

MODELO DE REQUISITOS PARA SISTEMAS EMBEBIDOS

Liliana González Palacio¹
Germán Urrego Giraldo²

Recibido: 03/03/2008

Aceptado: 04/10/2008

RESUMEN

En este artículo se presenta un modelo de requisitos como apoyo para la construcción de sistemas embebidos. En la actualidad, las metodologías de Ingeniería de Requisitos propuestas para este dominio no establecen continuidad en su proceso de desarrollo, ya que poseen una fuerte orientación a la etapa de diseño y un énfasis más débil en la etapa de análisis. Además, dichas metodologías ofrecen pautas para tratar los requisitos luego de que han sido obtenidos, pero no proponen herramientas; como por ejemplo, un modelo de requisitos, para la obtención de estos.

Este trabajo hace parte de un proyecto de investigación que tiene como objetivo proponer una metodología de Ingeniería de Requisitos (IR) para el análisis de Sistemas Embebidos (SE). El modelo de requisitos propuesto y su forma de utilización se ilustran mediante un caso de aplicación consistente en la obtención de requisitos para un sistema de sensado de movimiento, embebido en un sistema de alarma para hogar.

Palabras clave: Ingeniería de Software, Ingeniería de Requisitos, Metodologías, Sistemas Embebidos.

-
- 1 Ingeniera de sistemas de la Universidad de Antioquia. Magister (c) en Ingeniería con énfasis en Ingeniería de requisitos de la Universidad de Antioquia. Docente tiempo completo programa Ingeniería de Sistemas Universidad de Medellín. Teléfono: 3405529. E-mail: lignonalez@udem.edu.co
 - 2 Ingeniero Civil de la Universidad Nacional de Colombia, Aufbaustudium (equivalente a magister) en Informática aplicada de la Universidad Karlsruhe. (Alemania), Magíster en Teoría e Ingeniería de Bases de Datos en la Universidad de Paris I Panteón-Sorbona (Francia), PhD en Informática de la Universidad de Paris I Panteón-Sorbona, Profesor de la Universidad de Antioquia, Departamento de Ingeniería de Sistemas. Teléfono: 2105530. E-mail: gaurrego@udea.edu.co

MODEL OF REQUIREMENTS FOR EMBEDDED SYSTEMS

ABSTRACT

In this paper a model of requirements for supporting the construction of embedded systems is presented. Currently, the methodologies of Engineering of Requirements, in this field, do not let continuity in their development process, since they have a strong orientation to design stage and a weaker emphasis on the analysis stage. Furthermore, such methodologies provide guidelines for treating requirements after being obtained. However, they do not propose tools such as a model of requirements for obtaining them.

This paper is the result of a research project which objective is to propose engineering of requirements methodology for embedded systems analysis. The model of proposed requirements and its use are illustrated through an application case consisting on obtaining requirements for a movement sensing system, embedded in a home alarm system.

Keywords: Software Engineering, Engineering of Requirements, methodologies, embedded system.

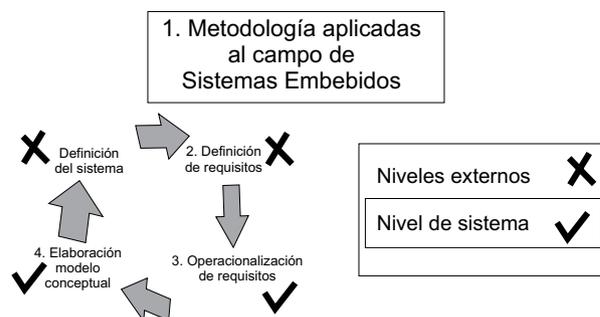
I. INTRODUCCIÓN

A menudo los diseñadores de sistemas cometen el error de comenzar a diseñar e implementar soluciones que no han sido completamente especificadas y que corresponden a problemas a los que les falta delimitación, lo cual conduce a la construcción de sistemas que no satisfacen las necesidades de los clientes y que incurrir en el aumento de los costos y en el incumplimiento de los plazos establecidos. Todo lo anterior refleja las carencias que existen en cuanto a la definición de requisitos como se describe en las estadísticas del Standish Group (Standish Group, 1994) (Standish Group, 2002) y en la literatura de Ingeniería de requisitos desde los trabajos pioneros de Ross (Ross y Schoman, 1977).

Esta problemática sigue existiendo tal como se ha mostrado en diferentes trabajos desde hace tres décadas (Zave y Jackson, 1997). En el campo de los sistemas embebidos tratados en Capel y Holgado (2004) y en (Ingham et al., 2006) dicha problemática general de los sistemas de información no solo se conserva sino que se agrava hasta el punto que el 60% de las componentes que integran hardware (HW) y software (SW) deben ser rediseñadas luego de haber sido programadas (Ganssle, 1999). La problemática expuesta para los sistemas en general tiene mayor impacto en el caso de los sistemas embebidos.

Dificultades adicionales en el caso de los sistemas embebidos surgen, entre otros, por los hechos siguientes: a) Estos sistemas se realizan generalmente 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 que no contienen ninguna adaptación a los sistemas embebidos. c) Las metodologías existentes en el campo de los sistemas embebidos no cubren todas las fases de la IR. Son orientadas a la fase de diseño con poco apoyo de las fases previas del ciclo

de vida de los sistemas. En la figura 1 se pueden observar las carencias de estas metodologías, en el nivel externo y en el nivel del sistema.



Fuente: elaboración propia.

Figura 1. Carencias encontradas en las metodologías de ingeniería de requisitos para sistemas embebidos

Algunas metodologías orientadas a sistemas embebidos sólo permiten expresar los requisitos que se encuentran en el nivel de sistema, es decir, los que tienen que ver con funcionalidades, pero olvidan los requisitos que están en niveles externos, en donde se encuentran las interrelaciones con el súper-sistema y otros sistemas embebidos. Son metodologías muy orientadas al diseño, ignorando casi por completo la fase de captura y análisis de requisitos, como la metodología basada en el análisis de estados tratada en Ingham et al. (2006), la cual permite la separación temprana de HW y SW. El hardware se modela con un diagrama de efectos de estado, y el software mediante la definición de metas y restricciones. La metodología CARA expuesta en Alur et al. (2004) también se ocupa del nivel del sistema y es concebida exclusivamente para sistemas médicos embebidos. En el proceso se transforman requisitos de diseño informales a lenguajes formales, como las máquinas de estado finito.

