

MODELO DE CONTEXTO Y DE DOMINIO PARA LA INGENIERÍA DE REQUISITOS DE SISTEMAS UBICUOS

Liliana González Palacio*
Germán Urrego Giraldo**

Recibido: 06/08/2010

Aceptado: 08/10/2010

RESUMEN

Este artículo presenta los modelos de contexto y de dominio de un ambiente ubicuo genérico, como instrumento para facilitar la transición del conocimiento de la fase de definición a la fase de análisis y servir de base para la obtención de los requisitos. El aporte es derivado de una investigación en metodologías para construcción de ambientes ubicuos, enmarcada dentro del macroproyecto para la construcción de una plataforma de cocreación de productos y servicios innovadores en el área de telecomunicaciones. Esta iniciativa hace parte del Centro de Excelencia ARTICA.

Palabras clave: ingeniería de software, ingeniería de requisitos, metodologías, sistemas ubicuos, modelo de contexto, modelo de dominio.

* Magíster en Ingeniería U. de A. Docente Ingeniería de Sistemas Universidad de Medellín. Estudiante de Doctorado en Ingeniería. Medellín-Colombia. Correo electrónico: lignonalez@udem.edu.co.

** Ph. D. en Informática Universidad de París I. Profesor Departamento de Ingeniería de Sistemas Universidad de Antioquia. Medellín-Colombia. Correo electrónico: gaurrego@udea.edu.co.

CONTEXT MODEL AND DOMAIN MODEL FOR REQUIREMENTS ENGINEERING UBIQUITOUS SYSTEMS

ABSTRACT

In this paper, the context and domain models, as an instrument to facilitate the knowledge transition between the definition phase and the analysis phase, and to serve as a basis to the requirements obtainment is presented. This contribution was obtained from a research project of building methodologies ubiquitous environments, within the macro project of construction of a platform for the co-creation of products and innovation services in telecommunications area. This initiative is part of the ARTICA excellence center.

Key words: software engineering, requirements engineering, methodologies, ubiquitous systems, context model, domain model.

INTRODUCCIÓN

En cualquier lugar, en cualquier momento y a través de cualquier dispositivo es la definición más sencilla para explicar el concepto de ubicuidad desde la perspectiva tecnológica, que ha de modificar la manera como el hombre experimenta el mundo [1].

Consecuentemente, los sistemas ubicuos (conocidos también como pervasivos) constituyen entornos donde los elementos tecnológicos son invisibles para los usuarios pero su funcionalidad continúa proporcionándose, y a la vez, los dispositivos inteligentes se insertan en las tareas diarias, haciendo que la interacción usuario-sistema sea natural y desinhibida en todo tipo de situaciones y circunstancias [2].

Una etapa inicial y muy importante en el proceso de desarrollo de cualquier sistema, incluyendo los denominados ubicuos, es la ingeniería de requisitos (IR), donde se lleva a cabo el proceso de descubrir, analizar, escribir y verificar los servicios y restricciones del nuevo sistema [3]. Su relevancia radica en que de la definición de los requisitos dependerán las etapas subsecuentes del desarrollo. Si esta fase no se lleva a cabo con el debido rigor puede provocar serios problemas en tiempos de entrega, aumento en presupuestos y expectativas insatisfechas de los clientes, ya que el producto final estará incompleto o poco funcional. De ahí el interés y la importancia de proponer metodologías que permitan la captura y tratamiento de requisitos de una manera sistemática y organizada.

Para lograr el reconocimiento de los requisitos es fundamental caracterizar el dominio al que pertenece el tipo de sistema a construir, buscando generalizar el conocimiento y facilitar posteriores desarrollos. En orden a cumplir este objetivo algunos autores [4] proponen la construcción de un modelo de contexto de utilización y un modelo de dominio para recopilar información sobre los servicios típicos que ofrecen los sistemas enmarcados en un determinado contexto de aplicación, y los agentes e interacciones que por defecto se presen-

tan al analizar un campo de conocimiento particular.

La fase de IR en el caso de los ambientes ubicuos se dificulta teniendo en cuenta algunas propiedades particulares que poseen y situaciones presentadas durante su desarrollo [5]. Como características relevantes vale la pena mencionar su orientación a la identificación, localización, detección de señales, marcada comunicación entre dispositivos, requerimientos adicionales de memoria, adaptación a cambios en el entorno donde están ubicados, entre otras [6], que aumentan el riesgo de construir sistemas ubicuos que proveen funcionalidades innecesarias con un consumo adicional de recursos, tan escasos en este dominio particular [7].

De otro lado, algunas situaciones que complican el proceso de IR en este dominio particular se enuncian a continuación [8]: a) El proceso es realizado por expertos en electrónica, pero alejados del uso de las metodologías de análisis y diseño de sistemas. b) La ausencia de metodologías específicas, que son reemplazadas por aquellas de propósito general sin ninguna adaptación a este tipo de sistemas. c) Las metodologías para construcción de sistemas ubicuos no proporcionan bases y guías sólidas para la definición del sistema, fase en la cual se adquiere conocimiento fundamental para la posterior definición y representación de requisitos. d) El desarrollo de un sistema ubicuo o pervasivo supone el uso de diversas y cambiantes tecnologías para satisfacer los requisitos de los usuarios. Por el contrario, con los métodos actuales, durante el modelamiento los desarrolladores quedan ligados a aspectos tecnológicos fijos, que impiden la adopción de otras tecnologías conduciendo a una rápida obsolescencia de las metodologías usadas y de los sistemas desarrollados con ellas.

En la actualidad son pocas las investigaciones dedicadas a la Ingeniería de Requisitos para este tipo de sistemas, y los aportes en cuanto a desarrollo de metodologías y de procesos de desarrollo son aún limitados [6-8, 11-13, 15, 16, 18, 19]. La mayoría de las iniciativas están centradas únicamente en

la etapa de diseño, particularmente en el estudio de la interfaz persona-ordenador, de los factores humanos y en la elaboración de recomendaciones generales.

