

Estímulos a la Investigación, Desarrollo Tecnológico e Innovación 2014

**Ingeniería Dirigida por Modelos para el Desarrollo de Software Auto-Adaptativo
en el Mundo Inteligente**

Dr. Germán Harvey Alférez Salinas

**Universidad de Montemorelos
Montemorelos, N.L., México
Junio 5 de 2014**

Tabla de Contenidos

| | | |
|-------|---|------|
| 1 | Generalidades | - 1 |
| 2 | Responsabilidades | - 1 |
| 3 | Justificación del Proyecto | - 2 |
| 3.1 | Motivación | - 2 |
| 3.2 | Objetivo del Proyecto | - 2 |
| 3.3 | Resultados Esperados | - 4 |
| 3.4 | Beneficios | - 5 |
| 4 | Análisis de Factibilidad del Proyecto | - 6 |
| 4.1 | Antecedentes | - 6 |
| 4.2 | Análisis del Entorno del Proyecto | - 7 |
| 4.3 | Estado del Arte | - 8 |
| 4.4 | Determinación de Recursos | - 10 |
| 4.4.1 | Recursos Humanos | - 10 |
| 4.4.2 | Recursos Financieros | - 11 |
| 4.4.3 | Recursos Técnicos | - 11 |
| 4.4.4 | Recursos Materiales | - 11 |
| 4.5 | Aportación del Proyecto | - 12 |
| 4.5.1 | Avances en la Ciencia y/o la Tecnología que se Prevén Alcanzar | -12 |
| 4.5.2 | Beneficios e Impacto Social, Económico y Ambiental en la Región, el País y el Mundo | - 14 |
| 4.5.3 | Productos de Investigación Esperados: Artículos, Libros, Tesis, Patentes, etc. | - 14 |
| 5 | Plan Detallado del Proyecto | - 15 |
| 5.1 | Generalidades | - 15 |
| 5.2 | Planificación de la Secuencia del Proyecto | - 15 |
| 5.3 | Estructura Organizacional y Personal Participante | - 17 |
| 6 | Presupuesto | - 17 |
| 7 | Control del Programa de Trabajo del Proyecto | - 19 |
| 7.1 | Identificación de Riesgos y Puntos Críticos | - 19 |
| 7.2 | Gestión de Cambios | - 19 |
| 7.3 | Medición de Avances | - 19 |
| 8 | Evaluación y Cierre del Proyecto | - 19 |
| 9 | Bibliografía | - 19 |

1. Generalidades

En un mundo abierto cada vez más dinámico, inteligente y descentralizado, las tecnologías y los tipos de aplicaciones evolucionan hacia la creación de eco-sistemas compuestos por una amplia variedad de dispositivos y servicios heterogéneos y distribuidos, de naturaleza claramente móvil y ubicua, y en constante evolución tecnológica. Gracias a la infraestructura que proporciona la Internet del Futuro (IF), a la Computación en la Nube, al avance de los dispositivos móviles (teléfonos inteligentes, tabletas, etc.) y los 'wearables' (gafas, relojes inteligentes, sensores en la ropa, etc.), se están desarrollando nuevas aplicaciones y servicios sofisticados con alto grado de volatilidad a través de la cooperación entre dispositivos (u objetos) interconectados, no conocidos en tiempo de diseño, para una sociedad de usuarios cuyas necesidades evolucionan y cambian rápidamente.

Ante esta situación, surge la necesidad de desarrollar sistemas que sean capaces de adaptarse de forma continua en tiempo de ejecución a nuevas condiciones del entorno, situaciones impredecibles, nuevos dispositivos y tecnologías con los/las que interactuar o nuevos servicios que consumir. Esta adaptación debe gestionarse de forma autónoma ya que no parece viable asumir que los desarrolladores intervengan activamente en materializar todas las adaptaciones en tiempo de ejecución.