Otras metodologías para sistemas embebidos consideran, además del nivel del sistema, el contexto determinado por el súper-sistema y otros sistemas

embebidos. En esta categoría están los trabajos de la metodología ECSAM (Lavi et al., 2005) centrados en el análisis de caja negra (detectar sólo interacciones entre el sistema y sistemas externos), lo cual permite que usuarios finales y diseñadores entiendan los requisitos capturados. Pertenece también a esta categoría la transformación de requisitos expresados en diversas notaciones fuente a un grafo conceptual realizado por Cyre (1997).

En los sistemas embebidos, los procesos de ingeniería de requisitos se complican a tal punto que en las áreas de aplicación típica más del 50% de los problemas se producen porque el sistema no cumple con las expectativas del usuario, debido a la captura errónea de los requisitos (Cheng et al., 2006). De esta manera surgen nuevas tareas de rediseño que conllevan aumento en los costos por la compra adicional de componentes hardware e incremento en los tiempos de entrega debido a la necesidad de desarrollar nuevo software de control (Kovitz, 2001) (Lavi y Kudish, 2005).

En consecuencia, el objetivo fundamental de la investigación asociada con este artículo es el de proponer una metodología de ingeniería de requisitos para sistemas embebidos que permita el análisis de requisitos y la generación de un modelo conceptual que facilite la entrada a un lenguaje de especificación de sistemas embebidos. En este trabajo se presenta el modelo de requisitos para el dominio de sistemas embebidos, el cual da cuenta del logro del primer objetivo específico de la mencionada investigación.

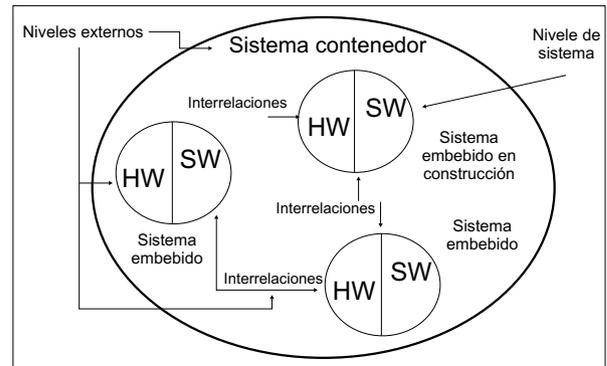
El presente artículo, además de la introducción, contiene: en la segunda sección, los materiales y métodos que fundamentan el trabajo. La tercera sección presenta los resultados en cuanto a la obtención del modelo de requisitos y un caso de aplicación de dicho modelo. Las conclusiones y trabajos futuros se enuncian en la cuarta sección. Las dos últimas secciones corresponden a los agradecimientos y a la bibliografía.

2. MATERIALES Y MÉTODOS

En esta sección se presentan los conceptos básicos que permiten entender las secciones siguientes. El proyecto se enmarca dentro de dos grandes áreas: la ingeniería de requisitos (IR) y los sistemas embebidos (SE).

Un sistema embebido es un sistema de procesamiento de información de uso específico integrado en otro sistema de mayor tamaño y conformado por componentes hardware y software (Marwedel, 2003).

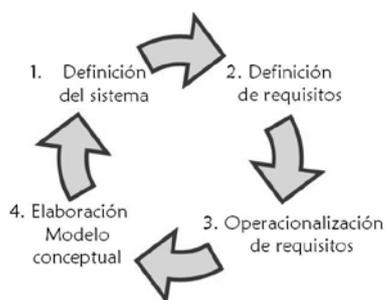
Los sistemas embebidos tienen algunas particularidades como la integración de componentes hardware y software (Marwedel, 2003), y su relación de jerarquía con un súper-sistema que se encarga de controlar la comunicación entre sistemas del mismo nivel. Esto se puede observar en la siguiente figura:



Fuente: elaboración propia.

Figura 2. Arquitectura básica de un súper-sistema de varios sistemas embebidos.

Por otro lado, la ingeniería de requisitos es una rama de la ingeniería de software que apoya al diseñador de sistemas en su tarea de traducir los objetivos del mundo real a funciones, restricciones de un sistema (Ross et al., 1977). Las fases de la ingeniería de requisitos se muestran en la figura 3:



Fuente: elaboración propia.

Figura 3. Fases de la ingeniería de requisitos.

En la fase de definición del sistema se obtiene el conocimiento de los contextos externos que expresan la problemática que se debe afrontar y fundamentan una solución. En la segunda fase se deberá obtener un documento con todos los requisitos. En la fase de operacionalización de requisitos se logra producir un documento detallado de funcionalidades y restricciones de bajo nivel de abstracción del sistema a construir; y en la cuarta fase se construye un modelo conceptual que contenga la solución acorde con los requisitos y restricciones.

Una metodología de IR pensada para este dominio debe permitir la representación de requisitos en el nivel de sistema, es decir, los que expresan funcionalidades, pero, además, debe plasmar requisitos que están en niveles externos, en donde se encuentran las interrelaciones con el súper-sistema y otros sistemas embebidos.

Como parte de los elementos básicos para construir una metodología orientada a sistemas embebidos se presenta una reseña de algunas metodologías de IR que han sido aplicadas en otros dominios y que presentan condiciones para permitir su transformación y adaptación al dominio de los sistemas embebidos, buscando que posibiliten el descubrimiento de requisitos que se encuentran en niveles diferentes al del sistema. Dichas metodologías pueden ser clasificadas de diversas formas, pero en la tabla 1 se han retenido sólo algunas de las que ofrecen mayores facilidades

para su adaptación y se han clasificado por su fundamento en los conceptos de escenario o de meta (Liu y Yu, 2001).

Tabla 1. Algunas metodologías orientadas por metas o por escenarios.

| Metodologías orientadas por escenarios | Metodologías orientadas por metas |
|--|---|
| ScenIC (Potts, 1999): Se basa en objetivos, tareas, obstáculos y acciones llevadas a cabo por actores. | KAOS (Dardenne et al, 1993) (Leiter y Lamsweerde, 2002): Se fundamenta en técnicas de Inteligencia Artificial para definir metas, acciones, estados de forma consistente y rigurosa |
| SCRAM: Propone una combinación de escenarios y prototipos tempranos (Sutcliffe, 2003). | ABC-Besoins (Urrego, 2005): Orientada por metas y agentes para lograr un mayor entendimiento del problema a resolver |

Fuente: elaboración propia.

Las metodologías orientadas por escenarios como ScenIC y SCRAM son más entendibles para el usuario, pero conllevan ambigüedades e inconsistencias, y no poseen una forma de determinar cuándo se han tomado suficientes escenarios como para cubrir toda la funcionalidad del sistema a construir.

