriented middleware implementation for pervasive smart spaces and environments based on sensor technology, which is called the nSOM (nano Service-Oriented Middleware) architecture. Section 5.1 overviews the proposed contribution. Section 5.2 presents its architectural modelling. Section 5.3 discusses the service workflow of this middleware for both inter-node and intra-node communications. Finally, main conclusions of this chapter are summarized in Section 5.4.

5.1 Overview of the nSOM Approach

This proposal represents the practical realization of the reference service-oriented framework formally modelled in Section 4.2, in order to build a distributed service delivery platform for dynamic provisioning of virtual sensor services in pervasive environments. Considering as design requirements those discussed in Chapters 3 and 4, this approach addresses platform and OS independence, support for general-purpose core functionalities, heterogeneous multi-service

composition, enhanced software application development and internetworking support with external service clouds and platforms. Additionally, semantically-driven adaptive composition and processing capabilities of services are also provided based on a contribution called nSOL (nano Semantics-Oriented Language). This represents the practical implementation of the formal service composition model presented in Section 4.3, which is discussed in Chapter 6.

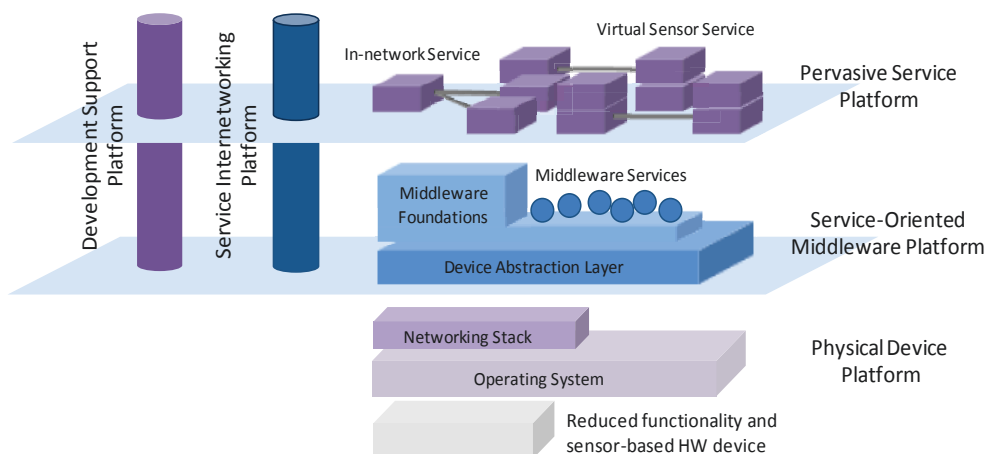


Figure 5.1: Overview of the nSOM architecture proposal.

As a result, the nSOM architecture comprises specific implementation details regarding the generic, abstract service-oriented model presented in the previous chapter. This is illustrated in Figure 5.1. Main research foundations of this SOA approach are presented in [99], including the definition of the Service-Oriented Middleware Platform, Pervasive Service Platform, Service Internetworking Platform and Development Support Platform.

The overview of this practical implementation is as follows. The Physical Device Platform encapsulates the hardware device, operating system and network protocol. This includes a broad variety of embedded networking equipment embracing personal and mobile handsets, smartphones, radio frequency and near-field communication tags, wearable computing and sensor networks, among others. It also covers specific operating systems and networking protocol stacks, specifically designed for those devices. In our implementation, Wireless Sensor Networks define the hardware platform imposing specific low-level hardware and software details. It is worth noting, however, that the related abstraction services provided by the Service-Oriented Middleware Platform enable hardware, operating system and network protocol heterogeneity by implementing a Device Abstraction Layer (DAL). This Service-Oriented Middleware Platform also defines a set of general-purpose core functionalities, which are part of the middleware service foundations. These include control services, low-level services, high-level services and cross services. Furthermore, it implements context-based intra-node communication mechanisms between software components and application entities, as well as inter-node communication procedures. These middleware core services are provided to the Pervasive Service Platform. This develops an agent-oriented software engineering approach for pervasive in-network services based on aspect-oriented programming, also defining an

adaptive virtual sensor composition approach based on semantic knowledge management. This composition-centric model for pervasive in-network services in the Pervasive Service Platform is performed interacting with the orchestration facilities built in the Service Internetworking Platform, which also provides brokering support for internetworking with external clouds and data networks. Finally, and regarding the Development Support Platform, specific aspects for enhancing software engineering tasks in embedded devices have been also addressed in our contribution. This includes the definition of contract interfaces for both agent-based services and middleware components. It provides the required programmatic structure for their implementation while defining functionalities in the middleware for managing their scheduling. This interface model has been integrated into an open-source framework considered for implementing, testing and installing the resulting software in the physical sensor nodes. In this regard, an Integrated Development Environment (IDE) with support for source code versioning control in a Linux-based virtual environment has been used in our work.

5.2 Middleware Architecture

The foundations of the Service-Oriented Middleware Platform in the nSOM contribution have been defined to address the design requirements previously justified, according to our previous research works in this area [98, 186]. Based on the abstract formalization approach provided in Section 4.2, a concrete formal model according to a UML Component Diagram of the Service-Oriented Middleware Platform is proposed. This is illustrated in Figure 5.2. The description of the abstraction layers and software components in the service-oriented middleware architecture is presented and discussed in the following sections.

5.2.1 Control Services Layer

The Control Services Layer represents the middleware backbone of this proposal. This enables the deployment of the components and their life cycle management, also allowing interaction and communication support between them based on data events. These functionalities are provided by the Container and Event Engine middleware components.

The Container component is the framework of the architecture. This schedules the execution of the middleware components based on the *IComponentManagement* interface, overwriting the methods defined in the *IComponent* interface. It is implemented by all the components for defining programmatic procedures to load, run, stop and destroy, as well as callback invocations based on a receptacle procedure. This interface also enables a context-aware mechanism for dynamic discovery and access of the components in the proposed framework. The Event Engine component implements the core message handler in the middleware. It defines a low-coupling, inter-component data flow mechanism based on a context-aware, event-driven architecture. In this regard, the middleware components use the *IEvent* interface, which interacts with the Container component using the *IDynamicContext* interface. This allows accessing the rest of the components in the framework and handling the generated events in a dynamic way. This component provides to the applications both a publish/subscribe mechanism and a source-based delivery procedure, according to the direction information of the potential consumer provided in the control part of the context-aware event.

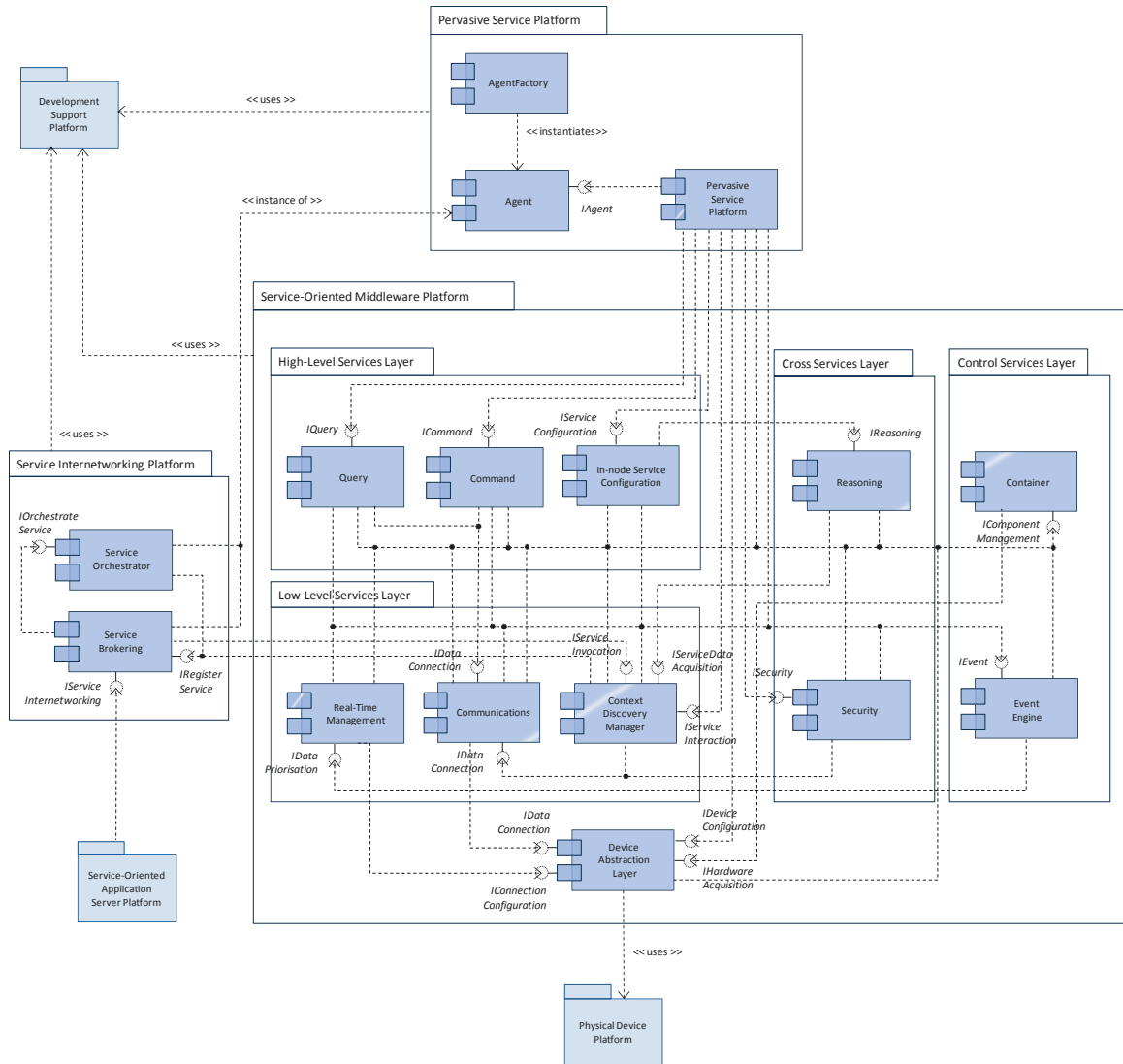


Figure 5.2: UML Component Diagram of the Service-Oriented Middleware Platform.

5.2.2 Cross Services Layer

The Cross Services Layer defines orthogonal services for the rest of the middleware software architecture for both low and high service levels, but also for the pervasive in-network services. In this regard, it implements cross-layer core functionalities. These are provided by the Reasoning and Security middleware components.

The Reasoning component defines and manages specific rules for automatic status and service inference in the local device node. These are dynamically instantiated according to the published service descriptions and capabilities, which are accessible through the *IServiceDataAcquisition* interface at the Low-Level Services Layer. These reasoning rules are offered in the middleware through the *IReasoning* interface providing dynamic information about the device setting-up in terms of activated services and functionalities. The Security component focuses on protecting and assuring privacy in service communications. This defines the *ISecurity* interface to provide data confidentiality based on cryptographic techniques applied to the data events generated in the

Pervasive Service Platform and propagated over the wireless network. The proposed security model defines a key exchange mechanism for authenticating the network devices with key distribution centres, as well as to distribute the cipher keys among them. The dialog for performing the key distribution process, as well as the transmission of the ciphered service agent events, is performed using the *IDataConnection* interface of the communication service.

5.2.3 Low-Level Services Layer

The Low-Level Services Layer defines a set of services strictly related to the Physical Device Platform, which can be intensively used during the architecture operation while enabling the core basis for developing high-level middleware functionalities. These are provided by the Real-Time Management, Communication and Context Discovery Management middleware components.

The Real-Time Management component focuses on enabling configuration support for the requirements of simple services based on the real-time capabilities of the operating system. This provides an *IDataPriorisation* interface to the Control Services Layer for defining the priority of the data events. This component also interacts with the Device Abstraction Layer for configuring the QoS parameters in the scheduler in case of provisioning of real-time capabilities by the operating system. The Communication component manages the networking protocol stack based on the support provided by the Device Abstraction Layer. This enables the *IDataConnection* interface, which is a uniform abstraction defining a set of primitives for implementing connection and connectionless communication procedures including data encapsulation, transmission and routing. The Context Discovery Management component implements middleware operation management for in-network service interactions, as well as context-aware service support based on semantics. In this regard, it interacts with the Service Internetworking Platform for processing service discovery advertisements and invocation messages, for both incoming and outgoing service data flows, using the *IServiceInteraction* interface. In addition, inferred service-based semantic annotations and relations based on reasoning procedures are provided through the *IServiceDataAcquisition* interface.

5.2.4 High-Level Services Layer

The High-Level Services Layer implements a programming support abstraction based on the middleware core, which is accessible to application business process entities in order to develop adaptable and heterogeneous domain services. This functionality is provided by the Query, Command and In-node Service Configuration middleware components.

The Query component provides support for executing specific information queries, enabling data gathering and configuration of periodic and historical measurements. This capability can be accessed from the pervasive in-network services in the Pervasive Service Platform via the *IQuery* interface. It enables a fine-grained, flexible mechanism for data acquisition based on specific query assertions to program the sensor devices. The Command component allows administration and operation invocation in scattered devices, thus providing inter-service support based on the execution of remote software procedures. This service provides an *ICommand* interface, which is accessed from the pervasive in-network services through the Pervasive Service Platform and

where the invocation of the remote operation is specified in terms of context-aware service information entities. Finally, the In-node Service Configuration component focuses on providing dynamic configuration and management of the service instances and device functionalities. This defines the *IServiceContiguration* interface to the Pervasive Service Platform, with the aim to provide adaptive behaviour capability to the business processes in their execution, using as an input the reasoning rules from the Cross Services Layer.

5.3 Middleware Workflow

Once the foundations of the nSOM architecture have been described, the proposed workflow is presented. This focuses on how the middleware components and the in-network agents interact to perform the service provisioning. It focuses on attending both the intra-node and inter-node interworking models, which make sense of the proposal in terms of local service computing and service provisioning to the wireless cloud, respectively. These points are covered in the following sections.

5.3.1 Intra-node Model

As illustrated in Figure 5.2, the Application Layer in this approach is represented by the Pervasive Service Platform, which develops the service model formalized in Section 4.3.3. In this regard, this abstraction comprises the discrete business logic for integrating a set of pervasive in-network services in the device as well as object references of the virtual sensor services in the network, in a specific environment and time instant, with three specific interfaces provided to rest of the architecture (i.e., Service-Oriented Middleware Platform, Service Internetworking Platform and Development Support Platform).

The Pervasive Service Platform is perceived as an agent-oriented container abstraction to deploy and execute pervasive in-network services based on aspect-oriented programming, being the access point to the rest of the Service Oriented Architecture. These develop the formal service model defined in Section 4.3.1 where a set of end-user, on-the-fly agent instances implementing the *IAgent* interface are defined. The use of this interface definition for developing agent-based services in the Pervasive Service Platform is twofold. On the one hand, a lightweight and uniform mechanism to schedule and execute the software agents is provided, as well as a callback receptacle for service interactions. On the other hand, it offers a high-level abstraction for pervasive in-network services based on agent technology for embedded wireless devices, where developers should just focus on the business logic of the service instead of the programmatic details related to the hardware platform. The API defined in the *IAgent* interface includes five abstract methods, which are the following: load, run, stop, destroy and receptacle. The load method allows the deployment of the agent instance in the Pervasive Service Platform. It receives as an input parameter of the invocation a `ContextElement_t` object reference, which is used to discover the middleware components at runtime. The run, stop and destroy methods do not receive any invocation parameter. These are responsible to start the agent execution and register its service description in the network, to stop the execution of the agent and unregister it, and to free all the resources assigned to the service in the Pervasive Service Platform, respectively. Finally, there is the receptacle method, which enables a callback

procedure for service invocations. It receives a `ServiceContext_t` object reference in the method call, where the service operation and parameters as well as the service result are encapsulated. This data structure also defines a control part where the direction information of the event producer is provided, and where the information of the potential consumer can also be included to perform source-based delivery. According to the previous, Figure 5.3 illustrates the UML Sequence Diagram of the proposed intra-node communication workflow in an event-driven, multi-agent interaction.

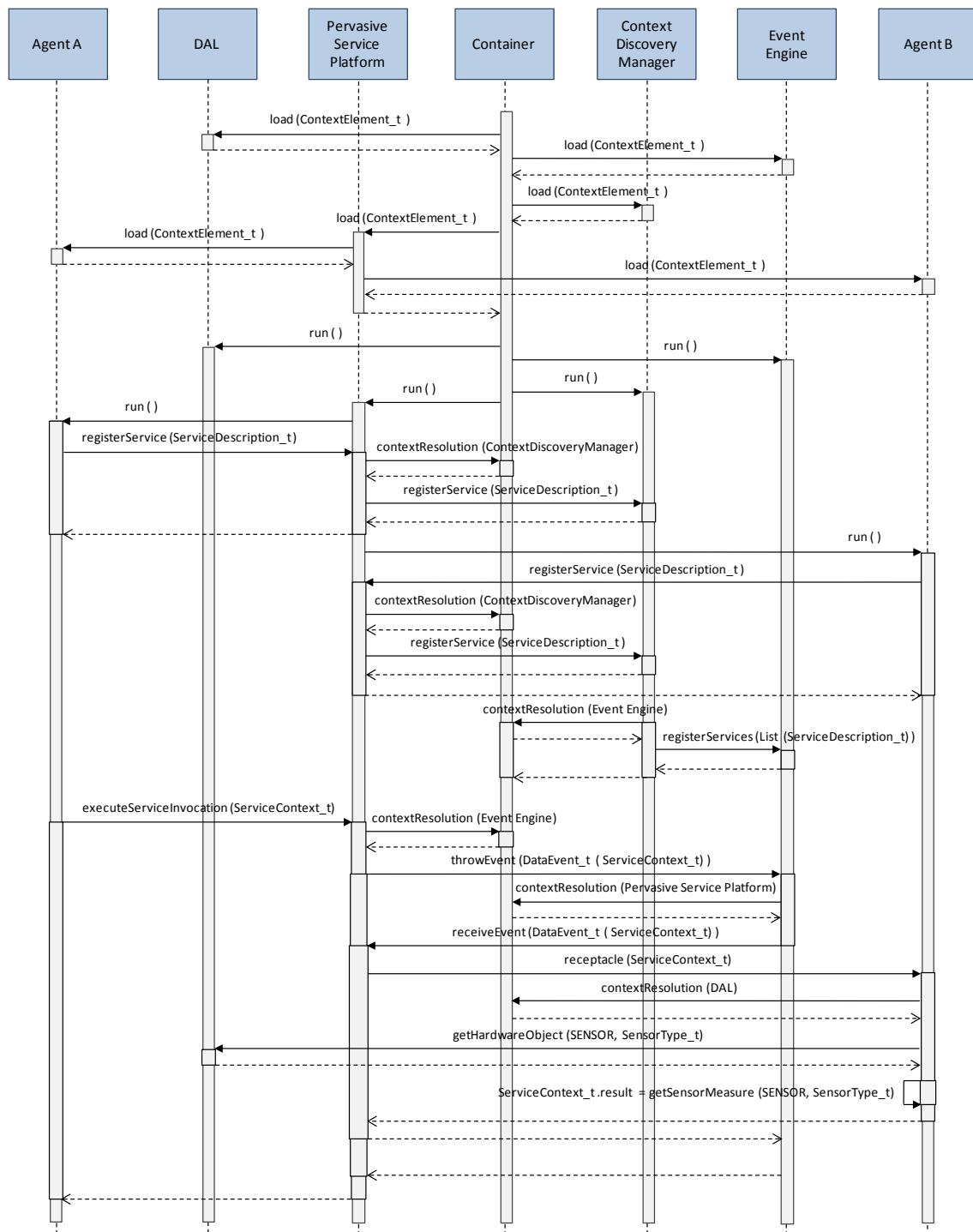


Figure 5.3: UML Sequence Diagram of the intra-node communication workflow.

As shown, the intra-node middleware workflow is initiated with the load process. According to this, all the middleware components are instantiated, configured and deployed in the architecture as well as the user agents are installed in the Pervasive Service Platform component. In the invocation of the load method implemented in the components and agents, a `ContextElement_t` element is provided. This context-aware data structure is referenced by the Container component at runtime, in order to provide on-the-fly discovery and access capabilities for the rest of components and user agents. Once the load process is finished, the Container component performs the invocation of the run method of all the components to start their execution. In the case of the Pervasive Service Platform component, this recursively triggers the invocation of the run method of all the instantiated agents. The run method of the user agents performs the registration process of the pervasive in-network services based on `ServiceDescription_t` data structures. In this way, these announcements are published to the Context Discovery Manager component, which periodically propagates the list of new `ServiceDescription_t` object references to the Event Engine component to support the publish/subscribe interworking service. At this point, the agent-based service interaction in the intra-node communication model is enabled. This is initiated when an agent invokes the `executeServiceInvocation` method of the Pervasive Service Platform component with a `ServiceContext_t` data structure as an input parameter. This context-aware object reference includes a control part and a data part. On the one hand, the control part defines the direction information of the origin agent and potential target agent (in the case of using source-based delivery), both the software agent and device identifiers. On the other hand, the data part specifies the operation, input and output parameters, preconditions and service result. The `ServiceContext_t` object generated by the origin agent is delivered to the Event Engine component, which invokes the receptacle of the target agent using the control part of the data structure. The target agent executes the service according to the information provided in the data part of the `ServiceContext_t` reference, incorporating into the result field of the data structure the service outcome (e.g., a simple sensor measure using the support of the Device Abstraction Layer component). Based on this callback mechanism, the result of the pervasive in-network service execution is received back in the origin user agent.

5.3.2 Inter-node Model

The core subsystem for implementing the inter-node communication paradigm in our approach is the Service Internetworking Platform. As presented in Section 4.2.4, this provides the composition-centric model in the nSOM architecture. This focuses on two of the basic requirements for service-oriented middleware frameworks as shown in Table 3.1, which are the multi-service composition support and the internetworking with external service platforms. To this end, the Service Internetworking Platform defines two network entities: the Orchestrator and the Broker. On the one hand, the Orchestrator executes the service composition procedure based on the available services in the sensor devices of the wireless network and the defined composition algorithm for pervasive in-network services. On the other hand, the Broker entity constitutes the interface between the wireless ad hoc domain and the external networks for exposing service descriptions to the Internet cloud via a service-oriented Application Server Platform (see Section 6.3.3 for further information) while providing support for service invocation. The design of the Orchestrator and Broker entities takes advantage of the MAS

approach provided by the nSOM architecture, being implemented as software agents in top of the Pervasive Service Platform in dedicated wireless sensor nodes. According to the previous, the network architecture involving these networking entities is illustrated in Figure 5.4.

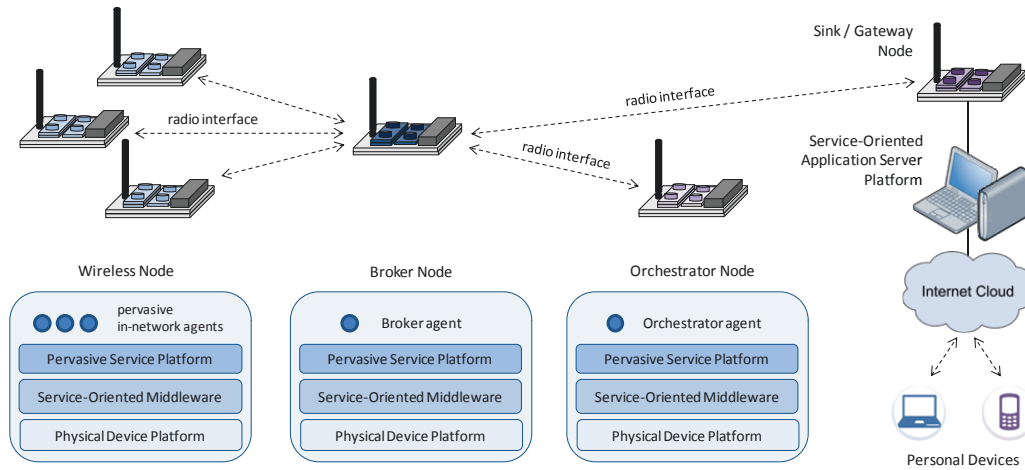


Figure 5.4: Inter-node internetworking model: proposed network architecture.