Como una contribución al mejoramiento de las tecnologías para el desarrollo de sistemas ubicuos y de las limitaciones en su definición y análisis, este artículo presenta los modelos de contexto y de dominio que proveen una base para la fase de definición de un sistema ubicuo genérico. Estos resultados provienen de un proyecto de investigación sobre la definición y conceptualizaron de sistemas ubicuos desarrollado en el marco de un macroproyecto denominado “Plataforma para la cocreación de productos y servicios innovados en el área de telecomunicaciones”, perteneciente al Centro de Excelencia ARTICA.

Este trabajo continúa con la generación de un modelo de requisitos y un modelo conceptual genéricos para ambientes ubicuos, y su posterior implantación en la plataforma de cocreación, como mecanismo para garantizar el acceso sin limitantes de tiempo y acceso por parte de los participantes en el proceso de innovación. El resto del artículo está constituido por las siguientes secciones: Posterior a la introducción, en la sección 2 de materiales y métodos se mencionan los conceptos relevantes para entender la problemática tratada, y algunas aproximaciones a la solución planteadas por otros autores. Luego se presenta, en la sección 3 la propuesta objeto del artículo. La discusión de resultados ocupa la sección 4. En la quinta sección se enuncian las conclusiones. Por último se presenta la bibliografía.

1. MATERIALES Y MÉTODOS

A continuación se presentan algunos conceptos básicos que permiten entender la problemática planteada. Inicialmente se introduce la terminología relevante, y luego una breve revisión de la literatura.

1.1 Revisión de términos

Este artículo se fundamenta en dos pilares: los ambientes ubicuos y la ingeniería de requisitos. Cada una de las áreas mencionadas tiene una serie de conceptos particulares que vale la pena resaltar.

La **computación ubicua** da nombre a un paradigma de computación novedoso en el que la tecnología aparece integrada en elementos cotidianos de la vida real y en el que la interacción usuario-sistema se lleva a cabo de forma transparente. Estas características suponen un cambio tanto en la concepción de los sistemas a desarrollar como en la forma de interactuar con éstos [8]. Se habla de **ubicuidad** por la posibilidad de acceso a los recursos sin limitantes de tiempo, medio de acceso ni lugar, y se acuña el término **transparencia** para referirse a tecnologías que entran en la trama de la vida cotidiana hasta no distinguirse. Para soportar estas propiedades, el sistema pervasivo debe contar con orientación a la identificación, mecanismos de localización de usuarios, detección de señales provenientes del ambiente, marcada comunicación entre dispositivos y variedad en estos (forma, tipo de acceso, tipo de conexión a redes), requerimientos adicionales de hardware, adaptación a cambios en el entorno donde están ubicados los usuarios, infraestructura provista de sensores, entre otras [9].

La construcción de este tipo de sistemas involucra complejidades adicionales, debido a sus características particulares [6], y desde la fase de ingeniería de requisitos (que incluye actividades de definición y análisis del sistema) deben existir métodos y técnicas para incorporar los elementos ya mencionados, garantizando de este modo que funcionalidades y servicios propios de este dominio sean modelados correctamente y tenidos en cuenta en etapas tempranas del desarrollo.

En la actividad de definición del sistema, previa a la captura de requisitos, se establecen los problemas a resolver, las fuentes de conocimiento que pueden ayudar en la búsqueda de las soluciones, los intereses y necesidades que aquellos interesados

(denominados en adelante agentes) desean resolver a partir de la implantación de la nueva aplicación o sistema, y los servicios/funcionalidades típicas para un dominio particular [10].

Como apoyo para esta actividad existen los modelos de contexto de utilización y el de dominio [4]. El primero representa las acciones e interacciones de los agentes implicados en el funcionamiento del sistema, en tanto que el segundo, también denominado modelo de objetos y servicios del dominio, se constituye en un medio para sintetizar y hacer útil el conocimiento de un dominio con miras a la especificación de las funcionalidades y de los atributos de calidad de un sistema.

El modelo de contexto de utilización se elabora haciendo un inventario de agentes involucrados en el sistema, para luego identificar interacciones entre ellos, y necesidades que deben suplir con la implementación de la nueva solución.

Los modelos mencionados aportan información importante para la posterior construcción del modelo de requisitos y conceptual del sistema [4].

Con este recorrido por el fundamento conceptual del artículo es posible ahora abordar la revisión de la literatura.

1.2 Revisión de la literatura

La búsqueda de propuestas en el tema estuvo orientada a metodologías de ingeniería de requisitos para el desarrollo de ambientes ubicuos o pervasivos, pero dados los hallazgos, se tuvo que extender a aproximaciones de dominio general que incluyeran formalismos y modelos de soporte a la fase de definición de sistemas. La clasificación de propuestas encontradas se resume en la figura 1. Cabe aclarar que para el alcance de este artículo solo se hará mención de algunas conclusiones importantes sin entrar en el detalle de cada propuesta encontrada.

