

Ingeniería de Requerimientos para Sistemas Sensibles al Contexto, un estudio comparativo

A.C.Viviana Castelli¹ - Mg. Pablo Thomas²- Mg. Rodolfo Bertone³
Facultad de Informática – UNLP
III-LIDI⁴

Abstract

The computing community has brought about the creation of Software Engineering after years of failure and despite the advances already made; it still requires more time –even decades- to become a mature engineering field necessary among the computing community. This also involves the fact that Requirement Engineering has not reached a maturity state either.

On the other hand, a new paradigm has emerged together with the technological development: Context-Aware Software, that is to say, the generation of applications able to react automatically to context changes. Then, it is possible to imagine a methodology for the development of these applications and, consequently, an “adaptation” on the part of the Requirement Engineering to be used in these new applications, even though there is still a long way to go. This paper shows a comparative research on several Requirement Engineering approaches for Context-Aware Applications.

Keywords: Requirement Engineering, Context-Aware Software, Pervasive Systems, Ubiquitous Computing.

Resumen

La comunidad informática propició el advenimiento de la Ingeniería de Software luego de años de fracasos, y aún a pesar de los progresos, le falta tiempo -quizás décadas- para convertirse en una disciplina de ingeniería madura que la sociedad de la información requiere. Esto implica que la Ingeniería de Requerimientos tampoco ha llegado a un estado de madurez plena.

Por otra parte, de la mano del avance tecnológico, ha surgido un nuevo paradigma: la Programación Sensible al Contexto, es decir, la generación de aplicaciones capaces de reaccionar a los cambios en el contexto de manera automática. Es posible, entonces, pensar en una metodología de construcción de estas aplicaciones, y por ende la “adecuación” de la Ingeniería de Requerimientos tradicional de modo tal de ser utilizada en este nuevo tipo de aplicaciones, aunque aun quede mucho camino por recorrer.

Este trabajo, por lo tanto, presenta un estudio comparativo de diversos enfoques de Ingeniería de Requerimientos para Aplicaciones Sensibles al Contexto.

Palabras Claves: Ingeniería de Requerimientos, Aplicaciones Sensibles al Contexto, Sistemas Pervasivos, Computación Ubicua.

¹ {castelli_viviana@hotmail.com} Facultad de Informática. Universidad Nacional de La Plata

² {pthomas@lidi.info.unlp.edu.ar} III-LIDI. Prof. Adjunto Dedicación Exclusiva. Facultad de Informática. Universidad Nacional de La Plata

³ {pbertone@lidi.info.unlp.edu.ar} III-LIDI. Prof. Titular Dedicación Exclusiva. Facultad de Informática Universidad Nacional de La Plata

⁴ Instituto de Investigación en Informática. Facultad de Informatica. Universidad Nacional de La Plata. 50 y 120. La Plata. Buenos Aires Argentina.

1. Introducción

El desarrollo de software es un proceso que comienza con una fase de aprendizaje, donde el propósito del Ingeniero de Software es comprender, entre otros, la organización, el problema, el cliente, el usuario y sus necesidades. Por otra parte, otro aspecto clave para cumplimentar con éxito el desarrollo de un sistema de software es lograr elucidar todo el contexto del problema, y así poder generar un planteo completo de objetivos que permita obtener una solución correcta y eficiente.

Las necesidades del cliente se plantean como un conjunto de **requerimientos** sobre el sistema a desarrollar. Según IEEE [8], un requerimiento es “una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo” o “una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal”.

Es muy frecuente observar que un gran número de los proyectos de software fracasan por no realizar una adecuada definición, especificación, y gestión de los requerimientos. Dentro de algunos problemas que se pueden encontrar en la administración de requerimientos, refieren entre otros factores a la falta de participación del usuario, requerimientos incompletos y cambios mal gestionados. De aquí la importancia en el proceso de **Ingeniería de Requerimientos** para obtener una especificación de requerimientos de un Sistemas de Software.

La **Ingeniería de Requerimientos** es una disciplina que ha evolucionado desde sus comienzos. Asimismo, cabe destacar que el desarrollo de Sistemas de Software ha avanzado de modo considerable como así también el hardware subyacente sobre el cual se ejecuta, a punto tal que las computadoras han dejado de ser máquinas extremadamente grandes para pasar a ser pequeños dispositivos del tamaño de una mano que conviven diariamente con las personas, o más aún, en algunos casos llegan a ser imperceptibles en el entorno.

Hace poco tiempo, y en relación a la evolución del hardware señalada, han surgido algunos conceptos nuevos en computación, tales como **computación ubicua, pervasiva y sensibilidad al contexto**.

A continuación se presenta un breve marco conceptual para la comprensión de los conceptos de computación ubicua, computación pervasiva y sistemas sensibles al contexto; como así también se plantean algunas cuestiones relacionadas al proceso de Ingeniería de Requerimientos en estos nuevos tipos de sistemas o paradigmas de computación (computación ubicua-pervasiva y sistemas sensibles al contexto), focalizándose particularmente sobre sistemas sensibles al contexto. Por último, se analiza el estado de la Ingeniería de Requerimientos en sistemas sensibles al contexto, a través del estudio comparativo de diferentes propuestas: [3], [9], [10], [6], [5].