The proposed inter-node communication model for service discovery and invocation has been defined according to a proposed nSOM-PDU for the exchange of messages between data producers and consumers. This PDU, which is illustrated in Figure 5.5, consists of a Protocol Control Information with the common header and the Service Data Unit with the payload area, which is composed by a finite set (n) of service objects. This protocol data structure focuses on providing a flexible encapsulation definition optimized for each specific communication, thus minimizing the network load and processing overhead on the wireless sensor nodes.

Protocol Version	Message Type	ACK Flag	Flow ID	Payload Length
------------------	--------------	----------	---------	----------------

(a)

Attribute	Payload Length
Value	

(b)

Figure 5.5: nSOM-PDU format: (a) common header; (b) service object.

Based on this previous PDU format, both the common header and the payload have been defined. On the one hand, and regarding the header, the following fields are included: Protocol version (4-bit), which currently is version number 1; Message type (8-bit), which identifies the propagated service-level protocol message; ACK Flag (1-bit) in order to support piggybacking

acknowledgements in request/response interactions; Flow ID (8-bit), which enables the data sources to correlate a specific sequence of packets in client-server, request/response interactions depending on the application requirements, or being set to 0 to disable this feature; and Payload Length (16-bit) where the length of the payload following this header is specified in octets. On the other hand, and as mentioned before, the payload area of the nSOM-PDU is composed by a set of service objects. Each of these includes three fields: Attribute (8-bit) to identify the service object data type; Payload Length (8-bit), which includes the length of the payload of the service object; and Value, which encapsulates the specific service-level payload. In addition, Table 5.1 shows the message definition in the nSOM service discovery protocol.

TABLE 5.1: *nSOM service discovery protocol: Messages specification.*

Message type	Source	Destination	TX mode	Description
HELLO	Wireless device node / Orchestrator node	Broker node	Multicast transmission	<i>HELLO message.</i> It supports service advertisement in the wireless ad hoc domain for both simple in-network pervasive services and composed services to the Broker nodes. Also, ACK piggybacking is used with unicast transmission from destination to source to confirm its reception during the association establishment procedure.
BYE	Any network node	Any network node	Unicast transmission	<i>BYE message.</i> It enables the advertisement of network disconnections of a node in an ordered manner. The payload includes the code of the disconnection reason (normal, abnormal with error code).
ACK	Any network node	Any network node	Unicast transmission	<i>ACK message.</i> It is used as a positive acknowledgment for providing an explicit confirmation mechanism to nodes when acknowledgements based on piggybacking are not possible.
SERV_SUBS_NOTIF	Orchestrator node	Broker node	Multicast transmission	<i>SERVICE SUBSCRIPTION NOTIFICATION message.</i> It performs the subscription of an Orchestrator for available publications of in-network pervasive services in a Broker. Also, ACK piggybacking by using unicast transmission from destination to source is used to confirm its reception during the association establishment procedure.
SERV_AVAIL_NOTIF	Broker node	Orchestrator node	Unicast transmission	<i>SERVICE AVAILABILITY NOTIFICATION message.</i> It announces the existence of requested in-network pervasive service. Based on previous subscriptions, the Broker entity sends to the Orchestrator the service advertisement contained in a HELLO message of a specific service published by a wireless device node.
SERV_UNSUBS_NOTIF	Orchestrator node	Broker node	Unicast transmission	<i>SERVICE UNSUBSCRIPTION NOTIFICATION message.</i> It announces the unsubscription of the Orchestrator to a Broker in order to avoid receiving more SERV_AVAIL_NOTIF messages for a concrete subscription correlated with a previous SERV_SUBS_NOTIF message.

SERV_SHUTD_NOTIF	Broker node	Orchestrator node	Unicast transmission	<i>SERVICE SHUTDOWN NOTIFICATION message.</i> This is used to advertise the Orchestrator that a previously subscribed in-network pervasive service has been disconnected (i.e., service no available) in a specific wireless device node.
SERV_REG	Broker node	Gateway node	Unicast transmission	<i>SERVICE REGISTERING message.</i> It supports service advertisement and registering from the wireless ad hoc domain to the Internet cloud for both simple in-network pervasive services and composed services.
SERV_UNREG	Broker node	Gateway node	Unicast transmission	<i>SERVICE UNREGISTERING message.</i> It supports service unregistering from the wireless ad hoc domain to the Internet cloud for both simple in-network pervasive services and composed services.
PROBE	Broker node	Wireless device node / Orchestrator node	Broadcast transmission	<i>PROBE message.</i> It performs an on-demand service search in the wireless ad hoc domain including simple in-network pervasive services and composed services where both specific service type and general-service lookups are supported
PROBEMATCH	Wireless device node / Orchestrator node	Broker node	Unicast transmission	<i>PROBEMATCH message.</i> It is the reply of a PROBE message. This contains the advertisements and definition of the requested service or services in the wireless ad hoc domain.
SERV_DATA_REQ	Gateway node / Broker node / Orchestrator node	Broker node / Orchestrator node / Wireless device node	Unicast transmission	<i>SERVICE DATA REQUEST message.</i> It defines an invocation request for a simple in-network service or a composed service. It is generated in the Gateway node being redirected to the specific Broker node for service resolution. In case of a simple service, it is sent to wireless device nodes while in case of a composite service it is sent to the specific Orchestrator node for service resolution, which send the service data request to the wireless nodes that implements the composed service.
SERV_DATA_RESP	Wireless device node / Orchestrator node / Broker node	Orchestrator node / Broker node / Gateway node	Unicast transmission	<i>SERVICE DATA RESPONSE message.</i> It is the reply of a SERV_DATA_REQ message. This contains the data obtained in the service invocation. In the case of a simple in-network service, this is the value from the wireless device node while in a composed service it represents the resulting data implemented in the Orchestrator according to the process composition template and the simple service pieces of the wireless device nodes.

According to the previous, the end-to-end modelling of the inter-node interworking approach is addressed in a formal manner. In this regard, Figure 5.6 illustrates the UML Sequence Diagram of the inter-node workflow based on the service discovery and invocation procedures, in keeping with the network topology in Figure 5.4. This high-level approach for the inter-node model lies on the management support provided by the Service-Oriented Middleware Platform for the intra-node interaction workflow, as detailed in the previous section.

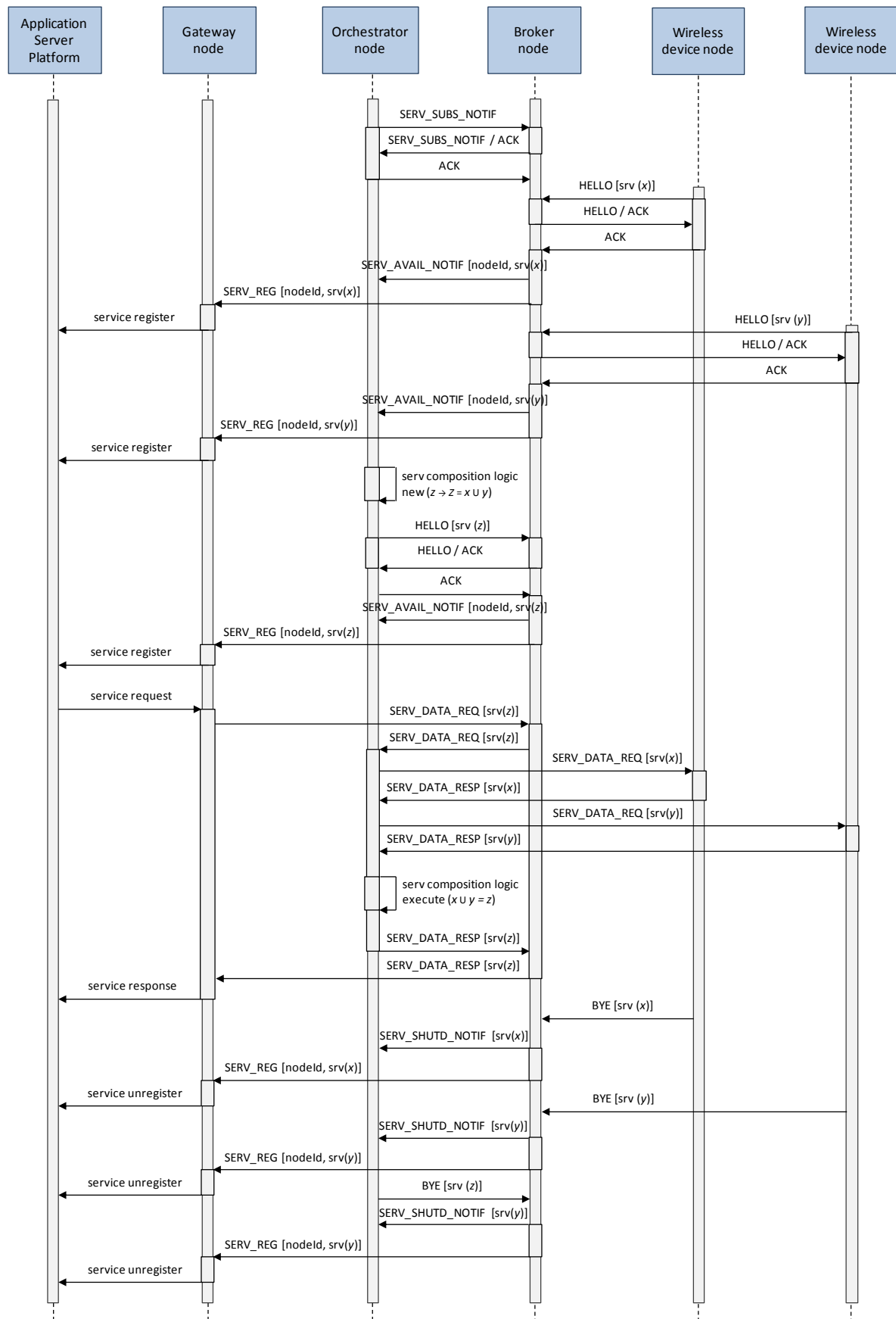


Figure 5.6: UML Sequence Diagram of the inter-node communication workflow.

During the service discovery phase associated to the control plane, wireless device nodes propagate periodically announcements of pervasive in-network services to the Broker nodes using HELLO messages, where the interface list of available services is encapsulated. These interfaces are modelled according to a semantic knowledge management approach based on service composition ontology. Orchestrator nodes also send periodically SERV_SUBS_NOTIF messages to the Broker nodes in order to subscribe to service announcements. Providing high-availability and load balancing, a three-way handshake protocol over multicast for association establishments in the ad hoc network is performed. It uses as service election criteria (i.e., policy applied in the aggregation procedure for identifying the best inputs between analogous service entities) of the composition algorithm the minimum round-trip time of the available service advertisements. These relations are defined between wireless devices and Broker nodes (i.e., HELLO, HELLO/ACK, and ACK interactions) as well as between Orchestrators and Broker nodes (i.e., SERV_SUBS_NOTIF, SERV_SUBS_NOTIF/ACK, and ACK interactions). Once the wireless device nodes and Orchestrator nodes have a designated Broker node, the communications with it are using unicast but keep performing the three-way handshake protocol for early detection of association disruptions, starting a new discovery phase if required. Both the reception of acknowledged HELLO messages in the designated Broker node and acknowledged SERV_AVAIL_NOTIF messages in the associated Orchestrator define a soft-state to preserve service registrations, which have to be refreshed periodically to maintain their availability in the network. Once the designated Broker node receives a HELLO message, it generates a SERV_AVAIL_NOTIF message to its associated Orchestrator node for advertising it about service availability, as well as a SERV_REG message sent to the Gateway node for service registering in external service cloud platforms. In addition, and according to the service composition logic in the Orchestrator node, when all the required pervasive in-network services are available and registered in the associated Broker node, the related composite service is instantiated. This generates a service announcement in the associated Broker node based on a HELLO message and the corresponding SERV_AVAIL_NOTIF message in the associated Orchestrator node, as in the case of a pervasive in-network service.

Regarding the service invocation phase in the data plane, service requests from the external service cloud are received in the Gateway node through a service-oriented Application Server Platform. It generates a SERV_DATA_REQ message to the Broker node where the service was previously announced, depending on the identifier of the requested service and the service registrations performed. From this Broker node, the SERV_DATA_REQ message can be sent to the wireless device node where the service is implemented (in the case of a simple, pervasive in-network service) or redirected to the associated Orchestrator node (in the case of a composite virtual sensor service). According to this last scenario, the service request will be received and processed in the corresponding agent-based composition service. This leads to generate (n) SERV_DATA_REQ messages to the wireless device nodes where the pervasive in-network services related to this composition are instantiated. When a SERV_DATA_REQ message is received in a wireless device node, it is dispatched to the agent implementing the corresponding pervasive in-network service. It performs the invocation of the service and returns back the result in a SERV_DATA_RESP message to the Broker node. On the one hand, and in the case of an external request of a pervasive in-network service, the SERV_DATA_RESP

message is sent back from the Broker node to the Gateway node where this will be delivered to the service-oriented Application Server Platform. On the other hand, and in the case of an external request for a virtual sensor service, the Broker node sends the corresponding (n) `SERV_DATA_RESP` messages to the related agent-based composition service in the associated Orchestrator node, where the business logic of the service integration algorithm for that composite service will be applied. This generates an expected result, which will be sent back to the service-oriented Application Server Platform through the Gateway node encapsulated in a `SERV_DATA_RESP` message.

5.4 Conclusions

In this chapter, the nSOM approach for service development and provisioning in pervasive smart ecosystems based on sensor network technology is discussed. As stated, this contribution represents the practical implementation of the reference service-oriented framework presented in Section 4.2. In this regard, the proposed SOA is justified. Additionally, the related agent-based service workflow for intra-node interactions and the service discovery protocol for inter-node communications are discussed. The rest of this section comprises a summary of the main issues regarding these aspects in the framework of this dissertation.

As presented, this solution adopts a component-based architecture in the Service-Oriented Middleware Platform, as well as a provisioning solution of pervasive in-network services using agent-oriented software engineering in the Pervasive Service Platform. The service-oriented middleware abstraction defines a set of general-purpose core functionalities. These include control services, cross services, low-level services and high-level services. Moreover, context-based intra-node communication mechanisms between software components and application entities, as well as inter-node communication procedures, are also implemented. First, the Control Services Layer supports the deployment, scheduling and interaction of software components in this architecture. The components identified in this middleware layer are the Container and Event Engine. The Container component manages the life cycle of the middleware, enabling context-aware dynamic discovery and access functionalities. In addition, context-based and event-driven communication support is provided by the Event Engine component. Second, the Cross Services Layer implements orthogonal functionalities. The components identified in this middleware layer are the Reasoning and Security. The Reasoning component defines handling support for status inference in the nodes based on the published services and the descriptions of the available resources. The Security component focuses on secure ubiquitous communications and authenticating network devices using key distribution centres. Third, the Low-Level Services Layer includes hardware-related functionalities as the base for the high-level middleware services. The components identified in this middleware layer are the Real-Time Management, Communication and Context Discovery Management. The setting-up for real-time parameters is performed using the Real-Time Management component. The Communication component provides a uniform abstraction for data communications. Additionally, the Context Discovery Management component enables context-aware service management based on semantics. Finally, the High-Level Services Layer defines a development support abstraction for programming heterogeneous business processes.

The components identified in this middleware layer are the Query, Command and In-node Service Configuration. The process for data acquisition based on specific query assertions is implemented in the Query component. Execution of remote context-aware procedures is supported by the Command component. The In-node Service Configuration component focuses on dynamic configuration of the service instances and functionalities.

The proposed service workflow focuses on intra-node and inter-node interaction models, defining how the middleware components and in-network agents interact to perform the service provisioning. Regarding the intra-node communication workflow, the Pervasive Service Platform represents an agent-oriented container abstraction to deploy and execute pervasive in-network services based on aspect-oriented programming, being the access point to the rest of the SOA. The scheduling and life cycle management (i.e., load, run, stop and destroy) of the agent-based services in this proposal are performed according to a programmatic interface model defined in the Pervasive Service Platform and the support of the Service-Oriented Middleware Platform. This also enables dynamic discovery capabilities of middleware components in the architecture at runtime, as well as a uniform procedure for service interaction based on context-aware information. It defines a callback procedure for service invocations where the service operation, parameters and potential service result are encapsulated, being delivered to the required participant in the service execution. Besides the previous, this architecture supports multi-service composition and internetworking with external service clouds, which relies on the inter-node communication workflow. This includes the definition of the Orchestrator and Broker network entities as part of the Service Internetworking Platform. These elements embrace the service composition logic and the interface between the wireless ad hoc domain and the external service networks via a Sink/Gateway node, respectively. The communication between them and the wireless device nodes is based on a service discovery protocol with an optimized nSOM-PDU definition. This protocol proposal for embedded environments covers the service advertisements and disconnections from the wireless device nodes, service subscriptions, notifications between the Orchestrator and the Broker nodes, service registration and unregistrations to the external Sink/Gateway node and service-oriented Application Service Platform, on-demand service search, and service request and response for specific invocations, both for pervasive in-network services and composite services.

nSOL: A Semantics-Oriented Language for Ubiquitous Knowledge Management

*B*ASED on the data-centric nature of most sensor networks, both the explosive growth of generated data and the rising availability of ubiquitous technology in our surrounding environments highlight the application of knowledge management principles and practices to the pervasive computing research as a key aspect. This issue is even more critical in ubiquitous service-oriented and composition infrastructures, where end-users might not be aware of the environment dynamicity, spontaneous communications and potential device interoperability, and where the management of knowledge assets and context-aware information in an adaptive and interoperable manner is essential.

This chapter describes our composition-centric implementation approach for virtual sensor services in ubiquitous smart ecosystems based on sensor network technology. This proposal has been called nSOL (nano Semantics-Oriented Language). Section 6.1 reviews the proposed contribution. Section 6.2 discusses our ontology solution for developing semantically-driven pervasive service composition. Section 6.3 presents its interface language and notation syntax. Finally, main conclusions of this chapter are summarized in Section 6.4.

6.1 Overview of the nSOL Approach

This proposal represents the practical implementation of the reference service composition framework formalized in Section 4.3. As reviewed, this model focuses on developing a dynamic composition-centric, semantically-driven solution that enables virtual sensor services based on sensor network technology in pervasive smart spaces and environments. As design requirements dissected in Chapters 3 and 4, this approach addresses the definition of a lightweight composition

language, development of adaptive service composition, semantic knowledge management based on ontologies, instantiation support for dynamic virtual sensor services and service-based integration with the Internet of Things paradigm. Main research foundations of this service composition-centric approach are based in our previous work [99].

The overview of this practical implementation is as follows. The nSOL approach develops the service-oriented, semantic knowledge management proposal for building virtual sensor services in the nSOM architecture. Its foundations are on the design of the Service-Oriented Middleware Platform, the Pervasive service Platform and the Service Internetworking Platform. This proposes a service composition language with a compact syntax based on Service Mapping Description (SMD) and JavaScript Object Notation (JSON), in order to define the service interface contract for the service discovery and interaction protocol. This schema is extended by providing Resource Description Framework triples that are embedded in the SMD/JSON descriptions. This enables an adaptive composition model of pervasive in-network services, which focuses on flexible service integration according to their availability in the network. In addition, this composition-centric approach is defined according to a knowledge management model based on a proposed ontology of pervasive in-network services. The resulting composition model has been implemented according to the OWL-S approach, to specify the service interfaces, invocation mechanisms and composition procedures. It provides a high, fine-grained expressiveness degree with enhanced service selection, as well as an improved service discovery and integration process. The previous issues lead to a dynamic composition schema of virtual sensor services providing support for decisions based on collaborative intelligence of multiple sensor instances, which would be impossible to create with isolated physical sensors. Additionally, and with the aim of enabling service-based internetworking with the IoT paradigm, this contribution allows the exposure of virtual sensor services to the service cloud using Internet-based standard interfaces, thus bridging sensor network infrastructures with Ambient Intelligence ecosystems.

6.2 Semantic-Driven Pervasive Service Composition

From the abstract formalization approach in Section 4.3 for architecting and composing pervasive in-network services, a concrete formal model based on an ontology definition has been built. This provides a formal vocabulary, covering all basic terms and relationships, which defines how pervasive in-network services are represented and aggregated using a composition-centric, semantic knowledge management proposal based on ontology. This leads to a concrete solution to model automatic computation and reasoning, centred on dynamic and adaptive instantiation and management of pervasive virtual sensor services.

The rest of this section is focused on presenting the data model specification based on the Resource Description Framework, which constitutes the knowledge management approach for implementing the nSOL ontology to model and compose pervasive in-network services. In this regard, the RDF approach has become a well-known and widely accepted formal language to describe the semantics of distributed approaches. This solution, which is based on first-order logic, defines 3-member tuples or triples that describe well-known relationships or properties between resources in a flexible, natural and concise manner.

6.2.1 Ontology-based Graph Data Model

The data model formalization for developing pervasive in-network services, as well as for composing them into virtual pervasive sensor services, is addressed according to the semantic digraph (i.e., directed graph) illustrated in Figure 6.1. In this graph $G = (V, E)$ we denote V as the set of graph vertex and E as the set of graph edges. The definition of the identified elements in these item sets is described in the following paragraphs.



Figure 6.1: *Semantic digraph of pervasive in-network service modelling and composition.*

On the one hand, V represents the finite set of subjects, predicates and objects in our proposed OWL-based ontology. This includes the following: `accessInformation`, `applicationEndpoint`, `applicationEndPointType`, `applicationEndpointValue`, `atomicProcess`, `atomicProcessSet`, `compositeProcess`, `compositionLogic`, `deviceEndpoint`, `deviceEndpointType`, `deviceEndpointValue`, `endpoint`, `hasAccessInformation`, `hasApplicationEndpoint`, `hasApplicationEndPointType`, `hasApplicationEndpointValue`, `hasAtomicProcess`, `hasAtomicProcessSet`, `hasCompositeProcess`, `hasCompositionLogic`, `hasDeviceEndpoint`, `hasDeviceEndpointType`, `hasDeviceEndpointValue`, `hasEndpoint`, `hasInput`, `hasOperation`, `hasOperationId`, `hasOperationSet`, `hasOutput`, `hasPrecondition`, `hasPreconditionSet`, `hasParameter`, `hasParameterSet`, `hasParameterType`, `hasParameterValue`, `hasResultType`, `hasResultValue`, `hasPreconditionType`, `hasPreconditionValue`, `hasProcessLogic`, `hasServiceId`, `hasServiceResult`, `input`, `operation`, `operationId`, `operationSet`, `output`, `parameter`, `parameterSet`, `parameterType`, `parameterValue`, `precondition`, `preconditionSet`, `preconditionType`,

preconditionValue, processLogic, resultType, resultValue, service, serviceId, serviceResult. On the other hand, E describes the finite set of directed node-arc-node links between the elements contained in the set of graph vertex. This includes the following: domain, range, subclass-of. This graph data model that implements our ontology is composed by a finite set (n) of RDF triples {subject, predicate, object}. Each one of these assertions expresses a specific relation between the resource that is denoted by the subject and the property specified in the semantic triple.

