

Especificación de Requerimientos para el Desarrollo de Software Automotriz en México

Jorge R. Aguilar Cisneros

Decanato de Ingenierías

Universidad Popular Autónoma del Estado de Puebla

Puebla, Pue., México

jorge.aguilar@upaep.mx

Carlos Alberto Fernández-y-Fernández

Instituto de Computación

Universidad Tecnológica de la Mixteca

Huajuapán de León, Oaxaca, México

caff@mixteco.utm.mx

Resumen — Actualmente, dentro del sector automotriz, México es reconocido a nivel mundial, como un excelente país ensamblador, sin embargo, países como Alemania, Estados Unidos y Japón, entre otros, son reconocidos mundialmente además de ensambladores como países innovadores de éste sector. Bajo este escenario, se propone que México incursione en la investigación e innovación automotriz, particularmente, en el área de desarrollo de sistemas de software automotriz embebido tanto críticos como no críticos. Para llevar a cabo esta propuesta, en este artículo se presenta la situación del desarrollo de software automotriz en México, así como una revisión de los principales modelos de ciclo de vida, estándares y herramientas utilizadas en la construcción de sistemas de software automotriz embebido. De manera particular, presentamos la especificación de requerimientos para el sector automotriz, como un área inicial de atención.

Palabras clave— Especificación de Requerimientos, ISO 26262, Software automotriz.

I. INTRODUCCION

El sector automotriz en México tiene un impacto económico importante [10], ya que representa el 3.6% del PIB Nacional. Adicionalmente, México en 2012, se posicionó en 8vo lugar a nivel mundial como país ensamblador y ocupó el 3er lugar en América [1]. Las principales actividades económicas de éste sector son: Fabricación de automóviles y camiones, fabricación de carrocerías y remolques y fabricación de partes para vehículos automotores [2].

Al igual que en México, en los países que se distinguen por ser innovadores y productores de vehículos se encuentran: China ensambló, en 2012, más de 20 millones de autos, USA con cerca de 10 millones, les siguen, Japón, Alemania y Corea. De estos países, Alemania es considerado el país más innovador en este sector y fue el que, en 2011, hizo la mayor inversión en I+D (Investigación y Desarrollo), 19.6 billones de euros [9], por otro lado, E.U se encuentra en la 3era posición a nivel mundial al invertir cada año alrededor de 13 billones de USD en I+D [8] por su parte, Japón invirtió en 2012, alrededor de 18.7 billones de USD [7].

Las áreas de innovación que se abordan son, por ejemplo: Eficiencia de los vehículos, confiabilidad, vehículos híbridos, comunicación, electrónica, software, entre otras.

El software necesario para las innovaciones tecnológicas debe ser desarrollado tomando en consideración: Estándares, modelos de desarrollo y arquitecturas de software específicas para el sector. Algunos de los estándares requeridos para los autos es el ISO 26262 [11], con respecto al modelo de desarrollo de software automotriz se sugiere el modelo V [12] y, se desarrolló AUTOSAR [13] para definir una arquitectura de software estándar para el sector automotriz.

El ISO 26262 es un estándar internacional que se utiliza para el desarrollo de sistemas automotrices con un alto grado de seguridad y confiabilidad. Esta estándar tiene como objetivo garantizar la seguridad funcional de sistemas eléctricos y/o electrónicos de un vehículo. El modelo de desarrollo V, es una serie de pasos secuenciales en el que cada fase debe ser probada y validada antes de pasar a la siguiente fase, la última fase corresponde al desarrollo del software. AUTOSAR, es una arquitectura de software abierta y estandarizada cuyo objetivo es crear y establecer estándares abiertos para la arquitectura de componentes electrónicos en el sector automotriz. Esta lista de estándares modelos y arquitecturas es tan sólo una muestra reducida de las herramientas que se pueden utilizar para hacer frente a la innovación en el sector automotriz.

En este artículo, presentamos la especificación de requerimientos para sistemas automotrices, como un área inicial de atención por parte de los interesados en desarrollar sistemas de software para este importante sector comercial.

II. LA INDUSTRIA AUTOMOTRIZ EN MÉXICO.

En un estudio realizado por el Boston Consulting Group y publicado por la Secretaría de Economía [3], se identificó a la industria automotriz mexicana como un sector estratégico para el desarrollo económico del país debido, entre otros aspectos a que México en el 2012 se ubicó como uno de los principales productores a nivel mundial, en ese mismo año, este sector exportó el 23.5% del valor de las exportaciones nacionales totales, ubicándose, incluso, por arriba de las exportaciones petroleras. Con respecto a las remuneraciones que la industria automotriz otorga a sus empleados, en promedio, equivalen a 2.3 veces más que las del resto de las industrias manufactureras de México [4].

A. Investigación y desarrollo automotriz

En México, algunas compañías cuentan con centros de ingeniería y diseño, entre ellas están: Nissan, Ford, Chrysler, VW, General Motors y Delphi.

Las actividades que se desarrollan en estos centros de ingeniería son: El centro de modelado de Nissan Design America, desarrolla prototipos automotrices, Ford México es un centro de desarrollo del producto, entre otras actividades, lleva a cabo pruebas de temperatura. En el centro de investigación de Chrysler se estudian materias primas amigables con el medio ambiente. VW de México, lleva a cabo pruebas y liberación de sistemas electrónicos. El Centro de Ingeniería de General Motors diseña y desarrolla productos automotrices innovadores [3].

Sólo se presentan los centros de ingeniería y diseño que se reportan en el documento “Programa Estratégico de la Industria Automotriz 2012-2020”, aunque posiblemente existan más centros.

B. Innovación y Desarrollo de Software Automotriz

Hasta ahora, la única compañía instalada en México en la que se pudo corroborar que se desarrollan sistemas de software para el sector automotriz es la empresa Delphi México [5]. Posiblemente existan más, pero en esta investigación documental, no se encontró otra empresa o institución que desarrollara software automotriz.

C. Desarrollo de Software Automotriz en Universidades

Dada la importancia del sector automotriz mexicano, se llevó a cabo una investigación [6] sobre el número de instituciones de educación superior en nuestro país que formarían capital humano para este sector. Se encontró que en el área de Tecnología e Ingeniería, existen 2319 carreras profesionales (Técnico Superior Universitario, Licenciatura, posgrado) de éstas, en 155 carreras se menciona el sector automotriz, ver figura 1.



Fig. 1. Carreras de Ingeniería y Tecnología.