2. Computación Ubicua, Computación Pervasiva y Sistemas Sensibles al Contexto

Se conoce como ‘padre’ de la computación ubicua a **Mark Weiser**, quien en 1988 formula lo que hoy se considera como una de las primeras nociones de computación ubicua:

La **Computación ubicua** tiene como objetivo mejorar el uso de la computadora de modo que muchas computadoras estén disponibles en todo el entorno físico, haciéndolas efectivamente invisibles para el usuario [7].

Computación ubicua se aproxima a lo opuesto a la realidad virtual. La realidad virtual pone a las personas dentro de un mundo generado por computadoras, mientras que las “fuerzas” de la computación ubicua llevan a la computadora a vivir aquí, en el mundo con la gente [7].

La idea subyacente de [7] está relacionada a la tendencia actual de embeber los dispositivos computacionales en el ambiente, haciéndolos desaparecer de la vista del usuario; de esta manera se utiliza la capacidad computacional del dispositivo de manera inconsciente; es decir, el usuario

interactúa con el dispositivo sin notar su existencia o presencia, y por lo tanto su atención no es distraída.

Es muy frecuente que los términos “*ubicuos computing*” (computación ubicua) y “*pervasive computing*” (computación pervasiva) sean utilizados como sinónimos. Algunos autores [13] consideran que ambos conceptos tienen un significado similar con alguna pequeña variante. Según esta visión, la computación pervasiva es una especialización del concepto de computación ubicua, y se refiere a computación ubicua como la utilización de dispositivos computacionales embebidos, brindando información a las aplicaciones en ámbitos relacionados con el comercio a través de la Web, entre otros; es decir computación ubicua utilizada con un fin particular.

2.1 Sistemas Sensibles al Contexto

En general las definiciones de **Sistemas Sensibles al Contexto** se pueden clasificar en dos grandes grupos. Aquellas que consideran la computación basada en contexto como el software que simplemente utiliza el contexto, y aquellas que la definen como la computación que no sólo utiliza el contexto sino que además es capaz de adaptarse a él [2].

Según [4] la **computación sensible al contexto** es un paradigma de computación en el cual las aplicaciones y los servicios pueden tomar ventajas de la información contextual tal como: información del usuario, de la ubicación del dispositivo, del estado del dispositivo, del tiempo del día, de la interacción usuario-servicio y sus actividades, entre otros.

¿Cuál es la relación entre computación ubicua/ pervasiva y los sistemas sensibles al contexto? La **computación sensible al contexto** consiste de dispositivos computacionales integrados en el medio ambiente ‘rodeando’ a las personas de modo tal, que proporcionan servicios aumentados con capacidades sensoriales: los dispositivos situados en un espacio físico pueden ver u oír (sentir) quién o qué se encuentra alrededor, qué actividad está realizando, la ubicación física de los objetos o personas del entorno y cuándo esta ocurriendo algo.

De algún modo los **sistemas sensibles al contexto** plantean una extensión de la visión presentada por [7]. Son dispositivos dispersos en el entorno del usuario que cooperan con algún software, brindando información del contexto en que se ejecuta la aplicación, permitiendo de este modo al sistema reaccionar en función de los cambios producidos en su entorno.

3. El Contexto: característica fundamental de un Sistema Sensible al Contexto

El rasgo más importante que caracteriza a una aplicación sensible al contexto es el modo en que la aplicación reacciona a los cambios en el contexto. Los desarrollos tradicionales reaccionan a los cambios en el contexto de una manera indirecta, es decir, alguna persona generalmente usuario del sistema, tiene que tomar la iniciativa de informar los cambios al sistema a través del ingreso de datos específicos y la aplicación reacciona en consecuencia. Por otra parte, las aplicaciones sensibles al contexto reaccionan automáticamente a los cambios en el contexto o ambiente donde se utilicen, mediante una respuesta acorde a la situación particular.

La Real Academia Española [17] considera como **contexto** al entorno físico o de situación, ya sea político, histórico, cultural o de cualquier otra índole, en el cual se considera un hecho.

En [2] se define **contexto** como cualquier información que se puede utilizar para caracterizar la situación de las entidades (persona, lugar u objeto) que se consideran pertinentes para la interacción entre un usuario y una aplicación, incluyendo al usuario y a la aplicación.

Otra visión sobre contexto se plantea en [1], considerado como el conjunto de los estados del medio ambiente y configuración del sistema, que determina el comportamiento de la aplicación o produce eventos en la aplicación que resultan interesantes para el usuario.

El contexto permite que la comunicación entre los seres humanos y los dispositivos informáticos sea mucho más eficiente. Si una aplicación logra adaptar su comportamiento al contexto actual, probablemente la satisfacción del usuario se verá incrementada.

Algunos autores [2], [15], [1], [11], [14] sugieren una extensión al concepto de contexto, proponiendo diferentes taxonomías de contexto.

En [2] y [15] se sugieren tres categorías de contexto.

- 1) **El contexto de la computación**, como la infraestructura de red, los dispositivos de entrada-salida, y los procesadores disponibles y demás.
- 2) **El contexto de usuario**, que incluye la ubicación del usuario, ubicación en relación a las personas, y la situación social del usuario.
- 3) **El contexto físico**, como la iluminación, el nivel de ruido, o la temperatura.