According to the modelling of a pervasive in-network service done in Section 4.3.1, its concrete ontology-based formalization can be defined as $PS : PS_i \equiv [\{ \text{“service”}, \text{“hasProcessLogic”}, \text{“processLogic”} \}, \{ \text{“processLogic”}, \text{“hasAtomicProcess”}, \text{“atomicProcess”} \}, \{ \text{“service”}, \text{“hasOperationSet”}, \text{“operationSet”} \}, \{ \text{“service”}, \text{“hasPreconditionSet”}, \text{“preconditionSet”} \}, \{ \text{“service”}, \text{“hasServiceResult”}, \text{“serviceResult”} \}, \{ \text{“service”}, \text{“hasAccessInformation”}, \text{“accessInformation”} \}]$. In this regard, a pervasive in-network service instance defines a discrete business logic processLogic according to a process atomicProcess, in order to generate a result serviceResult based on a set of operations operationSet and a set of preconditions preconditionSet, with a specific data accessInformation for deployment and invocation. The service result serviceResult is declared as a pair of triples [{“serviceResult”, “hasResultType”, “resultType”}, {“serviceResult”, “hasResultValue”, “resultValue”}]. The service preconditions are expressed in preconditionSet as a n -triple {“preconditionSet”, “hasPrecondition”, “precondition”} where each precondition item is defined as a pair of triples [{“precondition”, “hasPreconditionType”, “preconditionType”}, {“precondition”, “hasPreconditionValue”, “preconditionValue”}]. Also, an operationSet is expressed as a n -triple {“operationSet”, “hasOperation”, “operation”} where each operation is articulated with three triples [{“operation”, “hasOperationId”, “operationId”}, {“operation”, “hasInput”, “input”}, {“operation”, “hasOutput”, “output”}]. Both input and output objects are parameterized and characterized as a pair of triples {“input”, “hasParameterSet”, “parameterSet”}, {“output”, “hasParameterSet”, “parameterSet”}. Each parameterSet object is an n -triple {“parameterSet”, “hasParameter”, “parameter”} while each parameter can be enunciated as a pair of triples [{“parameter”, “hasParameterType”, “parameterType”}, {“parameter”, “hasParameterValue”, “parameterValue”}]. Finally, the accessInformation item provides a pair of triples [{“service”, “hasServiceId”, “serviceId”}, {“service”, “hasEndpoint”, “endpoint”}], which reference the unique identifier of the service and its related information for deployment and invocation, respectively. The endpoint is declared as a pair of triples [{“service”, “hasDeviceEndpoint”, “deviceEndpoint”}, {“service”, “hasApplicationEndpoint”, “applicationEndpoint”}] where deviceEndpoint specifies the device identifier where the pervasive service is located, and applicationEndpoint constitutes the identifier of the application in such device that implements the pervasive service. In addition, the deviceEndpoint is declared as a pair of triples [{“deviceEndpoint”, “hasDeviceEndpointType”, “deviceEndpointType”}, {“deviceEndpoint”, “hasDeviceEndpointValue”, “deviceEndpointValue”}] while for the applicationEndpoint these are [{“applicationEndpoint”, “hasApplicationEndpointType”, “applicationEndpointType”}, {“applicationEndpoint”, “hasApplicationEndpointValue”, “applicationEndpointValue”}].

Extending the conceptualization of a pervasive in-network service and according to the abstract formalization approach model for composition-centric pervasive services stated in Section 4.3.2, its concrete ontology-based formalization is proposed. In this regard, we define a pervasive virtual

sensor service as PVSS: $PVSS_i \equiv [\{ \text{"service"}, \text{"hasProcessLogic"}, \text{"processLogic"}, \{ \text{"processLogic"}, \text{"hasCompositeProcess"}, \text{"compositeProcess"} \}, \{ \text{"compositeProcess"}, \text{"hasCompositionLogic"}, \text{"compositionLogic"} \}, \{ \text{"compositeProcess"}, \text{"hasAtomicProcessSet"}, \text{"atomicProcessSet"} \}, \{ \text{"atomicProcessSet"}, \text{"hasAtomicProcess"}, \text{"atomicProcess"} \}, \{ \text{"service"}, \text{"hasOperationSet"}, \text{"operationSet"} \}, \{ \text{"service"}, \text{"hasPreconditionSet"}, \text{"preconditionSet"} \}, \{ \text{"service"}, \text{"hasServiceResult"}, \text{"serviceResult"} \}, \{ \text{"service"}, \text{"hasAccessInformation"}, \text{"accessInformation"} \}]$. As shown, a pervasive virtual sensor service instance is modelled as a service with a set of operations `operationSet` and a set of preconditions `preconditionSet`, while generating a service result `serviceResult` according to a discrete business logic `processLogic`. The semantic definition in terms of RDF triples for the `operationSet`, `preconditionSet`, `serviceResult` and `accessInformation` elements is analogous to the related specification in the case of a pervasive in-network service, but with a particular type of `processLogic`. It is characterized as a composite process `compositeProcess` specifying the orchestration procedure based on the integration of a finite set (n) of atomic processes `atomicProcessSet`.

6.2.2 Composition-centric Service Engineering in nSOL / nSOM

This graph-based data model establishes the semantic knowledge management approach for service composition in the nSOL contribution. In this regard, and in the context of our nSOM architecture, an ontology-aware proposal for the Service Internetworking Platform has been accomplished. In this regard, brokering agents are responsible for receiving and relaying service definition information, while Orchestrator agents process and manage these semantic annotations and properties explicitly during the instantiation of the virtual sensor service. This task is performed according to the mandatory and optional terms of the vocabulary defined in the ontology for the corresponding protocol messages. On the one hand, and as discussed in Section 5.3.2, service agents developing broker role provide an interface between the wireless ad hoc and the external service cloud domains. These handle the registering for both pervasive in-network and virtual sensor services announcements, as well as providing support for service disconnection, subscription advertisement, search capability and data request/response invocations. On the other hand, orchestrator agents manage the service data contained in service announcements and invocations/responses, according to the embedded semantic elements and the specific composition algorithm. In compliance with the ontology-based graph data model of nSOL, the following semantic elements of interest can be highlighted: `preconditionSet`, `preconditionType` and `preconditionValue` (optional semantic terms, which just in case of service execution has associated any precondition); `parameterSet`, `parameterType`, `parameterValue` (optional semantic terms, according to the potential parameterization of service operations); `serviceResultType`, `serviceResutValue` (mandatory semantic terms, which are required as part of the execution workflow of the specific service); `deviceEndpointValue`, `applicationEndpointValue` (mandatory semantic terms, where service execution requires access and in order to reference the invocation endpoint). In addition, orchestrator agents are focused on defining the discrete business logic of the virtual pervasive sensor service, which is modelled in terms of logic `compositeProcessLogic` for a set of atomic processes `atomicProcessSet`. The business logic of the composite process is developed based on the discrete business logic provided by the agent-based user services (i.e., pervasive in-

network services) executed on the wireless device nodes of the ad hoc network. These are selected according to the election service criteria, as an input of the composition algorithm implemented by the orchestrator agents. As stated in previous section, it is based on the minimum round-trip time for service advertisements using periodic re-calculation, according to the soft-state of the service registrations in the designated Broker node. The reference to deploy and access to a pervasive virtual sensor service is based on the orchestrator agent identifier and the network address of the Orchestrator node where it is implemented. These parameters are published in service announcements (i.e., HELLO message) in the designated Broker node of the sensor network. For further information in the context of the service annotation process and the composition workflow in the nSOL contribution, see Section 6.3.2.

6.3 Interface Language and Semantic Notation Syntax

In the last decade, XML has become the de facto standard for data modelling, representation and exchange. This issue led to improve the support for system and application heterogeneity, as well as to enhance their integration at Application Level, where Web Services become the dominant service-oriented middleware for service accessing and provisioning on the Internet. Nevertheless, most of the XML-based data representation technologies are, in general, not suitable for sensor networks. This section focuses on surveying our proposal for interface language applied to pervasive sensor services, as well as the abstract notation syntax for their semantic annotation according to the defined ontology-based graph data model. Additionally, our approach for pervasive service-oriented integration into the Internet cloud is reviewed.

6.3.1 Interface Definition using Service Mapping Description

Several works in the related literature discuss the suitability of the XML language as a data modelling format in embedded and reduced functionality computing devices [1, 61, 149, 233]. As the main limitation, these highlight its high degree of verbosity that results on an increased overhead associated with message size. This issue has a direct negative effect on the processing capacity required for encoding/decoding messages and extra battery consumption for data radiation, as well as an impact on the required transmission bandwidth and delay in propagation time. In this regard, the application of XML-related technologies such as WSDL to define the contract interface of a service or SOAP as message-oriented communication protocol does not fit, in most of cases, with the inherent resource-constrained nature of wireless ad hoc and sensor networks in ubiquitous computing.

In this regard, some state-of-the-art research initiatives to address the previous open issues have been performed. A novel direction in pervasive computing based on Wireless Sensor Networks is the application of JavaScript Notation Syntax (RFC 4627) [139], which is an approach for data modelling and management. As stated in this Request for Comments, JSON defines a lightweight, language-independent and self-describing text-based data representation and exchange format. Its syntax is specified in terms of collections of name/value pairs where the value field can be a number, a string, a boolean data type, an array, an object or null. The codification rules are simple (basically, names and primitive value types are placed into double quotes, colon separated; object records are into curly brackets; arrays are written into square

brackets), providing a high-compaction degree of the structures required for defining data assets and their relationships. In this way, several novel studies agree that, in most of cases, the use of JSON instead of XML can reduce data redundancy and the required analyzing time effectively, thus highlighting its suitability for programming, debugging and maintaining. In the particular case of embedded and sensor-based computing, the application of this proposal can lead to a significant reduction of the wireless network overhead, processing overload and power consumption [44, 246, 259].

According to the previous analysis, our proposal is founded on the application of JSON for service modelling and as ubiquitous data exchange format, implementing on top of it a service interface and invocation language based on the Service Mapping Description (SMD) [91] initiative. This novel approach enables a machine-readable, compact and flexible interface language specification for standard Web Services based on JSON-RPC. Regarding our nSOL contribution, the application of SMD in sensor-based ubiquitous computing is defined. This leads to provide a service interface language applied for both service discovery and invocation procedures, in the case of pervasive in-network services and composite virtual sensor services. The foundations of the SMD proposal are outlined in the next paragraph. Additionally, semantically-driven service composition is devised pursuant to RDF annotations, which are embedded into SMD definitions according to the ontology graph-based data model presented in Section 6.2.1. This is surveyed in the next section.

A Service Mapping Description instance is a JSON object including a finite set (n) of name/value pairs called service properties, which are used to describe the interworking model of available service methods in a specific location. One of these service properties is called services, which constitutes a JSON object representing the method definitions of service calls. The service properties can be detailed at root level of the SMD document, thus denoting common information that is inherited by all services, being possible to be overwritten by each service for a particular configuration. According to the SMD 2.0 specification, the proposed service properties are the following: transport, envelope, contentType, jsonCallbackParameter, target, additionalParameters, parameters, returns, services, SMDVersion, id and description. The description of the previous is as follows. The transport property establishes the protocol to enable service invocations, including GET, POST, REST, JSONP and TCP/IP methods (optional property, default value: POST). The envelope property indicates how the service parameters are encapsulated in the transport protocol, being its possible values URL, PATH, JSON, JSON-RPC-1.0 and JSON-RPC-2.0 (optional property, default value: URL). The contentType property is the type of the expected content item of a service, supporting any valid MIME type (optional property, default value: application/json). The jsonCallbackParameter property is used to point out the callback method contained in the response (optional property, default value: callback). The target property is the reference endpoint for a service invocation request, specifying a URL or a network address if the transport property is TCP/IP (mandatory property). The additionalParameters property enables to indicate if unnamed parameters are supported, in order to be included or not in the service requests. The possible values are true, false or a JSON object to specify additional parameters (optional property, default value: true). The parameters property defines the parameters of a service call, which can be appointed at root level to be inherited by all the services described in the SMD document, or being provided at

service level. The value of a parameter is a JSON array including the following additional parameters: name (name of the parameter), optional (defines if the parameter in the call is optional or not), and default (default value for the parameter). The returns property states the expected return type of a service call (mandatory property). The services property, as stated before, formulates the method calls of implemented services, being provided as a JSON object. Each name/value pair of the object is a service method, where the name is the name of the service and the value is the related description (mandatory property). The SMDVersion property declares the protocol specification that is used in the SMD instance (optional property). The id property defines a unique identifier of the service method (optional property). Finally, there is the description property, which provides a definition of the concrete service method (optional property).

6.3.2 Semantic Annotation for Service Architecting and Composition

Semantic annotations enable us to identify, in a formal and unequivocal manner, concepts and relationships in the context of specific knowledge domains with the aim of enhancing their integration into Machine to Machine (M2M) processes, leading to system interoperability. This technology can be applied to semantic composition workflows. This focuses on implementing integrated mechanisms that automatically bind the information of selected service resources for instantiating business process. Based on this notion, our nSOL approach focuses on semantic service annotation for lightweight composition of pervasive in-network services into virtual sensor services using sensor network technology. To this end, the underlying support of our nSOM architecture is proposed. This research contribution constitutes an innovation point in the field of ubiquitous computing, which foundations are provided in our previous work [99].

With the aim to define semantic service annotations in SMD service interfaces, a lightweight RDF-based approach has been applied. To this end, the RDF Notation3 (N3) [38] standard is adopted. Notation3 enables a compact syntax that provides a simple manner to express triples while reducing the language overhead, promoting smooth integration with standard RDF. In essence, this non-XML encoding format allows the previous by supporting namespaces and directives (e.g., URI references that are bound to a namespace using @prefix prefix, @base directive to reference the URI that is used as a base for relative URIs, etc.), as well as supporting abbreviations and shortcuts for common predicates (e.g., comma separator to represent a common object with the same subject and predicate in the triple; semicolon separator to represent a common predicate with the same subject, etc.). Additionally, this RDF subset also enables to represent unnamed nodes (including square brackets, blank node syntax and paths), formulae for including graphs, lists definition, boolean literals and notes in numbers, among other features.

According to it, the proposed schema for semantic service annotation in pervasive sensor-based services is presented. This is illustrated in Figures 6.2, 6.3 and 6.4, where examples of service advertisement, service data invocation and service data response are provided, respectively. As shown, service interfaces are defined using SMD language with JSON syntax, while semantic tags are in RDF Notation3 according to the ontology-based graph data model reviewed in Section 6.2.1. This annotation approach is based on in-line, embedded N3 triples in the service

contract interfaces. In this regard, the name of a service property in the SMD description is associated with a concrete object in the ontology-based graph data model, and likewise the latter with the corresponding value of the SMD property for such pervasive service. This information is processed by the Orchestrator entity as part of the execution of the service composition workflow, which is presented in the next paragraph.

```

01 {
02   @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
03   @prefix nsol: <http://ontology.repository/nsol/1.0/nsol_ontolgy.owl>
04   "transport": "TCP/IP",
05   "envelope": "JSON-RPC-2.0",
06   "target" a nsol:deviceEndPointValue: "7F00.0101.0000.1001",
07   "additionalParameters": "false",
08   "SMDVersion": "2.0",
09   "services": {
10     "service_one" a nsol:service: {
11       "target" a nsol:applicationEndPointValue: "/nsomServiceAgent_one",
12       "parameters": [
13         { "name" a nsol:parameter: "single_parameter",
14           "type" a nsol:parameterType: "char" } ],
15       "returns" a nsol:returnsType: "void"
16     },
17     "service_two" a nsol:service: {
18       "target" a nsol:applicationEndPointValue: "/nsomServiceAgent_one",
19       "parameters": [
20         { "name" a nsol:parameter: "first_parameter",
21           "type" a nsol:parameterType: "string" },
22         { "name" a nsol:parameter: "second_parameter",
23           "type" a nsol:parameterType: "string" } ],
24       "returns" a nsol:resultType: "string"
25     }
26   }
27 }

```

Figure 6.2: *Semantic annotation example for service announcement.*

```

01 {
02   @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
03   @prefix nsol: <http://ontology.repository/nsol/1.0/nsol_ontolgy.owl>
04   "transport": "TCP/IP",
05   "envelope": "JSON-RPC-2.0",
06   "target" a nsol:deviceEndPointValue: "7F00.0101.0000.1001",
07   "origin": "7F00.0101.0000.1002",
08   "additionalParameters": "false",
09   "SMDVersion": "2.0",
10   "services": {
11     "service_two" a nsol:service: {
12       "target" a nsol:applicationEndPointValue: "/nsomServiceAgent_one",
13       "origin": "/nsomBrokerAgent",
14       "parameters": [
15         { "name" a nsol:parameter: "first_parameter",
16           "value" a nsol:parameterValue: "parameter 1 value" },
17         { "name" a nsol:parameter: "second_parameter",
18           "value" a nsol:parameterValue: "parameter 2 value" } ]
19     }
20   }
21 }

```

Figure 6.3: *Semantic annotation example for service data invocation.*

```

01 {
02   @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
03   @prefix nsol: <http://ontology.repository/nsol/1.0/nsol_ontology.owl>
04   "transport": "TCP/IP",
05   "envelope": "JSON-RPC-2.0",
06   "target" a nsol:deviceEndPointValue: "7F00.0101.0000.1001",
07   "origin": "7F00.0101.0000.1002",
08   "additionalParameters": "false",
09   "SMDVersion": "2.0",
10   "services": {
11     "service_two" a nsol:service: {
12       "target" a nsol:applicationEndPointValue: "/nsomServiceAgent_one",
13       "origin": "/nsomBrokerAgent",
14       "parameters": [
15         { "name" a nsol:parameter: "first_parameter",
16           "value" a nsol:parameterValue: "parameter 1 value" },
17         { "name" a nsol:parameter: "second_parameter",
18           "value" a nsol:parameterValue: "parameter 2 value" } ],
19       "returns" a nsol:resultValue: " this is the result of the service"
20     }
21   }
22 }

```

Figure 6.4: *Semantic annotation example for service data response.*

The definition of the related process workflow for pervasive service architecting and composition based on semantic annotations is as follows. During the service discovery phase, information regarding registered services in each Broker node (using HELLO messages) is announced to its designated Orchestrator node (SERV_SUBS_NOTIF and SERV_AVAIL_NOTIF messages). As discussed in Section 5.3.2, the soft-state mechanism implemented in such nodes deals with spontaneous communications and environmental dynamicity present in ubiquitous and ad hoc networks. These pervasive in-network services are input candidates for the process workflow of the composition algorithm, according to the outcome of the service election criteria. The selected service candidates are passed on-the-fly to such composition workflow, which is responsible to instantiate and deploy the operational virtual sensor service process in the Orchestrator node at runtime, as well as the corresponding service registration in the designated Broker node. This workflow is based on a process composition template. This defines the required entity types of pervasive in-network services into the integration process, as well as their coordination activities and tasks, capturing the requirements model to obtain the virtual sensor service instance. In this regard, the proposed composition procedure is semantically driven, according to the proposed ontology-based graph data model and the workflow defined in the process composition template. This leads to implement a mechanism for identifying and binding interface elements of pervasive services according to their RDF annotations at runtime. This is done for both service discovery (i.e., service announcements) and invocation descriptions (i.e., service request and response), which participate in the business logic of the specific virtual sensor service. According to the processing capacities and the available network resources, such OWL ontology implementing the graph-based data model can be deployed in the service-oriented Application Server Platform, which is a full functionality device with significantly higher capacity than the wireless sensor nodes. In this regard, software agents in the Orchestrator nodes can perform on-demand query resolution against the ontology definition

through the Broker and Sink/Gateway nodes, in order to dynamically process the semantic annotations instead of maintaining multiple replicas of it in the memory region associated to each agent-based service.

6.3.3 Pervasive Service-Oriented Integration into the Internet Cloud

As discussed in Sections 3.1.4 and 3.2.4, a key aspect in both service-oriented middleware architectures and dynamic service composition frameworks is the internetworking support with external service networks on the Internet of Things world. In our approach, this is provided via the service-oriented Application Services Platform, as introduced in Section 5.3.2. On the one hand, this subsystem interacts with the Broker network entities through the Sink/Gateway nodes for managing the exchange of data messages with the wireless ad hoc sensor domain for pervasive sensor invocations. On the other hand, it provides user interfaces to the Internet cloud domain for enabling ubiquitous access to pervasive services. As part of our work, the identification of the ideal, main technical requirements for the service-oriented Application Server Platform has been accomplished. These include the provisioning of service-oriented abstractions for application interoperability, multi-interface support for heterogeneous and transparent service connectivity, as well as open and standard-based solutions to interact with the Internet. In consequence, this subsystem enables SOA foundations for effective, dynamic, agile and easy integration of business processes into scalable distributed networks.

Based on the previous discussion, the adoption of an Enterprise Application Integration (EAI) approach [110] is proposed. In this regard, and in the framework of the LifeWear European Research Project [169], an Enterprise Service Bus (ESB) solution was applied to implement the interface between the wireless and ad hoc sensor domain and the Internet cloud. An ESB, as an instantiation of an evolved EAI, is a highly-distributed and event-driven platform for providing messaging capability, web-based interfaces, data management support, as well as monitoring and coordination procedures for integrating heterogeneous applications and services across diverse networks and systems in a loosely-coupled manner [63]. According to this proposal, some previous works on pervasive computing such as [57, 207, 238] have researched on the application of this technology to sensor networks. From the point of view of our dissertation, this EBS proposal is defined as a black-box entity with a justified functionality, being out of the scope of the proposed research contributions. It is mainly used to enable several Sink/Gateway nodes, each one implementing a specific adaptor for concrete open heterogeneous interfaces such as standard Web Services, RESTful [37] and DPWS [213], to connect with the Internet while integrating pervasive in-network and virtual sensor services into external networks and end-user applications.

6.4 Conclusions

This chapter reviews the nSOL approach for enabling virtual sensor service provisioning according to a semantically-driven, composition-centric paradigm for building ubiquitous smart ecosystems based on sensor network technology. This contribution represents the practical implementation of the service composition model provided in Section 4.3. In this regard, the underlying ontology-based graph data model is presented. After that, the proposed service

support in the nSOM architecture for virtual service interworking is analyzed. Additionally, and based on the previous, the interface language for defining service descriptions as well as the semantic notation syntax for service composition introduced in our approach are discussed. Finally, main requirements for pervasive service integration with the Internet of Things cloud are identified. The rest of this section comprises a summary of the main issues regarding these aspects in the framework of this dissertation.

The composition-centric approach developed in the nSOL contribution has been devised around an ontology-based, semantic knowledge management model. This provides a purely formal, unequivocal vocabulary for representing pervasive in-network services while integrating them into virtual sensor service abstractions. To this end, a graph-based data model based on RDF triples for creating an OWL-compliant ontology is proposed. According to this, a pervasive in-network service is perceived as an abstract resource represented in terms of `preconditionSet`, `operationSet`, `processLogic`, `serviceResult` and `accessInformation` parameters. As stated, this ontology-based graph data model establishes the knowledge management schema applied for ubiquitous service composition in the nSOM architecture. In this regard, service interface descriptions, including both service discovery and invocation definitions, are semantically annotated using the subjects, predicates and objects defined in this ontology.

As discussed, data representation formats based on XML may not be suitable, in most cases, for the resource-constrained nature of embedded sensor network computing, due to its high degree of verbosity. This leads to reconsider the widespread use of Web-based technologies, such as WSDL and SOAP, in the domain of Wireless Sensor Networks. In this regard, we propose the employ of the JSON standard in these ad hoc networks, as a lightweight, language-independent and self-describing text-based data representation and interchange format. This proposal, which is based on name/value pairs, enhances data compaction and can effectively reduce the data redundancy and required time for analysis, network overhead and power consumption. Based on this syntax approach, the SMD proposal for defining service discovery descriptions and invocation interfaces in pervasive environments is applied. According to the previous, SMD instances are defined as a JSON object containing a finite set (n) of service properties describing heterogeneous models for service interworking. In this regard, service properties that are considered in the SMD v2.0 specification include the following: `transport`, `envelope`, `contentType`, `jsonCallbackParameter`, `target`, `additionalParameters`, `parameters`, `returns`, `services`, `SMDVersion`, `id` and `description`.

Once the knowledge domain for pervasive service composition is established and the related data representation and service interface description formats are defined, our semantic annotation approach is discussed. This highlights the application of semantic annotations to composition workflows, enabling specific procedures to bind the information of selected service resources in order to create and instantiate business processes in an automated manner. In this regard, the nSOL contribution leads to bring the foundations of semantic service annotations for lightweight service composition of pervasive in-network services into virtual sensor services using sensor network technology. To this end, SMD service interfaces are annotated according to the RDF N3 standard. This non-XML approach enables a compact syntax for providing a simple manner to express semantic triples while reducing the language

overhead, also allowing smooth integration with standard RDF. In this regard, and by using embedded Notation3 triples, service interfaces expressed in SMD and contained in service advertisements, service data invocations and service data responses are annotated according to the ontology-based graph data model previously reviewed.

