



Dyna

ISSN: 0012-7353

dyna@unalmed.edu.co

Universidad Nacional de Colombia
Colombia

Aguilar, Jose; Mendonça, Maribel; Jerez, Marxjhony; Sánchez, Manuel
Emergencia ontológica basada en análisis de contexto, como servicio para ambientes
inteligentes

Dyna, vol. 84, núm. 200, marzo, 2017, pp. 28-37

Universidad Nacional de Colombia
Medellín, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=49650910004>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Ontological emergence based on context analysis as a service for intelligent environments

Jose Aguilar ^a, Maribel Mendonça ^b, Marxjhony Jerez ^c & Manuel Sánchez ^d

^a Universidad de Los Andes, Mérida, Venezuela. Escuela Politécnica Nacional, Quito, Ecuador. aguilar@ula.ve

^b Universidad Centroccidental Lisandro Alvarado, Barquisimeto, Venezuela. mmendonca@ucla.edu.ve

^c Universidad de Los Andes, Mérida, Venezuela. marxjhony@ula.ve

^d Universidad Nacional Experimental del Táchira, San Cristóbal, Venezuela. mbsanchez@unet.edu.ve

Received: July 15th, 2016. Received in revised form: October 10th, 2016. Accepted: October 24th, 2016.

Abstract

An Intelligent Environment (IE) is composed by intelligent devices that interact with each other in carrying out their task, and they need to understand their contexts to provide adequate services to their users. An IE for Learning in the Cloud (IECL) is one that facilitates the location and use of educational services available in the cloud, which will be made available to users (teachers and students), resulting in an improvement of their teaching-learning process. One of the fundamental aspects for the development of IECLs, it is the contextual awareness. This paper integrates an IECL, based on Multi-Agent Systems, with services of Context Awareness, which enrich the Semantic Model of the IECL, characterized by an ontological knowledge base, which allows the agents of AmICL to take decisions adapted to the context. Particularly, through processes of ontological mining and using meta-ontologies, emerge ontological models appropriate to the context.

Keywords: Smart Environments; Context Awareness; Ontological Emergence; Cloud Learning; Knowledge as a Service.

Emergencia ontológica basada en análisis de contexto, como servicio para ambientes inteligentes

Resumen

En los Ambientes Inteligentes (AmI) existen dispositivos inteligentes que interactúan entre sí en el cumplimiento de sus tareas, que necesitan entender su contexto para poder ofrecer servicios adecuados a sus usuarios. Un AmI para aprendizaje en la nube (AmICL) es aquel que facilita la ubicación y uso de servicios educativos disponibles en la nube, que serán puestos a disposición de los usuarios (Profesores y Estudiantes), derivando en una mejora en sus procesos de enseñanza-aprendizaje. Uno de los aspectos fundamentales para su desarrollo, es la consciencia contextual. Este trabajo integra un AmICL basado en Sistema Multi-Agentes, con servicios de Consciencia de Contexto, enriqueciendo el modelo Semántico de AmICL a través de una base de conocimiento ontológica que permite a los agentes de AmICL tomar decisiones adaptadas al momento. Particularmente, a través de procesos de minería ontológica y el uso de meta-ontologías, para hacer emerger modelos ontológicos adecuados al contexto.

Palabras clave: Ambientes Inteligentes; Análisis de Contexto; Emergencia Ontológica; Aprendizaje en la nube; Conocimiento como Servicio.

1. Introducción

Los avances en las tecnologías de información y comunicación y en los ambientes inteligentes, ha permitido el desarrollo de entornos de aprendizaje ubicuos conscientes

del contexto (u-Learning), capaces de monitorear el estado de aprendizaje de los estudiantes y el contexto ambiental. Así, se puede proporcionar la información apropiada al estudiante individualmente, en el lugar correcto, y en el momento adecuado. Los trabajos realizados en esta área indican que

How to cite: Aguilar, J., Mendonça, M., Jerez, M. & Sánchez, M., Emergencia ontológica basada en análisis de contexto, como servicio para ambientes inteligentes. DYNA 84(200), pp. 28-37, 2017.

son varios los factores que deben tenerse en cuenta para desarrollar Ambientes inteligentes de Aprendizaje. Uno de esos factores fundamentales es la consciencia ambiental, la cual permite conocer que actividades realizan los usuarios en el ambiente; lo que permite al componente inteligente del AmI razonar para determinar cómo apoyar al usuario en el desarrollo de sus actividades, y cuales objetos del ambiente (objetos inteligentes y no inteligentes) pueden ser usados para dicho propósito, lo que conllevará a una mejora en el proceso de enseñanza-aprendizaje.

Uno de los requisitos de los AmI es ofrecer servicios inteligentes a los usuarios de acuerdo a sus necesidades, por lo tanto, es necesario que sean dinámicos y adaptables. Esto no es posible si se presentan modelos conceptuales estáticos que no permitan la inclusión de nuevos conceptos, por esa razón, es una necesidad para los AmI la posibilidad de adaptar sus modelos conceptuales al surgimiento de elementos y comportamientos emergentes. La propuesta que se presenta en este trabajo busca ofrecer una solución a este importante requerimiento, tal que el AmI educativo pueda responder ante situaciones imprevistas que puedan presentarse en el aula de clase.

Sobre AmI existen muchos trabajos, y casi todos concuerdan que los AmI deben ser conscientes del contexto [1], para servir a un gran número de personas, en diferentes ambientes y con diferentes propósitos. En su investigación, [1] presenta un middleware consciente del contexto compuesto por dos niveles, uno del agente y otro basado en patrones para reconocer una situación. Ellos construyen un modelo del contexto del sistema independiente de los agentes, que permite una comunicación entre vecinos a través del contexto, lo que provee descentralización y mejora el rendimiento.

Por otro lado, la representación y modelado del contexto puede ser realizado haciendo uso de ontologías, por su riqueza semántica. En [2] se presenta una ontología para representar la interacción entre los perfiles de usuarios y el contexto. El perfil del usuario es importante para la personalización del contexto, en base a sus requerimientos y sus necesidades. La interacción entre el perfil del usuario y el contexto requiere de la integración de la información, para generar un mapa de conceptos similares.

Los trabajos en [1, 2] no ofrecen propuestas concretas sobre la posibilidad de adaptar sus modelos conceptuales (ontologías) a nuevos requerimientos en el AmI y a comportamientos emergentes; en consecuencia, sus investigaciones no saben cómo responder ante situaciones imprevistas que se presenten en el ambiente. En este trabajo, con el uso de meta-ontologías, se tienen ontologías de contexto en un nivel más genérico, que pueden ser adaptadas, a partir de procesos de minería ontológica, para generar ontologías emergentes de acuerdo a las necesidades del AmI. Esto significa que las ontologías emergen de acuerdo a la situación de contexto actual, lo que permite al AmI responder ante situaciones para las cuales no se tiene una solución predefinida. De igual manera, en esta investigación se integran diversos servicios semánticos dentro de la arquitectura, que pueden ser invocados cuando se requiera adaptar el marco ontológico.