Sin embargo sólo 29 carreras de manera directa se enfocan al sector automotriz, ver figura 2, entre las carreras que se ofertan están: Mecánica Automotriz, Diseño Automotriz, Sistemas Automotrices, etc. y, de estas 29 carreras sólo existe una universidad, en todo el país, que dentro de sus programas de estudio incluye 1 materia llamada: Desarrollo de software automotriz.

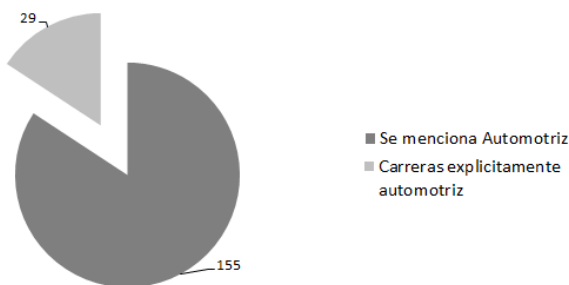


Fig. 2. Carreras del área automotriz.

Por los resultados obtenidos con esta investigación, pareciera que México, actualmente, no incursiona en el área de innovación en sistemas de software automotriz o no lo hace de una manera que le permita ser reconocido en este sector a nivel mundial. Uno de los factores que, deducimos, contribuye a este escenario es la casi nula preparación de recursos humanos en esta área.

En este sentido y dado que el desarrollo de software automotriz se identifica como un área de desarrollo tecnológico, encontramos un dato alentador, en el Programa Estratégico de la Industria Automotriz 1012-2020, en el apartado Estrategias y Líneas de Acción, se plantea la necesidad de Incrementar la Investigación y Desarrollo tecnológico, identificándose como áreas de acción: 1) Impulsar la especialización de universidades en el área de desarrollo

tecnológico automotriz, 2) Incrementar la interacción industria-academia y 3) Aumentar y mejorar los apoyos en becas para áreas de desarrollo tecnológico automotriz, en este sentido, se podrían buscar fondos federales para impulsar la investigación y desarrollo en el sector de software automotriz embebido de misión tanto crítica como no crítica.

D. Software Automotriz Embebido

La tasa de crecimiento del software automotriz embebido ha sufrido una aceleración en las últimas décadas. Esto lo podemos ver en los nuevos automóviles ya que tienen entre 20 y 70 ECUs (*Electronic Control Unit*) con más de 100 millones de líneas de código [28]. Por este motivo, gran parte de la funcionalidad que se agrega en autos modernos es gracias al software embebido, lo que conlleva a que entre el 35 y el 40% del valor comercial de un auto [29] se debe a este factor.

III. LA INDUSTRIA AUTOMOTRIZ EN EL MUNDO

Al igual que en México, la industria automotriz en otros países es un factor económico importante, en esta sección se mostrarán algunos datos de países como Alemania, Estados Unidos y Japón.

En Alemania, el sector automotriz es el sector económico más importante, en 2011, empleó a más de 712 mil personas. Alemania produce más del 17% de los autos que se venden en el mundo y es el país europeo con la mayor cantidad de plantas OEM (*Original Equipment Manufacturer*). Adicionalmente es un país líder en la investigación y desarrollo automotriz a nivel mundial [9].

Por su parte en Estados Unidos, ningún otro sector manufacturero genera tantos empleos como el sector automotriz, en 2014, empleó a más de 734 mil personas. USA ocupa el 3er lugar a nivel mundial en inversión para la investigación y el desarrollo automotriz. Tan sólo Chrysler, Ford y General Motors, juntos, invierten más de 13 billones de dólares en I + D, cada año. Por otro lado, se estima que en 2016 la venta anual de autos en USA será mayor a 16.5 millones de unidades, mientras que la producción, sólo en USA, alcanzará los 11.5 millones de vehículos construidos para ese mismo año [8].

En Japón, la industria automotriz e industrias relacionadas con ella, son las que más invierten en investigación y equipamiento, dando empleo al 8.7% de su población económicamente activa, lo que representa 5.47 millones de personas empleadas en estos sectores [7].

En esta sección no se incluyen datos de otros países que se distinguen en este sector como son China, Corea, India y Brasil, entre otros.

IV. HERRAMIENTAS DE DESARROLLO

En este apartado, usaremos la palabra herramienta para englobar estándares, modelos de ciclo de vida y arquitecturas. En las siguientes subsecciones describimos el ISO 26262, el Modelo V y la arquitectura AUTOSAR.

A. ISO 26262

El estándar de seguridad funcional ISO 26262 [14], publicado en 2011, está dirigido a vehículos con un peso máximo de 3.5 toneladas. Este estándar está conformado por un conjunto de procesos que se deben considerar con el fin de construir sistemas de misión crítica seguros (*safety critical systems*), ejemplos de estos tipos de sistemas son: El sistema de bolsas de aire, el sistema de frenado ABS, entre otros.

El objetivo de este estándar es reducir el riesgo de enfrentarse a peligros que son causados por un mal

comportamiento de los sistemas de seguridad críticos, los cuales, pueden ser eléctricos, electrónicos o programables.

El ISO 26262 se compone de 10 partes, ver figura 3, que se subdividen en cláusulas. Cada cláusula está integrada por requerimientos y cada requerimiento contiene un conjunto de sub-requerimientos.

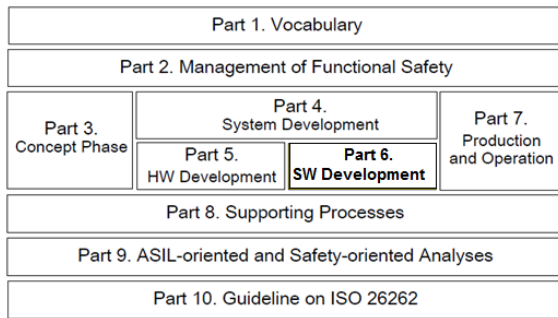


Fig. 3. Partes del ISO 26262.

Las partes que conforman el estándar ISO 26262 son: Vocabulario [11], Administración de la seguridad funcional [15], Fase conceptual [16], Desarrollo del producto: Nivel de Sistema [17], Desarrollo del producto: Nivel de Hardware [18], Desarrollo del producto: Nivel de Software [19], Producción y operación [20], Procesos de soporte [21], Orientación ASIL y Análisis Orientado a la seguridad [22] y Guías del ISO 26262 [23].

B. ISO 26262-6 Desarrollo de software automotriz

Para esta propuesta de innovación en el área de software, es de especial importancia la parte 6 [19] ya que ésta es la que de manera específica aborda los aspectos del desarrollo de sistemas de software para el sector automotriz. En este apartado se describen 7 fases que, con el fin de garantizar la seguridad y confiabilidad del sistema, se deben contemplar en el momento del desarrollo; ver tabla 1. A continuación se describen cada una de estas fases.