El objetivo principal de este proyecto es proporcionar métodos, técnicas y herramientas que permitan el desarrollo sistemático y productivo de software auto-adaptativo para el ámbito de la IF, donde múltiples dispositivos y servicios heterogéneos deben adaptarse para alcanzar una interacción coordinada y proporcionar a los usuarios servicios globalizados que se ajusten a sus necesidades. Este tipo de sistemas plantea nuevos retos tecnológicos que no pueden afrontarse mediante procedimientos tradicionales. Este proyecto persigue proporcionar nuevas soluciones desde el ámbito de la Ingeniería Dirigida por Modelos (MDE) y los Modelos en Tiempo de Ejecución (M@RT). En concreto, se proporcionarán soluciones que permitan abordar la evolución de estos eco-sistemas en un contexto de incertidumbre en cuanto a servicios, dispositivos y cambios en el contexto. Además, se ofrecerá soporte metodológico al proceso de desarrollo de este tipo de sistemas.

Finalmente, puesto que el presente proyecto se enmarca dentro de un contexto ingenieril, los resultados del proyecto se validarán a través del desarrollo de prototipos, infraestructuras software y casos de laboratorio, además de la experimentación en entornos académicos e industriales.

2. Responsabilidades

La Facultad de Ingeniería y Tecnología (FIT) ha designado a las siguientes personas para fungir como coordinadores principales para llevar a cabo el proyecto.

- Dr. Germán H. Alférez es designado líder del proyecto. Él coordinará el avance del proyecto y administrará al personal del equipo de trabajo. Además, él estará a cargo

de difundir los resultados del estudio y de la coordinación de la construcción de las herramientas software.

- Mtro. Alejandro Garrido es designado coordinador técnico del área de hardware. Él está a cargo de la gestión de la construcción de los prototipos basados en hardware en donde se probarán las soluciones propuestas.
- Mtro. Ignacio Cruz es designado asesor técnico del área de software. Él apoyará la gestión de la construcción de las herramientas software.
- Mtra. Mary Adame es designada en la elaboración de reportes y en la administración de los recursos financieros.
- Mtro. Raúl Rodríguez es designado coordinador del análisis estadístico de los resultados experimentales.

3. Justificación del Proyecto

3.1. Motivación

En un mundo abierto cada vez más dinámico, inteligente y descentralizado, las tecnologías y los tipos de aplicaciones evolucionan hacia la creación de eco-sistemas compuestos por una amplia variedad de dispositivos y servicios heterogéneos y distribuidos, de naturaleza claramente móvil y ubicua, y en constante evolución tecnológica [1, 2].

Estos sistemas están compuestos por partes construidas por diversas organizaciones. Además, estos sistemas deben ejecutarse de forma continua y en entornos cambiantes. Debido a esto, es necesario que estos sistemas deban adaptarse en tiempo de ejecución a nuevas condiciones del entorno, nuevos dispositivos o servicios, o nuevas necesidades del usuario [3-6].

Ante esta situación, se hacen necesarias métodos, técnicas y herramientas que permitan la adaptación de los sistemas de forma autónoma (sin la participación obligatoria de analistas de sistemas) así como mecanismos que faciliten la interacción e integración en tiempo de ejecución de múltiples sistemas y servicios heterogéneos. Todo ello, siguiendo métodos de desarrollo rigurosos que garanticen la calidad de los sistemas desarrollados y la adaptación obtenida en tiempo de ejecución.

3.2. Objetivo del Proyecto

El objetivo general de este proyecto es proporcionar métodos, técnicas y herramientas para afrontar el desarrollo y evolución de sistemas auto-adaptativos en el ámbito de la IF. En concreto, se persigue afrontar los retos que introduce:

- La construcción de eco-sistemas a partir de sistemas ya existentes, que sean capaces de adaptarse en todo momento a las necesidades de los usuarios para proporcionar los servicios que estos necesitan.

- La evolución de estos eco-sistemas en un contexto de incertidumbre en cuanto a servicios y dispositivos disponibles, así como la aparición de nuevas condiciones de contexto.