Por su parte, en [3] se presenta el término “Aprendizaje Inteligente”, como una característica de los sistemas, que permite proveer al estudiante no solo de un sitio para acceder a las clases (real o virtualmente), sino también la capacidad de generar prácticas de aprendizaje, herramientas y

sugerencias, en el lugar y momento oportuno, y de la forma correcta. El aprendizaje inteligente se percibe como un sistema de aprendizaje con tecnología, que es capaz de guiar y aconsejar a los aprendices a aprender en el mundo real, desde el acceso a los recursos del mundo digital. Las necesidades individuales de cada alumno, deben ser determinadas mediante el análisis de sus comportamientos de aprendizaje o estilos de aprendizaje, el rendimiento, y el contexto del mundo en el que se sitúan. Hasta ahora no se han encontrado trabajos que permitan la adaptación dinámica de las ontologías que se manejan en el contexto de un AmI. Con esta propuesta de emergencia ontológica contextual, se quiere ofrecer una respuesta a la necesidad de adaptación de los servicios que debe ofrecer un AmI educativo a los usuarios de acuerdo a su contexto, considerando que las ontologías se han convertido en componentes claves para el funcionamiento de los AmI, en el manejo de sus modelos conceptuales. La emergencia ontológica representa una forma dinámica de crear ontologías de contexto, lo que permite realizar un análisis mucho más acertado de la realidad, para tomar decisiones acordes a los requerimientos emergentes del AmI, y así ofrecer servicios personalizados en apoyo a las actividades académicas de sus usuarios.

En este trabajo se integra un AmICL [4, 5] cuya especificación y arquitectura es realizada usando el paradigma de Sistema Multi-Agentes (SMA), con servicios de Consciencia de Contexto, de tal manera de enriquecer los modelos semánticos del AmI, definidos como bases de conocimiento ontológico. Particularmente, las ontologías emergen para responder a los contextos presentes en un momento dado. El artículo se organiza de la siguiente manera: primero una sección en donde se ilustran los diferentes aspectos considerados en esta propuesta (AmICL, emergencia ontológica y conciencia del contexto), después se presenta la especificación de la propuesta, culminando con la experimentación, estudiándose dos casos. Finalmente se analizan los resultados y se presentan las conclusiones.

2. Arquitecturas bases

En este artículo se presenta una propuesta de un servicio que permite a un AmI educativo suplirse de las ventajas que proporciona la emergencia de ontologías basada en el contexto, para poder responder a situaciones imprevistas y cambiantes en un momento dado. Esta propuesta está basada en la integración de 3 middlewares que han sido desarrollados previamente, para ofrecer servicios de emergencia ontológica [6], de análisis de contexto [7], y de gestión de AmICL [4, 5]. Particularmente, en este trabajo se integran funcionalidades, servicios o tareas de estos middleware. En esta sección se describe la arquitectura de cada uno de los middleware que componen la propuesta.

2.1. Middleware para ambientes inteligentes de aprendizaje basados en C-Learning (AmICL)

Este middleware, denominado AmICL, caracteriza un AmI de aprendizaje basado en C-Learning, usando un grupo

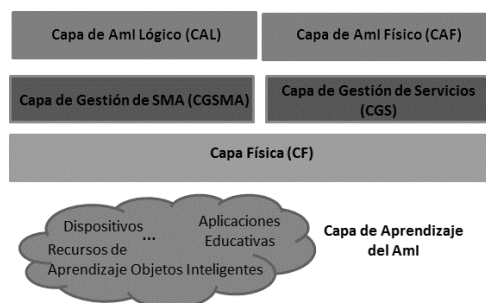


Figura 1. Arquitectura General de AmICL
Fuente: [5]

de agentes, entre los cuales están: Agente Perfil de Estudiante (APE), Agente Tutor (AT), Agente Dispositivo (AD), Agente de Gestión de Servicios (AGS), y Agente Servicio (AS). Dicho Middleware fue propuesto en [4, 5], y es definido por seis capas (ver Fig. 1). La Capa Física (CF) representa los diversos dispositivos físicos del ambiente, definidos como agentes dispositivos en la Capa de gestión de AmI Físico (CAF). CF está conectada con el Sistema Operativo y trabajan en conjunto para realizar las diferentes actividades que se llevarán a cabo en el entorno. La Capa de Gestión del SMA (CGSMA) está compuesta por una comunidad de agentes que soporta la ejecución de aplicaciones multi-agentes. Esta capa ha sido especificada en artículos previos [4, 5] y se basa en el estándar FIPA (Foundation for Intelligent Physical Agents, por sus siglas en inglés), organismo que establece estándares de software para agentes y define los elementos necesarios para que una sociedad de agentes pueda coexistir [8,15]. La Capa de Gestión de Servicios (CGS) es donde se registran, descubren, invocan, etc., los servicios requeridos por el ambiente en un momento determinado. La Capa de AmI Lógico (CAL) representa los diferentes componentes de software que son usados en la plataforma educativa. Adicionalmente, esta capa tiene dos agentes, el agente APE y el agente AT, que caracterizan a estudiantes y profesores, respectivamente. Finalmente, la capa de aprendizaje de AmI es donde se despliegan los diferentes dispositivos y herramientas de software del AmI de aprendizaje.

AmICL aporta ventajas sobre otras propuestas, ya que combina objetos inteligentes y no inteligentes, que integran el AmI con servicios académicos que provee el paradigma de aprendizaje en la nube (C-Learning), para adaptarse y responder a los requerimientos de enseñanza-aprendizaje de los usuarios. Esto permite poner a disposición de los usuarios de forma inteligente, los servicios apropiados que permitan mejorar el desarrollo de las actividades académicas que realizan en un momento específico, adaptado a la forma de aprender de los estudiantes (de acuerdo con su estilo de aprendizaje).

En particular, el aspecto que permite unir el AmI con el paradigma de computación en la nube se encuentra en la Capa de Gestión de Servicios (CGS). Esta capa contiene algunos agentes, que son una extensión del estándar FIPA, para gestionar tanto los recursos y aplicaciones presentes en el sistema, como los servicios educativos en la nube. Los detalles de AmICL pueden encontrarse en [4, 5].

En consecuencia, AmICL es un Middleware Reflexivo Autonomático que permite la integración de los objetos (inteligentes o no) del ambiente educativo, con recursos educativos en la nube, de una manera flexible, proactiva y adaptativa, tal que estos objetos pueden ajustar su comportamiento de acuerdo al contexto y requerimientos del usuario, posibilitando ubicar, analizar y compartir el conocimiento. Todas estas características se logran en mayor o menor grado, al hacer uso de un servicio de consciencia contextual que cuente con un mecanismo que permita hacer emerger las ontologías de contexto, dominio, etc., de acuerdo con la situación del momento; es allí donde los middleware CARMiCLOC y Semántico hacen su trabajo.

2.2. Middleware para la gestión del contexto (CARMiCLOC)

CARMiCLOC es un middleware orientado a servicios, que se encarga de observar el contexto y modelarlo. El objetivo de CARMiCLOC [7] es ofrecer a los agentes de un AmI servicios de Conciencia del Contexto, para que estos puedan tomar acciones oportunas acordes al contexto del momento. CARMiCLOC es una arquitectura de Middleware Reflexivo que incorpora la computación autónoma para la auto-adaptación, y usa el paradigma de la computación de la nube para permitir escalabilidad. Es una arquitectura basada en servicios web, lo que permite que sus servicios puedan ser consumidos por aplicaciones conscientes del contexto o no.