TABLA 1. ISO 26262-6: FASES EN EL DESARROLLO DE SOFTWARE.

Fase	Nombre de la fase
5	Inicio del desarrollo del software
6	Especificación de requerimientos de seguridad
7	Diseño de la arquitectura
8	Diseño e implementación
9	Pruebas unitarias
10	Pruebas de integración
11	Verificación de requerimientos de seguridad

Fase 5. Inicio del desarrollo del software. Durante esta fase, se seleccionan las herramientas y el proceso de desarrollo de software. Adicionalmente se crea un plan en el que se incluye, entre otras cosas, el modelo que se utilizará en el desarrollo del software así como los planes para verificar el desarrollo. También se define el lenguaje que se usará para el modelado y para la implementación del software y las herramientas que se utilizarán en las diferentes fases de desarrollo.

Fase 6. Especificación de requerimientos de seguridad. El objetivo principal de esta fase es definir todas las funciones que proporcionará el software y que potencialmente podrían impactar la seguridad del sistema ya sea por: a) Generar un estado inseguro, por ejemplo un módulo de software que controla la tracción en el proceso de frenado o, b) No tratar correctamente un fallo en hardware o software. Los requerimientos, como pueden ser: aspectos de tiempo de

respuesta para cada componente de software, interfaces entre componentes de software, etc, deben ser definidos.

Fase 7. Diseño de la arquitectura. Durante esta fase, se desarrolla un diseño de alto nivel para cada componente de software. La conformidad entre el diseño de la arquitectura y los requerimientos de seguridad debe ser verificada. Existen requerimientos específicos para el diseño de la arquitectura, por ejemplo: Uso de una notación apropiada, diseño modular, categorización de componentes, entre otros.

Fase 8. Diseño e implementación. En esta fase, cada subsistema se diseña e implementa. Los requerimientos para esta fase son: Definir notación de diseño, hacer una especificación unitaria, desarrollar una verificación unitaria. El Estándar recomienda algunos métodos para cumplir con los requerimientos de la fase, por cuestión de espacio estos métodos no se muestran en este artículo.

Fase 9. Pruebas unitarias. El objetivo de las pruebas unitarias es evaluar de manera individual la correcta operación de cada módulo de software. Para las pruebas unitarias, se deben cumplir los siguientes requerimientos: 1. Verificar componentes de seguridad, 2. Planificación y ejecución de pruebas, 3. Métodos de pruebas unitarias, 4. Generación de pruebas, 5. Métricas de pruebas. Las pruebas unitarias son útiles porque permiten identificar y corregir problemas en el código. Para llevar a cabo las pruebas unitarias, se recomienda el uso de métodos definidos en el estándar.

Fase 10. Pruebas de integración. Aunque en las pruebas unitarias se encontrarán la mayoría de los defectos de un sistema de software, es probable que surjan algunos defectos debido a la incorrecta comunicación entre los componentes del sistema. El objetivo de las pruebas de integración es demostrar que la integración de los diferentes componentes de software trabaja adecuadamente. Los métodos recomendados por el estándar para esta fase, son los mismos que para la fase de pruebas unitarias.

Fase 11. Verificación de requerimientos de seguridad. Esta fase se lleva a cabo con el fin de asegurar que el software embebido trabaja apropiadamente en el ambiente para el cual fue creado. Los requerimientos que deben ser cubiertos en esta fase son: 1. Planeación y ejecución de la verificación, 2. Ambientes de prueba, 3. Hardware objetivo, 4. Resultados de la evaluación.

C. Modelo V

Las fases de desarrollo de software definidas en el estándar internacional ISO 26262-6 se dividen en sub-fases que son mapeadas en el modelo V, ver figura 4.

En la fase de Especificación de Requerimientos de Seguridad del SW, se definen los requerimientos y las limitaciones funcionales que el sistema deberá cumplir. Los requisitos se obtienen del análisis de necesidades del usuario. El objetivo es generar un documento de especificación de requerimientos que se utilizará como entradas para la siguiente fase del modelo [24].

En la fase del diseño de la arquitectura se definen y describen los elementos, componentes, interfaces, interacciones, el hardware y el software. Esta fase implica la selección del sistema operativo y de la plataforma informática que será utilizada, adicionalmente, los componentes de software deben ser definidos para cumplir con los requisitos del usuario. El objetivo de esta fase es generar un documento de especificación de la arquitectura del sistema, que servirá como entrada para la fase del Diseño Unitario del Software y para la implementación.

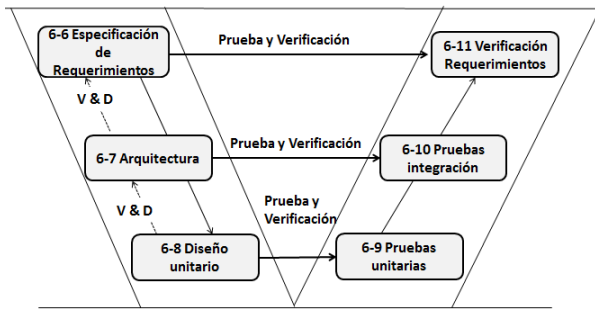


Fig. 4. El Modelo V [23].

En la fase del diseño unitario e implementación del software, se definen los principales bloques de software, las interfaces, las interacciones entre los módulos, los contenidos funcionales, los estados del sistema, las condiciones de error y los modos de diagnóstico. La salida de esta fase es un documento con el diseño de software, que es la base de la fase de codificación.

La fase de codificación tiene como objetivo la creación de los módulos. El sistema se desarrolla por primera vez en bloques pequeños denominados unidades que contienen sólo un aspecto funcional y se integran más tarde para formar el paquete completo de software.

En la fase de integración y verificación del software cada unidad se prueba de manera independiente. La verificación consiste en realizar pruebas funcionales a las interfaces de los módulos. Las pruebas más detalladas toman en cuenta la estructura interna de los módulos de software. Durante la integración, las unidades se unen para generar el sistema completo. Después de la integración, el sistema debe ser probado para verificar que los requerimientos del cliente se cumplen.

D. AUTOSAR

La Arquitectura de Sistemas Abierta para el sector Automotriz (AUTOSAR) [13] es una arquitectura de software abierta y estandarizada desarrollada por un grupo de proveedores, fabricantes de automóviles, y desarrolladores de herramientas de software.

El objetivo es crear y establecer estándares abiertos para el desarrollo de la arquitectura de los componentes electrónicos en el sector automotriz. Esto incluye la estandarización de funciones básicas de los sistemas de software, la portabilidad en diferentes plataformas, la portabilidad para la interconexión en red, la integración de múltiples proveedores, el mantenimiento al sistema y el soporte a nuevas versiones del software a lo largo del ciclo de vida del vehículo.