According to the previous, the modelling process implemented in the composition workflow of our proposal is defined. This procedure, which is executed on each Orchestrator node, receives the service descriptions with embedded semantic annotations, as well as the service data request invocations from its Broker node, sending to it the service data responses received from the wireless sensor nodes. Based on the service announcements published in the network, the composition template defining the required service entity types in the integration and coordination activities and tasks, as well as the service candidates obtained according to the election criteria, the virtual sensor service is instantiated in the Orchestrator node at runtime. These on-the-fly composite services are exposed to the Internet cloud as new pervasive services, through the service-oriented Application Server Platform. Main identified technical requirements of this subsystem are service abstraction for application interoperability, multi-interface support for heterogeneous and transparent service connectivity, as well as open and standard-based solutions to interact with the Internet. This leads to enable effective, dynamic, agile and easy integration of business processes into scalable distributed networks.

Performance Characterization

*U*BIQUITOUS computing is characterized by the enormous potential it promises, being called to be a key factor of the new information and communication era where it is just the beginning. However, the impact of pervasive solutions in our daily lives is rather limited, making difficult to bring this technology to the end-users in a natural way. In this regard, of special importance is the development of living labs for enhancing practice experiences, focusing on initiatives that involve research projects for leading proof-of-concept deployments and widespread demonstrators in real-world environments.

This chapter describes our prototype implementation and its performance characterization for validating the practical application contributions of this dissertation. In this regard, the following sections are covered. Section 7.1 surveys the testbed approach for the prototype. Section 7.2 inspects code size of the middleware components and software agents. Volatile memory footprint is discussed in Section 7.3. The study regarding CPU overhead is in Section 7.4, while service time delay measures are shown in Section 7.5. The battery lifetime analysis is provided in Section 7.6. Finally, main conclusions of this chapter are summarized in Section 7.7.

7.1 Testbed Definition

The practical application contributions presented in this dissertation have been validated carrying out a characteristics analysis of a virtual service prototype for personal environmental sensing. On the one hand, this implies the development of the nSOM architecture as the core platform for service-oriented middleware architecting and pervasive service interworking for both intra-node and inter-node communication modes. On the other hand, the implementation of the nSOL semantic composition paradigm has been also accomplished. This includes the composition-centric proposal for orchestrating pervasive in-network services based on agent

technology into virtual sensor services, as well as their exposure to the Internet cloud for building pervasive smart spaces and environments. In this regard, the present section overviews this sensor-oriented approach from the perspective of its research framework, as well as the related details of the sensor network platform used to build the proposed software prototype.

7.1.1 Prototype Modelling

The initial germ of this prototype appears in the context of the DiYSE European Research Project [90], with the main objective of building smart ecosystems on the Internet of Things according to a service creation paradigm developed by the end-users. These are provided with the necessary tools for searching for public services easily, mashing-up device functions discovered in the smart space with independence of the specific platforms details. This includes dealing with issues such as hardware diversity, data connectivity in resource-constrained devices or multi-modal communication protocols, among others. From the technical point of view, semantic technologies play a key role in this initiative providing the necessary support for integrating such ubiquitous services of diverse nature implemented on heterogeneous devices into context-aware smart infrastructures. The practical demonstrator including a preliminary version of our prototype contribution earned the ITEA2 Exhibition Award 2010, while further research efforts carried out in this project allow achieving the ITEA2 Silver Award 2012. In addition, additional work based on this approach has been carried out in the framework of the LifeWear European Research Project [169], which was introduced in Section 1.3.

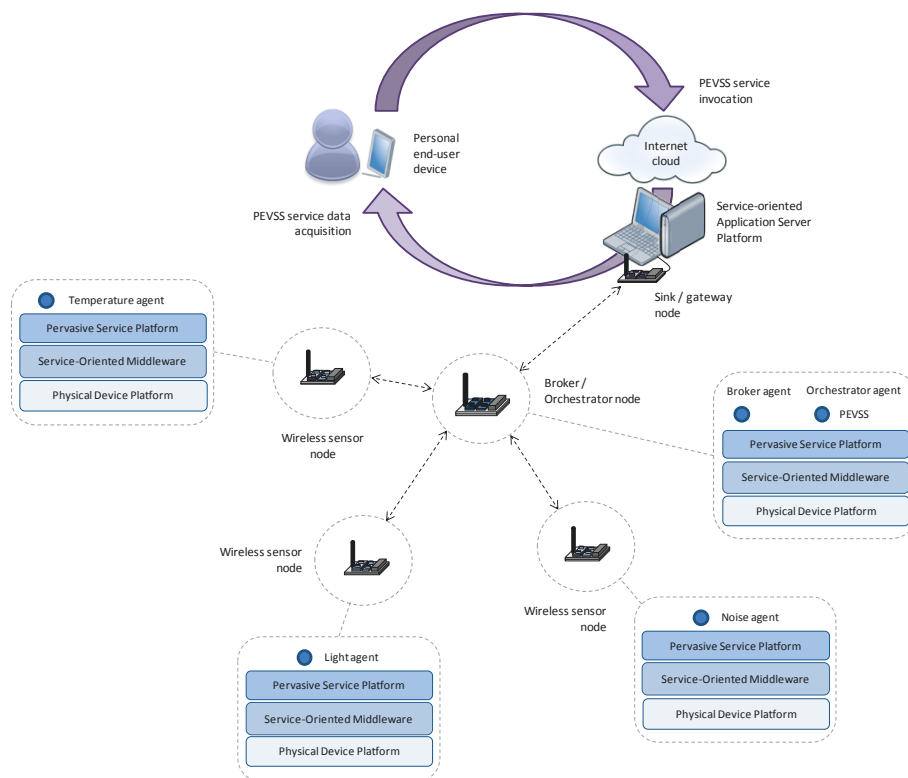


Figure 7.1: *Characterization scenario for prototype analysis.*

Our contribution to this research project was centred on the development of a dynamic service delivery platform based on the nSOM and nSOL architectures for enabling virtual sensor services to the Internet cloud. These are managed as an input for mashup-based frameworks of services using tools based on composition-oriented canvas deployed in the personal and mobile devices of the end-users. In the context of the domain of the wireless ad hoc sensor network, a Personal Environment Virtual Sensor Service (PEVSS) prototype was developed, which is illustrated in Figure 7.1. It is implemented in a sensor-based orchestrator agent that defines the associated business logic to the process workflow and the composition template in order to integrate environmental sensor services. The sensor network scenario in the project demonstrator was centred on a ubiquitous smart space equipped with three different types of pervasive in-network services: temperature, light and noise. Each one of these is implemented by a specific software agent deployed in a dedicated wireless sensor node. This defines operations to obtain measures in both quantitative (i.e., celsius degrees for temperature measures, luxes for light intensity, and decibels for sound level) and threshold-based (i.e., very low, low, medium, high, very high) modes. On the one hand, and from the perspective of the middleware backbone, these agents take advantage of the nSOM architecture, being built on top of it. According to the inter-node communication procedure presented in 5.3.2, a broker agent is developed. It performs service interworking within the wireless ad hoc domain and with the external service cloud via an ESB node infrastructure. In addition, the PEVSS agent for implementing orchestration and instantiation of virtual sensor service processes is defined. Both software agents are deployed in the same Broker/Orchestrator node, which is connected to the ESB using a Sink/Gateway node implementing a web-based interface. This enables service invocation and data acquisition by the end-users through a Web application installed on their mobile devices. On the other hand, and regarding the dynamic service composition approach, the SMD service descriptions published by the agents implementing the pervasive in-network services are registered in the designated broker agent and propagated to the PEVSS orchestrator agent. According to the composition workflow process, the available services in the network are filtered by the election criteria while the orchestrator agent instantiates the PEVSS artefact at runtime. The resulting composite process is built, as discussed in Section 6.3.2, according to the RDF annotations for the temperature, light and noise pervasive in-network services. In this regard, these determine the endpoint service references to obtain the inputs for the composition algorithm. The PEVSS instance is registered in the broker agent as the procedure to obtain the comfort measure perceived in a specific area, depending on the sensed values by the involved simple services. As in the case of the pervasive in-network services, both quantitative and threshold-based interface methods are defined by the PEVSS agent and published to the Internet cloud.

7.1.2 Development Platform

In order to carry out the analysis of the proposed software prototype, its implementation and characterization on real hardware has been done. Currently, there are several development platforms for wireless sensor-based embedded computing, where a detailed comparative study of hardware platforms for Wireless Sensor Networks can be found in [67]. Among the existing alternatives, we selected the Oracle Sun Small Programmable Object Technology (SPOT) [218] sensor network platform, which is illustrated in Figure 7.2. It provides a framework for rapid

prototyping of wireless sensor-based applications using the J2ME CLDC (Connected Limited Device Configuration) API v1.1 [275], differentiating three layers: processor board with radio, sensor board and battery. Additionally, and as described below, a programming environment is supplied with it. This comprises a set of tools designed to build Java-based applications on consumer electronics and embedded devices.



Figure 7.2: *Oracle Sun SPOT hardware platform* (Oracle Corporation).

The processor board with radio of the Sun SPOT nodes includes a 180MHz, 32-bits RISC ARM920T processor core, 4MB Flash and 512KB RAM memory, as well as a radio module equipped with a ChipCon 2420 wireless transceiver operating at 250kbps, 2.4GHz (IEEE 802.15.4 specification). Connection with personal computers and laptops is enabled through a USB interface. This processing layer supports local reads from a sensor board installed in the node, which is equipped with temperature, light and proper acceleration (i.e., accelerometer) sensors. It also offers general-purpose, input-output pins for connecting external data sources, an analog-to-digital converter and eight tri-colour LEDs. Finally, both the processor board with radio and the sensor board are powered by a 3.7 V, 750mAh lithium-ion battery module. As main limitation of this hardware platform, battery lifetime is quite reduced [279] leading to a high energy consumption, which is excessive in most cases for the required autonomy in production deployments of WSNs.

Apart from the previous hardware-related issues, the Sun SPOT platform includes a complete software development environment. In addition to the J2ME CLDC, several APIs such as hardware, demo sensor board, network, radio and desktop libraries are also built-in. Integrated Development Environment (IDE) integration is supported and simulation-based tools for code deployment, test and debugging are provided. Finally, as a runtime environment, Sun SPOT uses Squawk [256]. This is a Java Virtual Machine (JVM) with small footprint. It is mainly implemented in Java Language for improving portability and maintainability, while defining a compact byte-code instruction set with a simplified garbage collector for enhanced performance with a reduced computing overhead. In addition, thread scheduling for multi-programming, as well as isolation mechanisms to optimize the management of multiple Java applications, are also supported.

7.2 Code Size

In this section, the impact of code overhead from the point of view of both non-volatile memory storage space and source lines of code is discussed. The justification of using such software metrics is based on the resource-constrained nature of ad hoc sensor networks, in terms of code storage, as indicated in Section 1.1. This typically ranges from 48KB in early generation sensor nodes to 32MB in the most novel hardware platforms. Firstly, this analysis is performed attending to the Service-Oriented Middleware Platform, specifically to the software components implemented in the defined prototype. These data are collected in Table 7.1. Secondly, figures regarding code overhead for both the Pervasive Service Platform and Service Internetworking Platform are provided in Tables 7.2 and 7.3, respectively. In both cases, the obtained measures are contrasted with the Flash memory capacity available in the Sun SPOT development platform.

TABLE 7.1: *Code size for Service-Oriented Middleware Platform*

Software artefact	Flash size (bytes)	Source lines of code	Occupancy percentage
Communications component	12536	745	0.229
Container component	9758	502	0.233
Context Discovery Manager component	11427	554	0.272
Device Abstraction Layer component	1740	113	0.041
IComponent interface	209	31	0.005
Query component	4064	198	0.097

On the one hand, and regarding the Service-Oriented Middleware Platform approach, for this prototype we developed the software components listed in Table 7.2, based on the definition of the *IComponent* interface. These are the following: Communications, Container, Context Discovery Manager, Device Abstraction Layer and Query. As shown, the code size of these middleware components is very compact, thus resulting in a low implementation overhead. The total amount of code footprint for the Service-Oriented Middleware Platform is lower than the 0.9% of the 4MB non-volatile capacity of Sun SPOT. The code size for the greatest component is lower than 15KB while the average is around 7.7KB, which is the 0.188 % of the Flash storage space. On the other hand, and as shown in Table 7.2, the implementation for the Pervasive Service Platform in this prototype was focused on the Pervasive Service Platform

component, as well as a set of user agents based on the *IAgent* interface. These are the following: pervasive in-network light agent, pervasive in-network noise agent and pervasive in-network temperature agent. As in the case of the Service-Oriented Middleware Platform, a lightweight implementation of the software agent layer is observed. In the case of the Pervasive Service Platform component, its code footprint was 1.7KB, still below to the average of the middleware components. Regarding user agents, the observed data show an average size of a pervasive in-network agent around 3.7KB. This represents the 0.091% of the available non-volatile memory capacity, while in all cases was below to the 0.12% of the occupancy percentage. This figure reflects a very high degree of scalability for deploying user agents in the Pervasive Service Platform of this proposal. Finally, and in this same line, values obtained from the codification of the Service Internetworking Platform in the prototyped virtual sensor service indicate a low code overhead for both the service broker agent and PEVSS orchestrator agent. In this regard, and being deployed in the same physical sensor node, the total sum of their code sizes is around 9.3KB, which represents less than the 0.23% of the available Flash memory in such wireless device node.

TABLE 7.2: *Code size for Pervasive Service Platform*

Software artefact	Flash size (bytes)	Source lines of code	Occupancy percentage
IAgent interface	377	60	0.009
Pervasive in-network light agent	3348	181	0.080
Pervasive in-network noise agent	4761	277	0.114
Pervasive in-network temperature agent	3302	106	0.079
Pervasive Service Platform component	1740	113	0.041

TABLE 7.3: *Code size for Service Internetworking Platform*

Software artefact	Flash size (bytes)	Source lines of code	Occupancy percentage
PEVSS service orchestrator agent	6004	251	0.143
Service brokering agent	3480	118	0.083

7.3 Volatile Memory Footprint

As in the case of code storage capacity, volatile memory is a scarce resource in wireless sensor nodes. Typically, RAM size in these ad hoc devices varies from 10KB to 32MB depending on the platform, which imposes severe restrictions on data allocation and management procedures. According to the previous, we evaluate the footprint impact of volatile memory in our prototype using profiling analysis, in terms of average consumption of heap and non-heap memory in the Java Virtual Machine. On the one hand, heap memory defines the runtime and dynamic storage area for all class instances and arrays. Its usage is monitored by the garbage collector for freeing the memory associated with non-used objects. On the other hand, non-heap memory provides a shared memory area for all the threads, as well as for internal processing in the JVM. Additionally, per-class structures such as constants and method data, as well as code associated with methods and class constructors, are also stored in this memory area.

TABLE 7.4: *Volatile Memory Footprint*

Type of node	Heap memory (bytes)		Non-heap memory (bytes)	Occupancy percentage (worst-case)
	Maximum	Minimum		
Wireless sensor node	91208	78624	17048	20.648
Broker / Orchestrator node	131800	123376	17473	28.472

In this analysis, the average volatile memory overhead in both wireless sensor nodes and Broker/Orchestrator node is provided. The throughput in this test was one transaction per second (tps) in the wireless sensor nodes in terms of light, noise and temperature service data request/responses. This factor leads to a triple-transaction-rate per second (3tps) in the Broker/Orchestrator network node, according to the PEVSS abstraction and the deployment model discussed in Section 7.1.1. The obtained results appear in Table 7.4. This shows the minimum and maximum consumption thresholds of heap memory and average non-heap memory usage, also identifying the impact on occupancy percentage for the worst-case (i.e., maximum heap plus average non-heap) according to the features of the Sun SPOT platform. In the case of wireless sensor nodes, the average heap memory consumption was 84916 ± 6292 bytes, while the observed non-heap memory consumption was 17048 bytes. This led to a worst-case occupancy percentage equals to the 20.648% of the 512KB of available volatile memory. Regarding the Broker/Orchestrator node, the average heap memory occupancy raised up to 127588 ± 4212 bytes, while the observed non-heap memory consumption was 17473 bytes. This represents a worst-case occupancy percentage of volatile memory equals to the 28.472% of the available RAM. These data reflect a compact footprint of volatile memory consumption in sensor nodes, which is an essential requirement for such kind of ad hoc embedded computing

devices. Additionally, and according to the previous, there is a non-linear growth of volatile memory consumption in the case of the software on the wireless sensor nodes and for the Broker/Orchestrator node, where both the broker and PEVSS orchestrator agents coexist. This issue, together with the observed compact values regarding memory consumption in the nodes for both the non-volatile and volatile cases, enhances the potential deployment factor of sensor-based agents in our approach with a high expected degree of network scalability.

7.4 CPU Overhead

Novel sensor-based paradigms for the Information and Networked Society, as those listed in Section 1.1.2, underscore the need for devising more advanced and complex interaction patterns, which enable higher performance with in-network processing as a basic requirement. It demands suitable management of the pervasive business processes, maximizing their multi-programming and concurrence degrees while assuring an adequate usage of the processing resources. These improvements are founded on the evolution in processor technology for sensor networking. This typically goes from 8MHz, 8-bit architecture microcontrollers to 32-bits processors at more than 410MHz with dynamic voltage scaling techniques. In this section, our analysis is centred on the CPU performance in the wireless sensor and Broker/Orchestrator nodes, specifically in terms of processor overhead and threads management by using Java profiling analysis, as in the study of the previous section.

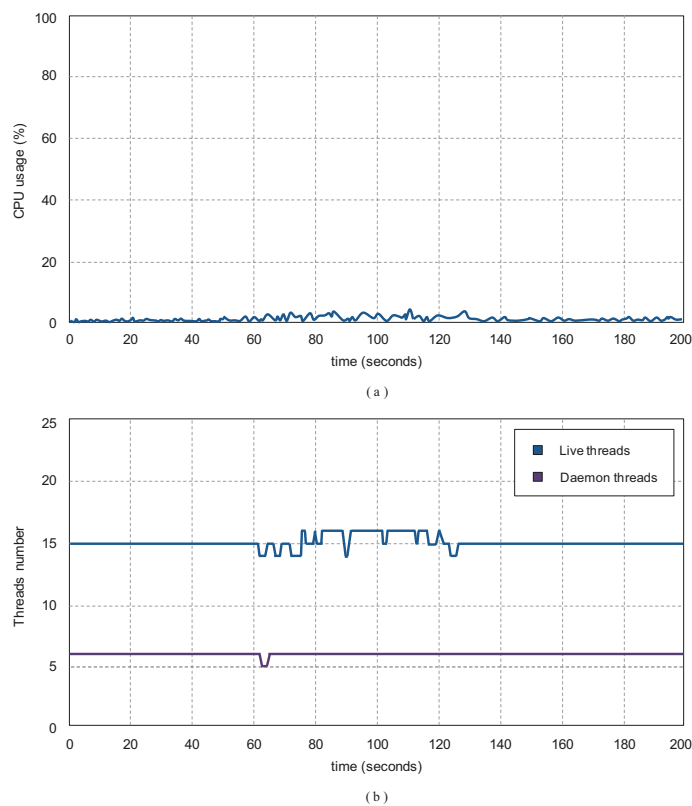


Figure 7.3: CPU overhead analysis in wireless sensor node: (a) processor usage; (b) threads management.

On the one hand, the CPU performance in the wireless sensor nodes during the evaluation is illustrated in Figure 7.3, according to two graphs regarding processor usage percentage and threads management, respectively. For both cases, there are two time intervals of interest: the first one is an idle period, which corresponds with time $t < 60s$ and $t > 120s$ in our evaluation, while the second one represents an operation period where the wireless sensor node is managing a traffic load equals to 1tps, corresponding with $60s < t < 120s$. First, and in the case of processor usage (see Figure 7.3.a), a very small delta in the CPU load between idle (peak processor usage: 2.9%) and traffic operation (peak processor usage: 4.6%) modes is appreciated. This issue highlights a negligible computing overhead for handling service requests/responses during transaction execution with the considered workload. Second, thread management is illustrated in Figure 7.3.b, differentiating between daemon threads (i.e., internal threads of the JVM) and live threads (i.e., daemon plus non-daemon ones). In the particular case of application or non-daemon threads, we observe that during the idle periods the number remains stable and equals to 9. This datum is related to the architectural design of the Service-Oriented Middleware Platform, the Pervasive Service Platform and the pervasive in-network agents implemented in the prototype. Furthermore, and during the traffic handling phase, the number of non-daemon threads balances between 8 and 10, being this fact closely associated with the management procedures used for handling incoming and outgoing traffic in the middleware. It is worth noting that the total number of threads keeps bounded during the evaluation, without noticeable signs of overflow or resource leakage.

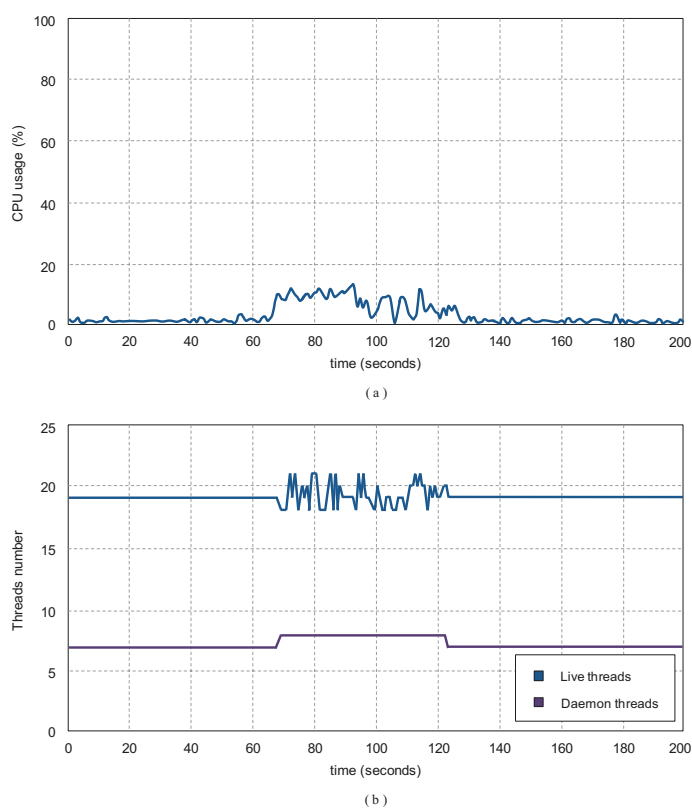


Figure 7.4: CPU overhead analysis in Broker/Orchestrator node: (a) processor usage; (b) threads management.

On the other hand, and in an analogous manner as in the previous case, Figure 7.4 illustrates the CPU performance considering the prototype evaluation of the Broker/Orchestrator node. First, and regarding processor usage (see Figure 7.4.a), we observe a remarkable increment between the idle (peak processor usage: 3.2%) and traffic operation (peak processor usage: 15.5%) periods compared with the previous study case. This issue is directly related to the number of tps that were served in the ad hoc wireless domain: 1tps in the case of wireless sensor nodes and 3tps in the case of Broker/Orchestrator nodes. However, the processing overhead introduced in this node was quite reduced, without any perceived performance degradation. Second, and regarding the thread management study (see Figure 7.4.b), during idle periods the number of non-daemon threads was 12, while in the case of periods of service traffic activity this value was between 11 and 14. As observed, such fluctuation occurs more frequently with a more pronounced sawtooth in this study than in the previous case. This is closely related to the action of the garbage collector, which is invoked for freeing resources. The number of threads during the test was bound without describing abrupt transitions or an asymptotic growth, showing a stable behaviour of the software.