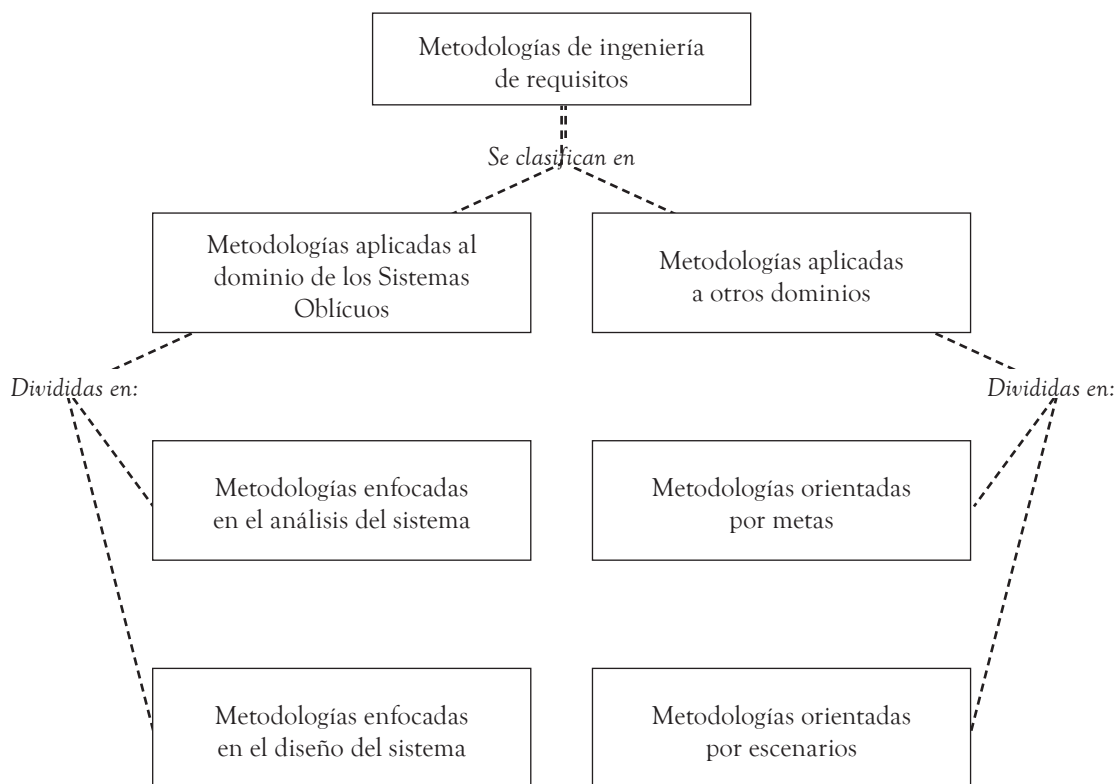


Figura 1. Clasificación de propuestas abordadas en la revisión de la literatura.

Fuente: elaboración propia.

En la rama de metodologías aplicadas al dominio de ambientes ubicuos vale la pena mencionar las siguientes: la aproximación de Tandler [7]: ofrece una arquitectura de *software* específica para ambientes de computación ubicua, y la valida mediante un caso de estudio denominado BEACH (Ambiente Básico para Colaboración Activa con Hipermedia). En esta arquitectura se ofrecen nuevas formas de interacción humano-computador, diferentes al uso de dispositivos como el mouse o el teclado. Antes de plantear la arquitectura, el autor propone 5 categorías de requisitos con sus respectivas subcategorías. También se incorpora una propuesta de modelo conceptual.

La propuesta de Giner y Torres [8]: Los autores proponen una clasificación de requisitos y su posterior representación basada en modelos para sistemas ubicuos, además mencionan aspectos clave para caracterizar este tipo de ambientes, tales como identificación de los elementos que participen en el sistema, heterogeneidad de interacción, Interoperabilidad entre sistemas. Posteriormente los autores proponen un conjunto de modelos que facilitan la representación de un ambiente ubicuo, y se complementan entre sí aportando información relevante de sistema.

El método propuesto por Muñoz y otros [11] incluye un formalismo para especificación de requisitos funcionales de sistemas pervasivos o ubicuos, que posteriormente son mapeados a modelos PervML, los cuales permiten: a) generación rápida de prototipos para validar los requisitos; b) definición de transformaciones que proveen mecanismos de trazabilidad para llevar los requisitos a la implementación, y viceversa. Se adapta la técnica CTT (ConcurTaskTree) para representar y describir información relevante de los sistemas pervasivos (como ubicación física, adquisición de datos del ambiente, entre otras), y con dicha información los autores definen la transformación desde el modelo de requisitos basado en tareas hasta PervML, que es un lenguaje de dominio específico para la especificación de sistemas pervasivos independiente de la tecnología.

La aproximación de Cetina [12] sigue el enfoque de desarrollo de *software* dirigido por modelos mediante el cual es posible la generación automática de código a partir de modelos para sistemas pervasivos. El autor se concentra en definir el lenguaje de modelado al que da soporte la herramienta para especificar estos sistemas (PervML) y un *framework* de implementación que permite la ejecución del código generado.

Kranz y otros [13] proponen una clasificación para sistemas de presencia ubicua de acuerdo con 5 criterios de clasificación relacionados con la forma de adquirir información del ambiente, las funcionalidades de entrada y salida, el medio de comunicación soportado, la extensión de la presencia, el modo de comunicación. Posterior a este aporte, proponen un conjunto de guías para el diseño de sistemas de presencia ubicua.

Según el estudio realizado, la mayoría de metodologías halladas poseen una fuerte orientación a la fase de diseño y un énfasis más débil hacia la fase de análisis que se ocupa de hacer una conceptualización del dominio de aplicación, y el posterior estudio de los requisitos que debe cumplir el ambiente ubicuo. Otro aspecto a resaltar de las metodologías encontradas es el hecho de que ofrecen pautas para tratar los requisitos luego de que han sido elicitados, pero no proporcionan herramientas para su captura, y esto también se debe a su fuerte orientación a la fase de diseño. Aunque existen 3 metodologías que proponen modelos de requisitos y algunas de ellas avanzan hasta la generación del modelo conceptual, se encuentra una desconexión entre los requisitos que se capturan y la forma en que se reflejan en el modelo conceptual. Otra debilidad detectada en los aportes recopilados es la ausencia de formalismos para la definición del sistema, previa a la captura de requisitos. No se halló evidencia de modelos que faciliten a los analistas y diseñadores de ambientes ubicuos incorporar un vocabulario común para este dominio en el cual se identifiquen los agentes participantes, sus interacciones típicas, lo mismo

que servicios y funcionalidades propias de este tipo de sistemas.

Las carencias detectadas marcaron la necesidad de explorar metodologías de IR aplicadas en otros dominios, buscando una que pueda ser transformada y adaptada al dominio de los sistemas ubicuos. Las aproximaciones encontradas son clasificadas de diversas formas, pero, un número apreciable de ellas están orientadas por escenarios o por metas, tal como se mostró en la figura 1 [14].