Con la metodología de AUTOSAR se puede desarrollar un sistema, para ello, se deben considerar los siguientes pasos [25]:

1. Configuración del sistema. En este paso se desarrolla el diseño de la arquitectura. La principal acción en este paso es mapear los componentes de software con su ECU correspondiente.
2. Identificar información de las ECUs. En éste paso se obtiene información de cada ECU y se genera un documento con las especificaciones.
3. Configuración de ECUs. Se genera un documento con la descripción de la configuración de las ECUs.
4. Genera archivo ejecutable. Los archivos se compilan y se ligan para generar los ejecutables.

Actualmente, el software ha adquirido un rol fundamental en la construcción de sistemas eléctricos automotrices. Aproximadamente, el 90% de la innovación en las ECUs están directa o indirectamente relacionadas con tecnologías de software [37]. Por lo que se genera la necesidad de desarrollar especificaciones de requerimientos para software automotriz. Sin embargo, existe una dificultad en ésta tarea, que se debe, principalmente, a 3 aspectos: complejidad de los datos, complejidad del comportamiento y complejidad del ambiente. Ante este hecho, a lo largo de los años se han propuesto diferentes métodos para el análisis y especificación de requerimientos de software para el sector automotriz.

Una de estas propuestas sugiere la formalización de requerimientos de software automotriz mediante modelos de pruebas [38] que son ejecutados mediante simulaciones model-in-the-loop y hardware-in-the-loop. Los autores proponen el Modelo de Uso Temporizado (*Timed Usage Model TUM*), que utilizan como una forma de especificar requerimientos formales los cuales, sirven de base para todas las actividades de pruebas. Este modelo proporciona la posibilidad de describir la sincronización y la dependencia de datos de los sistemas que serán probados. El modelo también permite la generación y planeación de los casos de pruebas. El Modelo TUM está basado en las cadenas de Markov (*Markov Chain Usage Models, MCUM*). El modelo TUM consiste de (sólo se muestran algunas definiciones):

- Un conjunto de estados $S = \{S_1, \dots, S_n\}$ que representan posibles estados de uso.
- Un conjunto de Arcos A , que representan transiciones.

Un arco del estado S_i a el estado S_j se denota por A_{ij}

Un ejemplo de un TUM, se presenta en la figura 5. Las rutas que van del estado inicial S_1 , al estado final S_4 , son casos de pruebas válidos.

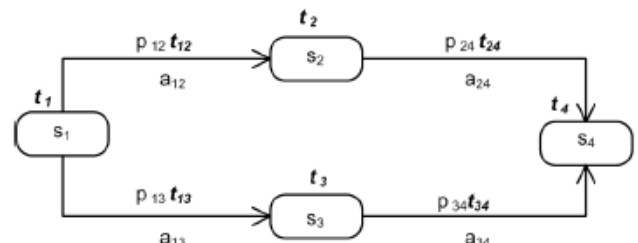


Fig 5. Ejemplo de TUM.

Para genera una muestra de casos de prueba, se puede utilizar un algoritmo determinístico. Los requerimientos del sistema que son escritos en lenguaje natural, se almacenan en un sistema de administración de requerimientos, posteriormente, son transformados a una representación TUM, que sirve de base para generar el conjunto de casos de prueba. La metodología para transformar requerimiento escritos en lenguaje natural a un TUM se muestra en la figura 6.

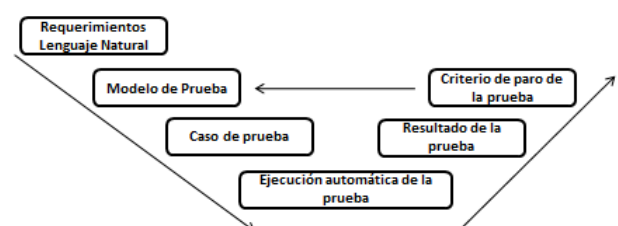


Fig. 6. Metodología de transformación de requerimientos a un TUM.

Existe una propuesta para transformar requerimientos funcionales del sistema E/E (*Electrical and Electronic Systems*) de automóviles, que están escritos en lenguaje natural en una notación semi-formal, usando una propuesta conocida como Patrones de Especificación de Tiempo Real (*Real Time Specification Patterns, RTSP*) [39]. Estos patrones se basan en el Sistema de Patrones de Especificación (*Specification Pattern System, SPS*). Algunos de los patrones utilizados en esta propuesta se muestran en la tabla 2.

Tabla 2. PATRONES RESTRINGIDOS DE LA LENGUA INGLESA.

Ámbito	scope	Globally
Cualitativo	universalityPattern	It is always the case that P holds
	responsePattern	, then S eventually holds.
	realtimeOrderCategory	If P holds, then S holds (boundedResponsePattern boundedInvariancePattern)
Tiempo Real	boundedResponsePattern	At least every c time unit(s)
	maxDurationPattern	Less than c time unit(s)

Los patrones se presentan en lengua inglesa.

Un ejemplo de un requerimiento en lenguaje natural es: “La señal *lowFuelLevelWarning* deberá pasar a su estado Activo, cuando la entrada *totalFuelLevel* esté por debajo del nivel predefinido”, cuando transformamos este requerimiento a una notación formal queda de la siguiente manera: “Si $totalFuelLevel \leq 10\%$, entonces, output *lowFuelWarning* deberá ser puesta a Activo”. Entonces, para este caso, utilizaríamos el patrón Response, tendríamos: “Globally, It is always the case that if ($totalFuelLevel \leq 10\%$) holds, then ($lowFuelWarning = \text{Activo}$) eventually holds”. (El requerimiento está escrito en inglés debido a que los patrones están definidos para ese idioma). La metodología para la transformación de requerimientos, se muestra en la figura 7.

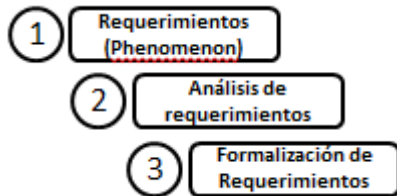


Fig. 7. Metodología de transformación de requerimientos a un TUM.

Otro trabajo presenta una propuesta para mejorar la especificación de requerimientos para software automotriz. La cual consiste en la integración de dimensiones conceptuales de calidad dentro del proceso de ingeniería de requerimientos (*Requirements Engineering Process, REP*) [40]. La calidad de la especificación de requerimientos puede ser definida y medida mediante varios criterios de calidad como son: estructura, completitud, correcto, consistencia, acuerdo, formalización y concretización. En esta propuesta sólo se toman en cuenta estos 2 últimos criterios, para desarrollar un framework. El cual transforma requerimientos incompletos e informales a una especificación precisa. Un diagrama de concretización de requerimientos se muestra en la figura 8.