7.5 Service Time Delay

The design of future smart spaces imposes remarkable engineering challenges regarding service responsiveness and latencies. The underlying heterogeneity of networks, devices and applications makes necessary to define mechanisms for achieving dynamic and adaptive data acquisition, processing and service execution. However, the resource-constrained nature of such ad hoc networks imposes to keep a close trade-off between the maximization of the throughput and the limited bandwidth of WSNs, typically operating with transceivers compliant with the IEEE 802.15.4 specification at 250kbps, 2.4GHz. In this regard, and using code instrumentation, we analyze our prototype in terms of pervasive in-network service interface compactness, and pervasive in-network service time delay for registering and round-trip times, as well as according to virtual sensor service instantiation and data acquisition times. The obtained results were calculated as the average of 1000 test executions for each corresponding use case.

First, service interface compactness is discussed in terms of size optimisation of the pervasive in-network services. To this end, our SMD/JSON proposal is compared with traditional WSDL and XML approaches, for both service description and service request/response data exchange, respectively. On the one hand, and regarding the service interface description comprising a normalized SMD definition of 93-bytes, the functional optimisation relationship with the corresponding WSDL document is equals to 2.6:1. This represents an overhead in the WSDL document of more than the double and a half of the SMD interface for the pervasive in-network service description at byte level. On the other hand, in the case of a service request/response message handshake of a 122-byte normalized SDM data interchange, the functional optimisation relationship against the related XML document instance is equals to 1.4:1. These data show a remarkable improvement obtained by using SMD interfaces to compact data volumes transferred over the radio interface, which is a critical aspect in order to minimize the impact on node computation, transmission delay, network overhead and battery consumption, as discussed in Section 6.3.1.

Second, we focus on service registering and round-trip times for data acquisition developed by the pervasive in-network sensor services within the wireless ad hoc domain, according to the nSOM inter-node communication model presented in Section 5.3.2. On the one hand, the average delay and related typical deviation for service registration in the Broker/Orchestrator node during the service discovery phase has been measured in 105.7 ± 30.1 ms. This figure involves the propagation of a HELLO message in the wireless network, the related processing in the broker agent for publishing and making it accessible at destination, as well as for sending to the orchestrator agent a SERV_AVAIL_NOTIF message for delivering the related RDF-compliant semantic annotations. On the other hand, the round-trip time is measured for the pervasive in-network services in our approach. This issue comprises a SERV_DATA_REQ, SERV_DATA_RESP message handshake of a simple service from the Broker/Orchestrator node to a wireless sensor node and back. According to the previous, the observed average delay and associated typical deviation in our evaluation is 171.2 ± 46.8 ms.

Finally, some additional measures of the PEVSS virtual sensor service for the completion time or virtual service instantiation and the virtual sensor acquisition time were performed. These tests follow the deployment assumption that both the brokering and PEVSS orchestrator agents were in the same node, as illustrated in Figure 7.1. On the one hand, the completion delay for virtual sensor service instantiation in the Broker/Orchestrator node comprises the time period from the reception of the pervasive in-network service descriptions in the broker agent until the service interface description of the virtual sensor service is received in the broker agent and published in the wireless network. This includes the required time for parsing the RDF annotations in SERV_AVAIL_NOTIF message and the instantiation of the orchestrator agent. The obtained figure for average delay and typical deviation in this test case is 41.9 ± 16.5 ms. On the other hand, the acquisition time for virtual sensor data includes the time period from the broker node send the SERV_DATA_REQ messages to the wireless sensor nodes until the related SERV_DATA_RESP messages are received and the virtual sensor measure is conformed in the orchestrator agent and dispatched to the broker agent. This has been measured in terms of average delay and typical deviation, being equals to 326.3 ± 80.4 ms. These values, together with the presented in the previous paragraph regarding discovery and data acquisition in the case of pervasive in-network services, show acceptable latencies and related typical deviations for satisfying the QoE expectations of end-users in their processes and transactions in terms of low delay and jitter.

7.6 Battery Lifetime

As highlighted in this dissertation, one of the most important limitations in ad hoc and sensor networks is energy consumption. In this regard, Wireless Sensor Networks have been devised for being suitable to work in remote or inaccessible geographical areas using power supplies with limited autonomy, where in most cases the node lifetime is highly dependent on the autonomy of the batteries. The resource-constraint and ubiquitous nature of these devices imposes the need for extremely low power consumption. Typically, sensor nodes are fed by AA or AAA batteries, while some novel wireless embedded platforms use power supply based on lithium-ion batteries. In this section, our analysis is focused on a theoretical estimation of battery

consumption in sensor nodes. In addition, this is completed with in-field data obtained from the evaluation study of the prototype using code instrumentation, as in the previous section.

According to the Oracle Sun SPOT specification [218], in active operation mode the CC2420 radio transceiver module consumes 20mA in RX and 18mA in TX at 0dBm. In the case of memory, the consumption of Flash and RAM during normal operation of the node is 22mA and 25mA, respectively. Based on the data sheet specification, normal operation of this platform leads to an average current consumption equals to 95mA in active mode, with all processors and radio running. Considering an average throughput equals to 1tps in the case of wireless sensor nodes, the resulting current consumption is 85,34mAh. Using a 750mA battery, as in the case of Oracle Sun SPOT nodes, the theoretical battery autonomy is around 8.79 hours working with a 100% duty-cycle. Applying the same analytical model to the Broker/Orchestrator node, which develops an average throughput of 3tps, the related current consumption is equals to 142.03mAh. This implies a battery lifetime of 5.28 hours with the same battery charge and duty-cycle percentage as in the previous case.

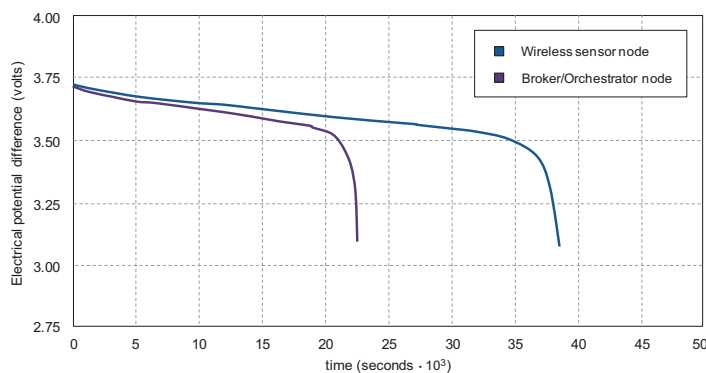


Figure 7.5: *Battery lifetime in wireless sensor and Broker/Orchestrator nodes.*

Obtained results of the prototype evaluation focussing on battery-autonomy metric are illustrated in Figure 7.5, for both the wireless sensor and the Broker/Orchestrator nodes. Using the *IBattery* API, the battery discharge curves were obtained based on the trace monitoring of the remaining capacity percentage. In this regard, the previous figure shows the electrical potential difference in the node battery versus the battery lifetime. In the case of the wireless sensor node, its autonomy was measured in around 10.64 hours, while in the case of the Broker/Orchestrator node it was around 6.24 hours. These values show a reduced autonomy in sensor nodes operation. It is worth noting, however, that this development platform is mainly focused on rapid prototyping of sensor-based applications. As stated in Section 7.1.2, one of the main limitations of the Oracle Sun SPOT is its short battery lifetime. According to the previous, the adoption of similar hardware platforms but using AA or AAA batteries in production deployments for providing higher current supply seems mandatory, in order to extend the network lifetime and autonomy. In case the previous were not enough, additional research on applying dynamic power management [292] and energy harvesting [267] strategies is proposed.

7.7 Conclusions

In this chapter, we reviewed the characterization analysis of the prototype approach regarding the nSOM and nSOL contributions, in terms of performance study. This represents the practical validation of the formalization and application models discussed in the previous chapters, with the aim of developing a dynamic service platform based on pervasive sensor networking. Besides the related deployment testbed and platform details, the obtained results are based on a set of software metrics widely accepted in embedded computing, according to the resource constraints of wireless sensor nodes. These metrics are code size, volatile memory footprint, CPU overhead, service time delay and battery lifetime. The rest of this section comprises a summary of the main issues regarding these aspects in the framework of this dissertation.

According to the research project demonstrator where this prototype is placed, a proof of concept regarding a Personal Environment Virtual Sensor Service (PEVSS) software artefact is provided. This is centred on building a ubiquitous smart space with simple, agent-based pervasive in-network services and their orchestration and exposure in a virtual sensor service. On the one hand, the nSOM architecture provides the middleware backbone for agent implementation, deployment and interworking in the wireless ad hoc domain and with the Internet cloud. On the other hand, semantically-driven service composition of pervasive in-network services into the PEVSS virtual sensor service is enabled with the nSOL proposal. This software prototype has been implemented using the Oracle Sun Small Programmable Object Technology (SPOT) platform. We highlight the most important strengths and limitations of this commercial hardware, including its development facilities. As a Java-based solution, Oracle Sun SPOT lies on a Java Virtual Machine that is called Squawk. This is a lightweight, small footprint JVM designed for consumer electronics and embedded devices.

Regarding the result analysis, we differentiate several use cases according to each specific performance metric analyzed in the characterization study. Following paragraphs highlight the most remarkable issues that underlie the validation of the proposed contributions.

First, we inspect the installed code size in the sensor nodes, which impacts on non-volatile memory overhead. This evaluation looks for code overhead and source lines of code of the Service-Oriented Middleware Platform, the Pervasive Service Platform and the Service Internetworking Platform. Obtained results illustrate a compact Flash footprint in the prototype implementation. In the case of the Service-Oriented Middleware Platform, the memory footprint was around 39KB (i.e., 0.9% of the available Flash memory in the Sun SPOT platform). Regarding the Pervasive Service Platform, the footprint measure was around 13KB (i.e., 0.3% of Flash memory). Finally, non-volatile memory overhead in the Service Internetworking Platform was around 9KB (i.e., 0.2% of the available Flash memory).

Second, an inspection of volatile memory footprint in the sensor devices was done. According to the memory model of the JVM, this study was centred on the average heap and non-heap memory overhead for both wireless sensor and Broker/Orchestrator nodes. This was accomplished with the aim to show the existing relationship between the allocated volatile memory and the available capacity in the nodes. Average value and typical deviation are provided in each measure. In the case of wireless sensor nodes, the average heap consumption

was around 83KB and the non-heap was below 17KB, with a worst-case consumption equals to the 20.6% of RAM occupancy of Sun SPOT nodes. In the case of Broker/Orchestrator node, the average heap consumption was around 125KB and the non-heap was equals to 17KB, thus leading to the 28.5% of the total amount of available volatile memory.

Third, an evaluation regarding CPU performance was conducted. This analysis includes idle and traffic operation modes, considering a throughput equals to 1tps in the case of wireless sensor nodes and 3tps in the case of Broker/Orchestrator node. In both cases, the processor load and threads management were monitored. In the case of wireless device node, the CPU load percentage was equals to 2.9% in idle mode and equals to 4.6% during traffic period, while in the case of Broker/Orchestrator node it was equals to 3.2% and 15.5%, respectively. Regarding management of user threads, a bounded behaviour without any sign of overflow or resource leakage was observed. In the case of wireless sensor nodes, the figure was balancing between 8 and 10, while in the case of Broker/Orchestrator node it was between 11 and 14.

Fourth, we study our prototype in terms of service time delay, focusing on analyze the trade-off between ubiquitous service responsiveness and bandwidth limitation in sensor devices. In the case of service compactness, we compared SMD/JSON with traditional WSDL and XML, where optimizations of 2.6:1 and 1.4:1 in the case of pervasive in-network services for service interface and request/response handshake, respectively, were reached. Regarding service registering and round-trip times for data acquisition by pervasive in-network sensor services in the wireless ad hoc domain, the average time for service registration in the Broker/Orchestrator node was 105.7 ± 30.1 ms, while the average round-trip time in pervasive in-network services was equals to 171.2 ± 46.8 ms. Additionally, the virtual sensor service instance was evaluated from the perspective of virtual service instantiation and virtual sensor acquisition times. The average completion delay for instantiating a virtual sensor service in the Broker/Orchestrator node was equals to 41.9 ± 16.5 ms, while the average acquisition time for virtual sensor data was 326.3 ± 80.4 ms.

Finally, a dissection of battery lifetime of the sensor network nodes was accomplished. The proposed study includes a theoretical analysis based on an energy model, as well as empirical data extracted from the in-field validation of the prototype. Provided results were obtained with a 100% duty-cycle, and traffic operation models of 1tps in the case of wireless sensor nodes and 3tps in the case of Broker/Orchestrator node. On the one hand, and considering the reduced current charge of the battery of the Sun SPOT platform, as well as its significant energy consumption during normal operation, the estimation of battery lifetime shows a constrained autonomy. In the case of wireless sensor nodes, the theoretical autonomy was around 8.8 hours while in the case of Broker/Orchestrator node it was 5.3 hours. These data were contrasted with the in-field results. The battery of the wireless sensor node was exhausted after 10.6 hours, while in the case of Broker/Orchestrator node it stops its execution after 6.2 hours of operation.

PART IV

CONCLUSIONS AND FURTHER
RESEARCH

Conclusions

*F*UTURE smart ecosystems for the Internet of Things should focus on redefining the way in which the users materialize their daily life experiences with themselves and with the others, and the rest will follow. A world of devices is certainly perceived as a world of connected people. In the coming years, additional achievements in pervasive computing will arrive but it is essential that those come by the hand of a progressive consolidation in end-user environments. This should be induced from both the academia and industry research domains with the aim of landing around us in an intimate and close manner, minimizing the distraction required to the users and anticipating to their needs and intentions, finally contributing to the germination of the so-called Information and Networked Society.

This chapter summarizes the main conclusions drawn from the contributions provided in this dissertation, while some related directions as further work are proposed. The rest is as follows. Section 8.1 highlights the major conclusions extracted from this doctoral thesis. Section 8.2 proposes some lines of further work. Finally, scientific publications as fundamental knowledge dissemination contributions of our research are listed in Section 8.3.

8.1 Conclusions

As one of the core findings of this dissertation, we research on the practical validation of how software and networking engineering based on the integration of service and agent orientation paradigms, semantics and knowledge management technologies can lead to enhance distributed service architecting and dynamic composition models for virtual service provisioning using sensor technology on the Internet of Things. Around the foregoing, more specific issues are discussed. In this regard, the following paragraphs present a summary of the main insights and proposed contributions of this essay.

Regarding the state-of-the-art analysis of research and development trends in Wireless Sensor Network technology, we focus on architecture and protocol engineering fundamentals, Service-Oriented Computing, Agent-Oriented Software Engineering, Semantic Knowledge Management and the Internet of Things.

The summary discussion of the previous is as follows. First, and regarding approaches in the sensor network protocol stack, novel proposals aim for energy-aware designs supporting data integration between layers while promoting collaboration. In this case, the real challenge is not to develop highly-optimised, specific-purpose solutions but to exploit all their potential for building adaptive, heterogeneous applications while optimizing their physical resources. Second, research roadmap in Service-Oriented Computing includes service foundations, service composition and service management. Service foundations are centred on service-oriented middleware for defining business-driven, vendor-neutral and heterogeneous core functionalities. Taking advantage of service composition, sensor virtualization based on intelligent and dynamic integration of services can be effectively implemented. In addition, service management enables self-adaptive properties and network resilience support in pervasive computing, thus promoting unattended operation and supervision. Third, Agent-Oriented Software Engineering creates loosely-coupled, automatic interactions of sensor-based business processes in a proactive manner. This allows decentralization of knowledge and computing, enhancing coordination between agent instances. Fourth, Semantic Knowledge Management engineering is motivated by the integration of several research areas, which include ontology management, knowledge discovery and human language technologies. The application of such in the development of distributed service platforms and service composition procedures enables more intelligent information access and management. Finally, the Internet of Things represents the framework for interweaving all the previous paradigms into pervasive smart ecosystems. This allows anytime/anywhere/anymedia communications in a wide range of application domains including transportation and logistics, healthcare, as well as personal and social.

In the context of the background discussion on wireless sensor technologies for developing distributed service platforms, we visit related work and approaches concerning service-oriented middleware and dynamic service composition techniques.

The summary discussion of the previous is as follows. On the one hand, middleware has been identified as a critical issue on emerging areas such as grid and cloud computing, as well as in mobile and pervasive networks. Traditional middleware architectures were usually devised for distributed environments with rich resources, so that new solutions specifically designed for addressing the complexity in sensor networks are required. However, most of sensor-based mainstream middleware proposals have been defined as ad hoc approaches. In order to deal with the previous, application of Service-Oriented Computing for middleware architecting is proposed. According to this paradigm, main requirements of future service-oriented middleware frameworks include platform and OS independence, general-purpose core functionalities, heterogeneous multi-service composition support, enable software application development and internetworking support with external service clouds. According to these challenges, some of the most remarkable proposals have been dissected. On the other hand, service composition is a property commonly highlighted of pervasive networking systems, where it is identified as a key

factor. Service composition in pervasive computing should address the inherent dynamicity of anytime/anywhere/anymedia information access, device heterogeneity as well as error-prone and unreliable communications between ad hoc nodes, where sensor-based services are coming up and down frequently. Such dynamic nature demands service delivery solutions able to adapt their composition by taking maximum advantage of the available services in the network at each moment. Regarding this, main requirements of future sensor-based dynamic composition platforms in service-oriented solutions have been identified. These include lightweight service composition languages, adaptive service composition, semantically-driven service integration based on ontologies, virtual sensor services, and service-based integration with the Internet of Things. Based on these topics, related proposals in this field are surveyed.

Based on the previous state-of-the-art analysis contributions, the corresponding formalization contribution of our research work is defined. It provides an abstract formalization model for both service-oriented architecting and dynamic composition in pervasive sensor networks, attending to the design challenges previously distilled.

The summary discussion of the previous is as follows. On the one hand, a reference service-oriented architecture model for heterogeneous sensor-based devices and embedded computing is proposed. This reference approach defines five abstraction levels or platforms: Physical Device Platform, Service-Oriented Middleware Platform, Pervasive Service Platform, Service Internetworking Platform and Development Support Platform. The Physical Device Platform focuses on vendor-neutrality based on the encapsulation of specific details regarding hardware, operating system and network protocol stack. The addressing a reusable middleware abstraction with general-purpose backbone functionalities is laid down in the Service-Oriented Middleware Platform. Multi-service composition support is implemented in the local nodes by the Pervasive Service Platform, enabling dynamic composition of virtual sensor services between nodes with the Internetworking Service Platform, which also allows internetworking with external service clouds. Finally, programmatic methodologies and practical approaches for making easier the implementation of the middleware and distributed sensor-based service entities in an open, rapid, low-cost evolvable and massively manner are covered by the Development Support Platform. On the other hand, a reference model for virtual sensor composition based on sensing and embedded computing is formalized as the service integration paradigm implemented in the reference service-oriented framework. It proposes, in a compliant manner with the concepts of service profile, process model and service grounding in the OWL-S specification, the notion of pervasive in-network service PS_i . Additionally, the definition of its capabilities in terms of discrete business logic, transition function, expected result, input and output operations, and service preconditions is included in this proposal. This approach is extended with the formal definition of a pervasive virtual sensor service $PVSS_i$, enabling composition of atomic pervasive in-network service processes, while providing access information. Finally, there is the formal definition of the Pervasive Service Platform PSP . This is proposed according to the discrete business logic of service composition, pervasive in-network and virtual sensor services, the scenario environment in a given time, and the set of interfaces with the adjacent platforms of the service-oriented framework.

As the first practical application contribution, the nSOM (nano Service-Oriented Middleware) is discussed. It represents our concrete realization of the reference service-oriented framework. It addresses the middleware design and implementation, including the agent-based workflow for both service exposure and discovery in the intra-node and inter-node interworking models.

The summary discussion of the previous is as follows. Our nSOM proposal adopts a component-based architecture in the Service-Oriented Middleware Platform, implementing a pervasive in-network service approach using intelligent software agents in the Pervasive Service Platform and Service Internetworking Platform. First, the service-oriented middleware defines a set of general-purpose core functionalities that include control services, low-level services, high-level services and cross services. These enable to the upper-layers a SOA framework for rapid, interoperable and easy assembly business application processes, where agent-based services are defined as autonomous, reusable, highly portable and vendor-neutral entities. Second, intra-node communication mechanisms between software components and application entities, as well as inter-node communication procedures, are implemented. Regarding the intra-node interaction, the Pervasive Service Platform constitutes an agent-oriented container abstraction to deploy and execute pervasive in-network services based on aspect-oriented programming design. This layer also provides capability to discover dynamically middleware components in the architecture at runtime, as well as a uniform procedure for service interaction based on context-aware information. Finally, this contribution addresses multi-service composition and internetworking with external service networks based on inter-node interaction. This support is implemented in the Orchestrator and Broker network entities of the Service Internetworking Platform, which are modelled as agent-based processes. According to the communication between the previous and wireless device nodes in the ad hoc domain, as well as with the service-oriented Application Service Platform via a Sink/Gateway node to the Internet cloud, a service exposure and discovery protocol for embedded networking has been accomplished, based on an nSOM-PDU definition optimized for radio interface transmission.

The above is completed with our second practical application contribution, which has been called nSOL (nano Semantics-Oriented Language). It defines a composition-centric solution for instantiating virtual sensor services according to an ontology model for semantically-driven pervasive composition. This relies on the definition of a ubiquitous service interface language and a lightweight notation syntax proposal for building ubiquitous services.

The summary discussion of the previous is as follows. The nSOL approach proposes an implementation for enabling ubiquitous service composition and provisioning in a dynamic and adaptive manner, with the aim of building smart infrastructures for the Internet of Things. The proposed concrete formalisation to model pervasive in-network services and aggregate them into virtual sensor services is provided, according to a scheme for ontology-based semantic knowledge management. This allows annotations to enhance the composition, discovery, and invocation of sensor-based ubiquitous services. In order to minimize the overhead in data transmission and processing, the use of the JSON standard as a lightweight, language-independent and self-describing data representation and exchange format is proposed. In addition, the Service Mapping Description solution for defining service descriptions and invocation interfaces in pervasive smart environments is applied. The SMD service interface

descriptions are annotated according to the ontology-based graph data model using RDF N3. This proposal enables a compact syntax that provides a simple manner to express triples while reducing the language overhead. According to the previous, the service composition workflow is defined. It is executed by the orchestrator agents, which instantiate the virtual sensor services at runtime according to the composition template, the advertisements stored in the designated broker agent and the service candidates obtained from the outcome of the election criteria algorithm. These virtual service abstractions are registered on the Internet cloud through the service-oriented Application Server Platform. This focuses on providing effective, dynamic, agile and easy integration of business processes into scalable distributed networks based on open, standard and multi-interface service connectivity.

To conclude, the performance characterization of the previous contributions is conducted. This addresses from the testbed definition for the prototype to the software engineering metrics applied in the experimental validation. These include code size, volatile memory footprint, CPU overhead, service time delay and battery lifetime.

The summary discussion of the previous is as follows. A prototype for a Personal Environmental Virtual Sensor Service (PEVSS) has been developed using the Oracle Sun SPOT programmable platform, according to the objectives of the research project demonstrator where it is placed. This composite service is based on the integration of some pervasive in-network services using agent technology that include temperature, light and noise. Based on the result analysis, a lightweight implementation with significant good performance has been developed for service architecting and dynamic virtual sensor service provisioning. It includes a compact interface definition with proficient sensor-based service responsiveness and low service latencies. Regarding code size, non-volatile memory footprint is less than the 1.5% of the available Flash memory in sensor nodes, while in the case of volatile memory the related measure for the worst-case of RAM consumption led to a occupancy percentage equals to the 20.6% in the case of wireless sensor nodes and the 28.5% in the case of Broker/Orchestrator node. In respect of CPU performance, at 1 transaction per second in the wireless sensor nodes the processor load rises to the 4.6%, and the 15.5% in the case of Broker/Orchestrator node at 3 transactions per second with a bounded thread and resource management. With regard to service time delay, a comparison between SMD/JSON and traditional WSDL and XML shows an optimization equals to 2.6:1 and 1.4:1 for service interface definition and data request/response handshake, respectively. The average service registration delay in the Broker/Orchestrator node was equals to 105.7 ± 30.1 ms, and the round-trip time for pervasive in-network service was equals to 171.2 ± 46.8 ms. In addition, the average completion time for virtual sensor service instantiation in the Broker/Orchestrator node was equals to 41.9 ± 16.5 ms, while the acquisition time for virtual sensor data was equals to 326.3 ± 80.4 ms. Finally, the battery lifetime of the sensor nodes shows a restricted autonomy when operating at 100% duty-cycle in this prototype-oriented platform for sensor networks, being approximately between 6.2 and 10.6 hours depending on the experimented number of transactions per second.