Las metodologías orientadas por escenarios proporcionan una descripción parcial del comportamiento de un sistema en una situación particular, y permiten representar con ejemplos concretos lo que el nuevo sistema hará. En las etapas tempranas de la ingeniería de requisitos, los escenarios son usados para soportar la definición de requisitos a alto nivel [15, 16], facilitando colaboración y entendimiento entre el equipo de desarrollo, clientes, usuarios y demás interesados. Los escenarios, además, facilitan la recopilación y representación de información en una forma entendible para los interesados.

Haumer [17] propone una metodología de este tipo al capturar los requisitos construyendo ejemplos del mundo real, y para ello hace uso de vídeos, entrevistas, esquemas, logrando un proceso de abstracción que conduce a la definición de modelos conceptuales más transparentes y trazables.

La metodología ScenIC [18] plantea un esquema de conocimiento relacionado con escenarios compuesto por metas, objetivos, tareas, obstáculos y acciones llevadas a cabo por actores. El método expresa los escenarios considerando si las metas pueden ser alcanzadas con las tareas definidas en el sistema, si los actores pueden realizar dichas tareas y cómo los obstáculos pueden impedir su normal ejecución por parte de los actores. En resumen, esta propuesta hace un análisis de medios y fines para garantizar que se cumplirán los requisitos.

En SCRAM (método de análisis de requisitos basado en escenarios) [19], los escenarios son combinados con prototipos tempranos para capturar

requisitos y elaborar un diseño preliminar, asegurando una comunicación activa entre usuarios y diseñadores lo que permite una mejor validación de los requisitos. El método tiene cuatro fases: captura de requisitos iniciales y familiarización con el dominio; visión inicial del diseño a partir de los requisitos; exploración de requisitos; validación de requisitos y prototipado. En el método no se propone ningún modelo de requisitos.

De otro lado, las metodologías basadas en metas proveen mecanismos para establecer la relación entre la funcionalidad esperada de un sistema y los procesos de negocio a los que éste dará soporte, ayudando a los agentes organizacionales en la realización de sus tareas.

Existen diversas propuestas metodológicas de este tipo. Entre ellas destaca poderosamente la notación i^* propuesta por Eric Yu en la primera mitad de la década de los 90 [14], que permite expresar de forma clara y sencilla las metas de los actores que aparecen en los modelos y las dependencias entre ellos. El uso de i^* incorpora un riesgo que se descubre pronto, pues no existe una definición única del lenguaje, lo cual genera cierta libertad y ambigüedad.

GBRAM (Método de Análisis de Requisitos Basado en Metas) [20] propone una serie de actividades a seguir para la obtención de un documento de requisitos a partir de metas de la organización. Presenta un proceso en el que se identifican metas organizacionales a partir de diversas fuentes (entrevistas, diagramas de flujo de trabajo, entre otros). Las metas se clasifican en metas de mantenimiento y metas de logro, y posteriormente se materializan en acciones del sistema.

La metodología KAOS [21, 22] permite construir modelos de requisitos a partir de las metas organizacionales. Esta aproximación está soportada por un marco formal basado en lógica temporal y en técnicas de refinamiento de Inteligencia Artificial, que define cada término (metas, acciones, estados) de forma consistente y rigurosa. La principal contribución de KAOS es la demostración de

que los requisitos se corresponden con las metas definidas para el sistema.

La propuesta de Ericsson [23] orientada por metas, postula que el modelado de negocio es una actividad de aprendizaje que ayuda al desarrollo del sistema. Esto implica que la primera actividad sería el modelado de la porción del negocio soportada por el sistema, lo cual ocasiona que cada nuevo desarrollo conlleve un nuevo modelado de parte de la organización.

La metodología ABC-Besoins [4] también está orientada por metas, y además tiene involucrado el concepto de agente. Si los requisitos se capturan pensando en los agentes que intervienen se obtendrá un mayor entendimiento del problema a resolver, porque aún sin que el sistema exista, los agentes deben interactuar para realizar determinadas actividades, y al pensar en la automatización de esas actividades se encontrará lo que el sistema debe hacer. Por su modelo para representar y utilizar el conocimiento del dominio, ABC-Besoins facilita la fase de captura de los requisitos. Se ofrecen para tal efecto dos formalismos de representación denominados modelo de contexto de utilización y estructura de objetivos y servicios del dominio.

Como último ejemplo de metodología orientada por metas, se encuentra B-SCP [24], un marco de ingeniería de requisitos basado en la conexión e integración de los conceptos de estrategia, contexto y proceso para apoyar la captura de requisitos organizacionales y su validación contra una estrategia de negocio. Este enfoque tiene implícito el concepto de meta.

1.3 Conclusiones de la revisión de literatura

Las metodologías orientadas al dominio de ambientes ubicuos presentan vacíos que son cubiertos por aproximaciones aplicadas actualmente en otros dominios y, particularmente, las orientadas por metas, que ofrecen propuestas claras para modelar el conocimiento de un dominio de aplicación determinado aportando instrumentos de tratamiento de información en la fase de definición del sistema.

2. SOLUCIÓN PROPUESTA

Teniendo en cuenta la revisión efectuada, y orientada no solo a metodologías de IR aplicadas en el dominio de los sistemas ubicuos, sino, a metodologías usadas en otros dominios, se propone intervenir la metodología ABC-Besoins [4], orientada por metas y agentes, que proporciona formalismos para representar el conocimiento básico de un dominio, información base para posteriormente proceder a la educación de requisitos y su posterior análisis, proporcionando continuidad en el proceso de IR, desde la elicitación de requisitos hasta la generación de un modelo conceptual y posterior generación de un modelo de diseño.