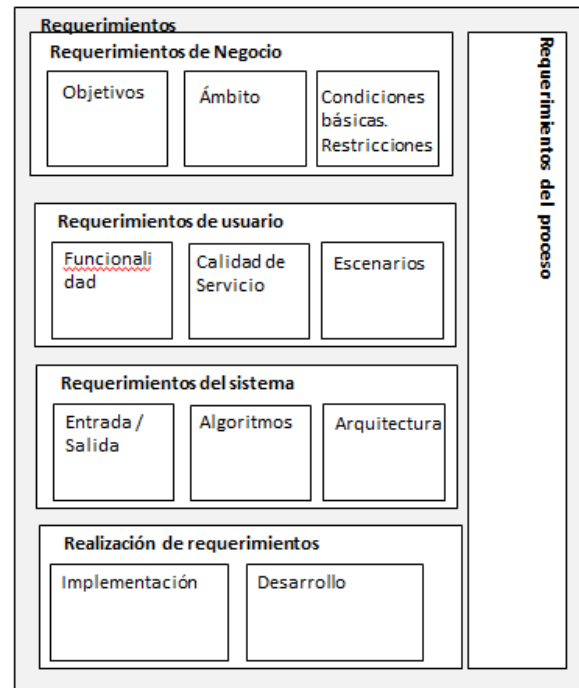


Fig. 8. Concretización (tipos) de requerimientos.

El framework de transformación de requerimientos, consiste de los siguientes pasos, ver figura 9.

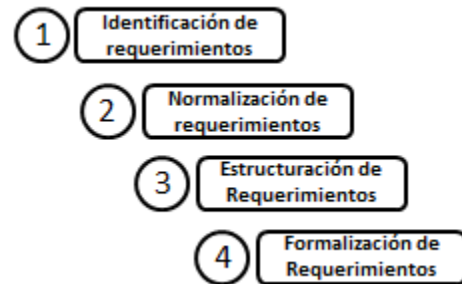


Fig. 9. Dimensiones conceptuales para la transformación de requerimientos.

En el primer paso se tienen requerimientos informales que se separan en requerimientos atómicos. En el segundo paso se generan requerimientos homogéneos, usan la misma terminología. En el tercer paso se lleva a cabo una agrupación de requerimientos. Finalmente, en el paso 4, se lleva a cabo la transformación a requerimientos formales, se usa el diagrama de estructura de sistemas llamado AUTOFOCUS. Para la formalización del comportamiento se utilizan diagramas de transición de estados. Para la formalización de la interacción, se utilizan patrones de interacción. Para la formalización de datos, se utiliza Tipos de definición de datos. Por ejemplo, el requerimiento: “Cuando el usuario frene su vehículo, el piloto automático se desactiva”, se transforma en el requerimiento semi-formal, ver figura 10.

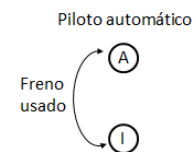


Fig. 10. Representación semi-formal de un requerimiento.

La especificación de requerimientos de seguridad (*Safety*) para vehículos terrestres, se puede hacer de 3 formas, ver tabla 3: notación informal, notación semi-formal y notación formal, dependiendo del nivel de integridad de seguridad asignado al de sistema que se desarrollará. El Nivel de Integridad Segura

del Automóvil (*ASIL, Automotive Safety Integrity Level*) [27], es una medida del riesgo que va desde la A, hasta la D. Donde, el valor A corresponde a una cantidad de riesgo mínimo y el D a una cantidad máxima.

TABLA 3. ESPECIFICACIÓN DE REQUERIMIENTOS DE SEGURIDAD. [6]
+ RECOMENDADO; ++ FUERTEMENTE RECOMENDADOS

Método de especificación de requerimientos	ASIL			
	A	B	C	D
1a Notación informal	++	++	+	+
1b Notación semi-formal	+	+	++	++
1c Notación formal	+	+	+	+

El estándar 26262, recomienda que la especificación de requerimientos se haga mediante una combinación de lenguaje natural, - algunos autores consideran al lenguaje natural una notación informal [41], y alguno de los métodos mostrados en la tabla 3. Para requerimientos de seguridad de alto nivel (ejemplo: Requerimientos de seguridad técnica y funcional), el lenguaje natural es el más apropiado. Mientras que para requerimientos de seguridad de bajo nivel (ejemplo: requerimientos de seguridad de software y hardware), notaciones mostradas en la tabla 3 son las más apropiadas.

El principal problema de escribir requerimientos en lenguaje natural es su ambigüedad, por lo que las alternativas que existen son: La notación informal, la notación semi-formal y la notación formal [42]. En la notación informal, los requerimientos se representan mediante diagramas de entrada/salida, la técnica se conoce como Análisis Estructurado (*Structured Analysis SA*). Incluso, existe un método llamado SADT [43] Adicionalmente, para notación informal, se puede utilizar gramáticas libres de contexto [44]. Para la notación semi-formal se puede utilizar UML[42], SysML, URN (Notación de Requerimientos de Usuario), entre otros. Para la notación formal, se puede utilizar la notación Z [45], KAOS que es un lenguaje formal orientado a objetivos, etc.

Por lo que podemos concluir, que la especificación de requerimientos para el sector automotriz se deberá hacer mediante una combinación de lenguaje natural, notación informal y notación semi-formal. La especificación de requerimientos formales para vehículos terrestres, de acuerdo al ISO 26262, no es obligatoria, sólo es recomendada [29].

VI. ÁREA DE INVESTIGACIÓN

Algunas de las áreas de investigación y desarrollo en el sector automotriz son [26]: Seguridad, Confort, carros autónomos, Sistemas: Electrónicos, eléctricos, neumáticos, mecatrónicos, mecánicos, frenos, química de materiales, diseño de vehículos, etc.

Si tomamos en consideración que actualmente, la innovación está directamente relacionada con el Software [27], podremos ver que existe una importante área de oportunidad en éste sector para la ingeniería de software.

A. Desarrollo de Software Automotriz en UPAEP

En UPAEP (Universidad Popular Autónoma del Estado de Puebla), se lleva a cabo un proyecto de investigación sobre el desarrollo de software automotriz en México, para ello, se estudia con detalle algunas herramientas que se usan en este sector y se promueve la importancia de la innovación automotriz en México.