El diseño de CARMiCLOC se muestra en la Fig. 2, en donde se observan sus niveles Base y Meta de la computación reflexiva. Los puntos de enlace se encuentran entre estos niveles. En el Nivel meta se observa el Gerente Autonomático y la Base de Conocimiento, cada de uno de ellos está compuesto por uno o más servicios. CARMiCLOC define siete servicios para realizar la gestión del contexto, los servicios pueden prestarse de manera interna (I) (requerido por otros servicios de CARMiCLOC) o llamados independientemente por servicios externos (E). Además, los Servicios de CARMiCLOC pueden ser consumidos por Aplicaciones Conscientes del Contexto (ACC) o Aplicaciones No Conscientes del Contexto. Los servicios que ofrece CARMiCLOC son: Se₁: Adquisición del Contexto y Pre configuración; Se₂: Modelado del Contexto; Se₃: Razonamiento del Contexto; Se₄: Distribución del Contexto; Se₅: Verificación y resolución de inconsistencia del Contexto; Se₆: Seguridad y privacidad del contexto; Se₇: Gestor de Servicios del Contexto. Los detalles de CARMiCLOC pueden encontrarse en [7].

El servicio de modelado del contexto de CARMiCLOC hace uso de ontologías, lo que permite razonar y enriquecer semánticamente el contexto. Sin embargo, para la gestión de la diversidad de conceptos heterogéneos, específicos para cada situación, se requiere de servicios de minería semántica, minería ontológica, entre otras, como los especificados en el middleware semántico [6]. En general, CARMiCLOC ofrecerá los servicios de análisis del contexto a cada uno de los dispositivos, aplicaciones y sensores registrados en el AmICL.

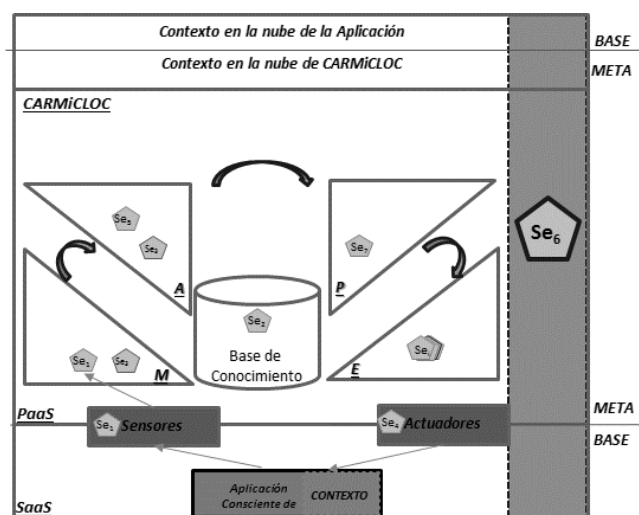


Figura 2. Arquitectura base de CARMICLOC
Fuente: [7]

2.3. Middleware Semántico

El objetivo del Middleware semántico es ofrecer un conjunto de servicios para la manipulación de ontologías en AmI [6, 9]. La arquitectura propuesta se implementa como un middleware reflexivo, que permite, entre otras cosas, la actualización automática de las ontologías para adaptarlas a la dinámica y evolución del AmI.

El middleware puede requerir en un momento dado realizar procesos de auto-gestión, para lo cual se basa en las meta-ontologías que componen su base de conocimientos interna, como un modelo genérico para componer y actualizar las ontologías requeridas por los diferentes procesos del AmI. Las Meta-ontologías definen una ontología general, que puede ser usada para definir ontologías de dominio [7]. La arquitectura de este middleware semántico se basa en la computación autónoma, y está diseñado como un middleware reflexivo autónomo. La propiedad de reflexión le da la posibilidad de monitorear los servicios ontológicos en el AmI, para adaptarlos y adecuarlos a nuevos requerimientos, y a su vez, a través de su marco ontológico basado en meta-ontologías, definir los elementos que deben tener las ontologías de un AmI en un momento dado. Posee dos marcos ontológicos: uno que contiene todo el modelo conceptual del AmI que se gestiona en la arquitectura, y otro interno del middleware, formado por meta-ontologías que permiten definir de forma genérica los conceptos y las reglas que rigen las ontologías del AmI, para apoyar el proceso evolutivo de esas ontologías (basado en reglas de aprendizaje, etc.). En este middleware existe un nivel base en el que se despliega una serie de servicios semánticos distribuidos, y un nivel meta donde se realiza el proceso de reflexión sobre el funcionamiento de estos servicios, para, de ser necesario, realizar las adaptaciones que requiera el sistema.

De manera general, los servicios que ofrece el middleware son los siguientes, Servicios de registro semántico y exploración de ontologías, Servicios de Actualización de Ontologías, Servicios de Integración de

Ontologías, y Servicios de Verificación y Análisis. Para más detalles de este middleware, ver [6].

3. Especificación del modelo de integración: conocimiento como un servicio

El “Conocimiento como un Servicio” es desplegado en AmICL, pues, es este quien requiere conocer el contexto para actualizar las ontologías desplegadas en él, de tal forma que los agentes tomen decisiones ante una situación particular. Para ello requiere los servicios de CARMiCLOC, quien provee los servicios de análisis del contexto. A su vez, este último requiere los servicios del Middleware semántico, pues éste será el encargado de gestionar los servicios de minería semántica, para hacer posible la emergencia ontológica en el AmI, según el contexto.

Debido a que AmICL posee una arquitectura orientada a agentes y CARMiCLOC posee una arquitectura orientada a servicios, es necesario que los servicios de CARMiCLOC sean consumidos usando el sub-sistema de gestión de la nube de AmICL, el cual es definido por la CGS y se explica a continuación.

3.1. Subsistema de gestión de la nube en AmICL

En la CGS coexisten cuatro agentes, dos agentes clásicos de FIPA, el Agente Gestor de Recursos (AGR) y el Agente Gestor de Aplicaciones (AGA), que manejan recursos y aplicaciones del sistema operativo en el SMA [6], y dos nuevos agentes introducidos en AmICL, que realizan el puente entre los agentes del AmI y los servicios en la nube:

- Agente Gestor de Servicios (AGS): Controla, registra y administra los servicios web disponibles en el Sistema; de tal forma que cuando un agente del SMA requiere un servicio específico, debe contactar al AGS, para que éste ubique al Agente de Servicio (AS) que caracteriza, en el SMA, al servicio web requerido. Adicionalmente, AGS tiene la función de descubrir los servicios web de C-Learning necesarios para mejorar el proceso de enseñanza-aprendizaje de los usuarios. AGS caracteriza lo que se conoce dentro de los Servicios Web como UDDI (Universal Description, Discovery and Integration, por sus siglas en inglés).
- Agente Servicio (AS): es la representación lógica de un servicio web. Éste agente caracteriza a un servicio web (un agente por cada servicio web), por esa razón, éste sabe cómo invocarlo (tiene su interfaz) y cuáles son los requerimientos necesarios para acceder al endpoint del servicio web. El AS tiene acceso al archivo de descripción del servicio (WSDL, por sus siglas en inglés). Existe una instancia de este agente por cada servicio web disponible en el SMA, de tal forma que varios agentes pueden consumir varios servicios web en un mismo tiempo. El diseño de estos agentes se describe con mayor detalle en [4, 5].