Todo lo anterior utilizando técnicas del MDE que hagan un uso crítico y avanzado de los modelos.

Para conseguirlo se proponen los siguientes **objetivos específicos**:

1. Definir Técnicas Para Gestionar la Incertidumbre en Sistemas Auto-Adaptativos: En primera instancia, en el proyecto se analizarán las diferentes técnicas y estrategias para capturar la incertidumbre y se propondrá un **marco conceptual** para representarla definiendo lenguajes específicos de dominio para tal propósito. El estado del arte actual carece de este marco conceptual.

Al haber culminado el marco conceptual, se propondrán **técnicas basadas en el uso de modelos en tiempo de ejecución** para describir cómo los sistemas auto-adaptativos pueden afrontar la gestión de la incertidumbre frente a situaciones no planteadas en tiempo de diseño.

Se construirán cuatro **prototipos** para validar la propuesta en cada uno de los siguientes escenarios en donde la incertidumbre se hace evidente: 1) composición de servicios Web de tamaño real; 2) maqueta de una ciudad inteligente; 3) maqueta de un edificio inteligente; y 4) robot de rescate.

2. Definir un Marco de Trabajo (Framework) y Herramientas Para Abordar la Evolución de Sistemas Auto-Adaptativos: Se proporcionará un **marco de trabajo (framework)** para expresar y desarrollar sistemas auto-adaptativos con capacidad de evolución.

Con el fin de facilitar la aplicación de el marco de trabajo en la industria, el siguiente paso consiste en desarrollar una **infraestructura software** que permita evolucionar los modelos que representan (y son causales con) el sistema en tiempo de ejecución, en dónde se aplicarán las técnicas para la gestión de la incertidumbre.

Además, se proporcionarán **herramientas de software para probar y validar** de manera exhaustiva que los cambios realizados son coherentes con los nuevos requisitos del sistema, garantizando al menos que el sistema opere de manera 'razonable' (atendiendo a la incertidumbre inherente y adquirida).

3. Definir un Marco Metodológico y Arquitectónico para el Desarrollo de Sistemas Auto-Adaptativos: Se definirá un **Marco Arquitectónico** basado en M@RT en el que la definición de grandes sistemas se materializará mediante la composición de componentes distribuidos más pequeños. Además se definirá un **Marco Metodológico** que permita la construcción de este tipo de sistemas desde una perspectiva MDE.

Para llevar a cabo la definición de dicho marco de forma exitosa, será necesario hacer uso de técnicas basadas en la Ingeniería de Métodos, las cuales facilitan la definición de métodos de desarrollo de software. De esta forma, también se cubrirá el vacío existente en el ámbito de la Ingeniería de Métodos en lo que respecta al diseño y a la implementación de métodos de desarrollo específicos para Sistemas Auto-Adaptativos.

3.3. Resultados Esperados

Los resultados esperados se presentan en términos de los objetivos del proyecto.

OBJETIVO 1: Definir Técnicas para Gestionar la Incertidumbre en Sistemas Auto-Adaptativos.

Los entregables que se obtendrán para este objetivo son los siguientes:

E1: Publicación científica de las técnicas existentes para la especificación de la incertidumbre.

E2: Publicación científica de Lenguaje Específico de Dominio (DSL) genéricos para la definición de incertidumbre.

E3: Ejemplos de uso del DSL en diferentes escenarios de la IF.

E4: Prototipos realizados para probar la propuesta.

OBJETIVO 2: Definir un Marco de Trabajo y Herramientas para Abordar la Evolución de Sistemas Auto-Adaptativos.

Los entregables que se obtendrán para este objetivo son los siguientes:

E5: Artículo científico sobre el estado del arte en evolución de software y sistemas adaptativos.

E6: Artículo científico acerca de la especificación de un marco de trabajo para describir los aspectos evolutivos de un sistema que sea procesable por el propio sistema software en tiempo de ejecución. Este marco estará apoyado por una infraestructura software y por herramientas para validar y probar posibles configuraciones de sistemas auto-adaptativos (ver E9).