8.2 Further Research and Directions

On the roadmap towards the consolidation of pervasive smart spaces and environments on the Internet of Things, an open issue to be addressed in the context of ubiquitous computing is the autonomic service management. According to a future perspective of a worldwide network composed by billions of personal devices, as well as pervasive environments where countless composition-centric services coexist, ensuring survivability and service availability is a critical issue. In our best knowledge, such advanced service management imposes the need for lightweight, dynamic and robust solutions implementing an autonomous monitoring plane with specific functions. These include procedures to assure self-healing (resilient mechanisms for diagnosis, self-repair and fault-correction), self-optimization (reconfiguring capabilities in order to optimize its related performance) and self-protection (proactive detection, identification and protection against security threads and attacks) properties on the network, among others.

In this regard, we devise the research on autonomic and pervasive resilience in wireless sensor computing for providing adaptive and intelligent control planes as a cutting-edge challenge, leading to enable autonomic self-management of data planes with service-orientation and inter-domain interoperability. Attending to the heterogeneous nature of personal and ubiquitous devices and networks, as well as to the potential application areas of smart infrastructures, the design and development of such autonomic control planes should balance between disparate service requirements in terms of traffic parameters (e.g., delay, jitter, bandwidth, packet loss) on the one hand, and the available pervasive network resources in terms of QoS parameters (e.g., queue length, radio link throughput, node energy) on the other hand. Around this knowledge area, several issues can be identified to build such vision. On the one hand, QoS-aware ontologies can be applied for describing both service requirements and sensor network resources according to a common model for defining Service Level Agreements (SLA). This enables service and network data to be processed, shared and reused between applications, allowing easier integration with advanced computing capabilities and related technologies. Additionally, compact query languages optimized for embedded computing can be applied for accessing and retrieving information related to service requirements and resource availability in the network. In this regard, semantic query languages designed for addressing the performance requirements in embedded and ubiquitous environments are needed. On the other hand, and with the aim of guaranteeing the dynamicity and elasticity of the interactions in the pervasive network, both service interfaces and node resources can be semantically annotated by using lightweight syntaxes in a compliant manner with the QoS-aware ontology. In this regard, research should be done not only on compact syntaxes optimized for ubiquitous computing in resource-constraint devices, but also on extending the previous languages for adding semantic value to them with embedded annotations. Besides, open and standard semantic approaches are requested for enabling native inter-domain interoperability. Finally, network paths to connect users and services fulfilling the specific QoS for each service flow should be calculated at runtime, according to the specific service requirements, node availability and resources, as well as network conditions. In this regard, multi-constraint path routing can be applied for building a QoS-aware overlay network for each service, in an on-demand and adaptive manner. The best path can be selected dynamically from the calculated overlay applying single-path, multi-path

or opportunistic routing in order to deliver the service data to the end-user timely, also with the possibility of early calculation of backup routes that can be used to enable path swapping for degraded service flows. This can lead to fast switching of routes used by the service flows with minimal disruption and data loss, as a resilient solution for addressing network failures in order to assure an acceptable degree of service provisioning.

8.3 List of Publications

Some of the main outcomes of this dissertation have appeared in the following publications. These constitute our fundamental knowledge dissemination contributions, which are the result of the research activity of the author in the area where this doctoral thesis has been carried out.

- Familiar, M. S., Martínez, J. F. and López, L., "Pervasive Smart Spaces and Environments: A Service-Oriented Middleware Architecture for Wireless Ad Hoc and Sensor Networks," *International Journal of Distributed Sensor Networks* 2012(725190) (2012) 1-11. Impact Factor: 0.203. JCR (70/78), category: Telecommunications.
- Familiar, M. S., Martínez, J. F., Corredor, I. and García-Rubio, C., "Building Service-Oriented Smart Infrastructures over Wireless Ad Hoc Sensor Networks: A Middleware Perspective," *Computer Networks* 56(4) (2012) 1303-1328. Impact Factor: 1.176. JCR (23/80), category: Telecommunications.
- Corredor, I., Martínez, J. F. and Familiar, M. S., "Bringing Pervasive Embedded Networks to the Service Cloud: A Lightweight Middleware Approach," *Journal of Systems Architecture* 57(10) (2011) 916-933. Impact Factor: 0.722. JCR (31/49), category: Computer Science, Hardware and Software.
- Corredor, I., Martínez, J. F., Familiar, M. S. and López, L., "Knowledge-Aware and Service-Oriented Middleware for Deploying Pervasive Services," *Journal of Network and Computer Applications* 35(2) (2012) 562-576. Impact Factor: 1.111. JCR (52/93), category: Computer Science, Software Engineering.
- Martínez, J. F., Familiar, M. S., Corredor, I., García, A. B., Bravo, S. and López, L., "Composition and Deployment of e-Health Services over Wireless Sensor Networks," *Mathematical and Computer Modelling* 53(3) (2011) 485-503. Impact Factor: 1.103. JCR (62/202), category: Applied Mathematics.
- Martínez, J. F., Bravo, S., García, A. B., Corredor, I., Familiar, M. S., López, L., Hernández, V. and Da Silva, A., "Pervasive Surveillance-Agent System based on Wireless Sensor Networks: Design and Deployment," *Measurement Science and Technology* 21(12) (2010) 1-10. Impact Factor: 1.317. JCR (20/79), category: Multidisciplinary Engineering.

References

- [1] Abangar, H., Ghader, M., Gluhak, A. and Tafazolli, R., "Improving the performance of web services in wireless sensor networks," In: Proc. Future Network and Mobile Summit, 2010, pp. 1-8.
- [2] AbdelSalam, H. S. and Olariu, S., "BEES: Bio-Inspired Backbone Selection in Self-Organizing Wireless Sensor Networks," IEEE Transactions on Parallel and Distributed Systems 23(1) (2012) 44-51.
- [3] AbdelSalam, H. S., Rizvi, S. R. and Olariu, S., "Energy-aware task assignment and data aggregation protocols in wireless sensor networks," In: Proc. IEEE Consumer Communications and Networking Conference, 2009, pp. 1-5.
- [4] Abdelzاهر, T., Blum, B., Cao, Q., Chen, Y., Evans, D., George, J., George, S., Gu, L., He, T., Krishnamurthy, S., Luo, L., Son, S., Stankovic, J., Stoleru, R. and Wood, A., "EnviroTrack: Towards an environmental computing paradigm for distributed sensor networks," In: Proc. IEEE International Conference on Distributed Computing Systems, 2004, pp. 582-589.
- [5] Abielmona, R., Petriu, E. M., Harb, M. and Wesolkowski, S., "Mission-Driven Robotic Intelligent Sensor Agents for Territorial Security," IEEE Computational Intelligence Magazine 6(1) (2011) 55-67.
- [6] Abrougui, A., Mercier, A., Occello, M. and Huget, M. P., "Recursive multi-agent system for dynamic and adaptative web services composition," In: Proc. International Conference on Management of Emergent Digital Ecosystems, 2009, pp. 295-299.
- [7] Achuthan, M., Balachandran, B. and Sharma, D., "Multi-agents for decision support," In: Proc. International Conference on Knowledge-Based Intelligent Information & Engineering Systems, 2004, pp. 469-475.
- [8] Ackoff, R. L., "From Data to Wisdom," Journal of Applied Systems Analysis 16(1) (1989) 3-9.
- [9] Adelstein, F., Gupta, S. K., Richard, G. G. and Schwiebert, L., Middleware for application development: Adaptation and agents," in: Chapman, S. S. (Ed.) Fundamentals of Mobile and Pervasive Computing, McGraw-Hill, New York, United States, 2005, pp. 113-135.

- [10] Ahmad, A., Ahmad, M. S. and Yusoff, M. Z., "An exploratory review of software agents," In: Proc. International Symposium on Information Technology, 2008, pp. 1-8.
- [11] Aijaz, F., Adeli, S. M. and Walke, B., "A service-oriented approach for in-network computations in wireless networks," In: Proc. IFIP International Conference on Wireless and Optical Communications Networks, 2009, pp. 1-6.
- [12] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. and Cayirci, E., "A Survey on Sensor Networks," IEEE Communications Magazine 40(8) (2002) 102-114.
- [13] Akyildiz, I. F., Melodia, T. and Chowdury, K. R., "Wireless Multimedia Sensor Networks: A Survey," IEEE Wireless Communications 14(6) (2007) 32-39.
- [14] Alavi, M. and Leidner, D. E., "Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues," MIS Quarterly 25(1) (2001) 107-136.
- [15] Alavi, M. and Leidner, D. E., Knowledge management systems: Theory and practice," in: Barnes, S. (Ed.) Knowledge Management Systems: Issues, Challenges and Benefits, Thomson Learning, Oxford, United Kingdom, 2002, pp. 15-35.
- [16] Aldewereld, H., Padget, J., Vasconcelos, W., Vazquez-Salceda, J., Sergeant, P. and Staikopoulos, A., "Adaptable, Organization-Aware, Service-Oriented Computing," IEEE Intelligent Systems 25(4) (2010) 26-35.
- [17] Alexander, R., "The Real Costs of Aspect-Oriented Programming," IEEE Software 20(6) (2003) 92-93.
- [18] Almalkawi, I. T., Guerrero, M., Al-Karaki, J. N. and Morillo-Pozo, J., "Wireless Multimedia Sensor Networks: Current Trends and Future Directions," Sensors 10(7) (2010) 6662-6717.
- [19] An, X. and Wang, W., "Knowledge management technologies and applications: A literature review," In: Proc. IEEE International Conference on Advanced Management Science, 2010, pp. 138-141.
- [20] Anand, A. and Singh, M. D., "Understanding Knowledge Management: A Literature Review," Journal of Engineering Science and Technology 3(2) (2011) 926-939.
- [21] Apel, S., Leich, T. and Saake, G., "Aspectual Feature Modules," IEEE Transactions on Software Engineering 34(2) (2008) 162-180.
- [22] Aquino-Lugo, A. A., Klump, R. and Overbye, T. J., "A Control Framework for the Smart Grid for Voltage Support Using Agent-Based Technologies," IEEE Transactions on Smart Grid 2(1) (2011) 173-180.
- [23] Arel, I., Liu, C., Urbanik, T. and Kohls, A. G., "Reinforcement Learning-based Multi-Agent System for Network Traffic Signal Control," IET Intelligent Transport Systems 4(2) (2010) 128-135.
- [24] Arnold, K., Sullivan, B., Scheifler, R., Waldo, J. and Wollrath, A., The Jini Specification. Addison-Wesley, Boston, United States, 2001.
- [25] Atzori, L., Iera, A. and Morabito, G., "The Internet of Things: A Survey," Computer Networks 54(15) (2010) 2787-2805.
- [26] Augusto, J. C., Nakashima, H. and Aghajan, H., "Ambient intelligence and smart environments: A state of the art," in: Nakashima, H., Aghajan, H. and Augusto, J. C. (Eds.) Handbook of Ambient Intelligence and Smart Environments, Springer-Verlag, New York, United States, 2010, pp. 3-31.

-
- [27] Avilés-López, E. and García-Macías, J. A., "TinySOA: A Service-Oriented Architecture for Wireless Sensor Networks," *Service Oriented Computing and Applications* 3(2) (2009) 99-108.
- [28] Bai, Y., Ji, H., Han, Q., Huang, J. and Qian, D., "MidCASE: A service oriented middleware enabling context awareness for smart environment," In: *Proc. International Conference on Multimedia and Ubiquitous Engineering*, 2007, pp. 946-951.
- [29] Bai, Y., Ji, H., Han, Q., Huang, J. and Zhang, Z., "Towards a Service-Oriented Middleware Enabling Context Awareness for Smart Environment," *International Journal of Ad Hoc and Ubiquitous Computing* 4(1) (2009) 24-35.
- [30] Bakhouya, M., "Adaptive Service Discovery and Composition in Ubiquitous and Pervasive Computing," *ACM Transactions on Autonomous and Adaptive Systems* 6(1) (2011) 1-3.
- [31] Bakhouya, M. and Gaber, J., "A propitious multi-agent system for spontaneous service emergence in pervasive computing environments," In: *Proc. IEEE International Conference on Pervasive Services*, 2007, pp. 409-414.
- [32] Bakhouya, M. and Gaber, J., "Service composition approaches for ubiquitous and pervasive computing: A survey," in: Li, E. Y. and Yuan, S. T. (Eds.) *Agent Systems in Electronic Business*, IRMA International, Pennsylvania, United States, 2008, pp. 323-350.
- [33] Bandyopadhyay, S., Sengupta, M., Maiti, S. and Dutta, S., "Role of Middleware for Internet of Things: A Study," *International Journal of Computer Science & Engineering Survey* 2(3) (2011) 94-105.
- [34] Baumer, C. and Magedanz, T., "Grasshopper: A mobile agent platform for active telecommunication," In: *Proc. International Workshop on Intelligent Agents for Telecommunication Applications*, 1999, pp. 19-32.
- [35] Becker-Asano, C., *WASABI: Affect Simulation for Agents with Believable Interactivity*. IOS Press, Amsterdam, The Netherlands, 2008.
- [36] Bellifemine, F. L., Caire, G. and Greenwood, D., *Developing Multi-Agent Systems with JADE*. John Wiley and Sons, Sussex, United Kingdom, 2007.
- [37] Belqasmi, F., Glitho, R. and Chunyan Fu, "RESTful Web Services for Service Provisioning in Next-generation Networks: A Survey," *IEEE Communications Magazine* 49(12) (2011) 66-73.
- [38] Berners-Lee, T., Connolly, D., Kagal, L. and Scharf, Y., "N3Logic: A Logical Framework For the World Wide Web," *Theory and Practice of Logic Programming* 8(3) (2008) 249-269.
- [39] Bhakti, M. A. and Abdullah, A., "Nature-inspired self organizing service oriented architecture: A proposal," In: *Proc. International Conference on Information Technology in Asia*, 2009, pp. 1-5.
- [40] Bhuvaneswari, A. and Karpagam, G. R., "Reengineering semantic web service composition in a mobile environment," In: *Proc. International Conference on Recent Trends in Information, Telecommunication and Computing*, 2010, pp. 227-230.
- [41] Blake, M. B., Tan, W. and Rosenberg, F., "Composition as a Service," *IEEE Internet Computing* 14(1) (2010) 78-82.
- [42] Bose, R., "Sensor Networks Motes, Smart Spaces, and Beyond," *IEEE Pervasive Computing* 8(3) (2009) 84-90.
-

- [43] Bose, R. and Helal, A., "Distributed mechanisms for enabling virtual sensors in service oriented intelligent environments," In: Proc. IET International Conference on Intelligent Environments, 2008, pp. 1-8.
- [44] Bouhouchi, R. and Ezzedine, T., "Wireless Sensors Networks Architecture," International Journal of Emerging Trends & Technology in Computer Science 1(4) (2012) 130-134.
- [45] Bouras, C., Philopoulos, A. and Tsiatsos, T., "A networked intelligent distributed virtual training environment: A first approach," In: Proc. Intelligent Multimedia Computing And Networking, 2000, pp. 604-607.
- [46] Brazier, F. M., Kephart, J. O., Van Dyke Parunak, H. and Huhns, M. N., "Agents and Service-Oriented Computing for Autonomic Computing: A Research Agenda," IEEE Internet Computing 13(3) (2009) 82-87.
- [47] Brittenham, P., Cutlip, R. R., Draper, C., Miller, B. A., Choudhary, S. and Perazolo, M., "IT Service Management Architecture and Autonomic Computing," IBM Systems Journal 46(3) (2007) 565-581.
- [48] Broll, G., Rukzio, E., Paolucci, M., Wagner, M., Schmidt, A. and Hussmann, H., "Perci: Pervasive Service Interaction with the Internet of Things," IEEE Internet Computing 13(6) (2009) 74-81.
- [49] Bronsted, J., Hansen, K. M. and Ingstrup, M., "Service Composition Issues in Pervasive Computing," IEEE Pervasive Computing 9(1) (2010) 62-70.
- [50] Brustoloni, J. C., *Autonomous Agents: Characterization and Requirements*. Carnegie Mellon Press, Pittsburgh, United States, 1991.
- [51] Buckl, C., Sommer, S., Scholz, A., Knoll, A. and Kemper, A., "Generating a tailored middleware for wireless sensor network applications," In: Proc. International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing, 2008, pp. 162-169.
- [52] Buettner, M., Yee, G. V., Anderson, E. and Han, R., "X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks," In: Proc. ACM International Conference on Embedded Networked Sensor Systems, 2006, pp. 307-320.
- [53] Buford, J., Kumar, R. and Perkins, G., "Composition trust bindings in pervasive computing service composition," In: Proc. International Conference on Pervasive Computing and Communications Workshops, 2006, pp. 261-266.
- [54] Burg, B., "Agent in the world of active web-services," in: Tanabe, M., Van den Beseelaar, P. and Ishida, T. (Eds.) *Digital Cities II: Computational and Sociological Approaches*, Springer-Verlag, Heidelberg, Germany, 2002, pp. 343-348.
- [55] Cañete, E., Chen, J., Díaz, M., Llopis, L. and Rubio, B., "A service-oriented middleware for wireless sensor and actor networks," In: Proc. International Conference on Information Technology: New Generations, 2009, pp. 575-580.
- [56] Cañete, E., Chen, J., Diaz, M., Llopis, L. and Rubio, B., "A Service-Oriented Approach to facilitate WSN Application Development," *Ad Hoc Networks* 9(3) (2011) 430-452.
- [57] Cao, J., "Flexible data collecting over distributed sensors on enterprise service bus," In: Proc. International Conference on Frontier of Computer Science and Technology, 2009, pp. 226-230.
- [58] Cao, J., and Siebert, J., "Service management in pervasive computing environments," in: Obaidat, M. S., Denko, M. and Woungang, I. (Eds.) *Pervasive Computing and Networking*, John Wiley & Sons, Sussex, United Kingdom, 2011, pp. 51-70.

-
- [59] Cao, Q., Abdelzaher, T., Stankovic, J. and He, T., "The LiteOS operating system: Towards unix-like abstractions for wireless sensor networks," In: Proc. International Conference on Information Processing in Sensor Networks, 2008, pp. 233-244.
- [60] Caporuscio, M., Raverdy, P. and Issarny, V., "ubiSOAP: A Service Oriented Middleware for Ubiquitous Networking," IEEE Transactions on Services Computing 5(1) (2012) 86-98.
- [61] Caputo, D., Mainetti, L., Patrono, L. and Vilei, A., "Implementation of the EXI schema on wireless sensor nodes using contiki," In: Proc. International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2012, pp. 770-774.
- [62] Chakraborty, D., Perich, F., Joshi, A., Finin, T. and Yesha, Y., "A reactive service composition architecture for pervasive computing environments," In: Proc. Personal Wireless Communications Conference, 2002, pp. 53-62.
- [63] Chappell, D. A., Introduction to the enterprise service bus," in: Hendrickson, M. (Ed.) Enterprise Service Bus, O'Really Media, Sebastopol, 2004, pp. 1-19.
- [64] Chaves, L. W. and Nochta, Z., "Breakthrough towards the internet of things," in: Ranasinghe, D. C., Sheng, Q. Z. and Zeadally, S. (Eds.) Unique Radio Innovation for the 21st Century, Springer-Verlag, Berlin, Germany, 2010, pp. 25-38.
- [65] Chen, J. and Huang, L., "Formal verification of service composition in pervasive computing environments," In: Proc. Asia-Pacific Symposium on Internetware, 2009, pp. 1-5.
- [66] Chen, X., Makki, K., Yen, K. and Pissinou, N., "Sensor Network Security: A Survey," IEEE Communications Surveys & Tutorials 11(2) (2009) 52-73.
- [67] Chien, T. V., Chan, H. N. and Huu, T. N., "A Comparative Study on Hardware Platforms for Wireless Sensor Networks," International Journal on Advanced Science, Engineering and Information Technology 1(2) (2012) 70-74.
- [68] Chopra, A. and Singh, M., "Elements of a business-level architecture for multiagent systems," in: Braubach, L., Briot, J. P. and Thangarajah, J. (Eds.) Programming Multi-Agent Systems, Springer-Verlag, Berlin, Germany, 2010, pp. 15-30.
- [69] Cios, K. J., Pedrycz, W. and Swiniarski, R. W., Data Mining: A Knowledge Discovery Approach. Springer, New York, United States, 2007.
- [70] Coloberti, M., Lombriser, C., Roggen, D., Troester, G., Guarneri, R. and Riboni, D., "Service discovery and composition in body area networks," In: Proc. International Conference on Body Area Networks, 2008, pp. 1-4.
- [71] Corredor, I., Martínez, J. F. and Familiar, M. S., "Bringing Pervasive Embedded Networks to the Service Cloud: A Lightweight Middleware Approach," Journal of Systems Architecture 57(10) (2011) 916-933.
- [72] Corredor, I., Martínez, J. F., Familiar, M. S. and López, L., "Knowledge-Aware and Service-Oriented Middleware for Deploying Pervasive Services," Journal of Network and Computer Applications 35(2) (2012) 562-576.
- [73] Cortada, J. W. and Woods, J. A., The Knowledge Management Yearbook 2000-2001. Butterworth-Heinemann, Massachusetts, United States, 2000.
- [74] Cucurull, J., Martí, R., Navarro-Arribas, G., Robles, S., Overeinder, B. and Borrell, J., "Agent mobility architecture based on IEEE-FIPA standards," Computer Communications 32(4) (2009) 712-729.
-

- [75] Cunningham, H., and Bontcheva, K., "Knowledge Management and Human Language: Crossing the Chasm," *Journal of Knowledge Management* 9(5) (2005) 108-131.
- [76] D'Mello, D. A. and Ananthanarayana, V. S., "Dynamic Web Service Composition Based on Operation Flow Semantics," *International Journal of Computer Applications* 1(26) (2010) 1-9.
- [77] Da Silva, R. I., Duca Almeida, V., Poersch, A. M. and Nogueira, J. M., "Wireless sensor network for disaster management," In: *Proc. IEEE Network Operations and Management Symposium (NOMS)*, 2010, pp. 870-873.
- [78] Dalkir, K., *Knowledge Management in Theory and Practice*. Butterworth-Heinemann, Oxford, United Kingdom, 2005.
- [79] Dar, K., Taherkordi, A., Rouvoy, R. and Eliassen, F., "Adaptable service composition for very-large-scale internet of things systems," In: *Proc. IFIP/USENIX International Middleware Conference*, 2011, pp. 1-6.
- [80] Darwish, A. and Hassanien, A. E., "Wearable and Implantable Wireless Sensor Network Solutions for Healthcare Monitoring," *Sensors* 11(1) (2011) 5561-5595.
- [81] Davenport, T. H., De Long, D. and Beers, M., "Successful Knowledge Management Projects," *Sloan Management Review* 39(2) (1998) 43-57.
- [82] Davidson, E. M., McArthur, S. D. J., McDonald, J. R., Cumming, T. and Watt, I., "Applying Multi-Agent System Technology in Practice: Automated Management and Analysis of SCADA and Digital Fault Recorder Data," *IEEE Transactions on Power Systems* 21(2) (2006) 559-567.
- [83] Davies, J., Grobelnik, M. and Mladenić, D., *Semantic Knowledge Management: Integrating Ontology Management, Knowledge Discovery, and Human Language Technologies*. Springer-Verlag, Heidelberg, Germany, 2009.
- [84] Davies, J., Lytras, M. and Sheth, A. P., "Semantic-Web-Based Knowledge Management," *IEEE Internet Computing* 11(5) (2007) 14-16.
- [85] Del Cid, P. J., Michiels, S., Joosen, W. and Hughes, D., "Middleware for resource sharing in multi-purpose wireless sensor networks," In: *Proc. International Conference on Networked Embedded Systems for Enterprise Applications*, 2010, pp. 1-8.
- [86] Delaney, K., Senturia, S., Delaney, K. and Dobson, S., "Augmenting materials to build cooperating objects," in: Delaney, K. (Ed.) *Ambient Intelligence with Microsystems*, Springer, New York, United States, 2009, pp. 19-46.
- [87] Di Flora, C., Gulp, J. C. and Prehofer, C., "Towards effective smart space application development: Impediments and research challenges," In: *Proc. International Conference on Pervasive Computing: Common Models and Patterns for Pervasive Computing Workshop*, 2007, pp. 1-2.
- [88] Di Nitto, E., Ghezzi, C., Metzger, A., Papazoglou, M. and Pohl, K., "A Journey to Highly Dynamic, Self-Adaptive Service-Based Applications," *Automated Software Engineering* 15(3) (2008) 313-341.
- [89] Dingsoyr, T., Bjornson, F. O. and Shull, F., "What Do We Know about Knowledge Management? Practical Implications for Software Engineering," *IEEE Software* 26(3) (2009) 100-103.
- [90] DiYSE European Research Project Do-it-Yourself Smart Experiences. ITEA2 code: 08005 (last accessed on: January, 2014) <www.dyse.org>.