Los agentes de AmICL usan el modelo de computación en la nube SaaS (Software as a Service, por sus siglas en inglés). Primero, los servicios web se registran en el AGS, esta acción permite conocer cuales servicios web están disponibles para ser usados por los agentes de AmICL.

Segundo, cuando un agente requiere un servicio web, consulta al AGS, para que éste último inicie una subasta que permita seleccionar el AS que satisface los requerimientos de servicio del agente solicitante. Una vez que el AS ha sido seleccionado, el agente que realizó la solicitud entabla conversación con éste para consumir el servicio web. AS envía mensajes (el formato de los mensajes lo conoce por medio del archivo WSDL) al endpoint del servicio web que caracteriza. De esta forma, el servicio web dará respuesta a la solicitud realizada al AS y retorna los resultados, los cuales serán remitidos por el AS al agente solicitante.

Los agentes AGS y AS son fundamentales para la integración de AmICL con CARMiCLOC y el Middleware Semántico, ya que éstos permitirán a los agentes de AmICL consumir los servicios de CARMiCLOC, logrando de esta manera proveer al AmI educativo de consciencia contextual, que permita a los agentes de la plataforma educativa responder a las necesidades académicas del entorno de manera apropiada.

3.2. Proceso de emergencia ontológica para AmI.

Las ontologías que participan en el modelado conceptual de AmICL deben ser dinámicas, es decir, deben cambiar y adaptarse a las necesidades y servicios del AmI. El marco ontológico de AmICL debe ser capaz de representar los nuevos objetos y comportamientos en el ambiente, para modelar adecuadamente las dinámicas del usuario y del contexto, detectando cambios en los patrones de comportamiento, y reflejándolos en sus ontologías. Algunos elementos son predefinidos, pero otros deben ser “emergentes”.

En particular, en este trabajo AmICL interactúa con el middleware semántico propuesto en [6] a través de los servicios de CARMiCLOC. En específico, los agentes de AmICL registran información sobre los eventos semánticos del sistema y sobre las ontologías que manejan, realizando anotaciones semánticas de las fuentes de conocimiento (a nivel de meta-datos) usadas en el AmI. Durante el proceso de registro semántico, si algún concepto no puede ser ubicado o relacionado con ninguno de los conceptos de las meta-ontologías del middleware a través de su descripción, deben analizarse sus propiedades, para detectar alguna relación, si aun así no puede ser ubicado, entonces es un concepto completamente desconocido, y por lo tanto se requiere una posible adaptación de la meta-ontología, lo que representa una evolución del marco conceptual del AmI.

3.2.1. Ontología emergente a partir de las meta-ontologías

El Middleware semántico ofrece sus servicios para, entre otras cosas, hacer emerger ontologías, de acuerdo a los nuevos elementos de contexto que se requieran manejar en AmICL. El contexto del AmI puede cambiar de acuerdo a la dinámica propia del ambiente y a los nuevos usuarios, actividades, recursos y condiciones ambientales, que se presenten y que sea necesario modelar. Las ontologías de contexto representan el conocimiento contextual que está presente en ese AmI. Este conocimiento puede estar disperso en diferentes fuentes de datos y dispositivos: bases de datos, documentos, datos monitoreados, dispositivos corporales, etc.

La base de las ontologías de contexto es la *Meta-Ontología*

del Contexto, la cual se usa para definir el modelo general del contexto, manteniendo la integridad y la coherencia entre los conceptos, ya que ella guarda la información global de cómo deben estructurarse las ontologías y como deben relacionarse entre sí (ver Fig. 3). Particularmente, esta meta-ontología enlaza y estructura las ontologías del contexto, y organiza la información semántica de las fuentes de datos que surgen en el AmI, por la incorporación al AmI de nuevos dispositivos, agentes, usuarios, etc.

Para poder determinar qué elementos hay en el ambiente, los dispositivos (sensores, equipos) de AmICL se registran en el middleware semántico, para poder ser detectados y modelados a través del servicio de CARMiCLOC.

Específicamente, en nuestro enfoque del Conocimiento como un Servicio, AmICL maneja varias ontologías de contexto distribuidas en los agentes del AmI, y una meta-ontología de contexto central. La meta-ontología central es manejada por el middleware semántico [6], que provee servicios ontológicos a las ontologías de los agentes del sistema, para permitir la emergencia ontológica. Los agentes utilizan sus ontologías locales para realizar sus servicios, interactuar con otros agentes, etc. Cuando se detecta la necesidad de actualizar su conocimiento (por ejemplo, por la llegada de un nuevo artefacto al ambiente, cambio de contexto, etc.) los agentes actualizan sus ontologías usando los servicios de CARMiCLOC. Además, al ingresar un nuevo elemento en el AmI, este debe sincronizar su ontología de contexto con la meta-ontología, usando el servicio de CARMiCLOC “Adquisición de Contexto”.

Ahora bien, para que la ontología pueda emerger, los servicios de CARMiCLOC invocan los servicios del middleware semántico, por ejemplo el de registro semántico para ingresar un nuevo elemento en el AmI. También consulta la Meta-Ontología del Contexto, a través del servicio “Exploración de ontologías”.

3.3. Especificación de los servicios para el análisis de contexto

En CARMiCLOC, todos los servicios pueden ser invocados de manera independiente, según lo requieran los agentes de AmICL. Es decir, el AmI educativo invoca los servicios de CARMiCLOC usando los agentes AGS y AS. En particular, los servicios de CARMiCLOC permiten el análisis de contexto, por lo que son usados por el AmI de la siguiente manera:

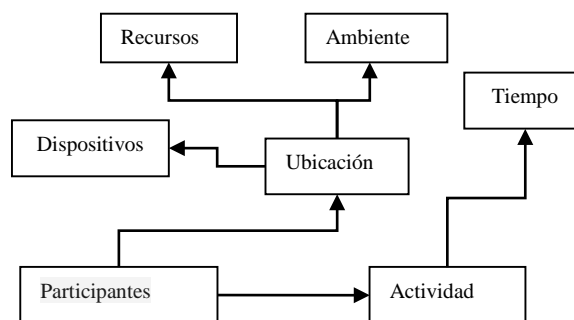


Figura 3. Meta-Ontología del Contexto
Fuente: Los Autores

- Los servicios de Adquisición del Contexto (Se_1) y Distribución del Contexto (Se_4), son usados para actualizar las ontologías del contexto.
- El servicio de Razonamiento del Contexto (Se_3) es usado para inferir si nuevos eventos han ocurrido que han modificado el contexto. Eventualmente, se llama al servicio Verificación y resolución de inconsistencia del Contexto (Se_5), si se detectan problemas en la información.
- Los servicios de Modelado del Contexto (Se_2) y Seguridad y privacidad del contexto (Se_6), son invocados implícitamente por los otros servicios de CARMiCLOC.

Estos servicios posibilitan proveer al conocimiento del contexto como un servicio, y tienen implícito un proceso de emergencia ontológica. Durante el proceso de emergencia ontológica, los servicios de CARMiCLOC invocan los servicios del middleware semántico que requiera.