En [1] se agrega:

1. **Contexto del tiempo**, tales como la hora del día, semana, año, entre otros.

La consideración más amplia sobre contexto es la propuesta por [11]:

- El **contexto espacio-temporal**, en el cual se describen atributos como la hora, lugar, velocidad, dirección, y el ámbito social.
- El **contexto del entorno**, en el que se describen las entidades “cercanas” al usuario, tales como los servicios, la temperatura, el ruido, las personas, y las redes.
- El **contexto personal**, en el que se describe los estados fisiológico y mental del usuario.
- El **contexto de la tarea**, que describe de manera explícita objetivos, tareas y acciones del usuario.
- El **contexto social**, que describe los aspectos sociales de los usuarios, tales como información acerca de amigos y familiares, así como una función del usuario (por ej. “en Trabajo”).
- El **contexto de la información**, que describe la información global y personal que está disponible.

A partir de la necesidad de utilizar dispositivos para obtener información contextual, algunos autores [14] realizan una división de la información contextual según sea proveniente de muestras tomadas por medio del sentido o no.

El contexto de una aplicación se descompone en entidades atómicas llamadas aspectos de contexto [14]. Un aspecto de contexto registra una parte del contexto general y está ligado a uno o más sensores.

Los aspectos de contexto se pueden clasificar según el origen de los datos en **Sensados y Derivados**. Los aspectos **Sensados** toman datos directamente desde sensores. Los aspectos **Derivados** hacen una agregación de datos. Los aspectos **Derivados** toman datos procedentes de sensores o de otros aspectos derivados y producen un aspecto resultante mediante la ejecución de un proceso [14].

4. Ingeniería de Requerimientos tradicional e Ingeniería de Requerimientos para Sistemas Sensibles al Contexto

La **Ingeniería de Requerimientos** es el proceso sistemático de desarrollo de requerimientos a través de un proceso cooperativo e iterativo de analizar el problema, documentar las observaciones resultantes en una variedad de formatos de representación, y controlar la precisión de la comprensión adquirida. Consiste en el ciclo de obtención de conocimiento, la representación y la validación. El éxito del proceso de Ingeniería de Requerimientos con frecuencia depende de la capacidad para transformar declaraciones informales confusas e individuales de los requerimientos en una especificación formal comprendida y aceptada por todas las partes interesadas. [12]

Se plantea la ejecución de tres etapas para llevar a cabo el proceso de Ingeniería de Requerimientos: **Elicitación de Requerimientos** (obtención del conocimiento), **Especificación de Requerimientos** (representación) y **Validación de Requerimientos** [12]. Las tres etapas planteadas no son independientes entre si, por el contrario se nutren una de otra. Como se aprecia en la **Figura 1**, se trata de un proceso iterativo de ejecución de las tres etapas, tantas veces como sea necesario.

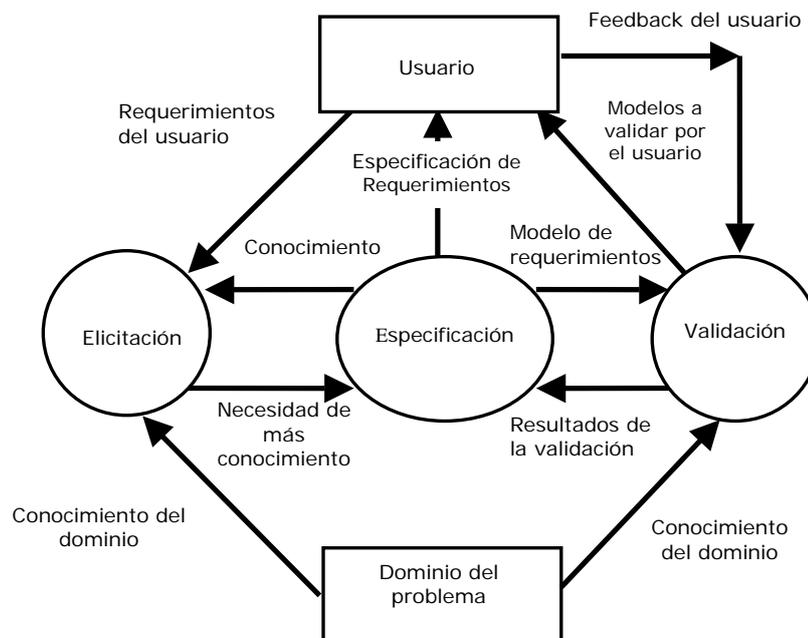


Figura 1 -Proceso de Ingeniería de Requerimientos [12]

Según [16], la meta del proceso de **Ingeniería de Requerimientos** es crear y mantener un documento de requerimientos del sistema. El proceso general comprende cuatro subprocesos de alto nivel. Los subprocesos tratan sobre la evaluación de la utilidad del sistema para el negocio (**estudio de viabilidad**); el descubrimiento de requerimientos (**obtención y análisis**), la transformación de estos requerimientos en formularios estándar (**especificación**), y la verificación de que los requerimientos realmente definen el sistema que quiere el cliente (**validación**).