-
- [91] Dojo Foundation Service Mapping Description Proposal (last accessed on: January, 2014) <www.dojotoolkit.org>.
- [92] Dorigo, M., Birattari, M. and Stutzle, T., "Ant Colony Optimization," IEEE Computational Intelligence Magazine 1(4) (2006) 28-39.
- [93] Dressler, F. and Akan, O. B., "Bio-inspired Networking: From Theory to Practice," IEEE Communications Magazine 48(11) (2010) 176-183.
- [94] Erl, T., SOA Design Patterns. Prentice Hall, New Jersey, United States, 2009.
- [95] Erman, A. T., Dilo, A. and Havinga, P., "A fault-tolerant data dissemination based on honeycomb architecture for mobile multi-sink wireless sensor networks," In: Proc. International Conference on Intelligent Sensors, Sensor Networks and Information Processing, 2010, pp. 97-102.
- [96] Erol-Kantarci, M., Mouftah, H. T. and Oktug, S., "Localization Techniques for Underwater Acoustic Sensor Networks," IEEE Communications Magazine 48(12) (2010) 152-158.
- [97] Faghih, M. M. and Moghaddam, M. E., "SOMM: A New Service Oriented Middleware for Generic Wireless Multimedia Sensor Networks Based on Code Mobility," Sensors 11(1) (2011) 10343-10371.
- [98] Familiar, M. S., Martínez, J. F., Corredor, I. and García-Rubio, C., "Building Service-Oriented Smart Infrastructures over Wireless Ad Hoc Sensor Networks: A Middleware Perspective," Computer Networks 56(4) (2012) 1303-1328.
- [99] Familiar, M. S., Martínez, J. F. and López, L., "Pervasive Smart Spaces and Environments: A Service-Oriented Middleware Architecture for Wireless Ad Hoc and Sensor Networks," International Journal of Distributed Sensor Networks 2012(725190) (2012) 1-11.
- [100] Fensel, D., "Forewords," in: Hepp, M., Leenheer, P. d., Moor, A. d. and Sure, Y. (Eds.) *Ontology Management: Semantic Web, Semantic Web Services, and Business Applications*, Springer-Verlag, Heidelberg, Germany, 2007, pp. 9-11.
- [101] Fragouli, C., "Network coding for sensor Networks," in: Haykin, S. and Liu, K. (Eds.) *Handbook on Array Processing and Sensor Networks*, Wiley-IEEE Press, 2010, pp. 645-667.
- [102] Franklin, S. and Graesser, A., "Is it an agent, or just a program?: A taxonomy for autonomous agents," In: Proc. International Workshop on Agent Theories, Architectures, and Languages, 1996, pp. 21-35.
- [103] García-Hernando, A. B., Martínez-Ortega, J. F., López-Navarro, J. M., Prayati, A. and Redondo-López, L., *Problem Solving for Wireless Sensor Networks*. Springer-Verlag, London, United Kingdom, 2008.
- [104] Garg, V. K., *Wireless Communications & Networking*. Morgan Kaufmann, San Francisco, United States, 2007.
- [105] Garruzzo, S., Rosaci, D. and Same, G. M., "ISABEL: A multi agent e-learning system that supports multiple devices," In: Proc. IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2007, pp. 485-488.
- [106] Georgakarakou, C. E. and Economides, A. A., "Agent technology applied to intelligent buildings," In: Proc. International Conference on Computers, 2006, pp. 770-774.
-

- [107] Gershenfeld, N. and Cohen, D., "Internet 0: Interdevice Internetworking - End-to-End Modulation for Embedded Networks," *IEEE Circuits and Devices Magazine* 22(5) (2006) 48-55.
- [108] Geyik, S. C., Szymanski, B. K., Zerfos, P. and Verma, D., "Dynamic composition of services in sensor networks," In: *Proc. IEEE International Conference on Services Computing*, 2010, pp. 242-249.
- [109] Geyik, S. C., Szymanski, B. K., Zerfos, P., Verma, D., Wright, J., Gibson, C. and Vincent, C., "Dynamic composition of services in sensor networks and its implementation under sensor fabric cached," In: *Proc. Annual Conference of International Technology Alliance*, 2010, pp. 1-8.
- [110] Gleghorn, R., "Enterprise Application Integration: A Manager's Perspective," *IT Professional* 7(6) (2005) 17-23.
- [111] Glisic, S. and Lorenzo, B., *Advanced Wireless Networks: Cognitive, Cooperative & Opportunistic 4G Technology*. John Wiley & Sons, Sussex, United Kingdom, 2009.
- [112] Grace, P., Hughes, D., Porter, B., Blair, G. B., Coulson, G. and Taiani, F., "Experiences with Open Overlays: A Middleware Approach to Network Heterogeneity," *ACM SIGOPS Operating Systems Review* 42(4) (2008) 123-136.
- [113] Grant, R. M., "Toward Knowledge based Theory of the Firm," *Strategic Management Journal* 17(1) (1996) 109-122.
- [114] Grobelnik, M. and Mladenić, D., "Knowledge discovery for ontology construction," in: Davies, J., Studer, R. and Warren, P. (Eds.) *Semantic Web Technologies: Trends and Research in Ontology-based Systems*, John Wiley and Sons, Sussex, United Kingdom, 2006, pp. 9-28.
- [115] Grosz, B. N. and Labrou, Y., "An approach to using XML and a rule-based content language with an agent communication language," in: Dignum, F. and Greaves, M. (Eds.) *Issues in Agent Communication*, Springer-Verlag, Heidelberg, Germany, 2000, pp. 96-117.
- [116] Grosso, W., *Java RMI*. O'Reilly Media, Sebastopol, United States, 2001.
- [117] Guinard, D., Trifa, V., Pham, T. and Liechti, O., "Towards physical mashups in the web of things," In: *Proc. Conference on Networked Sensing Systems*, 2009, pp. 1-4.
- [118] Gummadi, R., Gnawali, O. and Govindan, R., "Macro-programming wireless sensor networks using kairos," In: *Proc. IEEE International Conference on Distributed Computing in Sensor Systems*, 2005, pp. 126-140.
- [119] Gupta, R. and Kansal, G., "A Survey on Comparative Study of Mobile Agent Platforms," *International Journal of Engineering Science and Technology* 3(3) (2011) 1943-1948.
- [120] Gurgen, L., Roncancio, C., Labbé, C., Bottaro, A. and Olive, V., "SSStreamWare: A service oriented middleware for heterogeneous sensor data management," In: *Proc. ACM International Conference on Pervasive Services*, 2008, pp. 121-130.
- [121] Hadim, S. and Mohamed, N., "Middleware: Middleware Challenges and Approaches for Wireless Sensor Networks," *IEEE Distributed Systems Online* 7(3) (2006) 1-23.
- [122] Hashmi, S., Mougftah, H. T. and Georganas, N. D., "A new transport layer sensor network protocol," In: *Proc. Canadian Conference on Electrical and Computer Engineering*, 2006, pp. 1116-1119.

-
- [123] Hayes-Roth, B., "An Architecture for Adaptive Intelligent Systems," *Artificial Intelligence* 72(1) (1995) 329-365.
- [124] He, T., Krishnamurthy, S., Stankovic, J. A., Abdelzaher, T., Luo, L., Stoleru, R., Yan, T., Gu, L., Zhou, G., Hui, J. and Krogh, B., "VigilNet: An Integrated Sensor Network System for Energy-Efficient Surveillance," *ACM Transactions on Sensor Networks* 2(1) (2006) 1-38.
- [125] Heinzelman, W. B., Murphy, A. L., Carvalho, H. S. and Perillo, M. A., "Middleware to support Sensor Network Applications," *IEEE Network* 18(1) (2004) 6-14.
- [126] Hendler, J., "Agents and the Semantic Web," *IEEE Intelligent Systems* 16(2) (2001) 30-37.
- [127] Hewitt, C., "Viewing Control Structures as Patterns of Passing Messages," *Artificial Intelligence* 8(3) (1977) 323-364.
- [128] Hoffman, R. R., Ziebell, D., Fiore, S. M. and Becerra-Fernandez, I., "Knowledge Management Revisited," *IEEE Intelligent Systems* 23(3) (2008) 84-88.
- [129] Hong, S. H. and Kim, H. K., "A Multi-Hop Reservation Method for end-to-end Latency Performance Improvement in Asynchronous MAC-based Wireless Sensor Networks," *IEEE Transactions on Consumer Electronics* 55(3) (2009) 1214-1220.
- [130] Horton, F. W., *Information Resources Management: Concept and Cases*. Association for Systems Management, Ohio, United States, 1979.
- [131] Hughes, D., Thoelen, K., Horr , W., Matthys, N., Cid, J. and Michiels, S., "LooCI: A loosely-coupled component infrastructure for networked embedded systems," In: *Proc. Conference on Advances in Mobile Computing and Multimedia*, 2009, pp. 1-9.
- [132] Huhns, M. N. and Singh, M. P., "Service-Oriented Computing: Key Concepts and Principles," *IEEE Internet Computing* 9(1) (2005) 75-81.
- [133] Hunt, A. and Thomas, D., "The Art of Enbugging," *IEEE Software* 20(1) (2003) 10-11.
- [134] Hussain, S. A. and Gurkan, D., "Management and Plug and Play of Sensor Networks Using SNMP," *IEEE Transactions on Instrumentation and Measurement* 60(5) (2011) 1830-1837.
- [135] IEEE Foundation for Intelligent Physical Agents (FIPA) (last accessed on: January, 2014) <www.fipa.org>.
- [136] IEEE Standard for A Smart Transducer Interface for Sensors and Actuators—Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats, IEEE Std. 1451.4, 2004.
- [137] IEEE Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), IEEE Standard 802.15.4, 2006.
- [138] Iera, A., Floerkemeier, C., Mitsugi, J. and Morabito, G., "The Internet of Things - Guest Editorial," *IEEE Wireless Communications* 17(6) (2010) 8-9.
- [139] IETF RFC 4627: The application/json Media Type for JavaScript Object Notation (JSON) (last accessed on: January, 2014) <www.ietf.org>.
- [140] Ilarri, S., Trillo, R. and Mena, E., "SPRINGS: A scalable platform for highly mobile agents in distributed computing environments," In: *Proc. International Symposium on a World of Wireless, Mobile and Multimedia Networks*, 2006, pp. 632-637.
-

- [141] Iyengar, S. S., Wu, H. C., Balakrishnan, N. and Chang, S. Y., "Biologically Inspired Cooperative Routing for Wireless Mobile Sensor Networks," *IEEE Systems Journal* 1(1) (2007) 29-37.
- [142] Jacquot, A., Chanet, J. P., Hou, K. M., Diao, X. and Li, J. J., "A new approach for wireless sensor network management: LiveNCM," In: *Proc. International Conference on New Technologies, Mobility and Security*, 2008, pp. 1-6.
- [143] Jara, A. J., Zamora, M. A. and Skarmeta, A. F., "An architecture based on internet of things to support mobility and security in medical environments," In: *Proc. IEEE Consumer Communications and Networking Conference*, 2010, pp. 1-5.
- [144] Jennings, N. R., Sycara, K. P. and Wooldridge, M., "A Roadmap of Agent Research and Development," *Autonomous Agents and Multi-Agent Systems* 1(1) (1998) 7-38.
- [145] Jennings, N. R. and Wooldridge, M., *Applications of intelligent agents*, in: *Anonymous Agent Technology: Foundations, Applications, and Markets*, Springer-Verlag, Heidelberg, Germany, 1998, pp. 3-28.
- [146] Josuttis, N. M., *SOA in Practice: The Art of Distributed System Design*. O'Reilly Media, Sebastopol, United States, 2008.
- [147] Jung, E. H., Kim, Y. P., Park, Y. J. and Han, S. Y., "A smart sensor overlay network for ubiquitous computing," In: *Proc. International Conference on Ubiquitous Convergence Technology. Lecture Notes in Computer Science*, 2007, pp. 220-231.
- [148] Jung, S., "Self-organizing clusters for routing algorithm by diffusing an interest in wireless sensor networks," In: *Proc. International Conference on Software Engineering Research, Management & Applications*, 2007, pp. 702-710.
- [149] K bisch, S., Peintner, D., Heuer, J. and Kosch, H., "XML-based web service generation for microcontroller-based sensor actor networks," In: *Proc. International Workshop on Factory Communication Systems*, 2010, pp. 181-184.
- [150] Kalasapur, S., Kumar, M. and Shirazi, B. A., "Dynamic Service Composition in Pervasive Computing," *IEEE Transactions on Parallel and Distributed Systems* 18(7) (2007) 907-918.
- [151] Kang, S., Lee, Y., Min, C., Ju, Y., Park, T., Lee, J., Rhee, Y. and Song, J., "Orchestrator: An active resource orchestration framework for mobile context monitoring in sensor-rich mobile environments," In: *Proc. IEEE International Conference on Pervasive Computing and Communications*, 2010, pp. 135-144.
- [152] Khedo, K. and Subramanian, R. K., "A Service-Oriented Component-Based Middleware Architecture for Wireless Sensor Networks," *International Journal of Computer Science and Network Security* 9(3) (2009) 174-182.
- [153] King, I., Li, J. and Chan, K. T., "A brief survey of computational approaches in social computing," In: *Proc. Joint Conference on Neural Networks*, 2009, pp. 1625-1632.
- [154] Ko, J., Lu, C., Srivastava, M. B., Stankovic, J. A., Terzis, A. and Welsh, M., "Wireless Sensor Networks for Healthcare," *Proceedings of the IEEE* 98(11) (2010) 1947-1960.
- [155] Kock, N. and McQueen, R., "Knowledge and Information Communication in Organizations: An Analysis of Core, Support and Improvement Process," *Knowledge and Process Management* 5(1) (1998) 29-40.
- [156] Kortuem, G., Kawsar, F., Fitton, D. and Sundramoorthy, V., "Smart Objects as Building Blocks for the Internet of Things," *IEEE Internet Computing* 14(1) (2010) 44-51.

-
- [157] Kranz, M., Holleis, P. and Schmidt, A., "Embedded Interaction: Interacting with the Internet of Things," *IEEE Internet Computing* 14(2) (2010) 46-53.
- [158] Lange, D. B. and Mitsuru, O., *Programming and Deploying Java Mobile Agents Aglets*. Addison-Wesley, Boston, United States, 1998.
- [159] Lashkari, Y., Metral, M. and Maes, P., "Collaborative interface agents," in: Huhns, M. N. and Singh, M. P. (Eds.) *Readings in Agents*, Morgan Kaufmann Publishers, San Francisco, United States, 1998, pp. 111-116.
- [160] Le, T., Hu, W., Corke, P. and Jha, S., "ERTP: Energy-efficient and Reliable Transport Protocol for data streaming in Wireless Sensor Networks," *Computer Communications* 32(7) (2009) 1154-1171.
- [161] Lee, C. S. and Teytaud, O., "Intelligent agents for games and computer go," In: *Proc. IEEE Symposium on Intelligent Agent*, 2011, pp. 1-2.
- [162] Lee, S. H., Lee, S., Song, H. and Lee, H. S., "Wireless sensor network design for tactical military applications: Remote large-scale environments," In: *Proc. IEEE Military Communications Conference (MILCOM)*, 2009, pp. 1-7.
- [163] Levis, P. and Culler, D., "Mate: A tiny virtual machine for sensor networks," In: *Proc. ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, 2002, pp. 85-95.
- [164] Levis, P. and Gay, D., *TinyOS Programming*. Cambridge University Press, Cambridge, United States, 2009.
- [165] Li, H. and Li, W., "Application of ontology to question-answer knowledge management system," In: *Proc. IEEE International Symposium on Information Technologies and Applications in Education*, 2007, pp. 23-26.
- [166] Li, M., Lou, W. and Ren, K., "Data Security and Privacy in Wireless Body Area Networks," *IEEE Wireless Communications* 17(1) (2010) 51-58.
- [167] Li, W., Luo, J., Liu, B., Zheng, X. and Cao, J., "Multi-agent based QoS-aware service composition," In: *Proc. IEEE International Conference on Systems Man and Cybernetics*, 2010, pp. 3125-3132.
- [168] Li, Y. and Bontcheva, K., "Hierarchical, Perceptron-like Learning for Ontology-based Information Extraction," In: *Proc. International Conference on World Wide Web* (2007) 777-786.
- [169] LifeWear European Research Project *Mobilized Lifestyle with Wearables*. ITEA2 code: 09026 (last accessed on: January, 2014) <www.lifewear.es>.
- [170] Liu, H., Zhang, B., Mouftah, H., Shen, H. and Ma, J., "Opportunistic Routing for Wireless Ad Hoc and Sensor Networks: Present and Future Directions," *IEEE Communications Magazine* 47(12) (2009) 103-109.
- [171] Liu, J. and Zhao, F., "Composing Semantic Services in Open Sensor-Rich Environments," *IEEE Network* 22(4) (2008) 44-49.
- [172] Liu, L., Xu, J., Antonopoulos, N., Li, J. and Wu, K., "Adaptive Service Discovery on Service-Oriented and Spontaneous Sensor Systems," *Journal of Adhoc & Sensor Wireless Networks* 14(1) (2012) 107-132.
- [173] Liu, L., Zhang, X. and Ma, H., "Optimal Node Selection for Target Localization in Wireless Camera Sensor Networks," *IEEE Transactions on Vehicular Technology* 59(7) (2010) 3562-3576.
-

- [174] Liu, T., Sadler, C. M., Zhang, P. and Martonosi, M., "Implementing software on resource-constrained mobile sensors: Experiences with impala and ZebraNet," In: Proc. ACM Conference on Mobile Systems, Applications, and Services, 2004, pp. 256-269.
- [175] Liu, L., Antonopoulos, N., Xu, J., Webster, D. and Wu, K., "Distributed Service Integration for Disaster Monitoring Sensor Systems," *IET Communications* 5(12) (2011) 1777-1784.
- [176] Liutkevicius, A., "A survey of wireless sensor network interconnection to external networks," in: Sobh, T., Elleithy, K. and Mahmood, A. (Eds.) *Novel Algorithms and Techniques in Telecommunications and Networking*, Springer, London, United Kingdom, 2010, pp. 41-46.
- [177] Lopez, G., Custodio, V. and Moreno, J. I., "LOBIN: E-Textile and Wireless-Sensor-Network-Based Platform for Healthcare Monitoring in Future Hospital Environments," *IEEE Transactions on Information Technology in Biomedicine* 14(6) (2010) 1446-1458.
- [178] Lu, G., Krishnamachari, B. and Raghavendra, C. S., "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks," In: Proc. IEEE International Parallel and Distributed Processing Symposium, 2004, pp. 224-232.
- [179] Luo, H., Tao, H., Ma, H. and Das, S. K., "Data Fusion with Desired Reliability in Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems* 22(3) (2011) 501-513.
- [180] Ma, X. and Luo, L., "The analysis of 6LoWPAN technology," In: Proc. Pacific-Asia Workshop on Computational Intelligence and Industrial Application, 2008, pp. 963-966.
- [181] Madden, S. R., Franklin, M. J., Hellerstein, J. M. and Hong, W., "TinyDB: An Acquisitional Query Processing System for Sensor Networks," *ACM Transactions on Database Systems* 30(1) (2005) 122-173.
- [182] Madeyski, L. and Szala, L., "Impact of Aspect-Oriented Programming on Software Development Efficiency and Design Quality: An Empirical Study," *IET Software* 1(5) (2007) 180-187.
- [183] Maes, P., "Artificial Life meets Entertainment: Lifelike Autonomous Agents," *Communications of the ACM* 38(11) (1995) 108-114.
- [184] Mahmoud, Q. H., *Middleware for Communications*. John Wiley and Sons, Sussex, United Kingdom, 2004.
- [185] Martínez, J. F., Bravo, S., García, A. B., Corredor, I., Familiar, M. S., López, L., Hernández, V. and Da Silva, A., "Pervasive Surveillance-Agent System based on Wireless Sensor Networks: Design and Deployment," *Measurement Science and Technology* 21(12) (2010) 1-10.
- [186] Martínez, J. F., Familiar, M. S., Corredor, I., García, A. B., Bravo, S. and López, L., "Composition and Deployment of e-Health Services over Wireless Sensor Networks," *Mathematical and Computer Modelling* 53(3) (2011) 485-503.
- [187] Mason, A., Al-Shamma'a, A. I. and Shaw, A., "Wireless sensor network for intelligent inventory management for packaged gases," In: Proc. IEEE International Conference on Developments in eSystems Engineering (DESE), 2009, pp. 413-417.
- [188] Massey, A. P. and Montoya-Weiss, M., "A performance environment perspective of knowledge management," In: Proc. International Conference on System Sciences, 2003, pp. 1-10.