4. Experimentación

4.1. Salón de clases inteligente (SaCI).

En esta sección se usa como caso de estudio el Aula de Clases Inteligente, denominado SaCI, especificado en [10, 11]. SaCI usa el middleware AmICL para desplegarse, y propone dos tipos de agentes, uno para caracterizar los componentes de software, y el otro para definir los componentes de hardware en el ambiente. Básicamente, los agentes definen las capas CAF y CAL de AmICL.

- Capa de AmI Físico (CAF): En esta capa se caracterizan los diversos dispositivos presentes en el SaCI (inteligentes o no) como agentes. Ejemplos de posibles agentes en CAL son Pizarra Inteligentes, Robots, pupitres inteligentes, etc.
- Capa de AmI Lógico (CAL). En esta capa se caracterizan los diversos componentes de software en SaCI como agentes. Ejemplos de software es el sistema recomendador de contenidos digitales educativos, el ambiente virtual de aprendizaje (VLE por sus siglas en inglés) o el sistema académico.

Para especificar esos agentes, se ha propuesto en [10] un modelo general de agente basado en la metodología MASINA [12]. Por otro lado, la lista completa de esos agentes se encuentra en [10].

4.2. Instanciación de la arquitectura.

Con el fin de mostrar el proceso de integración de los 3 middleware para proveer “el conocimiento como un servicio”, se describe a continuación su funcionamiento en el SaCI. Este servicio permite la emergencia de ontologías en el entorno inteligente de aprendizaje basado en el paradigma de la nube.

En particular, AmICL se integra con los servicios en la nube por medio de los agentes AS y AGS, de tal manera que a través de estos dos agentes pueden usarse todos los servicios especificados por CARMiCLOC y el middleware semántico. Igualmente, los servicios de CARMiCLOC y el middleware semántico pueden ser invocados entre ellos.

A continuación se plantean varios escenarios en SaCI, para analizar el uso “del conocimiento como un servicio”.

4.2.1. Situación 1: adecuación del ambiente para una clase

En este escenario (Fig. 4) se muestra el proceso de adaptación del SaCI para una actividad académica particular, por ejemplo, para una clase de programación. El proceso se inicia cuando un Agente Tutor consciente del contexto, el cual determina que la próxima clase a dictarse en el aula corresponde a la clase del profesor que él caracteriza. El AT inicia el proceso de adecuación del aula, que permitirá aprovechar el ambiente de la mejor manera. Los pasos para llevar a cabo este proceso se detallan a continuación (ver Fig. 4):

1. AT solicita a AGS un AS que permita consumir los servicios de CARMiCLOC.
2. AGS realiza el descubrimiento del servicio y retorna el resultado al AT.
3. AT invoca los servicios de conciencia contextual de CARMiCLOC por medio del AS.
4. AD envía la información que CARMiCLOC requiera para llevar a cabo la tarea de descubrimiento del contexto.
5. CARMiCLOC se encarga de adquirir y razonar sobre el contexto, para lo cual requiere el modelo del contexto actual (Para ese AmI Educativo en ese momento). Para ello, solicita al middleware semántico el servicio de búsqueda de ontologías, en este caso requiere la ontología de componentes de ese SaCI, y la ontología de contexto para la actividad a realizar (clase de programación).
6. CARMiCLOC, en el proceso de adquisición de contexto, captura datos sobre todos los componentes (dispositivos, sensores, recursos, agentes, etc.), presentes en el AmI, y ofrece estos datos para poblar la meta-ontología, a través del servicio “Registro Semántico” del middleware semántico.
7. Las ontologías de componentes y de contexto son retornadas a CARMiCLOC, para razonar en base a ellas, y luego distribuir el contexto al AmICL, para su uso.
8. El AT toma esa información, con la cual decide los componentes que deben ser inicializados y los servicios que serán activados, para que la clase se desarrolle adecuadamente.

La Ontología emergerá, dependiendo de la información semántica que AmICL y CARMiCLOC hayan registrado en el Middleware Semántico. Si el profesor dicta una clase de programación, entonces se retornará una ontología poblada como la de la Fig. 5, de esta forma AT puede conocer que el curso cuenta, por ejemplo, con el siguiente software: un sistema recomendador de contenidos, un VLE, un software de reconocimiento de rostro para autenticar a los usuarios, entre otras cosas.

4.2.2. Situación 2: ingreso de usuarios al salón inteligente

Con este escenario (Fig. 6) se quiere mostrar como es el proceso de acceso de los usuarios al SaCI, para autenticarlos y adecuar el contexto. Una vez que el ambiente ha sido adecuado a la clase que será dictada, se inicia el proceso de autenticación de usuarios. Este proceso es importante no solo para conocer que estudiantes asistieron a clase, sino para que el SaCI pueda tomar decisiones acordes al perfil del usuario, lo que permitirá ofrecer servicios académicos adecuados al contexto (en este caso, estudiantes), que mejoren el proceso de enseñanza-aprendizaje adaptado al estilo de aprendizaje de cada estudiante particular. Los pasos que se llevan a cabo en esta situación se detallan a continuación (ver Fig. 6):