Ambas propuestas [12] [16] consideran que la obtención, especificación y validación de las necesidades del usuario son etapas necesarias. No obstante, sería deseable una consideración más, comprender y controlar los cambios en los requerimientos, debido a que generalmente parte de los requerimientos son volátiles.

Para aplicar los conceptos de **Ingeniería de Requerimientos en Aplicaciones Sensibles al Contexto**, el enfoque tradicional debe ser mejorado / adaptado. Si bien en las aplicaciones tradicionales existe el problema de una especificación de requerimientos vaga, incompleta o confusa debido a que existen situaciones en las que poder especificar todo no es factible, si a eso se añade que la aplicación debe reaccionar de manera diferente en función de su entorno, obtener una especificación de los requerimientos adecuada se torna mas complicado aún.

5. Enfoques de Ingeniería de Requerimientos para Sistemas Sensibles al Contexto

5.1 Enfoque de Finkelstein y Savigni [3]

En este enfoque se presenta un framework de Ingeniería de Requerimientos para sistemas sensibles al contexto. En el framework se abordan los principales problemas de los sistemas sensibles al contexto, como el cambio de contexto y la evolución de las necesidades.

Los autores utilizan un concepto innovador en Ingeniería de Requerimientos, la **reflexión**. Un sistema **reflexivo** mantiene, en tiempo de ejecución, estructuras de datos que materializan algunos aspectos del propio sistema. Para sus propósitos, la **reflexión explícita** significa que, en tiempo de ejecución se mantiene una representación del comportamiento del sistema, luego los cambios en el comportamiento actual se reflejan en una descripción del meta nivel. La **reflexión** es la clave en este ámbito y la manipulación de los metadatos es esencial en el contexto altamente dinámico de los servicios, ya que los servicios deben ser capaces de adaptarse dinámicamente a los cambios de contexto y a la evolución de los requerimientos que sufren los sistemas sensibles al contexto.

Los puntos básicos del enfoque son las metas u objetivos y el entorno. Se transforman las metas en requerimientos y se representa la información del entorno dentro de un contexto. Luego se enfatiza la descripción del servicio, como puente entre el mundo y la máquina.

En la figura 2 se pueden observar los componentes del framework (en idioma origen para no perder expresividad, más algunas aclaraciones).

Una cuestión importante es cómo describir las descripciones de los meta niveles. Se sugiere el uso de XML, debido a que es un estándar ampliamente utilizado, no se requiere crear analizadores, existen APIS y algunos productos para tal fin; XSLT, para el manejo de archivos XML, y se podría determinar si el comportamiento del sistema en tiempo de ejecución todavía está alineado con los requerimientos, basándose en la especificación obtenida utilizando XML.