Por su parte, las metodologías orientadas por metas tienen un grado de dificultad mayor, pero permiten obtener resultados más formales e incluir requisitos de niveles externos, esto es, que expresan no sólo funcionalidades. KAOS tiene la desventaja de hacer el tratamiento de requisitos de manera aislada, mientras que ABC-Besoins, con el concepto de “Intervención de agentes” logra integrar y relacionar los requisitos de diferentes contextos. Como ventaja adicional ABC-Besoins propone pautas para el análisis de requisitos desde la fase de captura hasta la fase de especificación, logrando la operacionalización de dichos requisitos y la construcción de un modelo conceptual. Todas estas ventajas se deben a un modelo de requisitos expresivo que permite relacionar requisitos de

diferentes niveles y capturar las necesidades de los agentes.

3. RESULTADOS

Esta sección se subdivide para mostrar inicialmente la forma en que se obtuvo el modelo de requisitos para sistemas embebidos, y luego pasar a un caso de aplicación.

Obtención del modelo de requisitos para sistemas embebidos

Teniendo en cuenta la carencia que en general presentan las metodologías descritas en la revisión de la literatura, se propone intervenir la metodología ABC-Besoins, que fue diseñada para el dominio de sistemas web, adaptándola y transformándola para tener en cuenta aspectos del dominio de sistemas embebidos, y construir un modelo conceptual que facilite la entrada a un lenguaje de especificación como SystemC. La decisión de retomar esta y no otra metodología se fundamenta principalmente en que ABC-Besoins ofrece soporte para todas las fases del análisis de requisitos y posee un modelo de requisitos bien estructurado que permite descubrir requisitos de los niveles de sistema y los niveles externos. En las sub-secciones siguientes se presenta el proceso y posterior resultado de adaptar el modelo de requisitos de la metodología seleccionada al dominio de sistemas embebidos.

Para lograr dicha adaptación fue necesario estudiar, primero, las características propias de los sistemas embebidos, las cuales se enuncian en la siguiente sub-sección.

A) *Identificación de las características propias de los sistemas embebidos*

La característica más importante de los sistemas embebidos es su interacción con el mundo exterior en función del tiempo o en función de

la presencia de estímulos. Para garantizar una interacción exitosa con el ambiente, el sistema debe incorporar algunas características, tales como la disponibilidad, fiabilidad y seguridad. Otras características propias de un sistema embebido son (Marwedel, 2003) (Lavi y Kudish 2005):

- Compuesto por hardware y software, con la característica de que el software tiene una interacción directa con los elementos hardware, pues se encarga de controlarlos y comunicarlos. Para esta composición debe ser posible representar: comportamiento (estados, eventos, y señales) y estructura o composición física.
- Relaciones jerárquicas, en las cuales se incluyen las interrelaciones entre el sistema embebido y su súper-sistema, el sistema embebido y sistemas del mismo nivel que se encargan de otras funciones específicas.
- Comportamiento basado en el estado de las componentes, por ejemplo, si X puerto no está disponible, entonces no se podrá enviar la señal que activa el proceso Y.
- Manejo de eventos, que son los que permiten constatar el cambio de estado de las componentes. Un evento puede ser externo (causado por el ambiente) o interno (causado por componentes del sistema).
- Recursos limitados en cuanto al tamaño, el consumo de energía, la memoria, y demás recursos que permitan garantizar la portabilidad del sistema embebido.
- Mínima interacción con el usuario, por lo tanto, son sistemas que deben funcionar durante años sin errores y ser capaces de recuperarse por sí mismos en caso de que estos ocurran. Deben ser sistemas con un alto grado de autonomía.
- Presencia de sincronización y comunicación, para permitir el flujo de información entre los diferentes sistemas embebidos que hacen

funciones específicas y contribuyen a la realización de la función del súper-sistema.

- Propiedades no funcionales, tales como la tolerancia a fallas, el tamaño, el consumo de potencia, el peso, la disponibilidad, la seguridad, la fiabilidad, deben definirse desde etapas tempranas de la construcción del sistema.
- La controlabilidad es otra característica de los sistemas embebidos, pues son sistemas pensados, en su mayoría, para el control, y además, por sus restringidas y muy específicas funciones, también son fácilmente controlables.

La construcción de un modelo de interacciones entre los agentes relacionados con el sistema embebido constituye el segundo paso en la elaboración del modelo de requisitos, tal como se describe en la sub-sección siguiente.

B) Construcción de un modelo de interacciones de agentes

Luego de conocer las características propias de los sistemas embebidos, se construyó, con la

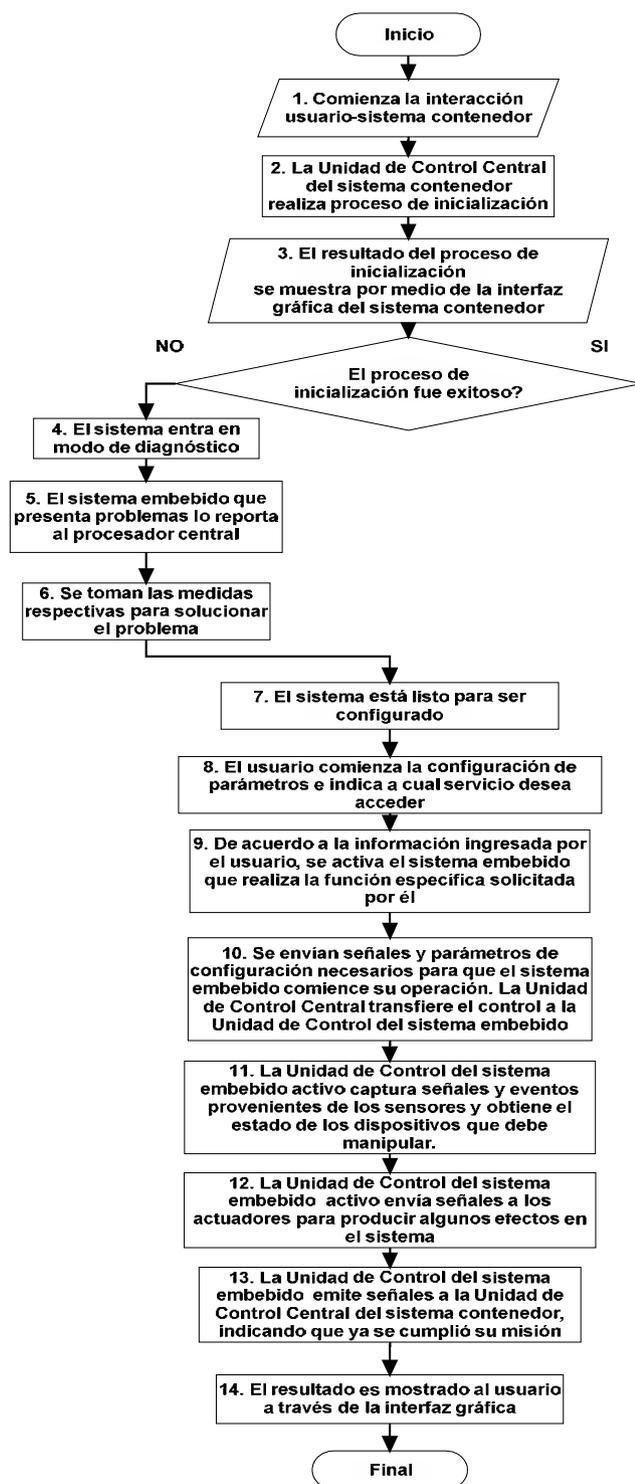
colaboración de estudiantes y profesores del grupo de Microelectrónica y Control de la Universidad de Antioquia, un modelo de interacción entre los agentes relacionados con el sistema embebido. Esta construcción se fundamentó en los dos elementos siguientes:

- Construcción del diagrama general de funcionamiento de un sistema embebido, que describe el proceso general que ocurre desde que el usuario comienza su interacción con el súper-sistema, hasta que el sistema embebido cumple una función específica y retorna resultados a dicho súper-sistema para que éste los haga visibles al usuario. Este diagrama se muestra en la figura 4.
- Identificación de las interacciones típicas en el interior de un sistema embebido. Con base en el diagrama general de funcionamiento se identifican en detalle las interacciones que soportan el funcionamiento general del sistema embebido. Dichas interacciones se representan en la tabla 2.

Tabla 2. Interacciones típicas en el interior de un sistema embebido

| |
|--|
| 1. El sistema embebido (SE) realiza un proceso de inicialización de sus componentes para determinar su estado. |
| 2. El SE activa sus interfaces de comunicación para recibir parámetros que le permitan desarrollar su función. |
| 3. El súper-sistema envía los parámetros necesarios para que el SE comience a ejecutar su función específica. |
| 4. El SE monitorea las señales provenientes de los sensores |
| 5. El SE envía señales a los actuadores para que ejecuten acciones como convertir señales de corriente o voltaje y activar otros dispositivos electrónicos o mecánicos |
| 6. El SE almacenan los resultados obtenidos en la memoria |
| 7. El SE envía señales que deben ser mostradas al usuario por medio de la interfaz gráfica |
| 8. El SE envía las señales que requieren otros sistemas embebidos para completar su función específica. |
| 9. El SE finaliza su función y transfiere el control al súper-sistema |

Fuente: elaboración propia.



Fuente: elaboración propia.

Figura 4. Diagrama de flujo de la interacción usuario- súper-sistema y sistema embebido

C) Transformación y adaptación del modelo de requisitos

Con base en las interacciones típicas y en el diagrama general de funcionamiento se identificaron los elementos esenciales propios de los sistemas embebidos a ser considerados en la definición de requisitos conducentes a la especificación de las interacciones típicas antes determinadas. Se definieron requisitos de los agentes identificados en el diagrama general de funcionamiento que determinarían de forma clara, precisa y completa las interacciones típicas. Los requisitos identificados se cotejaron con las trece categorías de la taxonomía de requisitos de ABC-Besoins para constatar si esta taxonomía cubría todos los requisitos detectados, o si, por el contrario, era necesario agregar nuevas categorías. En las categorías coincidentes con los requisitos se analizó la necesidad de subdividir las categorías para definir requisitos más específicos que incorporaran los elementos propios de los sistemas embebidos antes identificados. De esta manera se constató que las categorías de la taxonomía de ABC-Besoins eran suficientes para cubrir los posibles requisitos de los sistemas embebidos y se desagregaron dichas categorías para producir una taxonomía detallada propia de estos sistemas. El modelo de requisitos para sistema embebidos, obtenido mediante el anterior procedimiento, se presenta en la tabla 3.

Tabla 3. Modelo de requisitos metodología ABC-Besoins-SEM

MODELO DE REQUISITOS PARA SISTEMAS EMBEBIDOS

| Categorías modelo original ABC-Besoins | Subcategorías adicionadas para el dominio de SE |
|--|--|
| Disponibilidad de objetos | <ul style="list-style-type: none"> • Señales |
| Convocación y demostración | <ul style="list-style-type: none"> • Interfaces |
| Selección de alternativas | <ul style="list-style-type: none"> • Modos de operación • Submodos de operación |
| Interacción de agentes | <ul style="list-style-type: none"> • Interacciones con súper-sistema • Interacciones con otros SE y el ambiente. |
| Acción del agente | <ul style="list-style-type: none"> • Funcionalidades de entrada • Funcionalidades de proceso • Funcionalidades de salida |
| Transferencia/actualización | Sin subcategorías |
| Interrupción/restitución/conservación | <ul style="list-style-type: none"> • Prevención de fallas • Detección de fallas |
| Desempeño/cambio | Sin subcategorías |
| Precio y costos | <ul style="list-style-type: none"> • Tamaño • Consumo de energía |
| Descripción o caracterización de los agentes | <ul style="list-style-type: none"> • Disponibilidad • Fiabilidad • Seguridad frente al exterior • Seguridad frente a ataques • Eficiencia |

Fuente: elaboración propia.

Como se puede observar, no fue necesario eliminar ni adicionar ninguna categoría al modelo original ABC-Besoins, pero, sí se incluyeron algunas subcategorías, que se explican detalladamente en cada una de las tablas siguientes:

Tabla 4. Descripción de la categoría “Disponibilidad de objetos” y subcategorías asociadas

| Categoría: Disponibilidad de objetos | |
|---|---|
| Son requisitos que permiten indicar la forma en que se deben comportar y comunicar los componentes del sistema de acuerdo a los diferentes modos de operación, que se dan de acuerdo a los valores de determinadas señales que fluyen entre procesos y entre componentes del sistema. | |
| Subcategoría | Descripción de la subcategoría |
| Señales | Una señal comunica la ocurrencia de un evento que genera consecuencias como un cambio de estado (Lavi et al., 2005). Representa una condición necesaria para que un proceso de ejecute. |

Fuente: elaboración propia.