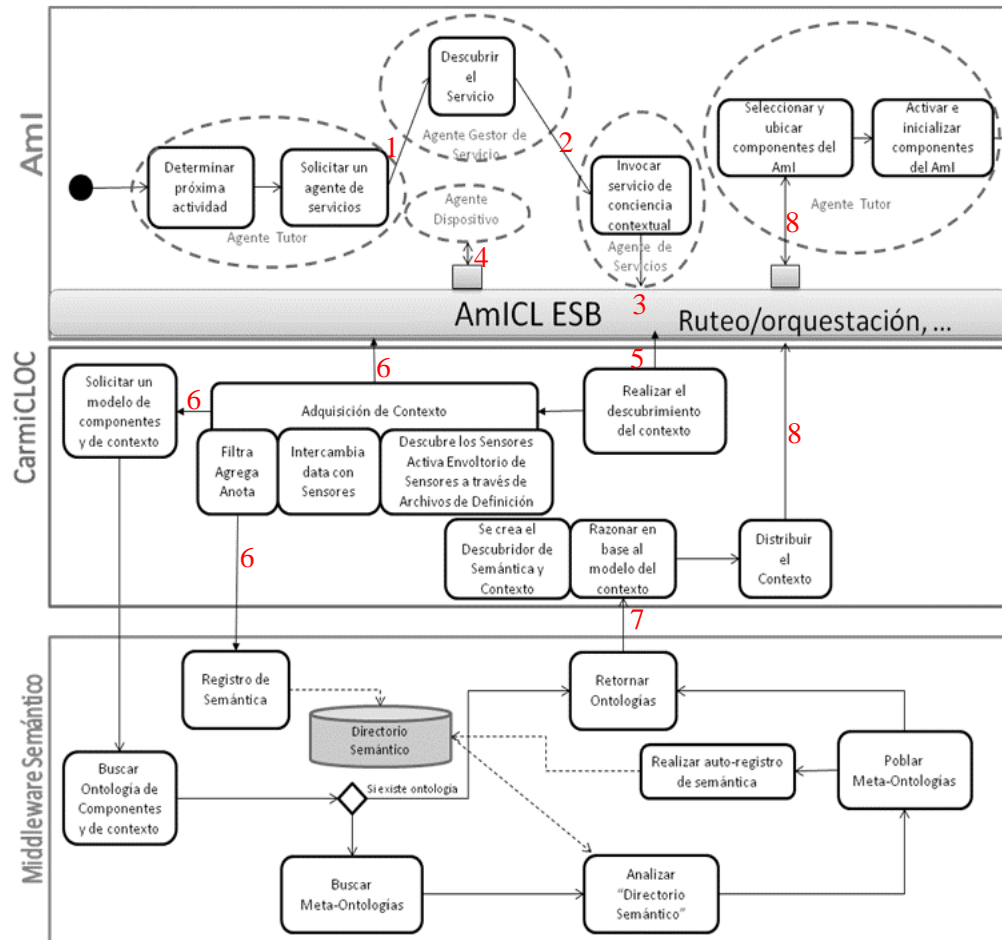


Figura 4. Adecuación del Ambiente para una clase
Fuente: Los Autores

1. En el SaCI se activan algunos agentes de acuerdo con la adecuación del ambiente: como el Agente VLE, y un Agente Dispositivo (AD) instanciado con un dispositivo de tipo cámara que ayudará en el proceso de autenticación de los usuarios. De igual manera, se registra en el AGS el agente AS que caracteriza al servicio de reconocimiento de rostro.
2. Todas estas configuraciones son posibles debido a que en la ontología de componentes retornada por CARMiCLOC se encontraba tal información, la cual utiliza el agente consciente de contexto AT para adecuar el ambiente, para que el AmI responda de manera inteligente ante la situación actual.
3. La autenticación de un usuario se puede realizar de forma virtual o física. Cuando un usuario (estudiante o profesor) se conecta de manera virtual, el Agente VLE lo identifica y autentica, utilizando, por ejemplo, el AS de reconocimiento de rostro que fue registrado en el AGS. Si el estudiante/profesor ingresa físicamente al aula, el AD (cámara) lo identifica y autentica, usando de igual manera el AS de reconocimiento de rostro.
4. El AD luego de identificar al usuario, solicita el servicio de búsqueda y verificación de usuarios al Agente VLE (o al AT), para verificar si son usuarios que les corresponde ese ambiente en ese momento, y poder darles acceso al lugar. Este agente requiere conocer la información del contexto sobre: cuál actividad corresponde a ese ambiente, para ese momento, y cuáles son los usuarios autorizados a entrar. Para ello invoca los servicios de conciencia contextual de CARMiCLOC por medio del AS (4.1).
5. De acuerdo a la información de contexto obtenida, se determina si se puede autorizar el acceso al usuario. Si el usuario no está definido para ese contexto (5.1), el Agente VLE consulta la planificación para determinar si es un usuario especial (que no estaba predefinido), para autorizar su acceso (5.2), o en caso contrario, denegar su acceso (5.3).
6. Si se autoriza el acceso, se invoca nuevamente el servicio de conciencia contextual, para poder adecuar el ambiente y los dispositivos a las características y perfil del usuario.
7. El AS envía la información que CARMiCLOC requiera para llevar a cabo la tarea de descubrimiento del contexto. CARMiCLOC se encarga de razonar sobre el contexto y con la información recibida del AmI, en este caso sobre los usuarios, realiza el registro semántico, para que el middleware semántico pueda poblar la ontología, retornando la ontología de contexto adecuada, para que CARMiCLOC pueda distribuir el contexto al SaCI.

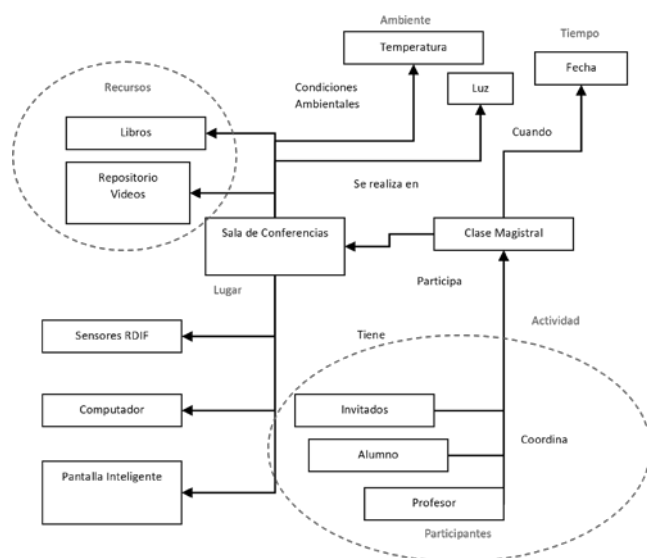


Figura 5. Modelo del Contexto Emergente (Caso: Clase Práctica de Programación)

Fuente: Los Autores

8. CARMiCLOC, a través del servicio de modelado de contexto, puede gestionar cambios en el modelo de contexto. Este servicio de modelado puede ser solicitado por el AmI, cuando no logra entender el contexto en un momento dado, y requiere solicitar al CARMiCLOC que adquiera un nuevo modelo de contexto. Para ello, CARMiCLOC solicita al middleware semántico el servicio de actualización de las ontologías. El Middleware semántico realiza la población de la meta-ontología, para asociar de forma adecuada nuevos conceptos en la ontología, y agregar nuevos atributos o axiomas. Si no posee la información requerida para la extensión de la ontología, puede necesitar la consulta de fuentes externas con otros modelos ontológicos.
9. Luego de autorizar el acceso, se registra en el SMA un agente que lo caracteriza: en el caso de los profesores un AT, y para el caso de los estudiantes se instancia un APE para cada uno. Seguidamente, el APE o el AT invocará los servicios de CARMiCLOC a través del AS correspondiente, lo que permitirá determinar actividades o servicios que pueden ser ofrecidos al estudiante acorde con el dominio y perfil del mismo. Por ejemplo, si el usuario tiene una discapacidad visual, el APE podría tomar la decisión de activar las opciones de accesibilidad en el dispositivo que esté utilizando ese estudiante.

4.3. Análisis comparativo.

A continuación se presenta un análisis comparativo con estudios anteriores que tratan de proponer soluciones en el ámbito de la consciencia contextual y del uso de ontologías en ambientes inteligentes educativos.

En [1] proponen un diseño por capas de un AmI, con una capa genérica que maneja agentes conscientes del contexto de forma independiente y descentralizada. En nuestro caso, además de manejar una capa independiente con los servicios de consciencia contextual, se manejan servicios del

middleware semántico, lo que permite al AmI responder a situaciones emergentes que pueden presentarse en un momento dado.

Por su parte, [2] presentan una ontología para representar la interacción entre los perfiles de usuarios y el contexto, en un entorno de aprendizaje colaborativo. En este caso, se expone una ontología para modelar el perfil del usuario y una ontología para modelar el contexto. Además, en nuestra propuesta se hace uso de meta-ontologías, lo que significa tener ontologías de contexto en un nivel más genérico, que puedan ser adaptadas, a partir de procesos de minería ontológica, de acuerdo a las necesidades del AmI.