E7: Artículo científico con una propuesta de diseño arquitectónico de componentes de software decorados con elementos de procesado y evolución de los modelos de E6.

E8: Ejemplos de aplicación del marco de trabajo de E6 y del diseño arquitectónico de E7 en diferentes escenarios de la IF.

E9: Dos productos software: 1) infraestructura software que apoye la evolución de sistemas; y 2) herramienta para validar y probar posibles configuraciones de sistemas auto-adaptativos.

OBJETIVO 3: Definir un Marco Metodológico y Arquitectónico para el Desarrollo de Sistemas Auto-adaptativos.

Los entregables que se obtendrán para este objetivo son los siguientes:

E10. Artículo científico con la especificación de un método para el desarrollo y evolución de sistemas auto-adaptativos que se adecue a las particularidades de este tipo de sistemas.

E11. Herramientas que permitan dar soporte al método especificado.

3.4. Beneficios

El objetivo principal de este proyecto es proporcionar métodos y técnicas que permitan el desarrollo sistemático y productivo de software auto-adaptativo en el ámbito de la IF. En la IF, diversos dispositivos y servicios heterogéneos deben adaptarse para alcanzar una interacción coordinada y proporcionar a los usuarios servicios globalizados que se ajusten a sus necesidades. Este tipo de sistemas plantean nuevos retos tecnológicos como la gestión de la incertidumbre, su evolución o la necesidad de un soporte metodológico adecuado a las nuevas características del proceso de desarrollo.

Las contribuciones científico-técnicas que se realizan son de corte tecnológico, proponiendo métodos, herramientas y técnicas dirigidas por modelos que permitan mejorar la productividad y calidad de los sistemas desarrollados. Los resultados del proyecto introducirán algún tipo de avance científico-técnico en las siguientes áreas:

1. La **Ingeniería Dirigida por Modelos**, donde a) se crearán nuevas abstracciones y lenguajes específicos de dominio para especificar la incertidumbre, b) se propondrán estrategias para evolucionar sistemas que hacen uso intensivo de los M@RT para auto-adaptarse y c) se construirán métodos de desarrollo dirigidos por modelos para el diseño e implementación de sistemas auto-adaptativos;.
2. Los **Sistemas Auto-Adaptativos** y la **Computación Autónoma**, donde a) se propondrán, a un nivel de abstracción alto, mejoras expresivas y técnicas concretas para especificar y tratar la incertidumbre, b) se incorporarán capacidades evolutivas a los sistemas auto-adaptativos, c) se propondrá un patrón arquitectónico para el desarrollo de sistemas auto-adaptativos que incorporará la capacidad de evolución y será aplicable a distintos dominios y escenarios, y por último d) se dotará a este tipo de sistemas de procesos de desarrollo bien definidos y soportados por herramientas.

3. **Dominios y Escenarios de Aplicación** (Internet de las Cosas, Ciudades Inteligentes, etc.) donde se podrán aplicar los avances desarrollados en el proyecto. A nivel industrial, los resultados pueden ser aplicables a entornos reales. Se buscarán socios tecnológicos tanto en la industria como en el gobierno con quienes se puedan usar las herramientas que se desarrollen favoreciendo la aplicación de los resultados a corto-medio plazo.
4. Se creará el **Laboratorio de Ciencias Computacionales** en la FIT para desarrollar las herramientas que soporten software adaptativo. Este Laboratorio será creado para abarcar las siguientes áreas relacionadas con las Ciencias Computacionales: Ingeniería de Software, Computación Autónoma, e Inteligencia Artificial. Se espera extender estas áreas a futuro con nuevos proyectos de investigación. El **Laboratorio de Ciencias Computacionales** dependerá del Centro de Investigación e Innovación en Ingeniería de la FIT.
5. Las **Herramientas** a desarrollar se construirán usando técnicas de ingeniería basada en modelos. De acuerdo al análisis del estado del arte, el **Laboratorio de Ciencias Computacionales** será el primero en su tipo en universidades latinoamericanas. En este sentido, seremos pioneros en Latinoamérica, y unos de los pocos en el mundo, en el desarrollo de sistemas inteligentes para el mundo abierto usando ingeniería basada en modelos.
6. Este proyecto servirá como base para **Impulsar el Quehacer Científico de la FIT**. Específicamente, alumnos de ingeniería podrán realizar sus proyectos de fin de carrera para resolver diferentes áreas de este proyecto. Además, alumnos en la Maestría en Ciencias Computacionales relacionarán sus tesis a las tareas a realizar en este proyecto. Más aún, los proyectos resultantes de las materias de Ingeniería de Software, Arquitecturas de Software y de Metodologías Orientadas a Objetos de la Maestría en Ciencias Computacionales se relacionarán con temas tomados de este proyecto.