Tabla 5. Descripción de la categoría “Convocación y demostración” y subcategorías asociadas

| Categoría: Convocación y demostración | |
|---|---|
| Descripción de la categoría: Estos requisitos tienen que ver con la comunicación que debe establecer el sistema con otros sistemas, tales como su súper-sistema y demás sistemas embebidos que se encuentran dentro de él. Se relacionan con las interfaces provistas para comunicarse hacia el exterior. | |
| Subcategoría | Descripción de la subcategoría |
| Interfaces | Es importante resaltar que los sistemas embebidos usualmente no tienen una interfaz propia con el usuario humano, sino que operan a través de una interfaz provista por el sistema global o el súper-sistema. Como consecuencia, la especificación de requisitos de interfaz de usuario del sistema embebido será similar a la del sistema global o contenedor. |

Fuente: elaboración propia.

Tabla 6. Descripción de la categoría “Selección de alternativas” y subcategorías asociadas

| Categoría: Selección de alternativas | |
|---|--|
| Descripción de la categoría: Son requisitos relacionados con la forma en que el sistema debe desplegar al usuario las posibles alternativas de uso. Este tipo de requisitos da paso a los modos y submodos de operación. | |
| Subcategoría | Descripción de la subcategoría |
| Modos de operación | Un modo de operación es un conjunto de funcionalidades del sistema que se activan o desactivan con la ocurrencia de un evento, envío de una señal y posterior entrada a un modo de operación. Los modos de operación típicos de un sistema embebido son: encendido, apagado (Lavi et al., 2005). |
| Submodos de operación | Pensando en jerarquía, cada modo de operación puede ser refinado en submodos. Por ejemplo, si el sistema embebido está en modo encendido, a su vez puede estar en uno de los siguientes submodos: en configuración, en operación normal, en mantenimiento (Lavi et al., 2005) |
| Categoría: Solicitud de acceso | |
| Descripción de la categoría: Luego de conocer las posibilidades que ofrece el sistema, el usuario selecciona cual es la de su interés. Estos requisitos tienen entonces que ver con dicha selección, y continuando con los modos de operación, tienen que ver con las funcionalidades que deben estar activas por cada modo. | |
| Sin subcategorías | |
| Categoría: Aporte (entrada-respuesta) | |
| Descripción de la categoría: Siguiendo con el proceso para el ofrecimiento y posterior selección de servicios, estos requisitos reflejan las acciones que el usuario debe realizar para acceder a alguna de las alternativas que ofrece el sistema. Por ejemplo, si el sistema es un cajero automático, luego de seleccionar la opción de ingresar al sistema, el cajero pide el ingreso de una contraseña para poder continuar y desplegar otras opciones. | |

| Sin subcategorías |
|---|
| Categoría: Verificación/decisión |
| Descripción de la categoría: posterior al aporte del usuario para obtener el servicio o funcionalidad requerida, el sistema hace verificaciones internas y continúa con el proceso para la prestación del servicio o la activación de la funcionalidad. |
| Sin subcategorías |

Fuente: elaboración propia.

Tabla 7. Descripción de la categoría “Interacción de agentes” y subcategorías asociadas

| Categoría: Interacción de agentes | |
|---|--|
| Descripción de la categoría: Estos requisitos, al igual que los denominados “Acción del agente”, hacen parte de los más conocidos como “requisitos funcionales”. De manera general, un requisito funcional indica cuales son las funciones que debe proveer el sistema a los agentes que lo usan, incluyendo como agentes los otros sistemas con los que debe interactuar. Aunque los sistemas embebidos están más orientados a la acción que a la interacción, existen relaciones que se establecen con los sistemas interactuantes y que aportan al cumplimiento de la función general del súper-sistema. | |
| Subcategoría | Descripción de la subcategoría |
| Interacciones con súper-sistema | Aquí solo se detallan las funciones que permiten la interacción del sistema embebido y el súper-sistema. |
| Interacciones con otros SE y el ambiente | Como ya se mencionó en las características propias de los sistemas embebidos, la presencia de jerarquía permite que cada sistema embebido incluido en un súper-sistema tenga asignada una función específica, que aporta al logro de una función general. En esta subcategoría se tendrán en cuenta las interrelaciones con los sistemas embebidos incluidos dentro del súper-sistema y la comunicación con el ambiente. |

Fuente: elaboración propia.

Tabla 8. Descripción de la categoría “Acción del agente” y subcategorías asociadas

| Categoría: Acción del agente | |
|--|---|
| Descripción de la categoría: Estos requisitos también hacen parte de los “requisitos funcionales”, pero no se ocupan de las interacciones, sino de las funcionalidades y servicios que debe proveer el sistema para cumplir con su propósito específico dentro del súper-sistema (Fuentes y Zúñiga, 2003). | |
| Subcategoría | Descripción de la subcategoría |
| Funcionalidades de entrada | Para determinar estas funcionalidades es preciso observar el sistema embebido como una caja negra, y analizar cuales son las entradas que debe recibir, tanto del súper-sistema, como de los otros sistemas embebidos, para comenzar su funcionamiento. |
| Funcionalidades de proceso | En este punto es preciso observar el sistema embebido como una caja blanca, para encontrar las funciones que debe proveer y lograr las salidas que se conectarán con otros sistemas. Entre las funciones típicas de proceso en un sistema embebido se cuentan: iniciación, configuración, diagnóstico, almacenamiento, terminación. |
| Funcionalidades de salida | Al igual que las funcionalidades de entrada, para esta subcategoría se observa el sistema como una caja negra y se determina que debe entregar a los demás sistemas para cumplir con su función específica. |

Fuente: elaboración propia.

Tabla 9. Descripción de la categoría “Transferencia/ actualización” y subcategorías asociadas

| Categoría: Transferencia/ actualización | |
|--|--|
| Descripción de la categoría: Son requisitos que reflejan cambios o actualizaciones a realizar luego de que el usuario lleve a cabo determinada acción. | |
| Sin subcategorías | |

Fuente: elaboración propia.