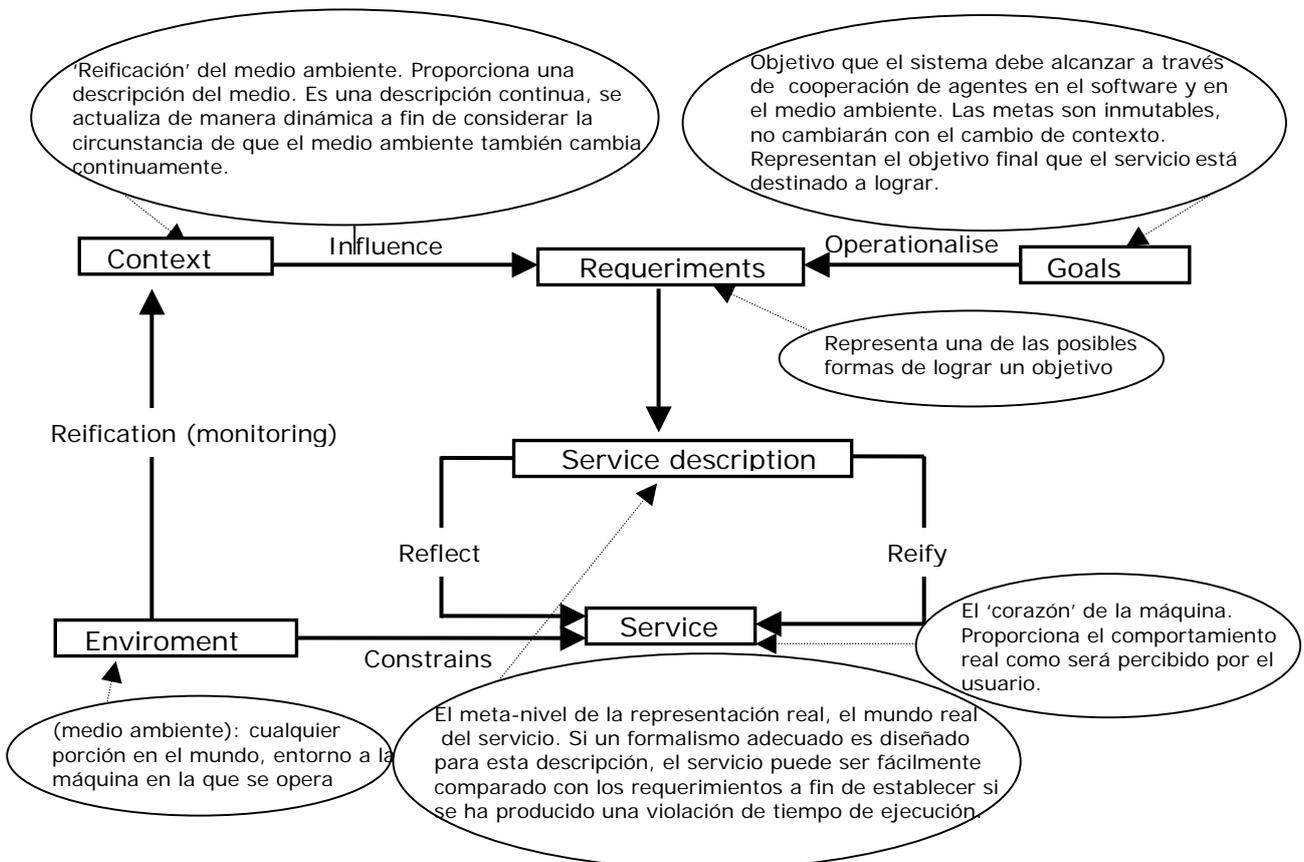


Figura 2 Vista general del framework [3]

5.2 Enfoque de Kammanahalli, Gopalan, Sridhar y Ramamritham [9]

En este caso, el enfoque de recuperación de información del contexto se basa en la **agregación de una secuencia previa de los contextos individuales**, y en **asociar la actividad actual de los usuarios a esos contextos** para determinar el mejor modo de entregar la información relevante sobre la actividad actual. Debido a la continuidad de las actividades del usuario en pos de lograr un determinado objetivo, la continuidad en contextos podría establecerse y explotarse para la determinación y recuperación de información más relevante. La agregación de contextos se logra haciendo uso de los patrones repetitivos inherentes a las actividades realizadas por el usuario.

Básicamente en función de algunos atributos mensurables, como por ejemplo el tipo de documento más seleccionado por el usuario, el apoyo de sistemas de recomendación y los contextos analizados (contextos previos del usuario o grupo); todo esto en adición al contexto actual, permitirán recomendar de manera más eficiente el/los documentos predilectos en el formato más conveniente para el usuario en cuestión.

La idea general y las contribuciones de este enfoque incluyen:

1. Definir el contexto a fin de incluir la información relativa a las actividades y el dispositivo utilizado por el/los usuario/s.
2. Identificar las preferencias del usuario en el dominio.
3. Identificar la atención inmediata del equipo (grupo de usuarios).
4. Recomendar la extensión estándar para recuperar la información pertinente al objetivo inmediato del equipo o para el contenido de interés de un usuario particular.
5. Presentar la información en el formato más relevante.

5.3 Enfoque de Kolos, Mazuryk, Poulisse y Van Eck [10]

Según este enfoque, se considera como punto de partida determinar las propiedades distintivas de los sistemas sensibles al contexto. Para ello se lleva a cabo una separación de las propiedades del sistema en función de su relación directa (o no) con el contexto, en **propiedades contextuales** y **propiedades no contextuales**. Son ejemplos de propiedades no contextuales la limitada atención del usuario sobre una aplicación por un largo periodo de tiempo, la necesidad de que la aplicación esté inmediatamente disponible para su uso, entre otros. Por otra parte, dentro de las propiedades contextuales se encuentran entre otras el entorno (completamente dinámico), cambios en los dispositivos de visualización de la aplicación, cuestiones relacionadas al ancho de banda.

Los autores utilizan la siguiente taxonomía para clasificar los diferentes contextos:

- El **contexto espacio-temporal**, en el cual se describen atributos como la hora, lugar, velocidad, dirección, y el ámbito social.
- El **contexto del entorno**, en el que se describen las entidades alrededor del usuario, tales como los servicios, la temperatura, el ruido, personas, y las redes.
- El **contexto personal**, en el que se describe los estados fisiológico y mental del usuario.
- El **contexto de la tarea**, que describe de manera explícita objetivos, tareas y acciones del usuario.
- El **contexto social**, que describe los aspectos sociales de los usuarios, tales como información acerca de amigos y familiares.
- El **contexto de la información**, que describe la información global y personal que está disponible.

El método de Ingeniería de Requerimientos propuesto por este enfoque consiste en observar las distintas partes interesadas del sistema en desarrollo, y tratar de determinar sus necesidades con respecto al sistema a partir de diversas fuentes y/o técnicas (entrevistas, diarios, pruebas de usuario, talleres, entre otros). Se utilizan las entrevistas semi-estructuradas para obtener más información

sobre los entornos de los usuarios, y la taxonomía de contexto enunciada previamente sirve de orientación en el desarrollo de estas entrevistas.

El siguiente paso es la construcción de un modelo más detallado y completo del medio ambiente.

Otra actividad utilizada son los talleres. El objetivo del taller es facilitar la creación de conceptos para los futuros servicios. Se divide a los participantes en grupos de brainstorming, posteriormente se seleccionan las dos mejores ideas y se procede a la discusión de estas ideas.