Los modelos de contexto y de dominio propuestos en [4] no fueron concebidos para el ámbito de ambientes ubicuos, por lo tanto, se deben adaptar para que reflejen un conocimiento profundo de este dominio de aplicación, incorporando el estudio de agentes y sus interacciones en orden a lograr sus metas.

A continuación se presentan los resultados de la adaptación.

2.1 Modelo de contexto de utilización para ambientes ubicuos

Se construye pensando en los agentes que interactúan y necesitan hacer operaciones sin contar todavía con la existencia de un sistema ubicuo, pero con un objetivo claro de poder comunicarse en cualquier tiempo, lugar o circunstancia con todo tipo de medios y de posibilidades de acceso. Los agentes requieren *comunicarse* sin preocuparse por asuntos como la distancia, el medio de acceso, la hora, y hacer manipulación de dispositivos acortando el concepto de distancia, por ejemplo, si se piensa en una mujer que es ama de casa y también trabaja, ella desearía poder apagar o encender la estufa ubicada en la casa de una forma remota desde su oficina. Si se piensa en la manipulación de dispositivos sin hacer uso de servicios ubicuos, los agentes tienen limitaciones

para controlarlos a distancia, pero en caso extremo lo pueden hacer de forma local. Los agentes emisor/receptor y receptor/emisor, además, requieren hacer intercambio de mensajes para conocer sus necesidades. En la figura 2 se muestra el modelo adaptado.

La lectura del diagrama se hará bajo dos escenarios: suponiendo la ausencia de un sistema ubicuo, es decir, indicando la forma en que los agentes emisor/receptor y receptor/emisor logran su objetivo de comunicarse por métodos tradicionales como contactar servicios de telefonía, mensajería instantánea, entre otros, y un segundo escenario en el cual se tiene incorporado el uso de un sistema ubicuo que facilite la comunicación de

los agentes interesados. En este último escenario entran nuevos agentes como las organizaciones que prestan servicios de ubicuidad.

Para describir el primer escenario se cuenta con las siguientes interacciones:

El agente emisor/receptor o receptor/emisor requiere hacer la configuración de los dispositivos que interviene o solicitar un cambio de estado en el dispositivo (por ejemplo, de ocupado a disponible, o de apagado a encendido). Estas operaciones deben hacerse de manera local si no se cuenta con servicios ubicuos.

- Los dispositivos manipulados, de acuerdo con la solicitud que reciben de otros agentes, envían mensajes de información que permiten

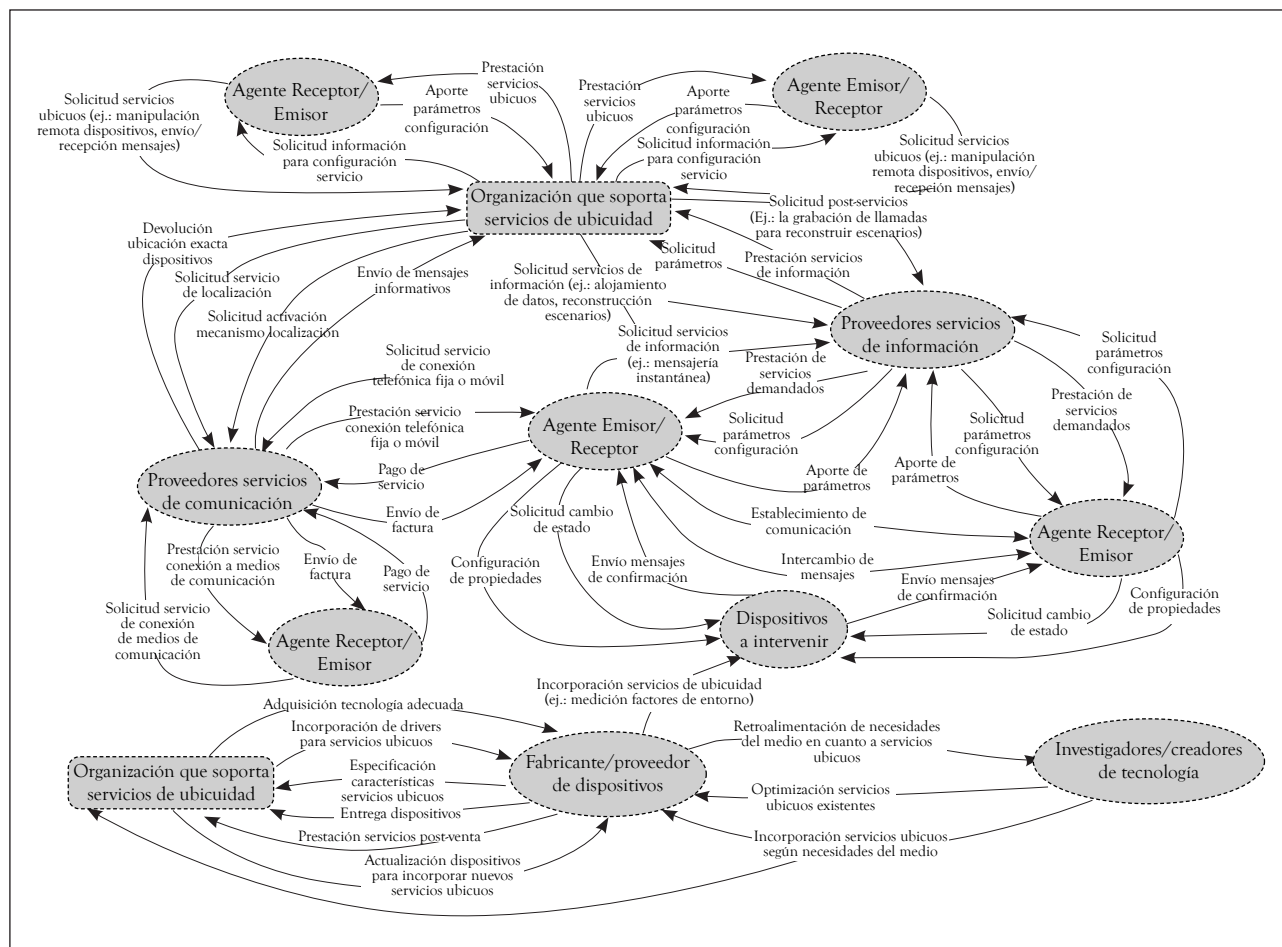


Figura 2. Modelo de contexto de utilización para ambientes ubicuos.