Tabla 10. Descripción de la categoría “Interrupción/ restitución/ conservación” y subcategorías asociadas

| Categoría: Interrupción/ restitución/ conservación | |
|---|--|
| Descripción de la categoría: Son requisitos relacionados con la reacción del sistema ante errores y fallos, por lo tanto es la forma de convertir en funcional un requisito como la fiabilidad del sistema. | |
| Subcategoría | Descripción de la subcategoría |
| Prevención de fallas | Estos requisitos expresan características que deben ser incluidas para minimizar las posibilidades de error del sistema antes de su entrada en funcionamiento. Por ejemplo, es importante incorporar componentes hardware fiables, y usar metodologías de especificación y diseño rigurosas |
| Detección de fallas | A pesar de definir características para evitar que ocurran fallas, éstas pueden ocurrir, y es preciso tener mecanismos que permitan determinar cuando el sistema entró en un estado de error y que acciones se deben emprender para recuperarse. La incorporación de facilidades de auto-chequeo y el uso de módulos del sistema redundantes son ejemplos. La inclusión de planes de contingencia también aumenta la tolerancia ante fallas, esto implica identificar las funciones críticas del sistema, y por cada función determinar las posibles fallas que puedan ocurrir, y por último, indicar posibles soluciones ante cada falla. Es importante hacer énfasis en estos requisitos, si se tiene en cuenta que será el propio sistema embebido el que se recupere, teniendo en cuenta que tiene poca interacción con el usuario humano. |

Fuente: elaboración propia.

Tabla 11. Descripción de la categoría “Desempeño/cambio”

| |
|---|
| Categoría: Desempeño/cambio |
| Descripción de la categoría: Estos requisitos permiten incluir formas de optimizar tiempos para realizar una operación. |
| Sin subcategorías |

Fuente: elaboración propia.

Tabla 12. Descripción de la categoría “Precio y costos” y subcategorías asociadas

| Categoría: Precio y costos | |
|---|--|
| Descripción de la categoría: Las principales características que determinan el precio de un sistema embebido son: su tamaño y el consumo de energía, factores que pueden sacar a un competidor del mercado, por lo cual, es preciso hallar un punto de equilibrio entre lo que cuestan las componentes a incluir y su ventaja frente a tamaño y consumo de energía. | |
| Subcategoría | Descripción de la subcategoría |
| Tamaño | Son requisitos que indican cual debe ser el tamaño de las componentes del sistema. La mayoría de los sistemas embebidos deben ser fáciles de transportar por el usuario y además, hacen parte de dispositivos que deben ser en esencia pequeños (por ejemplo, teléfonos celulares, PDA). Los diseñadores deben tener especial cuidado con los componentes que seleccionan. |
| Consumo de energía | El consumo de energía es una de las características que indica el grado de portabilidad de un sistema embebido, y también depende de los componentes que el diseñador escoja a la hora de construir e implementar el sistema embebido. |

Fuente: elaboración propia.

Tabla 13. Descripción de la categoría “Descripción o caracterización de los agentes” y subcategorías asociadas

| Categoría: Descripción o caracterización de los agentes | |
|---|--|
| Descripción de la categoría: Son requisitos no funcionales que definen propiedades emergentes del sistema, tales como disponibilidad, eficiencia, seguridad, confiabilidad. Pueden especificar también la utilización de una herramienta particular, un lenguaje de programación o un método de desarrollo durante la construcción del sistema. En los sistemas embebidos pueden ser más críticos que los requisitos funcionales. | |
| Subcategoría | Descripción de la subcategoría |
| Disponibilidad | Requisitos que manifiestan la necesidad de que un sistema, en un punto del tiempo, sea operativo y capaz de ofrecer los servicios requeridos por los usuarios (Humanes et al., 2004). |
| Fiabilidad | Requisitos creados para garantizar que el sistema ejecute sus funcionalidades libre de fallos en un tiempo especificado y en un ambiente determinado (Humanes et al., 2004). Para cumplir con estos requisitos puede ser necesario especificar requisitos adicionales relacionados con la gestión de fallas. |
| Seguridad frente al exterior | Requisitos relacionados con aquellos atributos que debe tener el sistema para operar sin provocar daños en el ambiente y sin entorpecer las funciones de los demás sistemas con los cuales interactúa. |
| Seguridad frente a ataques | Requisitos orientados a evitar ataques externos que pongan en peligro las funcionalidades del sistema, o que permitan comportamientos como: Denegación de servicio (que pone en peligro la disponibilidad), corrupción de programas o datos manejados por el sistema (lo cual puede afectar el comportamiento del sistema y, por tanto, la fiabilidad y la seguridad), revelación de información manejada por el sistema. Si no se garantiza este tipo de seguridad, el sistema puede ser corrompido y puede comportarse de una forma inesperada |
| Eficiencia | Requisitos que permiten lograr un efecto determinado optimizando los recursos disponibles. |

Fuente: elaboración propia.

Caso de aplicación del modelo de requisitos

Para validación del modelo de requisitos de sistema embebidos se toma como caso de aplicación un sistema de sensado de movimiento, inmerso en un sistema de alarma para hogar, que fue utilizado en la metodología ECSAM, propuesta en Lavi et al. (2005). Para este caso de aplicación se consideraron dos agentes principales: el comprador del sistema para alarma de hogar, quien da los requisitos del súper-sistema y el fabricante de sistemas de alarma, encargado de indicar los requisitos para el sistema de sensado de movimiento.

Siguiendo las categorías y subcategorías de requisitos propuestas en este trabajo, representadas en la tabla 3, y teniendo como base las interacciones típicas descritas en la tabla 2, se obtuvieron los requisitos para el sistema de sensado, que aparecen en la tabla 17.

Los requisitos obtenidos fueron comparados con los requisitos obtenidos con la metodología ECSAM expuestos en Lavi et al. (2005), observándose que efectivamente el modelo de requisitos propuesto amplía y diversifica los requisitos para el sistema de sensado.

Tabla 14. Caso de estudio para aplicación del modelo de requisitos ABC-Besoins-SEM

| Categorías modelo original ABC-Besoins | Subcategorías adicionadas para el dominio de SE | Requisitos obtenidos para el caso de estudio |
|--|---|---|
| Disponibilidad de objetos | Estados | <ul style="list-style-type: none"> Los estados de los componentes del sistema de sensado de movimiento serán idénticos a los modos de operación generales del sistema (ver subcategoría Modos de operación). |
| | Eventos | <p>El sistema de sensado de movimiento cambiará de modos de operación de acuerdo a los siguientes eventos:</p> <ul style="list-style-type: none"> Si se detecta la entrada “enable” con un valor de 0 el sistema está inactivo. Si el sistema está en modo inactivo y la entrada “enable” cambia a 1, en los primeros 3 segundos estará en modo activo y submodo de diagnóstico, pero luego pasará al submodo de operación normal. Si el sistema está en modo activo y submodo de operación normal y la entrada “enable” cambia a 0, cambiará a modo inactivo. Si el sistema está activo y en operación normal y falla el sensor, se debe entregar a la salida Q1 el valor de 5.5 voltios, y entrar al submodo de mantenimiento. Si el sistema está activo y en operación normal y se interrumpe el cable de alimentación, la salida Q1 debe tomar el valor de alta impedancia y entrar al submodo alta impedancia (nivel lógico que no se comprende ni como un 0 ni como un 1). |
| Convocación y demostración | Interfaces | <ul style="list-style-type: none"> El sistema de sensado de movimiento deberá contener una interfaz para calibrar la sensibilidad del sensor. |
| Selección de alternativas | Modos de operación | <ul style="list-style-type: none"> El sistema de sensado de movimiento debe tener dos modos de operación básicos: Encendido o apagado, o en otras palabras: activo o inactivo. |
| | Submodos de operación | <ul style="list-style-type: none"> Para el modo de operación encendido o activo se deben considerar los siguientes submodos: operación normal, alta impedancia, diagnóstico, mantenimiento. |