5.4 Enfoque de Hong- Chiu y Shen [6]

Este enfoque está basado en dos objetivos muy importantes para la interacción del usuario con el sistema: **usabilidad** y **experiencia del usuario**. Para cumplir con el **objetivo de usabilidad**, un sistema interactivo tiene que ofrecer servicios eficaces, y debe ser sencillo para que el usuario pueda aprender a utilizarlo y recordar cómo usarlo. Otra de las metas es la **experiencia del usuario** mediante su interacción con el sistema; con el fin de atraer a las personas a utilizar productos interactivos, éstos deben ser diseñados para que sean divertidos, agradables, placenteros, y estéticamente atractivos de usar.

Se utiliza el concepto de **contexto ampliado** a tres principales categorías de contexto: los **contextos de computación**, el **contexto de usuario**, y los **contextos físicos**.

1. **Contexto de la computación** se refiere a la configuración de hardware utilizado, tales como los procesadores disponibles para una tarea, los dispositivos de acceso para el usuario, la pantalla de entrada, y el ancho de banda.
2. **Contexto del usuario** representa a todos los factores humanos, tales como el perfil de usuario, calendarios, entre otros.
3. **Contexto físico** se refiere a aquello no relacionado a la computación, información proporcionada por el medio ambiente del mundo real como la localización, tiempo, la iluminación, los niveles de ruido, las condiciones de tráfico y la temperatura.

Las tres categorías son igualmente cruciales y en su conjunto determinan la interacción personalizada entre el usuario y la aplicación. Por lo tanto se propone el uso de este concepto de contexto ampliado, como base para la obtención de los requerimientos en el diseño de las aplicaciones ubicuas.

Con la adopción del concepto de contexto ampliado, el meta-modelo propuesto (figura 3) proporciona una base para que el sistema seleccione las características de sensibilidad al contexto en tiempo de ejecución, de acuerdo con el contexto actual de cada sesión de usuario, de modo que el sistema utilice los mismos criterios para proporcionar interacciones más apropiadas a los usuarios.

Se propone una serie de actividades (determinar grupos de usuarios, contextos típicos de los usuarios que componen los grupos, entre otros) explicando la metodología a seguir para obtener los requerimientos, y luego realizar una iteración sobre las actividades planteadas hasta obtener todas las características de los requerimientos de sensibilidad al contexto.

Los autores de este enfoque consideran que aún hay muchos desafíos para futuros estudios, tales como interfaz natural, el razonamiento de contexto, y las nuevas metodologías de evaluación. En particular, su investigación está centrada en la elicitación de requerimientos en aplicaciones sensibles al contexto, especialmente sobre cómo se compone la información de contexto y cómo los requerimientos pueden ser efectivamente elicitados con la consideración de posibles conflictos e implicaciones. Además sugieren el estudio de aplicaciones para colaboración ubicua y la computación basada en agentes, en función de este nuevo entorno informático.

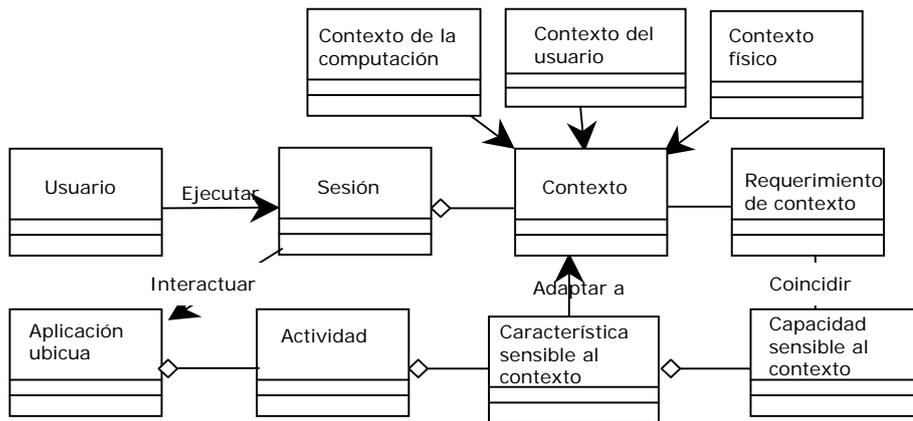


Figura 3 - Meta-modelo para obtención de requerimientos sensibles al contexto [6]

5.5 Enfoque de Godbole y Smari [5]

Se presenta esta visión con ayuda de un caso de estudio sobre la utilización de teléfonos celulares, que ilustra cómo la información de los usuarios eliciteda puede ser usada para desarrollar un modelo sensible al contexto, que proporcione la funcionalidad deseada mientras mantiene las preferencias del usuario y las características del escenario del sistema “en mente”.

Se propone la **adquisición directa de datos centrado en el usuario con el fin de crear reglas, modelos de aprendizaje u ontologías para la situación y la sensibilización al contexto**. Esta visión puede servir de complemento a las **técnicas de adquisición de datos basadas en sensores** y puede incluso sustituir los sensores en determinadas situaciones.

Los sensores en un sistema sensible al contexto proporcionan datos en bruto sobre sus usuarios, su comportamiento y características, y sus diversos entornos. Debido a que la perspectiva de solo usar sensores tiene algunas desventajas (no son buenos en escenarios grandes dado el número de muestras que necesita, su precisión, su nivel de intrusión, entre otras características), se propone utilizar una combinación de técnicas para elicitar la información de contexto.