4. Análisis de la Factibilidad del Proyecto

4.1. Antecedentes

En un mundo abierto [1] cada vez más dinámico, inteligente y descentralizado, las tecnologías y los tipos de aplicaciones evolucionan hacia la creación de eco-sistemas compuestos por una amplia variedad de dispositivos y servicios heterogéneos y distribuidos, de naturaleza claramente móvil y ubicua, y en constante evolución tecnológica. Gracias a la infraestructura que proporciona la Internet del Futuro (IF) [2], a la Computación en la Nube, al avance de los dispositivos móviles (teléfonos inteligentes, tabletas, etc.) y los 'wearables' (gafas, relojes inteligentes, sensores en la ropa, etc.), se están desarrollando nuevas aplicaciones y servicios sofisticados con alto grado de volatilidad a través de la cooperación entre dispositivos (u

objetos) interconectados, no conocidos en tiempo de diseño, para una sociedad de usuarios cuyas necesidades evolucionan y cambian rápidamente.

Ante esta situación, surge la necesidad de desarrollar sistemas que sean capaces de adaptarse continuamente en tiempo de ejecución a nuevas condiciones del entorno, situaciones impredecibles, necesidades cambiantes de sus usuarios, nuevos dispositivos y tecnologías con los que interactuar o nuevos servicios que consumir. Esta adaptación debe gestionarse de forma autónoma (auto-adaptatividad) ya que no parece viable asumir que un desarrollador intervenga activamente en materializar todas las adaptaciones en tiempo de ejecución.

Actualmente, existen multitud de trabajos en este ámbito [3-6], donde se considera la adaptatividad como una capacidad necesaria de los sistemas sensibles al contexto altamente dinámicos [1], para adecuarse a su contexto [7]. Numerosos trabajos [8-11] tratan de abordar la descripción en tiempo de diseño de la adaptatividad de los sistemas introduciendo la noción de variabilidad. En [12-16] se aplica al ámbito de los procesos de negocio. Sin embargo, estas soluciones se han centrado en el desarrollo de sistemas desplegados en entornos controlados y con un conjunto de opciones de adaptación finito y acotado conocido en tiempo de diseño.

Los sistemas software desarrollados con los actuales avances tecnológicos (Internet of Things (IoT), Computación en la Nube, teléfonos inteligentes, objetos “wearables”, etc.) deben diseñarse y desarrollarse asumiendo un cierto grado de incertidumbre “manejable o controlable” permitiendo el diseño de sistemas incompletos que se adaptarán y evolucionarán en tiempo de ejecución. Así pues, en el contexto de los sistemas Auto-Adaptativos, surgen una gran cantidad de retos científicos considerados “hot topics” en las comunidades científicas de la Ingeniería del Software para Sistemas Adaptativos y Auto-Gestionados [4, 6] y en la Ingeniería Dirigida por Modelos, en concreto los Modelos en Tiempo de Ejecución (MDE y M@RT) [17, 18]. De forma transversal, en el ámbito de la Ingeniería de Métodos, también surgen retos científicos relacionados con la necesidad de disponer de nuevas herramientas y métodos para abordar la construcción de este tipo de sistemas.