Fuente: elaboración propia.

verificar si la operación solicitada se llevó a cabo o cuáles fueron los inconvenientes que impidieron el cumplimiento de la petición por parte de otro agente.

- Los agentes emisor/receptor y receptor/emisor buscando medios de comunicación contactan a proveedores de servicios de comunicación para solicitar servicios de conexión de todo tipo (por ejemplo, telefónica móvil o fija).
- En respuesta a la solicitud hecha por los agentes emisor/receptor y receptor/emisor, los proveedores de servicios de comunicación prestan el servicio solicitado, y posterior a esto envían detalles de cobro, que serán respondidos con el pago.
- En la misma búsqueda de formas de comunicación entre agentes emisor/receptor y receptor/emisor, se hace la solicitud de servicios tales como mensajería instantánea y correo electrónico, que son aportados por los proveedores de servicios de información, preguntando algunos parámetros de configuración como el nombre del correo electrónico, la contraseña, posteriormente aportados por los agentes como condición para acceder al servicio.

Bajo el segundo escenario se tienen las siguientes interacciones, aclarando que el contexto de utilización de la solución incorpora el uso de servicios ubicuos:

- Los agentes emisor/receptor y receptor/emisor buscando servicios ubicuos (manipulación remota de dispositivos, envío y recepción de mensajes, modificación remota del estado del sistema, configuración de parámetros) los solicitan a las empresas especializadas en este tipo de servicios, y en retorno dichas empresas solicitan algunos parámetros requeridos para la prestación del servicio; por ejemplo, para prestar el servicio de manipulación remota de dispositivos es necesario conocer qué tipo de dispositivo desea usar el agente, o, para prestar el servicio de envío y recepción de mensajes se aportan datos como el nombre de la cuenta del remitente, del destinatario, la contraseña de acceso, el mensaje a enviar, etc.
- Las organizaciones que soportan servicios de ubicuidad solicitan a los proveedores de servicios de comunicación, servicios tales como el servicio de localización, necesario para poder ofrecerles a los agentes receptor/emisor y emisor/receptor la posibilidad de hacer manipulación remota de dispositivos, y los proveedores de este servicio entregan como respuesta la localización de cada dispositivo vinculado al sistema.
- Con miras a satisfacer las necesidades de agentes receptor/emisor y emisor/receptor, las organizaciones que prestan servicios ubicuos requieren alojar toda la información pertinente de sus clientes y solicitan el servicio a los proveedores de servicios de información, los cuales piden algunos parámetros tales como la capacidad deseada, y proceden a prestar el servicio.
- Los proveedores de servicios de comunicación informan continuamente a las organizaciones que requieren sus servicios sobre cualquier cambio en el portafolio o en la configuración actual.
- Los dispositivos, para acomodarse a las nuevas necesidades de los agentes emisor/receptor y receptor/emisor, deberán *incorporar nuevos servicios* comunicación e información, y para esto es necesario que los fabricantes de dispositivos y las organizaciones que soportan servicios de ubicuidad establezcan convenios de *intercambio de conocimiento* para dotar a estos dispositivos de nuevas funcionalidades que faciliten entre otras la comunicación remota. La interacción planteada supone comunicación entre los fabricantes/proveedores de información y los proveedores de servicios de comunicación e información, buscando incorporar los nuevos servicios demandados por las organizaciones que soportan servicios ubicuos.
- Posterior a la inclusión de nuevos servicios en los drivers que controlan los dispositivos, los

fabricantes/proveedores de dispositivos deben ofrecer servicio post-venta buscando identificar aspectos a mejorar en los servicios provistos para soportar ubicuidad.

- Y por último, siempre se deberá generar investigación para conocer la forma de optimizar los servicios ubicuos, y de esto se encargan los investigadores y creadores de tecnología con la respectiva retroalimentación a los fabricantes de dispositivos y a las organizaciones que soportan servicios de ubicuidad.

2.2 Estructura de servicios de un sistema ubicuo

Posterior al modelamiento de contexto de utilización es posible establecer los servicios que debe ofrecer un sistema ubicuo, resumidos en la figura 6. Se identificaron seis grandes categorías:

administración de servicios, suministro de información, localización, gestión de intervención de agentes, intervención del entorno y gestión de infraestructura. A continuación se desglosan.

1. Administración de servicios: este servicio facilita labores de configuración y administración que deban hacer aquellos usuarios con perfiles de administrador. Por ejemplo, el chequeo de dispositivos activos e inactivos es una responsabilidad del administrador, lo mismo que la solución de problemas posterior al informe de errores que entregue el sistema.
2. Gestión de infraestructura: mediante este servicio los administradores tienen posibilidad de manipular toda la infraestructura que compone un sistema ubicuo, desde sus dispositivos, hasta sus formas de conexión, sensores, actuadores, sistemas de localización, entre otros.