La principal ventaja de nuestra propuesta es la posibilidad de dar respuesta a la necesidad de adaptación de los servicios que ofrece un AmI educativo a los usuarios de acuerdo a su contexto, a través de la emergencia ontológica contextual, considerando que las ontologías se han convertido en componentes muy importantes en el manejo de los modelos conceptuales de los AmI. La emergencia ontológica representa una forma dinámica de crear ontologías de contexto, por lo tanto, es una forma dinámica de manejar los modelos conceptuales, permitiendo actualizarlos con los nuevos conceptos que surgen en el AmI y así tener una representación mucho más acertada de la realidad.

5. Conclusiones

La consciencia contextual es un elemento fundamental que debe estar presente en un ambiente inteligente, ya que le permite al mismo tomar decisiones de forma inteligente y adaptadas al usuario, lo cual posibilita a éste último interactuar con su entorno de forma personalizada [13, 14]. Es por ello que en este artículo se presentó la integración del Middleware Semántico y CARMiCLOC con AmICL, lo cual dotará al Ambiente Inteligente de Aprendizaje con servicios que permitirán conocer el contexto, para que los Agentes de software que sirven como capa inteligente de AmICL puedan descubrir servicios de la nube que potencien el aprendizaje personalizado de los estudiantes, y apoyen las tareas de enseñanza de los profesores.

La emergencia ontológica beneficia al proceso de análisis de consciencia contextual, ya que permite crear ontologías de contexto en base a una ontología más general, denominada meta-ontología, lo cual posibilita responder a situaciones emergentes (imprevistas) al poblar la meta-ontología por medio de procesos de minería ontológica (alineación y mezclas) a partir de ontologías creadas en situaciones similares (ver [6] para más detalles de ese proceso); basándose en todo momento, en la información semántica que AmICL y CARMiCLOC hayan registrado en el Middleware Semántico.

Los agentes de la capa de gestión de AmI lógico, específicamente los agentes APE y AT, son agentes conscientes del contexto, ya que ellos son capaces de invocar el servicio de consciencia contextual que provee CARMiCLOC, por medio del agente SA, y tomar decisiones inteligentes basándose en los resultados del contexto actual obtenido, personalizando y mejorando el desarrollo de las actividades académicas realizadas por los usuarios que caracterizan.

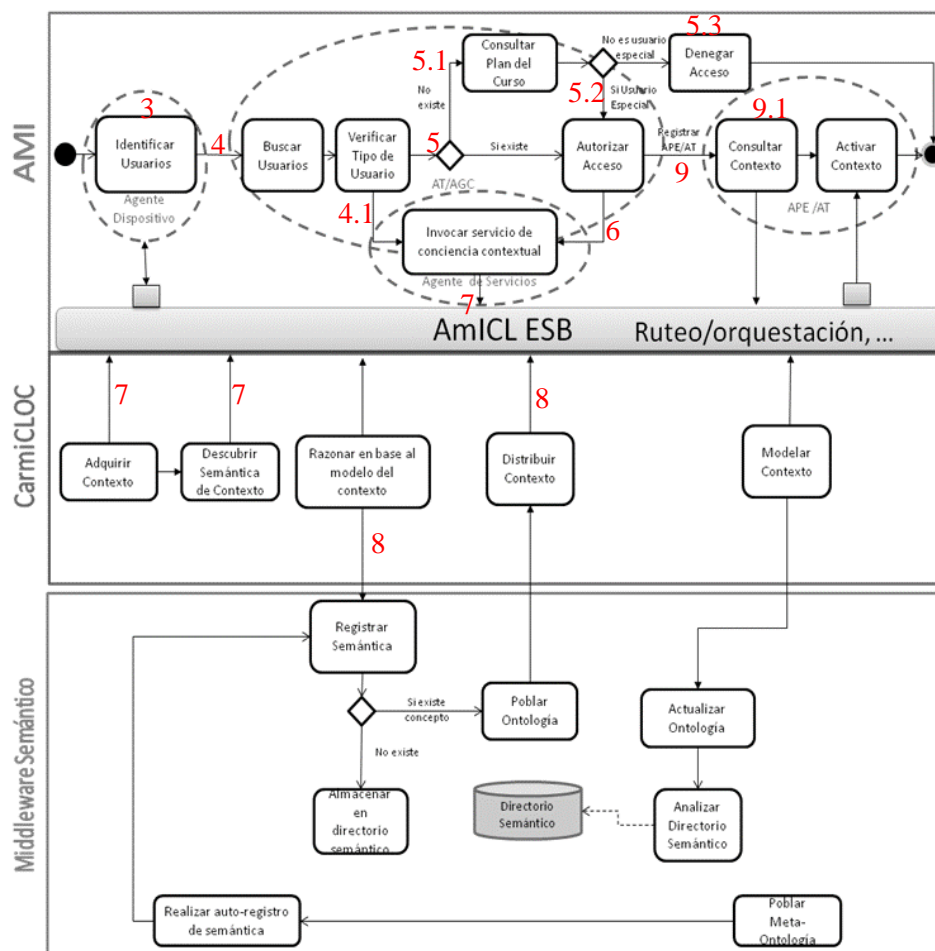


Figura 6. Autenticación de Usuarios en el Ambiente
Fuente: Los Autores

El proceso de integración presentado en este artículo se basa en un modelo desacoplado que sigue las recomendaciones de una arquitectura SOA, ya que tanto CARMiCLOC como el Middleware Semántico ofrecen sus funcionalidades como servicios, y AmICL puede acceder a ellas por medio del componente de la nube del mismo. De esta forma, AmICL puede consumir los servicios de análisis contextual ofrecidos por CARMiCLOC, pero sin ser totalmente dependiente, ya que CARMiCLOC y el Middleware Semántico pueden ofrecer sus servicios a otras aplicaciones. Igualmente, AmICL podría utilizar otro servicio de consciencia contextual, siempre y cuando ofrezca las ventajas que ofrecen tanto CARMiCLOC como el Middleware Semántico.

Los casos estudiados en la sección 4 muestran como AmICL puede adaptarse y cambiar su comportamiento interno de acuerdo a las necesidades del ambiente, ya que al usar los servicios de CARMiCLOC y el Middleware Semántico, puede no solo conocer que componentes se encuentran en el ambiente, sino que también puede responder a situaciones imprevistas, por medio del uso de las ontologías emergentes, mejorando así las capacidades del AmI, siempre orientado a mejorar el proceso de enseñanza-aprendizaje por medio del uso de servicios en la nube, como los sistemas de

recomendación de contenidos, los servicios de analítica de datos para conocer como ha sido el proceso de aprendizaje de los estudiantes, entre otros.

El trabajo futuro está orientado a la implementación del proceso de integración en un SaCI real, con lo cual se podrán detectar y corregir fallas que permitan mejorar el rendimiento y las capacidades de integración, en cuanto a escalabilidad, interoperabilidad, entre otros.

Agradecimientos

Aguilar ha sido parcialmente financiado por el Proyecto Prometeo del Ministerio de Educación Superior, Ciencia, Tecnología e Innovación de la República de Ecuador.