Actualmente se desarrollan propuestas de sistemas automotrices en 2 áreas: Sistemas de ventanas y seguros eléctricos. En particular, se está bosquejando el comportamiento que estos sistemas deben presentar en situaciones de riesgo para los pasajeros. Para esta

investigación, una situación de riesgo se presenta cuando el conductor o los pasajeros necesitan recibir ayuda externa, y ellos, por sí mismos, no pueden bajar las ventanas o quitar los seguros, es en ese momento en que los sistemas mencionados deberán actuar de manera automática para que puedan recibir auxilio de una manera expedita y libre de riesgos.

Dada la naturaleza del proyecto, se está trabajando con un equipo multidisciplinario que está conformado por personas de áreas como: Ingeniería de Software, Ingeniería en Computación, Ingeniería en Diseño Automotriz e Ingeniería Mecatrónica.

Para el desarrollo del proyecto, inicialmente, nos hemos enfocado en la especificación semi-formal de requerimientos que es el trabajo que se presenta a continuación.

B. Propuesta para la especificación de requerimientos automotrices.

En el proyecto que se ha estado trabajando en el área de desarrollo de software automotriz, se utiliza la metodología mostrada en la figura 11. Para definirla, se tomó como base las recomendaciones del estándar ISO 26262-9. En este apartado del estándar se define el tipo de notación recomendada, dependiendo del nivel de seguridad de los sistemas automotrices que se desarrollarán.

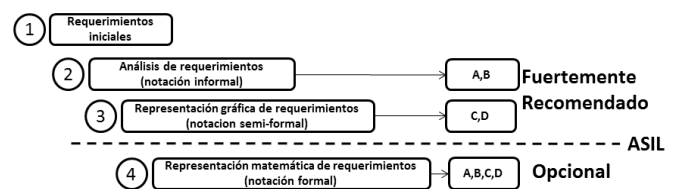


Fig. 11. Metodología propuesta en este artículo.

La metodología propuesta, se conforma de cuatro pasos que son descritos a continuación.

En el primer paso. Se obtienen los requerimientos iniciales del cliente y se escriben en lenguaje natural. Por ejemplo: “Necesitamos un sistema que se active al detectar una situación de riesgo para el conductor y/o sus acompañantes”. Como se puede observar, este requerimiento es muy general.

En este primer paso se lleva a cabo la obtención-escritura de requerimientos. La escritura se hace en lenguaje natural, ver figura 12. Este paso se lleva a cabo independientemente del nivel de seguridad del sistema a desarrollar, ver figura 11.

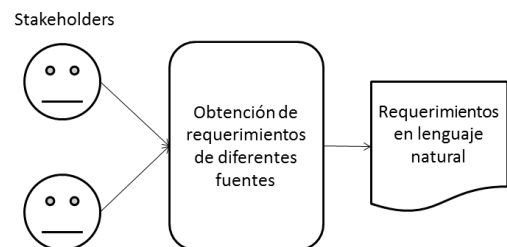


Fig. 12. Requerimientos iniciales

En el segundo paso, los requerimientos son analizados, reescritos y clasificados (requerimientos de negocio, funcionales, no funcionales, de sistema). Para la escritura de requerimientos, se deben tomar en cuenta, entre otras, las siguientes reglas [46]: Usar voz activa, no usar conectores lógicos para unir más de un requerimiento, evitar usar “etcétera”, utilizar “deberá” o “debe” para indicar la obligatoriedad del requerimiento, evitar términos que

impliquen posibilidad, por ejemplo: “puede”, “podrá”, “podría”, entre otros, ver figura 13. Lo que se busca en esta fase es obtener una representación de requerimientos que tome en consideración los atributos de calidad que deben estar presentes en su especificación. La escritura se realiza también, en lenguaje natural pero siguiendo cierta sintaxis, ejemplo: “El sistema de auxilio al conductor deberá activarse cuando la entrada *setWarning* sea igual a 1”. Este requerimiento se obtuvo después de hacer una clasificación de los requerimientos obtenidos en el paso 1 y después de aplicar las reglas de sintaxis para su escritura. El requerimiento anterior se clasificó como un requerimiento de sistema.

En este paso se genera la notación informal que el estándar 26262 recomienda fuertemente para cuando se desarrollan sistemas automotrices que se clasifican como A ó B.

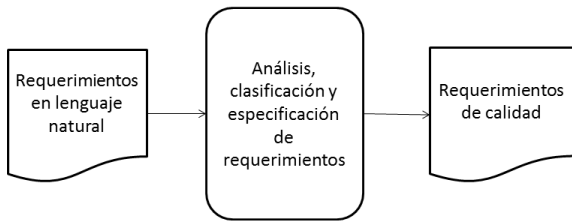


Fig. 13. Análisis de requerimientos

En el tercer paso, ver figura 14, los requerimientos son reescritos mediante diagramas lógicos [47], en los que se representan transiciones de estados, líneas de tiempo, etc. Para llevar a cabo esta tarea, en el proyecto se utilizó UML y SysML. Estas herramientas nos permitieron desarrollar diagramas de secuencia, diagramas de casos de usos etc., ver figura 16. El diagrama de estado de la figura 16, corresponde a la representación gráfica del requerimiento de sistema mostrado en el segundo paso de nuestra metodología. En este paso se genera la notación semi-formal que el estándar ISO 26262 recomienda fuertemente cuando se desarrollan sistemas automotrices que se clasifican como C ó D.

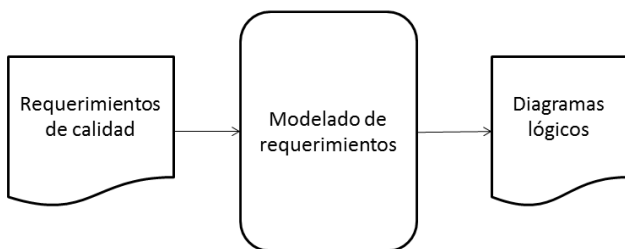


Fig. 14. Modelado de requerimientos.

En el cuarto paso, ver figura 15, los requerimientos semi-formales son transformados a una notación formal. En nuestra metodología, este paso se define como opcional, debido a que de acuerdo al estándar ISO 26262-9 [28], de dice que la escritura de requerimientos en notación formal, no se recomienda fuertemente para sistemas automotrices, por lo que los ingenieros de requerimientos pueden o no usar la notación formal. Independientemente, si se desea expresar requerimientos en notación formal, se podría utilizar, por ejemplo, el lenguaje Z. Ver figura 17. El esquema de la figura 17, representa el requerimiento de sistema que se escribió en el segundo paso de nuestra metodología.