| Categorías modelo original ABC-Besoins | Subcategorías adicionadas para el dominio de SE | Requisitos obtenidos para el caso de estudio |
|--|---|--|
| Solicitud de acceso | Sin subcategorías | <ul style="list-style-type: none"> • Para el modo inactivo, solo deben estar presentes: la funcionalidad que permite conexión con la fuente de alimentación y la función que mantiene contacto con el sistema central de control que permite el cambio al modo activo. • En el modo activo, submodo de operación normal deben estar activas todas las funcionalidades descritas en las categorías “Interacción de agentes” y “acción del agente”. • Para el modo activo, submodo de alta impedancia solo se activa la posibilidad de cambiar de valor la salida Q1. • En el modo activo, submodo de diagnóstico, debe estar activa la funcionalidad que ejecuta la rutina de detección de fallas. • En el modo activo, submodo de mantenimiento solo se activa la posibilidad de cambiar de valor la salida Q1. |
| Aporte (Entrada-respuesta) | Sin subcategorías | <ul style="list-style-type: none"> • El sistema de sensado de movimiento no tiene ningún requisito de este tipo. • El súper-sistema del sistema de sensado tiene como requisito de aporte el ingreso de una contraseña para permitir el cambio de modo de operación. |
| Verificación/Decisión | Sin subcategorías | <ul style="list-style-type: none"> • El sistema de sensado de movimiento no tiene ningún requisito de este tipo. • El súper-sistema del sistema de sensado tiene como requisito de verificación la existencia de una base de datos para el almacenamiento de usuarios, claves y tipos de usuario. |
| Interacción de agentes | Interacciones con súper-sistema | <p>El sistema de sensado de movimiento:</p> <ul style="list-style-type: none"> • Se comunicará con su súper-sistema a través de la entrada I1 o “enable”. • Entregará resultados por medio de las salidas: Q1 y Q3. |
| | Interacciones con otros SE y el ambiente. | <p>El sistema de sensado de movimiento:</p> <ul style="list-style-type: none"> • No se comunicará con otros sistemas embebidos. • Recibirá estímulos del ambiente por medio de señales físicas como el movimiento. |
| Acción del agente | Funcionalidades de entrada | <p>El sistema de sensado de movimiento deberá:</p> <ul style="list-style-type: none"> • Tomar como principio de funcionamiento el efecto Doppler (básicamente consiste en que una frecuencia sufre una variación proporcional a su velocidad de desplazamiento). • Recibir una entrada proveniente del sistema central de control que le indique su activación o desactivación (denominada I1 o “enable”). • Recibir estímulos del ambiente por medio de señales ultrasónicas que corresponden a movimientos de objetos o personas (denominada I2). • Recibir una entrada proveniente del mecanismo que permite calibrar la sensibilidad del sensor ante los movimientos (conocida como I3). • Recibir una entrada proveniente de la fuente de alimentación (conocida como I4). |

| Categorías modelo original ABC-Besoins | Subcategorías adicionadas para el dominio de SE | Requisitos obtenidos para el caso de estudio |
|--|---|--|
| Acción del agente | Funcionalidades de proceso | <p>El sistema de sensado de movimiento deberá:</p> <ul style="list-style-type: none"> • Proporcionar un mecanismo de calibración que permita graduar la sensibilidad del sensor ante el movimiento (revisar que no esté repetido). • Convertir una señal eléctrica de 40 kHz en una onda ultrasónica. • Proporcionar un mecanismo para tratar las ondas ultrasónicas provenientes del ambiente y convertirlas en señales eléctricas. • Tomar la señal eléctrica y amplificarla. • Tomar la señal amplificada y convertirla de analógica a digital. • Tomar el valor convertido y compararlo con el valor almacenado como referencia para determinar si hubo movimiento. |
| | Funcionalidades de salida | <p>El sistema de sensado de movimiento deberá:</p> <ul style="list-style-type: none"> • Entregar una señal eléctrica al sistema de control central que indique si hubo movimiento o si algún otro evento se presentó (tomará el nombre de Q1). Los valores que tomará esta señal son: 0 voltios, si no se detectó movimiento; 5 voltios si hay movimiento; 5.5 voltios si la rutina de detección de fallas detectó que el sensor tiene problemas; otro valor diferente si hay alta impedancia. • Entregar una señal ultrasónica de manera continua para hacer la comparación de frecuencias y determinar si se ha presentado movimiento (denominada Q2). • Entregar el resultado de ejecutar la rutina de reconocimiento que permite determinar si hay fallas en el sistema (conocida como Q3). |
| Transferencia/actualización | Sin subcategorías | <p>El sistema de sensado de movimiento deberá:</p> <ul style="list-style-type: none"> • Comparar la onda ultrasónica que llega al sensor de movimiento con la que se obtiene desde el mecanismo de calibración, para determinar si se presentó movimiento. |
| Interrupción/restitución/conservación | Prevención de fallas | <p>El sistema de sensado de movimiento deberá:</p> <ul style="list-style-type: none"> • Implementar una rutina de conteo de la vida útil del sensor. • Contar con un mecanismo para evaluar la calidad de la señal de alimentación para evitar conexiones y desconexiones sucesivas del sensor. |
| | Detección de fallas | <ul style="list-style-type: none"> • El sistema de sensado de movimiento deberá implementar una rutina de detección de fallas para chequear el estado del sensor. |
| Desempeño/cambio | Sin subcategorías | <ul style="list-style-type: none"> • El sistema de sensado de movimiento debe detectar un movimiento con un retardo máximo de 100 microsegundos. |
| Precio y costos | Tamaño | <ul style="list-style-type: none"> • El sistema de sensado debe tener un tamaño máximo de 10 cm de alto x 5 cm de ancho x 5 de profundo |
| | Consumo de energía | <ul style="list-style-type: none"> • Para que sean intrínsecamente seguros, los componentes del sistema no deben consumir más de 20 mA, porque de lo contrario, en áreas clasificadas (químicos) pueden generar incendios. |
| Descripción o caracterización de los agentes | Disponibilidad | <ul style="list-style-type: none"> • En modo activo, el sistema debe estar disponible los 7 días de la semana y las 24 horas del día. |