-
- [189] Mattern, F. and Floerkemeier, C., "From the internet of computers to the internet of things," in: Sachs, K., Petrov, I. and Guerrero, P. (Eds.) *From Active Data Management to Event-Based Systems and More*, Springer-Verlag, Berlin, Germany, 2010, pp. 242-259.
- [190] Maybury, M., "Human language technologies for knowledge management: Challenges and opportunities," In: *Proc. Workshop on Human Language Technology and Knowledge Management*, 2001, pp. 1-8.
- [191] McArthur, S. D. J., Davidson, E. M., Catterson, V. M., Dimeas, A. L., Hatziaargyriou, N. D., Ponci, F. and Funabashi, T., "Multi-Agent Systems for Power Engineering Applications—Part I: Concepts, Approaches, and Technical Challenges," *IEEE Transactions on Power Systems* 22(4) (2007) 1743-1752.
- [192] Mendes, L. and Rodrigues, J., "A Survey on Cross-Layer Solutions for Wireless Sensor Networks," *Journal of Network and Computer Applications* 34(2) (2011) 523-534.
- [193] Mendes, M., Falsarella, O., Fontes, I., Krause, S., Loyolla, W. and Mendez, C., "Architectural considerations about open distributed agent support platforms," In: *Proc. International Symposium on Autonomous Decentralized Systems*, 1997, pp. 359-366.
- [194] Metzger, M. and Polakow, G., "A Survey on Applications of Agent Technology in Industrial Process Control," *IEEE Transactions on Industrial Informatics* 10(1) (2011) 1-12.
- [195] Meyer, G. G., Främling, F. and Holmström, J., "Intelligent Products: A Survey," *Computers in Industry* 60(3) (2009) 137-148.
- [196] Mian, A. N., Baldoni, R. and Beraldi, R., "A Survey of Service Discovery Protocols in Multihop Mobile Ad Hoc Networks," *IEEE Pervasive Computing* 8(1) (2009) 66-74.
- [197] Microsoft Distributed Component Object Model (DCOM) Remote Protocol specification v11.0.1, 2011.
- [198] Milojicic, D., Breugst, M., Busse, I., Campbell, J., Covaci, S., Friedman, B., Kosaka, K., Lange, D., Ono, K. and Oshima, M., "MASIF: The OMG Mobile Agent System Interoperability Facility," *Personal and Ubiquitous Computing* 2(2) (2000) 117-129.
- [199] Mohamed, N. and Al-Jaroodi, J., "Service-oriented middleware approaches for wireless sensor networks," In: *Proc. International Conference on System Sciences*, 2011, pp. 1-9.
- [200] Mohamed, N. and Al-Jaroodi, J., "A Survey on Service-Oriented Middleware for Wireless Sensor Networks," *Service Oriented Computing and Applications* 5(2) (2011) 71-85.
- [201] Monowar, M. M., Rahman, M. O., Pathan, A. S. and Hong, C. S., "Congestion control protocol for wireless sensor networks handling prioritized heterogeneous traffic," In: *Proc. Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*, 2008, pp. 1-8.
- [202] Mubarak, H., "Developing Flexible Software Using Agent-Oriented Software Engineering," *IEEE Software* 25(5) (2008) 12-15.
- [203] Mu-kun, C., "E-commerce automated negotiation based on agent reasoning," In: *Proc. International Conference on Management of E-Commerce and E-Government*, 2010, pp. 31-34.
- [204] Nagi, K., *Transactional Agents: Towards a Robust Multi-Agent System*. Springer-Verlag, Heidelberg, Germany, 2005.
-

- [205] Nassehi, A., Newman, S. T. and Allen, R. D., "The Application of Multi-Agent Systems for STEP-NC Computer Aided Process Planning of Prismatic Components," *International Journal of Machine Tools and Manufacture* 46(5) (2006) 559-574.
- [206] Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T. and Swartout, W., "Enabling Technology for Knowledge Sharing," *AI Magazine* 12(3) (1991) 36-56.
- [207] Ngoc, T. Q., Lee, J. H., Gil, K. J., Jeong, K. and Lim, S. B., "An ESB based micro-scale urban air quality monitoring system," In: *Proc. International Conference on Networking, Architecture and Storage*, 2010, pp. 288-293.
- [208] Nissen, M., Kamel, M. and Sengupta, K., "Integrated Analysis and Design of Knowledge Systems and Processes," *Information Resources Management Journal* 13(1) (2000) 24-43.
- [209] Nonaka, I., "A Dynamic Theory of Organizational Knowledge Creation," *Organization Science* 5(1) (1994) 14-37.
- [210] Nonaka, I. and Takeuchi, H., *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press, Oxford, United Kingdom, 1995.
- [211] Nwana, H. S. and Ndumu, D. T., "A brief introduction to agent software technology," in: Jennings, N. R. and Wooldridge, M. (Eds.) *Agent Technology: Foundations, Applications, and Markets*, Springer-Verlag, Heidelberg, Germany, 1998, pp. 29-48.
- [212] Nwana, H. S. and Ndumu, D. T., "A Perspective on Software Agents Research," *The Knowledge Engineering Review* 12(2) (1999) 1-16.
- [213] OASIS Devices Profile for Web Services (DPWS) specification (last accessed on: January, 2014) <www.oasis-open.org>.
- [214] OASIS Web Services Business Process Execution Language (WS-BPEL) specification (last accessed on: January, 2014) <www.oasis-open.org>.
- [215] OMG Mobile Agent System Interoperability Facilities (MASIF) specification, TC Document orbos/97-10-05, 1997.
- [216] OMG Minimum CORBA specification v1.0, 2002.
- [217] OMG Common Object Request Broker Architecture (CORBA/IIOP) specification v3.1, 2008.
- [218] Oracle SunTMSPOT: Theory of Operation specification v5.0. Red Release, 2009.
- [219] OSGi Open Services Gateway initiative (last accessed on: January, 2014) <www.osgi.org>.
- [220] O'Sullivan, D. and Lewis, D., "Semantically driven service interoperability for pervasive Computing," In: *Proc. ACM International Workshop on Data Engineering for Wireless and Mobile Access*, 2003, pp. 17-24.
- [221] Padmanabhan, K. and Kamalakkannan, P., "A Study on Energy Efficient Routing Protocols in Wireless Sensor Networks," *European Journal of Scientific Research* 60(4) (2011) 517-529.
- [222] Pakkala, D., Paakkonen, P. and Sihvonen, M., "A generic communication middleware architecture for distributed application and service messaging," In: *Proc. Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services*, 2005, pp. 22-29.

-
- [223] Papazoglou, M. P., *Web Services: Principles and Technology*. Pearson Education, Essex, United Kingdom, 2008.
- [224] Papazoglou, M. P., Traverso, P., Dustdar, S. and Leymann, F., "Service-Oriented Computing: State of the Art and Research Challenges," *Computer* 40(11) (2007) 38-45.
- [225] Patil, R. S., Fikes, R. E., Patel-Schneider, P. F., McKay, D., Finin, T., Gruber, T. and Neches, R., "The DARPA knowledge sharing effort: Progress report," in: Huhns, M. N. and Singh, M. P. (Eds.) *Readings in Agents*, Morgan Kaufmann, San Francisco, United States, 1998, pp. 243-254.
- [226] Penya, Y. N., Mahlknecht, S. and Rossler, P., "Lightweight agent platform for high-performance fieldbus nodes," In: *Proc. IEEE International Workshop on Factory Communication Systems*, 2002, pp. 9-13.
- [227] Phung, T., Winikoff, M. and Padgham, L., "Learning within the BDI framework: An empirical Analysis," In: *Proc. International Conference on Knowledge-Based & Intelligent Information & Engineering Systems*, 2005, pp. 282-288.
- [228] Polanyi, M., *Personal Knowledge: Toward a Post-Critical Philosophy*. University of Chicago Press, Chicago, United States, 1958.
- [229] Polovina, S. and Andrews, S., "A transaction-oriented architecture for structuring unstructured information in enterprise applications," in: Sugumaran, V. (Ed.) *Intelligent, Adaptive and Reasoning Technologies: New Developments and Applications*, IGI Global, Hershey, United States, 2011, pp. 285-299.
- [230] Poslad, S., Buckle, P. and Hadingham, R., "The FIPA-OS agent platform: Open source for open standards," In: *Proc. International Conference and Exhibition on the Practical Application of Intelligent Agents and MultiAgents*, 2000, pp. 355-368.
- [231] Poslad, S. and Charlton, P., "Standardizing agent interoperability: The FIPA approach," in: Luck, M., Marik, V., Stepankova, O. and Trappl, R. (Eds.) *Multi-Agent Systems and Applications*, Springer-Verlag, Heidelberg, Germany, 2001, pp. 98-117.
- [232] Preece, A., Flett, A., Sleeman, D., Curry, D., Meany, N. and Perry, P., "Better Knowledge Management through Knowledge Engineering," *IEEE Intelligent Systems* 16(1) (2001) 36-43.
- [233] Priyantha, N. B., Kansal, A., Goraczko, M. and Zhao, F., "Tiny Web Services: Design and Implementation of Interoperable and Evolvable Sensor Networks," *International Conference on Embedded Networked Sensor Systems* 253(266) (2008) .
- [234] Rao, A. and Georgeff, M., "Modeling rational agents within a BDI-architecture," In: *Proc. International Conference on Principles of Knowledge Representation and Reasoning*, 1991, pp. 473-484.
- [235] Rao, A. and Georgeff, M., "BDI agents: From theory to practice," In: *Proc. International Conference on Multi-Agents Systems*, 1995, pp. 312-329.
- [236] Rashid, J. and Broxvall, M., "Indirect reference: Reconfiguring distributed sensors and actuators," In: *Proc. IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, 2010, pp. 284-290.
- [237] Rathnayaka, A. J., Potdar, V. M., Sharif, A., Sarencheh, S. and Kuruppu, S., "Wireless sensor network transport protocol - A state of the art," In: *Proc. IEEE International Conference on Broadband, Wireless Computing, Communication and Applications*, 2010, pp. 812-817.
-

- [238] Raza, U., Whiteside, B. R. and Fun Hu, "An enterprise service bus (ESB) and google gadgets based micro-injection moulding process monitoring system," In: Proc. Conference on Wireless Sensor Systems, 2012, pp. 1-6.
- [239] Reilly, D., Welsman-Dinelle, M., Bate, C. and Inkpen, K., "Just point and click ? using handhelds to interact with paper maps," In: Proc. International Conference on Human Computer Interaction with Mobile Devices and Services, 2005, pp. 239-242.
- [240] Rezgui, A. and Eltoweissy, M., "Service-Oriented Sensor-Actuator Networks: Promises, Challenges, and the Road Ahead," *Computer Communications* 30(13) (2007) 2627-2648.
- [241] Rhee, I., Warrier, A., and Aia, M., "Z-MAC: A hybrid MAC for wireless sensor networks," In: Proc. ACM Conference on Embedded Networked Sensor Systems, 2005, pp. 90-101.
- [242] Rogers, A., Corkill, D. D. and Jennings, N. R., "Agent Technologies for Sensor Networks," *IEEE Intelligent Systems* 24(2) (2009) 13-17.
- [243] Rolf H., W., "Internet of Things: New Security and Privacy Challenges," *Computer Law and Security Review* 26(1) (2010) 23-30.
- [244] Roman, M., Ziebart, B. and Campbell, R. H., "Dynamic application composition: Customizing the behavior of an active space," In: Proc. IEEE International Conference on Pervasive Computing and Communications, 2003, pp. 169-176.
- [245] Rosen, M., *Applied SOA: Service-Oriented Architecture and Design Strategies*. John Wiley and Sons, Sussex, United Kingdom, 2008.
- [246] Rouached, M., Baccar, S. and Abid, M., "RESTful sensor web enablement services for wireless sensor networks," In: Proc. World Congress on Services, 2012, pp. 65-72.
- [247] Rozner, E., Seshadri, J., Mehta, Y. and Qiu, L., "SOAR: Simple Opportunistic Adaptive Routing Protocol for Wireless Mesh Networks," *IEEE Transactions on Mobile Computing* 8(12) (2009) 1622-1635.
- [248] Russell, S. J. and Norvig, P., *Artificial Intelligence: A Modern Approach*. Prentice Hall, New Jersey, United States, 1994.
- [249] Sardis, E., Doulamis, A. and Matsatsinis, N., "Sensor Networks and Multi-Agents in Industrial Workflows," *International Journal of Machine Learning and Computing* 1(2) (2011) 205-212.
- [250] Satyanarayanan, M., "Pervasive Computing: Vision and Challenges," *IEEE Personal Communications* 8(4) (2001) 10-17.
- [251] Schuldt, H., "Agents and databases: A symbiosis?" In: Proc. International Workshop on Cooperative Information Agents, 2008, pp. 24-34.
- [252] Shaikh, F. K., Khelil, A., Ali, A. and Suri, N., "TRCCIT: Tunable reliability with congestion control for information transport in wireless sensor networks," 2010, pp. 1-9.
- [253] Shelby, Z. and Bormann, C., *6LoWPAN: The Wireless Embedded Internet*. John Wiley & Sons, Sussex, United Kingdom, 2009.
- [254] Shen, G. and Liu, B., "The visions, technologies, applications and security issues of internet of things," In: Proc. IEEE International Conference on E-Business and E-Government, 2011, pp. 1867-1870.
- [255] Shu, H., Liang, Q. and Gao, J., "Wireless Sensor Network Lifetime Analysis Using Interval Type-2 Fuzzy Logic Systems," *IEEE Transactions on Fuzzy Systems* 16(2) (2008) 416-427.

-
- [256] Simon, D., Cifuentes, C., Cleal, D., Daniels, J. and White, D., "Java on the bare metal of wireless sensor devices: The squawk java virtual machine," In: Proc. International Conference on Virtual Execution Environments, 2006, pp. 78-88.
- [257] Sindhgatta, R., Sengupta, B. and Ponnalagu, K., "Measuring the quality of service-oriented design," In: Proc. International Joint Conference on Service-Oriented Computing, 2009, pp. 485-499.
- [258] Singh, M. P. and Huhns, M. N., *Service-Oriented Computing: Semantics, Processes, Agents*. John Wiley and Sons, Sussex, United Kingdom, 2005.
- [259] Sleman, A. and Moeller, R., "Micro SOA model for managing and integrating wireless sensor network into IP-based networks," In: Proc. International Conference on Computational Intelligence, Communication Systems and Networks, 2010, pp. 137-142.
- [260] Smith, D. C., Cypher, A. and Spohrer, J., "KidSim: Programming Agents without a Programming Language," *Communications of the ACM* 37(7) (1994) 55-67.
- [261] Sollacher, R., Niedermeier, C., Vicari, N. and Osipov, M., "Towards a service oriented architecture for wireless sensor networks in industrial applications," In: Proc. IFAC Symposium on Information Control Problems in Manufacturing, 2009, pp. 2103-2108.
- [262] Soroka, A. J., "Agent-based system for knowledge acquisition and management," in: Benyoucef, L. and Grabot, B. (Eds.) *Artificial Intelligence Techniques for Networked Manufacturing Enterprises Management*, Springer-Verlag, London, United Kingdom, 2010, pp. 43-82.
- [263] Souto, E., Guimaraes, G., Vasconcelos, G., Vieira, M., Rosa, N., Ferraz, C. and Kelner, J., "Mires: A Publish/Subscribe Middleware for Sensor Networks," *Personal and Ubiquitous Computing* 10(1) (2006) 37-44.
- [264] Staab, S., "Human Language Technologies for Knowledge Management," *IEEE Intelligent Systems* 16(6) (2001) 84-94.
- [265] Straßer, M. and Schwehm, M., "A performance model for mobile agent systems," In: Proc. International Conference on Parallel and Distributed Processing Techniques and Applications, 1997, pp. 1132-1140.
- [266] Strimpakou, M., Roussaki, I., Pils, C., Angermann, M. and Anagnostou, M., "Context modelling and management in ambient-aware pervasive environments," In: Proc. Workshop on Location and Context Awareness, 2005, pp. 2-15.
- [267] Sudevalayam, S. and Kulkarni, P., "Energy Harvesting Sensor Nodes: Survey and Implications," *IEEE Communications Surveys & Tutorials* 13(3) (2011) 443-461.
- [268] Sun Microsystems Enterprise JavaBean (EJB) specification v3.0, 2006.
- [269] Sun, L., Long, C., Chen, C. and Guan, X., "A multicast-query-based data dissemination protocol for wireless sensor networks with multiple mobile sinks," In: Proc. World Congress on Intelligent Control and Automation, 2010, pp. 1240-1244.
- [270] Svensson, D., Hedin, G. and Magnusson, B., "Pervasive applications through scripted assemblies of services," In: Proc. IEEE International Conference on Pervasive Services, 2007, pp. 301-307.
- [271] Swan, J., Scarborough, H. and Preston, J., "Knowledge management - the next fad to forget people?" In: Proc. European Conference on Information Systems, 1999, pp. 668-678.
- [272] Sycara, K. P., "Multiagent Systems," *AI Magazine* 19(2) (1998) 79-92.
-

- [273] Sycara, K. P., "Agent based aiding of human teams," In: Proc. International Joint Conferences on Web Intelligence and Intelligent Agent Technologies, 2009, pp. 4-5.
- [274] Timm, C., Schmutzler, J., Marwedel, P. and Wietfeld, C., "Dynamic web service orchestration applied to the device profile for web services in hierarchical networks," In: Proc. International Conference on Communication System Software and Middleware, 2009, pp. 1-6.
- [275] Topley, K., *J2ME in a Nutshell*. O'Reilly Media, Sebastopol, United States, 2002.
- [276] Trappey, C. V., Trappey, A. J., Huang, C. J. and Ku, C. C., "The Design of a JADE-based Autonomous Workflow Management System for Collaborative SoC Design," *Expert Systems with Applications* 36(2) (2009) 2659-2669.
- [277] Urbieto, A., Barrutieta, G., Parra, J. and Uribarren, A., "A survey of dynamic service composition approaches for ambient systems," In: Proc. International Conference on Ambient Media and Systems, 2008, pp. 1-8.
- [278] uSWN European Research Project Solving Major Problems in MicroSensorial Wireless Networks. FP6 code: IST-034642 (last accessed on: January, 2014) <www.uswn.eu>.
- [279] Van Den Akker, D., Smolderen, K., De Cleyn, P., Braem, B. and Blondia, C., "TinySPOTComm: Facilitating communication over IEEE 802.15.4 between sun SPOTs and TinyOS-based motes," In: Proc. International Conference on Sensor Networks Applications, Experimentation and Logistics. 2010, pp. 177-194.
- [280] Van der Burg, S. and Dolstra, E., "Automated deployment of a heterogeneous service-oriented system," In: Proc. EUROMICRO Conference on Software Engineering and Advanced Applications, 2010, pp. 183-190.
- [281] Van der Hoek, W. and Wooldridge, M., "Multi-agent systems," in: Van Harmelen, F., Lifschitz, V. and Porter, B. (Eds.) *Handbook of Knowledge Representation*, Elsevier, Amsterdam, The Netherlands, 2008, pp. 887-928.
- [282] Vanitha, V., Palanisamy, V., Johnson, N. and Aravindhbabu, G., "LiteOS based Extended Service Oriented Architecture for Wireless Sensor Networks," *International Journal of Computer and Electrical Engineering* 2(3) (2010) 432-436.
- [283] Vasseur, J. P. and Dunkels, A., *Interconnecting Smart Objects with IP: The Next Internet*. Morgan Kaufmann Publishers, Massachusetts, United States, 2010.
- [284] Vaughan-Nichols, S. J., "Wireless Middleware: Glue for the Mobile Infrastructure", *Computer* 37(5) (2004) 18-20.
- [285] Vinoski, S., "Where is Middleware?" *IEEE Internet Computing* 6(2) (2002) 83-85.
- [286] Vinyals, M., Rodriguez-Aguilar, J. A. and Cerquides, J., "A Survey on Sensor Networks from a Multiagent Perspective," *The Computer Journal* 54(3) (2011) 455-470.
- [287] Völkel, M., Schaffert, S. and Oren, E., "Personal knowledge management with semantic Technologies," in: Rech, J., Decker, B. and Ras, E. (Eds.) *Emerging Technologies for Semantic Work Environments: Techniques, Methods, and Applications*, 2008, pp. 138-153.
- [288] W3C Resource Description Framework (RDF): Concepts and Abstract Syntax, Recommendation, 2004.
- [289] W3C SPARQL Protocol for RDF, Recommendation, 2008.
- [290] W3C OWL-S Semantic Markup for Web Services (last accessed on: January, 2014) <www.w3.org>.

-
- [291] Wall, J., Platt, G., James, G. and Valencia, P., "Wireless sensor networks as agents for intelligent control of distributed energy resources," In: Proc. International Symposium on Wireless Pervasive Computing, 2007, pp. 547-551.
- [292] Walteneus, D., "Dynamic Power Management in Wireless Sensor Networks: State-of-the-Art," IEEE Sensors Journal 12(5) (2012) 1518-1528.
- [293] Wang, X., Wang, J., Zheng, Z., Xu, Y. and Yang, M., "Service composition in service-oriented wireless sensor networks with persistent queries," In: Proc. Consumer Communications and Networking Conference, 2009, pp. 1-5.
- [294] Wei, Y. and Blake, M. B., "Service-Oriented Computing and Cloud Computing: Challenges and Opportunities," IEEE Internet Computing 14(6) (2010) 72-75.
- [295] Weiser, M., "The Computer for the 21st Century," Scientific American 265(3) (1991) 94-104.
- [296] Weiss, G., Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. The MIT Press, Massachusetts, United States, 1999.
- [297] Welbourne, E., Battle, L., Cole, G., Gould, K., Rector, K., Raymer, S., Balazinska, M. and Borriello, G., "Building the Internet of Things using RFID: The RFID Ecosystem Experience," IEEE Internet Computing 13(3) (2009) 48-55.
- [298] Wiig, K. M., Knowledge Management Foundations: Thinking about Thinking — how People and Organizations Create, Represent, and use Knowledge. Schema Press, Arlington, United States, 1993.
- [299] Winikoff, M., "JACK intelligent agents: An industrial strength platform," in: Bordini, R. H., Dastani, M., Dix, J. and Seghrouchni, A. F. (Eds.) Multi-Agent Programming: Languages, Platforms and Applications, Springer, New York, United States, 2005, pp. 175-196.
- [300] Wooldridge, M., An Introduction to MultiAgent Systems. John Wiley and Sons, Sussex, United Kingdom, 2009.
- [301] Wooldridge, M. and Jennings, N. R., "Intelligent Agents: Theory and Practice," Knowledge Engineering Review 10(2) (1995) 1-46.
- [302] Xia, J., Chang, C. K., Kim, T. H., Yang, H. I., Bose, R. and Helal, S., "Fault-resilient ubiquitous service composition," In: Proc. IET International Conference on Intelligent Environments, 2007, pp. 108-115.
- [303] Yan, Y., Zhang, B., Zheng, J. and Ma, J., "CORE: A Coding-Aware Opportunistic Routing Mechanism for Wireless Mesh Networks," IEEE Wireless Communications 17(3) (2010) 96-103.
- [304] Yang, N., "Research on the evaluation of knowledge management performance of high-tech enterprise," In: Proc. IEEE Symposium on Web Society, 2010, pp. 462-465.
- [305] Yang, Q. and Shieh, J. S., "A multi-agent prototype system for medical diagnosis," In: Proc. International Conference on Intelligent System and Knowledge Engineering, 2008, pp. 1265-1270.
- [306] Yick, J., Mukherjee, B. and Ghosal, D., "Wireless Sensor Network Survey," Computer Networks 52(12) (2008) 2292-2330.
- [307] Yuce, M. R., "Implementation of Wireless Body Area Networks for Healthcare Systems," Sensors and Actuators A: Physical 162(1) (2010) 116-129.
-

- [308] Zafeiropoulos, A., Spanos, D. E., Arkoulis, S., Konstantinou, N. and Mitrou, N., "Data management in sensor networks using semantic web technologies," in: Jin, H. and Lv, Z. (Eds.) *Data Management in Semantic Web*, Nova Science Publishers, New York, United States, 2011, pp. 97-118.
- [309] Zarras, A., Fredj, M., Georgantas, N. and Issarny, V., "Engineering reconfigurable distributed software systems: Issues arising for pervasive computing," in: Butler, M., Jones, C. B., Romanovsky, A. and Troubitsyna, E. (Eds.) *Rigorous Development of Complex Fault-Tolerant Systems*, Springer-Verlag, Berlin, Germany, 2006, pp. 364-386.
- [310] Zhai, L., Li, C. and Sun, L., "Research on the Message-Oriented Middleware for Wireless Sensor Networks," *Journal of Computers* 6(5) (2011) 1040-1046.
- [311] Zhang, D., Zhu, M., Cheng, H., Koh, Y. and Mokhtari, M., "Handling heterogeneous device interaction in smart spaces," in: Ma, J., Jin, H., Yang, L. and Tsai, J. (Eds.) *Ubiquitous Intelligence and Computing*, Springer-Verlag, Berlin, Germany, 2006, pp. 250-259.
- [312] Zhang, J., Orlik, P. V., Sahinoglu, Z., Molisch, A. F. and Kinney, P., "UWB Systems for Wireless Sensor Networks," *Proceedings of the IEEE* 97(2) (2009) 313-331.
- [313] Zhang, Z., Chen, Q., Bergarp, T., Norman, P., Wikstrom, M., Yan, X. and Zheng, L. R., "Wireless sensor networks for logistics and retail," In: *Proc. IEEE International Conference on Networked Sensing Systems*, 2009, pp. 1-4.
- [314] Zhengxia, W. and Laisheng, X., "Modern logistics monitoring platform based on the internet of things," In: *Proc. International Conference on Intelligent Computation Technology and Automation*, 2010, pp. 726-731.
- [315] Zhou, H., Huang, Z. and Zhao, G., "A service-centric solution for wireless sensor networks," In: *Proc. International ICST Conference on Communications and Networking in China*, 2010, pp. 1-5.
- [316] Zhou, J., Riekkki, J. and Ylianttila, M., "Modeling service composition and exploring its characteristics," In: *Proc. World Conference on Services*, 2009, pp. 446-451.