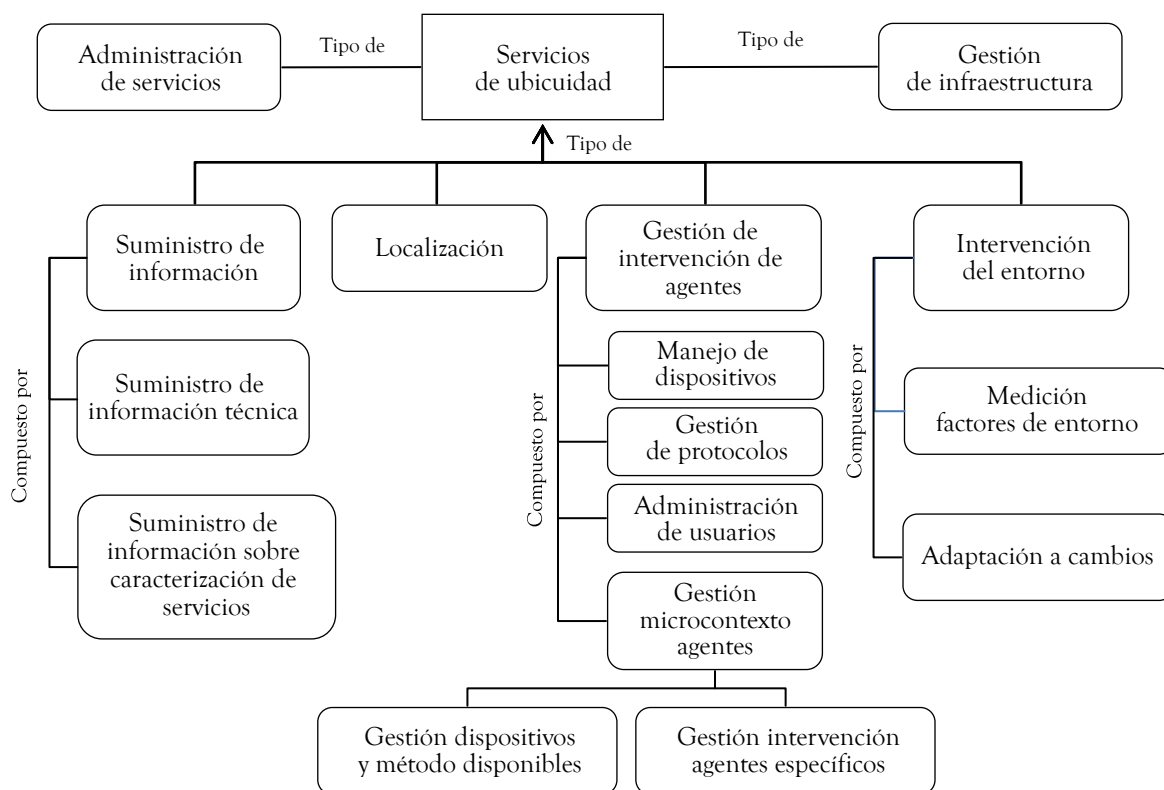


Figura 3. Estructura de servicios y objetos del dominio para ambientes ubicuos.

Fuente: elaboración propia.

3. Suministro de información: mediante este servicio, tanto los administradores de un sistema ubicuo como los usuarios acceden a conocimiento que les facilita interactuar con el sistema. En el caso de los administradores se desplegará información técnica para configuración de servicios, y para los usuarios, será mostrada información para guiar el uso de los diferentes servicios, lo mismo que una breve introducción a cada uno para facilitar la toma de decisiones de posibles compradores del sistema.
4. Localización: cualquier sistema de este tipo debe facilitar la identificación del lugar en donde se encuentra cada uno de los dispositivos asociados.
5. Gestión de intervención de agentes: un sistema ubicuo debe facilitar servicios para manejo de dispositivos de diferentes tipos, lo cual incluye manejo de interfaces de entrada y salida, además de gestión de protocolos que permitan la conexión de los dispositivos, y, a su vez, administración de usuarios para determinar, de acuerdo a su perfil, las funcionalidades que tendrán disponibles. En el servicio denominado “gestión microcontexto agentes” están ubicados aquellas particularidades que hacen parte del mundo propio del agente, incluyendo las funcionalidades disponibles de acuerdo al perfil asignado, lo mismo que otras condiciones excepcionales que dicho agente quiera incorporar en su interacción con el sistema ubicuo. Por ejemplo, un agente que tiene su hogar domótico con operadores con un operador de telefonía como UNE, si desea, ante condiciones diferentes como su estancia en un pueblo a donde solo llega señal de celular, debe poder conmutar de operador, independiente de que el elegido no esté predeterminado para los demás usuarios.
6. Intervención del entorno: gracias a su disposición e infraestructura, un sistema ubicuo estará en capacidad de censar cambios en el

entorno para desplegar servicios de acuerdo a una situación particular, además de capturar preferencias y parámetros comunes asociados a cada usuario, con el fin de ofrecer retroalimentación y disminuir la cantidad de información solicitada en las siguientes sesiones que se inicien. La intervención del entorno también permitirá que el sistema cambie y se adapte de acuerdo a sus condiciones de funcionamiento.

Los primeros niveles de esta estructura tienen relaciones “tipo de”, esto es, relaciones de generalización-especialización. Los niveles intermedios aceptan cualquier tipo de relación entre elementos, generalmente relaciones de composición.

Teniendo identificados los agentes que intervienen en la construcción de un sistema ubicuo, las interacciones entre ellos y los servicios típicos que debe proveer un sistema de este tipo, se facilita la determinación de los requisitos, tema de otro artículo, en el cual se define el modelo de requisitos. Los modelos presentados formalizan el paso de la fase de definición a la fase de análisis de los sistemas ubicuos y marcan el inicio de esta última fase.

3. DISCUSIÓN Y ANÁLISIS DE RESULTADOS

La revisión hecha como soporte a la investigación fue dirigida en dos sentidos: exploración de metodologías aplicadas al dominio de sistemas ubicuos y análisis de metodologías que son usadas en otros dominios. Con respecto a las primeras, se encontró que poseen una fuerte orientación a la fase de diseño y un énfasis más débil hacia la fase de análisis, y ofrecen pautas para tratar los requisitos solo luego de su educación, sin proporcionar herramientas para la obtención de los requisitos, la representación del conocimiento del contexto y el dominio, como tampoco métodos para hacer la transición entre las fases de definición y análisis [7, 8, 11-13].

Por su parte, las propuestas orientadas al tratamiento de sistemas en otros dominios proveen

métodos más completos para abordar la fase de análisis y conceptualización de las aplicaciones a construir, pero no fueron concebidas incluyendo las características particulares de los sistemas ubicuos [15, 16, 18, 19]. Por esta razón se seleccionó una metodología ubicada en esta línea, con fortaleza en la definición del sistema, para adaptarla a las particularidades de la ubicuidad.