| Categorías modelo original ABC-Besoins | Subcategorías adicionadas para el dominio de SE | Requisitos obtenidos para el caso de estudio |
|--|---|--|
| Descripción o caracterización de los agentes | Fiabilidad | <ul style="list-style-type: none"> Los requisitos necesarios para garantizar la fiabilidad ya fueron expresados en la categoría “Interrupción/restitución/conservación” |
| | Seguridad frente al exterior | <ul style="list-style-type: none"> Sin requisitos. |
| | Seguridad frente a ataques | <ul style="list-style-type: none"> El súper-sistema debe tener un sistema de batería adicional para evitar que el sistema quede aislado. |
| | Eficiencia | <ul style="list-style-type: none"> El sistema de sensado de movimiento deberá optimizar los recursos para detectar todos los movimientos sobrepasen el nivel de tolerancia definido en el proceso de calibración. |

Fuente: elaboración propia.

4. CONCLUSIONES Y TRABAJOS FUTUROS

El modelo de requisitos propuesto en este trabajo explota y especializa las categorías del modelo de la metodología ABC-Besoins, mostrando que las grandes categorías de esta metodología cubren los requisitos de los sistemas embebidos, confirmando de esta manera la universalidad y expresividad de dicho modelo de requisitos probado anteriormente para sistemas web. Las subcategorías de requisitos propias obtenidas para los sistemas embebidos dan mayor expresividad al modelo y constituyen una guía útil para la captura de los requisitos específicos de los sistemas embebidos. Esta validación y extensión del modelo de requisitos de la metodología ABC-Besoins hace parte de una nueva metodología denominada ABC-Besoins-SEM, cuyos modelos de análisis y de diseño hacen parte de una investigación en curso y que serán presentados en otro trabajo futuro. De la misma manera se está trabajando en la construcción de metodologías para otros sistemas, tales como los sistemas ubicuos.

5. AGRADECIMIENTOS

Especial agradecimiento a los estudiantes y profesores del grupo de Microelectrónica y Con-

trol de la Universidad de Antioquia, por su colaboración y orientación durante la aplicación del caso de estudio y por la financiación del proyecto.

6. REFERENCIAS

- ALUR, R, ARNEY, D GUNTER, E, LEE, I, LEE, J, NAM, M, PEARCE, F, VAN, E, & ZHOU, J. (2004). Formal specifications and analysis of the computer-assisted resuscitation algorithm(CARA) Infusion Pump Control System”. *Int J Softw Tools Technol Transfer*. 5: 308–319.
- CAPEL, M, HOLGADO, J. (2004). A New Design Procedure for a Real-Time Hybrid System Model. *Memorias IV Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento*. Universidad Politécnica de Madrid, 1: 191– 204.
- CHENG, B, KONRAD, S, KAMDOUN, S. (2006). Enabling a Roundtrip Engineering Process for the Modeling and Analysis of Embedded Systems. *Proceedings of the ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS 2006)*. Italia.
- CYRE, W. (1997). Capture, Integration, and Analysis of Digital System Requirements with Conceptual Graphs. *IEEE Transactions on Knowledge and Data Engineering*, 9(1).
- DARDENNE, A, LAMSWEERDE, V, FICKAS, S. (1993). Goal Directed Requirements Acquisition. *Science of Computer Programming*, 20: 3–50.
- FUENTES, A, HARDINGS, J, ZÚÑIGA, M. (2003). Software libre en sistemas embebidos. *Seminario de software libre*.

- GANSSE, J. (1999). ICE Technology Unplugged. *Embedded Systems Programming*, 1: 103-109.
- HUMANES, D, LÓPEZ, J, ROBLES, I, SÁNCHEZ, C. (2004). *Sistemas Críticos*. Ingeniería Técnica en Informática de Gestión. Escuela Politécnica. Universidad de Alcalá de Henares. España.
- INGHAM, M, RASMUSSEN, R, MATTHEW, B, MONCADA, A. (2006). Generating requirements for complex embedded systems using State Analysis. *Acta Astronautica* 58: 648 - 661.
- KOVITZ, B. (2001). Is backtracking so bad? The role of learning in software development. *Proceedings of Fifth IEEE International Symposium on Requirements Engineering*, Toronto, Canada, pp. 272.
- LAVI, J, KUDISH, J. (2005). Systems modelling & requirements specification using ECSAM: an analysis method for embedded & computer-based systems". *Innovations Syst Softw Eng.* 1: 100-115.
- LEITER, E, LAMSWEERDE, A. (2002). Deriving Operational Software Specifications from System Goals. *Proceedings 10th ACM Symposium on the Foundations of Software Engineering*, Charleston.
- LIU, L, YU, E. (2001). From Requirements to Architectural Design: Using Goals and Scenarios. *Proceedings of the First International Workshop on From Software Requirements to Architectures*. Canadá.
- MARWEDEL, P. (2003). *Embedded system design*. Kluwer Academic Publishers University of Dortmund. Germany. 2003.
- POTTS, C. (1999). ScenIC: A Strategy for Inquiry-Driven Requirements Determination. *4th IEEE International Symposium on Requirements Engineering*, 4: 58-65.
- ROSS, D, SCHOMAN, K. (1977). Structured Analysis for Requirements Definition. *Transactions on Software Engineering, IEEE*, 3(1): 6-15.
- STANDISH GROUP., 1994. CHAOS Report. (1994). Disponible en: http://www.standishgroup.com/sample_research/updated.php
- STANDISH GROUP. CHAOS Report. (2002). Disponible <http://www.standishgroup.com/sample_research/updated.php> Recuperado el 7 de febrero de 2008.
- SUTCLIFFE, A. (2003). Scenario-Based Requirements Engineering. *11th IEEE International Requirements Engineering Conference (RE'03)* 11: 320.
- URREGO, G. (2005). ABC-Besoins : Une approche d'ingénierie de besoins fonctionnels et non-fonctionnels centrée sur les Agents, les Buts, et les Contextes. Tesis Doctoral, Universidad Paris 1, Pantéon Sorbona, Francia.
- ZAVE, P, JACKSON, M. (1997). Four dark corners of requirements engineering. *ACM Transactions on Software Engineering and Methodology*, 6(1): 1-30.