4.2. Análisis del Entorno del Proyecto

El mundo computacional es cada vez más abierto [1]. La IF, la Computación en la Nube, los dispositivos móviles, y los “wearables” han despertado el desarrollo de nuevas aplicaciones y servicios. Esta situación genera la necesidad de desarrollar sistemas que sean capaces de auto-adaptarse continuamente en tiempo de ejecución.

La relevancia y actualidad de la necesidad de la auto-adaptación de software se puede constatar por la actividad de la comunidad científica en estos últimos años. Algunas de las conferencias más importantes son SEAMS, ICAC, SASO, CAC o el workshop Models@run.time de la conferencia MODELS (centrado en el uso de modelos para la construcción de sistemas auto-adaptativos); la revista de referencia en esta comunidad es TAAS. Además, esta temática

se ha introducido en conferencias como BPM o CoopIS, donde se estudia cómo gestionar la adaptatividad en procesos de negocio y sistemas de información cooperativos.

En el entorno de la FIT, el líder del proyecto, el Dr. Germán Harvey Alférez Salinas, tiene un amplio conocimiento en MDE, M@RT, y Computación Autónoma. De hecho, él ha publicado en varias conferencias y revistas científicas internacionales de alto impacto en estas áreas. Una lista completa de las publicaciones, proyectos y prototipos del Dr. Alférez se encuentra en su sitio Web: www.harveyalferez.com. Además, el proyecto contará con el apoyo de expertos clave de la FIT en diferentes áreas: Mtro. Alejandro Garrido (experto en hardware y en electrónica), Mtro. Ignacio Cruz (experto en software), Mtra. Mary Adame (experta en administración de proyectos), y el Mtro. Raúl Rodríguez (experto en estadística).

4.3. Estado del Arte

El estado del arte del proyecto se organiza de acuerdo a los siguientes retos de los sistemas auto-adaptativos:

Gestionar la Incertidumbre

Para afrontar los retos que plantea la visión de un mundo abierto, el concepto de incertidumbre [19] puede ofrecer un mecanismo para abordar el problema de no poder prever todas las situaciones de variabilidad en tiempo de diseño. La incertidumbre permite definir información en base a medidas poco precisas o incompletas, especificar funciones de utilidad, e introducir nuevos mecanismos de análisis. Aplicada a especificar la adaptatividad, este concepto conduce a la construcción de sistemas 'resistentes' (*resilient systems*) [19] que estarán preparados para hacer frente a situaciones excepcionales para las que no habían predefinidas reglas de adaptación, pero en los que no se podrá asegurar la corrección de una adaptación, aunque sí al menos se podrán garantizar adaptaciones 'útiles' [20, 21].

Existen trabajos que introducen la incertidumbre en tiempo de diseño [22, 23] y mecanismos para analizar y razonar sobre esta incertidumbre, la mayoría basados en la definición de metas [24, 25, 26]. Sin embargo, o bien no definen mecanismos para adaptar los sistemas en tiempo de ejecución, o bien, el análisis es muy complejo en sistemas grandes.

Abordar la Evolución de Sistemas Auto-Adaptativos

En la última década han surgido numerosas aproximaciones para dar soporte a la auto-adaptación de sistemas software, en respuesta a cambios en el contexto de ejecución y en el propio sistema [27-31].

Sin embargo, existe una opinión consensuada de que no se pueden diseñar sistemas auto-adaptativos pensando en los requisitos no conocidos en tiempo de diseño (*unknown unknowns*). Sólo se pueden definir auto-adaptaciones sobre situaciones que podrían ocurrir en tiempo de ejecución, que se puedan prever en tiempo de diseño (*unknown knowns*). Por

tanto, las aproximaciones actuales de auto-adaptación son insuficientes para abordar los nuevos retos que plantea el mundo abierto donde cambios no esperados pueden aparecer sin previo aviso [32].