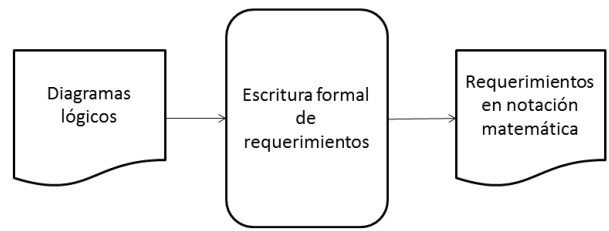


Fig. 15. Escritura formal de requerimientos.

La aplicación de nuestra metodología ha guiado la obtención y transformación de los requerimientos de los sistemas automotrices que se están desarrollando. Esta transformación va desde la obtención de requerimientos hasta su representación semi-formal.

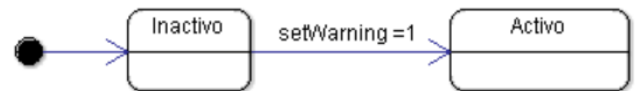


Fig. 16. Diagrama de estados del sistema de auxilio al conductor.

Los sistemas que ahora se están desarrollando no se encuentran clasificados en ningún nivel de ASIL, debido a que no son considerados de misión crítica, sin embargo, se están desarrollando como si estuvieran clasificados en el nivel C de ASIL, esto, con el fin de adquirir experiencia en la especificación semi-formal de los requerimientos automotrices.

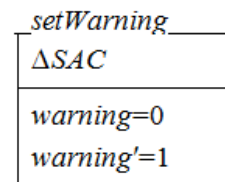


Fig. 17. Notación Formal de requerimientos

VII. CONCLUSIONES

México se encuentra dentro de los principales países manufactureros de la industria automotriz, esto es un indicativo del excelente talento humano y condiciones sociales existentes en el país, por lo que se puede inferir que existen las condiciones necesarias para que nuestro país se pueda posicionar entre los mejores países en el área de innovación automotriz. Otro factor que puede apoyar en este sentido es el Programa Estratégico de la Industria automotriz 2012-20120 que presenta dentro de sus estrategias y líneas de acción: Incrementar la investigación y desarrollo tecnológico dentro del sector automotriz.

Para que México aspire a ser referente en innovación automotriz, se requiere la formación de recursos humanos profesionales en áreas no tradicionales de la industria automotriz mexicana. Estas áreas son, sólo por nombrar algunas: desarrollo de sistemas automotrices de misión crítica y no crítica, estudio y aplicación de estándares automotrices como el ISO26262, Desarrollo de aplicaciones que tomen en consideración AUTOSAR, Formación de programadores especialistas en desarrollo de software embebido, Arquitectos de software, etc. Para lograrlo es necesario incrementar la participación y colaboración entre la industria, la academia y el gobierno.

La especificación de requerimientos es una etapa crucial, que determina el éxito o fracaso de los sistemas de software y para el caso del software automotriz es un área fundamental, sobre todo cuando se desarrollan sistemas en los que la

seguridad es fundamental (sistemas de misión crítica). En este trabajo, presentamos una propuesta inicial para transformar requerimientos expresados en lenguaje natural a una representación semi-formal. Se debe resaltar que la notación formal de requerimientos no es fuertemente recomendada, por el ISO 26262-9, en el desarrollo de sistemas para el sector automotriz, por lo que su uso es opcional.

Los proyectos que desarrollamos actualmente en UPAEP están orientados a incursionar de manera gradual en el desarrollo de sistemas embebidos automotrices por lo que hemos iniciado con sistemas de misión no crítica. Para incursionar en sistemas de misión crítica necesitamos incrementar nuestra experiencia en el área.

VIII. TRABAJO A FUTURO

Para incrementar nuestra experiencia en este sector y contribuir a la propuesta presentada en este artículo, hemos planeado las siguientes acciones: a) Acondicionar un laboratorio de desarrollo de software automotriz, en el que se llevarán a cabo proyectos que permitan mostrar que en México existe potencial para incursionar en el área, b) Identificar un nicho dentro del desarrollo de software automotriz en la que se desarrollarán, principalmente, los proyectos del grupo. c) Incrementar la experiencia del grupo mediante la participación o impartición de cursos, seminarios, talleres, conferencias, diplomados etc., d) Buscar relaciones con empresas, instituciones y el gobierno para sumar esfuerzos en el área de investigación y desarrollo tecnológico del sector automotriz. y finalmente, e) Probar nuestra metodología de especificación de requerimientos para sistemas automotrices.