4. CONCLUSIONES Y TRABAJOS FUTUROS

La poca comprensión del dominio del sistema es una de las principales causas de fracaso de un proyecto. Para tener una comprensión profunda sobre un dominio, es necesario entender los intereses, prioridades de los agentes, además de conocer los conceptos relacionados con el dominio.

Los modelos de contexto y de dominio presentados en este artículo aportan al analista elementos para entender el dominio de los sistemas ubicuos, mostrando su lógica de funcionamiento en el nivel macro incluyendo a los participantes (agentes) desde su construcción hasta su etapa de uso, a la vez que ofrece una herramienta para la captura de requisitos. Por otro lado, el modelo de **estructura de servicios y objetos del dominio** presenta un listado predeterminado de los servicios que debe ofrecer cualquier sistema ubicuo.

Como trabajo futuro se propone: a) Refinar el modelo de requisitos y el modelo conceptual, productos en curso que hacen parte de esta investigación. b) Construir un software para apoyar y facilitar el uso de los modelos enunciados.

REFERENCIAS

- [1] C. Gamboa *et al.*, "Descubriendo los retos de las Ciudades Ubicuas: Experiencias en Andicom 2008," *Revista Colombiana de Telecomunicaciones*, vol. 16, no. 51, pp. 15-21, 2009.
- [2] M. Weiser, "The future of Ubiquitous Computing on Campus," *Scientific American*, vol. 265, no. 3, pp. 94-104, 1991.
- [3] A. Lamsweerde, "Requirements Engineering in the Year 00: A Research Perspective," presentado a International Conference on software Engineering, Limerick, 2000.
- [4] G. Urrego, "ABC-Besoins: Une approche d'ingénierie de besoins fonctionnels et non-fonctionnels centrée sur les Agents, les Buts, et les Contextes," Tesis doctoral, Mathématiques Informatique et applications, Université Paris I, Panthéon Sorbonne, Sorbona, 2005.
- [5] E. Yu, "Towards modelling and reasoning support for early-phase requirements engineering," presentado a Third IEEE International Symposium on Requirements Engineering Annapolis, MD, USA, 1997, pp. 226 - 235.
- [6] H. Pham *et al.*, "Applying Model-Driven Development to Pervasive System Engineering," presentado a 1st International Workshop on Software Engineering for Pervasive Computing Applications, Systems, and Environments Minneapolis, Minnesota 2007.
- [7] P. Tandler, "Software Infrastructure for Ubiquitous Computing Environments: Supporting Synchronous Collaboration with Heterogeneous Devices," presentado a Proceedings of UbiComp 2001: Ubiquitous Computing, Atlanta, Georgia, 2001, pp. 96-115.
- [8] P. Giner, y V. Torres, "Una Propuesta Basada en Modelos para la Construcción de Sistemas Ubicuos que den Soporte a Procesos de Negocio," presentado a IDEAS: 10° Workshop iberoamericano de Ingeniería de Requisitos y ambientes de software, Venezuela, 2007.
- [9] J. Muñoz *et al.*, "Requirements Engineering for Pervasive Systems: A Transformational Approach," presentado a 14th IEEE International Requirements Engineering Conference (RE'06), Minneapolis/ ST Paul, Minnesota, 2006.
- [10] M. Kranz *et al.*, "Ubiquitous presence systems," presentado a Symposium on Applied Computing, Dijon, France, 2006, pp. 1902 - 1909
- [11] G. Hadad *et al.*, "Construcción de Escenarios a partir del Léxico Extendido del Lenguaje," presentado a Jornadas Argentinas de Informática e Investigación Operativa, Buenos Aires - Argentina, 1997.
- [12] K. Benner *et al.*, "Utilizing scenarios in the software development process," *Information System Development*

- Process- Elsevier Science Publisher*, pp. 117-134, 1993.
- [13] C. Potts, "ScenIC: A Strategy for Inquiry-Driven Requirements Determination," presentado a 4th IEEE International Symposium on Requirements Engineering, 1999, pp. 58-65.
- [14] A. Sutcliffe, "Scenario-Based Requirements Engineering," presentado a 11th IEEE International Requirements Engineering Conference (RE'03) Kyoto, 2003, pp. 320-331.
- [15] I. Kumaran, y S. Kumaran, *Jini Technology: An Overview*, New Jersey: Prentice Hall PTR Upper Saddle River, 2001.
- [16] L. González, "Metodología de Ingeniería de Requisitos para el análisis de sistemas embebidos," Tesis de maestría, Ingeniería de sistemas, Universidad de Antioquia, Medellín, 2008.
- [17] C. Cetina, "Diseño y Desarrollo de una Herramienta CASE para la Generación Automática de Código para Sistemas Pervasivos," Tesis doctoral, Centro de Investigación ProS, Universidad Politécnica de Valencia, Valencia, 2006.
- [18] P. Haumer *et al.*, "Requirements Elicitation and Validation with real world scenes," *IEEE Transactions on Software Engineering*, vol. 24, no. 12, pp. 1036-1054, 1998.
- [19] A. Anton, "Goal Identification and Refinement in the Specification of Software-Based Information System," Tesis doctoral, Computational Science & Engineering, Georgia Institute of Technology, Atlanta, 1997.
- [20] A. Dardenne *et al.*, "Goal Directed Requirements Acquisition," *Science of Computer Programming*, vol. 20, pp. 3-50, 1993.
- [21] H. Ericsson, y M. Penker, *Business Modeling with UML: Business Patterns at Work*, New York: Wiley Computer Publishing, 2000,
- [22] S. Bleistein *et al.*, "B-SCP: A requirements analysis framework for validating strategic alignment of organizational IT based on strategy, context, and process," *Information and Software Technology*, vol. 48, pp. 846-868, 2006.