Referencias

- [1] Olaru, A., Florea, M. and Seghrouchni, A., A context-aware multi-agent system as a middleware for ambient intelligence, *International Journal of Mobile Networks and Applications*, 18(3), pp. 429-443, 2013. DOI: 10.1007/s11036-012-0408-9
- [2] Luna, V., Quintero, R., Torres, M., Moreno-Ibarra, M., Guzmán, G. and Escamilla I., An ontology-based approach for representing the interaction process between user profile and its context for collaborative learning environments, *International Journal of Computers in Human Behavior*, 51, pp. 1387-1394, 2015. DOI:

- 10.1016/j.chb.2014.10.004
- [3] Hwang, G., Definition, Framework and research issues of smart learning Environments-A Context-Aware ubiquitous learning perspective, *International Journal of Smart Learning Environments*, 1(1), pp. 1-14, 2014. DOI: 10.1186/s40561-014-0004-5
 - [4] Sánchez, M., Aguilar, J., Cordero, J. and Valdiviezo, P., Basic features of a reflective middleware for Intelligent Learning Environment In The Cloud (IECL), *Proceedings of 2015 Asia-Pacific Conference on Computer Aided System Engineering (APCASE'15)*, 2015. DOI: 10.1109/APCASE.2015.8
 - [5] Sánchez, M., Aguilar, J., Cordero, J. and Valdiviezo, P., A smart learning environment based on cloud learning, *International Journal of Advanced Information Science and Technology (IJAST)*, [online]. 39 (39), pp. 39-52, 2015. Available at: <http://ijast.com/index.php/pub/item/404-a-smart-learning-environment-based-on-cloud-learning>
 - [6] Mendonça, M., Aguilar, J. y Perozo, N., *Middleware reflexivo semántico para ambientes inteligentes*, *Proceedings of Segunda Conferencia Nacional de Computación, Informática y Sistemas (CoNCISa)*, Caracas, Venezuela, 2014.
 - [7] Aguilar, J., Jerez, M., Exposito, E. and Villemur, T., CARMiCLOC: Context Awareness Middleware in Cloud computing, *Proceedings of Latin American Computing Conference (CLEI)*, 2015. DOI: 10.1109/CLEI.2015.7360013
 - [8] Aguilar, J., Ríos, A., Ríos, F. and Cerrada, M., *Sistemas multiagentes y sus aplicaciones en automatización industrial*, 2nd ed., Mérida: Talleres Gráficos de la Universidad de Los Andes, 2013.
 - [9] Guizzardi, G., On ontology, ontologies, conceptualizations, modeling languages, and (meta) models, *Proceedings of the 2007 conference on Databases and Information Systems IV: Selected Papers from the Seventh International Baltic Conference DB&IS'2006*, 2007, pp. 18-39.
 - [10] Valdiviezo, P., Cordero, J., Aguilar, J. and Sánchez, M., Conceptual design of a smart classroom base on multiagent system, *Proceedings of the 2015 International Conference on Artificial Intelligence (ICAI'15)*, 2015.
 - [11] Aguilar, J., Sánchez, M., Valdiviezo, P. y Cordero, J., Mecanismos de coordinación en un salón inteligente, *Proceedings of the 6to Congreso Iberoamericano de Estudiantes de Ingeniería Eléctrica (VI CIBELEC 2015)*, 2015.
 - [12] Aguilar, J., Besembel, I., Cerrada, M., Hidrobo, F. y Narciso, F., Una metodología para el modelado de sistemas de ingeniería orientado a agentes, *Revista Iberoamericana de Inteligencia Artificial*, 12 (38), pp. 39-60, 2008.
 - [13] Castro, M., *Modelado de interacciones sensibles al contexto mediante reglas en ambientes inteligentes*, Phd. Thesis, Departamento de Informática, Universidad de Córdoba, Córdoba, España, 2013.
 - [14] Tapia, D.I., Fraile, J.A., Rodríguez S., Alonso R.S. y J. and Corchado, M., Integrating hardware agents into an enhanced multi-agent architecture for ambient intelligence systems, *International Journal of Information Sciences*, 222, pp. 47-65, 2013. DOI: 10.1016/j.ins.2011.05.002
 - [15] Bravo, V., Aguilar, J., Rivas, F. y Cerrada, M., Diseño de un medio de gestión de servicios para sistemas multiagentes, *Proceeding of XXX Conferencia Latinoamericana de Informática*, pp. 431-439, 2004.



J. Aguilar, received the BSc. degree in System Engineering in 1987 (Universidad de los Andes-Venezuela), the MSc. degree in Computer Sciences in 1991 (Universite Paul Sabatier-France), and the PhD. degree in Computer Sciences in 1995 (Universite Rene Descartes-France). He was a Postdoctoral Research Fellow in the Department of Computer Sciences at the University of Houston (1999-2000). He is a titular professor in the Department of Computer Science at the Universidad de los Andes, Mérida, Venezuela, and Prometeo Research at the Universidad Técnica Particular de Loja. He is member of the Mérida Science Academy and of the IEEE CIS Technical Committee on Neural Networks. He has published more than 400 papers and 10 books, in the field of parallel and distributed systems, computational intelligence, science and technology management, etc. His research interest includes Artificial Intelligence, Semantic Mining, Big Data, Emergent System, and Smart Environment.
ORCID: 0000-0003-4194-6882



M. Mendonça, received the BSc. degree in Informatics Engineering in 1998 from Universidad Centroccidental Lisandro Alvarado, Venezuela. The MSc. degree in Computer Science from Universidad Centroccidental Lisandro Alvarado, Venezuela, in 2004. Currently doing Dr. degree in applied science (Ontological Learning) in Universidad de los Andes, Venezuela. She is an aggregate professor in the Department of System at the Universidad Centroccidental Lisandro Alvarado, Barquisimeto, Venezuela. Her research interest includes Artificial Intelligence, Ontological Learning, Emergent Computing and Smart Environment.
ORCID: 0000-0003-2512-4125



M. Jerez, received the BSc. degree in System Engineering in 1998 from Universidad de los Andes, Venezuela. The MSc. degree in Computer Science from Universidad de los Andes, Venezuela, in 2007. Currently doing Dr. degree in applied science (Context Awareness) in Universidad de los Andes, Venezuela. He is a researcher in the Centre of Microelectronics and Distributed Systems (CEMISID) at the Universidad de Los Andes, Mérida, Venezuela. His research interest includes Artificial Intelligence, Context awareness, Cloud Computing, Autonomic Computing, and Smart Environment.
ORCID: orcid.org/0000-0002-5582-6876



M. Sánchez, received the BSc. degree in System Engineering in 2002 from Universidad de los Andes, Venezuela. The MSc. degree in Computer Science from Universidad de los Andes, Venezuela, in 2012. Currently doing Dr. degree in applied science (Ambient Intelligence) in Universidad de los Andes, Venezuela. He is an aggregate professor in the Department of Informatics Engineering at the Universidad Nacional Experimental del Táchira, San Cristóbal, Venezuela. His research interest includes Artificial Intelligence, Learning Environments, Graphic Computing, Programming languages, among others.
ORCID: orcid.org/0000-0002-3310-3394



UNIVERSIDAD NACIONAL DE COLOMBIA

SEDE MEDELLÍN
FACULTAD DE MINAS

Área Curricular de Ingeniería
de Sistemas e Informática

Oferta de Posgrados

Especialización en Sistemas
Especialización en Mercados de Energía
Maestría en Ingeniería - Ingeniería de Sistemas
Doctorado en Ingeniería- Sistema e Informática

Mayor información:

E-mail: acsei_med@unal.edu.co
Teléfono: (57-4) 425 5365