REFERENCIAS

- [1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," *Phil. Trans. Roy. Soc. London*, vol. A247, pp. 529–551, April 1955. (*references*)
- [2] J. Clerk Maxwell, *A Treatise on Electricity and Magnetism*, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
- [3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in *Magnetism*, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
- [4] K. Elissa, "Title of paper if known," unpublished.
- [5] R. Nicole, "Title of paper with only first word capitalized," *J. Name Stand. Abbrev.*, in press.
- [6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," *IEEE Transl. J. Magn. Japan*, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
- [7] M. Young, *The Technical Writer's Handbook*. Mill Valley, CA: University Science, 1989.
- [8] E. J. Solís, "La Industria Automotriz: Retos y Oportunidades", XXV Simposium IMEF, Cd. de México, Junio 2013.
- [9] "La Industria Automotriz en México 2014", Instituto Nacional de Estadística y Geografía: INEGI, 2014.
- [10] "Programa Estratégico de la Industria Automotriz 2012-2020", Subsecretaría de Industria y Comercio, Dirección General de Industrias Pesadas y de Alta Tecnología, Secretaría de Economía, 2012.
- [11] "Encuesta mensual de la Industria Manufacturera", Instituto Nacional de Estadística y Geografía: INEGI, 2012.
- [12] E. Wiebe-Quintana, and S. Canales, "Making Power Windows Safe", *dSPACE Magazine*, 2010. Pp. 34-17.
- [13] "The Motor Industry of Japan 2014", Japan Automobile Manufacturers Association, Inc. 2014.
- [14] "State of the U.S Automotive Industry: Investment, Innovation, Jobs and America's Economic Competitiveness", American Automotive Policy Council, June 2014.
- [15] "The Automotive Industry in Germany", Germany Trade & Invest, 2013.
- [16] A. C. Valdenebro, "Explosión de la Industria Automotriz en México: De sus Encadenamientos Actuales a su Potencial Transformador", Friedrich-Ebert-Stiftung, Marzo 2014.
- [17] "ISO 26262-1:2011 Road Vehicles –Functional Safety – Part 1: Vocabulary". The International Organization for Standardization, 2011.
- [18] N. M. A. Munassar, and A. Govardhan, "A Comparison between Five Model of Software Engineering", *IJSI International Journal of Computer Science Issues*, Vol. 7, Issue 5, September 2010.
- [19] "AUTOSAR – AUTomotive Open System ARchitecture", www.autosar.org, Última visita enero 2015.
- [20] "ISO 26262, Functional safety for road vehicles", International Standard, November 2011.
- [21] "ISO 26262-2:2011 Road Vehicles –Functional Safety – Part 2: Management of functional safety". The International Organization for Standardization, 2011.
- [22] "ISO 26262-3:2011 Road Vehicles –Functional Safety – Part 3: Concept phase". The International Organization for Standardization, 2011.
- [23] "ISO 26262-4:2011 Road Vehicles –Functional Safety – Part 4: Product Development at the System Level". The International Organization for Standardization, 2011.
- [24] "ISO 26262-5:2011 Road Vehicles –Functional Safety – Part 5: Product Development at the Hardware Level". The International Organization for Standardization, 2011.
- [25] "ISO 26262-6:2011 Road Vehicles –Functional Safety – Part 6: Product Development at the Software Level". The International Organization for Standardization, 2011.
- [26] "ISO 26262-7:2011 Road Vehicles –Functional Safety – Part 7: Production and operation". The International Organization for Standardization, 2011.
- [27] "ISO 26262-8:2011 Road Vehicles –Functional Safety – Part 8: Supporting processes". The International Organization for Standardization, 2011.
- [28] "ISO 26262-9:2011 Road Vehicles –Functional Safety – Part 9: Automotive Safety Integrity Level (ASIL)–Oriented and Safety–Oriented Analyses". The International Organization for Standardization, 2011.
- [29] "ISO 26262-10:2011 Road Vehicles –Functional Safety – Part 10: Guideline on ISO 26262". The International Organization for Standardization, 2011.
- [30] K. Wolfgang, "An Integrated System Development Process including Hardware and Logistics based on a Standard Software Process Model". RTO IST Symposium, Technology for Evolutionary Software Development, 2002.
- [31] N. Li, H. Li, J. Wu, X. Zhong, Z. Sun, and W. Bao, "AUTOSAR Based Automatic GUI Generation", 13th IEEE International Symposium on Object/Component/Service –Oriented Real-Time Distributed Computing, IEEE, Carmona, Seville, 5-6 Mayo 2010, pp. 156-162.
- [32] "Innovation Trends and Challenges and Cooperation Possibilities with R&D in Automotive Industry", Automotive Network for Innovation: ACSEE, Slovakia, 2010.
- [33] E. Bringmann, and A. Krämer, "Model-based Testing of Automotive Systems", 1st International Conference on Software Testing, Verification, and Validation, IEEE, Lilehammer, November 2008, pp. 485-493.
- [34] E. Christof, and J. Capers, "Embedded Software: Facts, Figures, and Future", IEEE Computer Society, april 2009, pp. 42-52.

- [35] R. N. Charette, "This car run on code", IEEE Spectrum, February 2010.
- [36] E. Broy, I. Kruger, A. Pretschner, y, Salzman, C., "Engineering Automotive Software". Proceedings of The IEEE. 95(2): pp 356-373. 2007.
- [37] Siegel, S., Hielscher, K., and German, R., (2011). Formal Specification and Systematic Model-Driven Testing of Embedded Automotive Systems.
- [38] Filipovikj, P., Nyberg. M., and Rodriguez-Nava, G., (2014). Reassessing the Pattern-Based Approach for Formalizing Requirements in the Automotive Domain, IEEE.
- [39] Fleischmann, A., Hartmann, J., Pfaller, C., Rappl, M., Rittmann, S., and Wild, D. () Concretization and Formalization of Requirements for Automotive Embedded Software Systems Development.
- [40] Johannisson, K., () Formal and Informal Software Specification. Doctoral of Philosophy Thesis.
- [41] Sládeková, V., (2007). Methods used for requirements engineering.
- [42] Ross, D, T., and Schoman, K.E., (1977) Structured Analysis for Requirements Definition. IEEE Transaction on Software Engineering, Vol. SE-3, No. 1.
- [43] Scott, W., Cook, S., and Kasser, J., (2004) Development and Application of a Context-free Grammar for Requirements. The systems engineering Test and Evaluation (SETE) Conference.
- [44] Spivey, J. M., (1998) The Z Notation: A Reference Manual.
- [45] Turk, W., "Mission Possible...With Good Requirements". Defense AT&L: Septiembre-octubre 2005.
- [46] Gómez, C. C.E., Modeling Embedded Systems Using SysML. Master degree Thesis. 2009.



Jorge R. Aguilar Cisneros recibió el grado de Doctor en Ingeniería de Software por parte de la Universidad Popular Autónoma del Estado de Puebla (UPAEP), en Puebla, México, en el 2009. Actualmente es profesor-investigador en la misma universidad y está certificado, desde 2012, como SCRUM Master por parte de la Agile Alliance y desde 2007, como Desarrollador PSP (Personal

Software Process) por parte del Software Engineering Institute. Sus áreas de investigación son adopción de procesos de desarrollo de software en micro y pequeñas empresas y desarrollo de software para el sector automotriz. Actualmente es miembro de la IEEE y fue miembro del the International Council on Systems Engineering (INCOSE). También pertenece a la Red Temática Mexicana de Ingeniería de Software (RedMIS).



Carlos Alberto Fernández y Fernández es egresado de la Facultad de Informática de la Universidad Veracruzana. Más tarde realizó la Maestría en Ciencias de la Computación en la Fundación Arturo Rosenblueth. Recibió el grado de Doctor en Ciencias de la Computación en la Universidad de Sheffield, Inglaterra. Se encuentra adscrito al Instituto de Computación de la

Universidad Tecnológica de la Mixteca, desempeñándose como profesor e investigador de tiempo completo. Ha sido coordinador de la Universidad Virtual y de la Maestría en Computación con especialidad en Sistemas Distribuidos. Recientemente finalizó una estancia sabática en el Laboratorio Nacional de Informática Avanzada. Trabaja dentro del área de Ingeniería de Software, particularmente en las líneas de modelado visual, métodos de desarrollo y especificación formal de software. Ha sido responsable del Cuerpo Académico de Ingeniería de Software en la UTM y miembro del Verification and Testing Research Group en la Universidad de Sheffield. En la actualidad es miembro de los núcleos académicos de los estudios de posgrado de Maestría y Doctorado en Tecnologías de Cómputo Aplicado y de la Maestría en Computación con especialidad en Sistemas Distribuidos.