La adquisición basada en la perspectiva humana y el aprendizaje, básicamente se centra en el usuario. Se intenta recolectar información sobre las necesidades del usuario a partir de la interacción entre el usuario y el analista, o el usuario y el sistema, a través de encuestas, entrevistas y preferencias del usuario. Si bien esta técnica representa mejor ciertas características del contexto que el solo uso de sensores, la idea de combinar técnicas sirve para mejorar la calidad de la información de contexto obtenida.

Observar los usuarios, realizar entrevistas y encuestas, es adecuado si el escenario es de pequeña o mediana escala. Para los escenarios más grandes, las herramientas automatizadas pueden desempeñar un papel importante.

Una vez eliciteda la información de contexto por medio de las dos técnicas propuestas, la información obtenida puede ser utilizada para generar normas, políticas y modelos de aprendizaje de contexto y situaciones, como así también para generar ontologías de dominio que describen las relaciones entre los usuarios, situaciones, ambientes y eventos. Las ontologías obtenidas pueden ser empleadas para proporcionar las funcionalidades de sensibilidad e inteligencia de los sistemas sensibles al contexto.

La idea de utilizar la adquisición de contexto basada en perspectiva humana, el aprendizaje y el conocimiento, puede beneficiar al rápido desarrollo de recursos sensibles al contexto para un gran número de usuarios, mientras se mantiene el tiempo de diseño y bajo costo.

5.6 Cuadro comparativo de los enfoques presentados

Enfoque	Finkelstein- Savigni [3]	Kammanahalli-Gopalan-Sridhar-Ramamritham [9]	Kolos-Mazuryk-Poulisse-Van Eck [10]	Hong- Chiu- Shen [6]	Godbole- Smari [5]
Aportes	<p>1-Mantener en tiempo de ejecución una representación de comportamiento del sistema</p> <p>2-Monitorear violaciones de los requerimientos en tiempo de ejecución</p> <p>3-Brindar información de la arquitectura y representar información de contexto, en tiempo de ejecución</p>	<p>1-Manejar contextos pasados del usuario para mejorar el modo de entregar documentos en el contexto actual (uso de los patrones repetitivos inherentes a las actividades realizadas)</p> <p>2-Derivar el contexto actual del usuario mediante el análisis de patrones de acceso del usuario a páginas Web</p>	<p>1-Realizar la separación de las propiedades según se relacionen de manera directa o indirecta con el contexto.</p> <p>2-Proponer un enfoque de investigación destinado a obtener una metodología para Ingeniería de Requerimientos sensibles al contexto (no se presenta un método en si).</p>	<p>1-Mostrar un meta-modelo para la elicitación de requerimientos y una metodología para utilizar el modelo.</p> <p>2- Utilizar concepto de contexto ampliado: computación, usuario, y físico.</p> <p>3-Focalizar en la usabilidad del sistema y la experiencia del usuario</p>	<p>1-Separa claramente contextos sensados, de contextos obtenidos en cooperación del usuario.</p> <p>2-Plantea utilizar la adquisición de contexto sensados y obtenidos por interacción con usuario para crear modelos, ontologías y contextos globales y locales</p>
Ventajas	<p>1-Capacidad de un servicio particular para adaptarse a los cambios en el contexto de modo automático</p> <p>2-Uso de estándares para representación de datos</p>	<p>1-Integra el proceso de recuperación de información con la secuencia de contextos pasados, facilitando la recuperación de información sensible al contexto</p> <p>2-Genera a partir de las preferencias reglas de BD, favorece la próxima interacción del usuario usando una simple búsqueda de BD</p>	<p>1-Utiliza técnicas de estimulación para la interacción del usuario con el analista/desarrollador</p> <p>2-Método genérico aplicable a diferentes tipos de sistemas sensibles al contexto.</p> <p>3-Utilizar la ontología de contexto facilita la tarea del analista.</p>	<p>1-Mejora notablemente la satisfacción del usuario.</p> <p>2-Presenta una base para que el sistema seleccione las características sensible al contexto en tiempo de ejecución.</p> <p>3- La sencillez del modelo presentado, y la utilización de ontologías de contextos.</p>	<p>1-Favorece la comunicación usuario-analista</p> <p>2- Centra la Ingeniería de Requerimientos en el usuario</p> <p>3- Mediante la unión de las dos técnicas de adquisición de contexto incrementa la satisfacción del usuario.</p>
Desventajas	<p>1-No fomenta la comunicación entre usuario y analista/desarrollador.</p> <p>2-Difícil de comprender debido al nivel de abstracción con que se presenta el framework, poca información sobre implementación de los componentes del framework.</p>	<p>1-No hace hincapié en la privacidad de los usuarios, es intrusivo.</p> <p>2-Se focaliza en un sistema particular de manejo de documentos, poco general.</p>	<p>1- Modelo de investigación propuesto, complejo y poco explicitado.</p>		<p>1-La propuesta está orientada a sistemas de mediana escala, aunque sugiere la necesidad de automatización para sistemas más grandes.</p> <p>2-Realiza una descripción de alto nivel sobre como hacer Ingeniería de Requerimientos, pero no formaliza ningún método.</p>

6. Conclusiones

En la generación de un nuevo método de Ingeniería de Requerimientos para Sistemas Sensibles al Contexto, los enfoques estudiados abarcan cuestiones relacionadas al constante aprendizaje de los sistemas sobre la información del contexto, que serían interesante de considerar:

- 1) La reflexión sobre la información de contexto que va obteniendo la aplicación de modo de poder ir adaptando/ mejorando la respuesta del sistema, en función de los cambios sufridos en el entorno de ejecución.
- 2) La utilización del concepto de contexto ampliado: dividir al contexto en una serie de contextos particulares permitirá centralizarse en los atributos del subconjunto de contexto que se está evaluando, minimizando así el número de errores al momento de obtener los requerimientos del sistema a desarrollar.
- 3) Basarse en taxonomías de contexto, el analista tiene una guía a seguir sobre cuáles atributos del contexto debería evaluar en aplicaciones sensibles al contexto.
- 4) La participación del usuario en el proceso de Ingeniería de Requerimientos: el éxito de una aplicación depende en gran parte de la satisfacción del usuario y para poder satisfacerlo se debe conocer sus gustos y preferencias. Se mejora de este modo la interacción entre el sistema y el usuario.
- 5) La incorporación de herramientas de aprendizaje automático, serían de gran utilidad para este tipo de sistemas, generalizando comportamientos a partir del suministro de información no estructurada (las interacciones del usuario, por ejemplo) y patrones repetitivos.

Particularmente en el transcurso de este trabajo, se encontraron varias aplicaciones o sistemas sensibles al contexto, pero con poco o casi ningún formalismo asociado a los métodos de Ingeniería de Requerimientos para este tipo de aplicaciones.

Por otra parte, a pesar de que los enfoques analizados promueven distintos métodos de Ingeniería de Requerimientos para Sistemas Sensibles al Contexto, ninguno de ellos los expresa con suficiente detalle, dando lugar a cierta incertidumbre y desafío al mismo tiempo, en pos de obtener un nuevo método que satisfaga las características para este tipo de sistemas.

7. Trabajos futuros

Se plantea como meta definir un nuevo método de Ingeniería de Requerimientos para aplicaciones sensibles al contexto extendiendo y/o modificando los métodos existentes, ya que aún a pesar de varios años de investigación, no hay un consenso al respecto entre los académicos del tema.

Otro desafío importante consiste en la posible utilización de agentes inteligentes, interactuando para obtener la base de conocimiento necesaria, de modo tal de brindar la reacción más adecuada de la aplicación sensible al contexto en una circunstancia determinada.

Por último, está en consideración la integración con otra rama de la Inteligencia Artificial (IA), específicamente el aprendizaje automático.

Referencias

[1] G. Chen ,D. Kotz, A survey of context-aware mobile computing research. Technical Report TR2000-381, Department of Computer Science, Dartmouth College, 2000.

[2] A. K. Dey, G. D. Abowd, Towards a better understanding of context and context-awareness, Workshop on the What, Who, Where, When, and How of Context-Awareness, The Hague, the Netherlands, Abril 2000.

- [3] Finkelstein and A. Savigni, A Framework for Requirements Engineering for Context-Aware Services, First International Workshop from Software Requirements to Architecture-23d International Conference on Software Engineering, 2001.
- [4] A. Galis, A. Juhola, D. Raz, J. Serrat-Fernandez, Fast and Efficient Context-Aware Services, John Wiley & Sons Ltd-2006.
- [5] Godbole, Smari, Human perspective based context acquisition, learning and awareness in the design of context aware systems, Department of Electrical and Computer Engineering, University of Dayton, Military Communications Conference, 2006.IEEE.
- [6] Hong, Chiu, Shen, Requirements elicitation for the design of context-aware applications in a ubiquitous environment, Proceedings of the 7th international conference on Electronic commerce, 2005
- [7] <http://www.ubiq.com/hypertext/weiser/UbiCompHotTopics.html>
- [8] IEEE standard glossary of software engineering terminology, IEEE Std 610.12-1990- 1990.
- [9] R. Kammanahalli, S. Gopalan, Sridhar V, K. Ramamritham ,Context Aware Retrieval in Web-Based Collaborations, *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops (PERCOMW'04)*
- [10] Kolos, Mazuryk, Poulisse, van Eck, Requirements Engineering for Pervasive Services, 16 Octubre 2005. 2nd Workshop on Building software for pervasive computing.
- [11] J. Krogstie, Requirement Engineering for Mobile Information Systems, Proceedings of the Seventh International Workshop on Requirements Engineering, Interlaken, Switzerland, 2001.
- [12] P. Loucopoulos, V. Karakostas; System Requirements Engineering; McGraw-Hill International series in Software Engineering, 1995;
- [13] F. Mattern: Visión y fundamentos técnicos de la Computación Ubicua, Novática núm.153, sept.-oct. 2001, <http://www.ati.es/novatica/>.
- [14] G. Rossi, J. Diab, Trabajo de Tesis de Magister: “Inteligencia en aplicaciones sensibles a contexto”. Facultad de Informática. 2006
- [15] B. Schilit, N. Adams, R. Want, Context-Aware Computing Applications, IEEE Workshop on Mobile Computing Systems and Applications, 1994.
- [16] Sommerville, Ian. Ingeniería de Software. 7ª Edición, Pearson Educación, Madrid 2005. ISBN: 8478290745
- [17] www.rae.es