Uno de los retos será ofrecer un marco de trabajo y una infraestructura de soporte que permita introducir el concepto de auto-evolución, de manera que posibilite tanto “evolucionar” los modelos de estos sistemas auto-adaptativos en tiempo de ejecución, como probar y validar que estos cambios son coherentes con los requisitos del sistema [4]. La aplicación de patrones de cambio puede contribuir a asegurar una correcta evolución del sistema [33].

Existen trabajos [34, 35] que abordan la adaptación dinámica, argumentando que son capaces de gestionar ciertas situaciones excepcionales. En [36], por ejemplo, se propone una metodología MDE para dar soporte al diseño y desarrollo de aplicaciones auto-adaptativas que tienen en cuenta cambios en el contexto 'semi-anticipados' en base a metas. En [37] se presenta un enfoque para evolucionar modelos y realizar adaptaciones en base a la definición de unos parámetros analizados en tiempo de ejecución. En ambos trabajos, se identifican en tiempo de diseño los conceptos, tanto las interfaces como los parámetros, sobre los que el sistema razona en tiempo de ejecución, pero no permiten incorporar conceptos desconocidos en tiempo de diseño.

Definir un Marco Metodológico y Arquitectónico

Cada vez más los sistemas software actuales y del futuro están compuestos por partes construidas por diversas organizaciones, que deben ejecutarse de forma continua y operar en entornos cambiantes donde la adaptación se vuelve imprescindible. Estos sistemas dan servicio a usuarios móviles a través de dispositivos inteligentes, dotándoles de una completa ubicuidad. La IF constituye la infraestructura, la plataforma y el entorno de ejecución de estos 'sistemas de sistemas' [38] que deben desarrollarse de forma que soporten la auto-adaptación y los retos tecnológicos planteados anteriormente.

A nivel ARQUITECTÓNICO la mayoría de soluciones actuales siguen el espíritu de la computación autónoma [39] donde se introduce el bucle de control MAPE-K. Este enfoque centralizado es muchas veces complejo y pesado de implementar en sistemas ubicuos, aunque más sencillo de integrar en sistemas ya existentes. Otras aproximaciones [40] reconocen que en un mundo eminentemente distribuido se necesitan sistemas descentralizados capaces de proporcionar una inteligencia y una adaptación colectiva. Sin embargo, estas aproximaciones se enfocan en la implementación de soluciones algorítmicas novedosas, y no en proporcionar arquitecturas y métodos que faciliten el diseño de estos sistemas.

A nivel METODOLÓGICO son escasas las aproximaciones que se han preocupado por ofrecer métodos para el desarrollo de sistemas auto-adaptativos: en los seminarios de Dagstuhl [41] se comenta por primera vez la necesidad de reconceptualizar el proceso de desarrollo de estos sistemas, distinguiendo entre actividades en tiempo de diseño y en tiempo de ejecución;

se identifican una serie de retos para adaptar el modelo de proceso pero no se detalla cómo llevarlo a cabo. De forma similar, aunque específica de dominio, en el proyecto europeo ASCENS se propone un método para el ensamblado de componentes que integra patrones de adaptación en el desarrollo de sistemas [42]. En el ámbito de la IoT se han propuesto algunos métodos de MDE como el presentado en el proyecto europeo ELLIOT [43] que integra los métodos GenIoT y Aloha, donde se proporcionan herramientas orientadas a la captura de requisitos y escenarios en el ámbito de la IoT, y el método PMG-Pro [44] orientado a modelar e integrar “cosas” (vistas como servicios Web), donde no se considera la adaptación de dichos sistemas.

Analizando las propuestas actuales a nivel arquitectónico y metodológico, es necesario replantearse cómo se diseñan y desarrollan este tipo de sistemas para poder abordar los retos propuestos.

4.4. Determinación de Recursos

4.4.1. Recursos Humanos

El presente proyecto requiere de conocimiento en diversas áreas tales como MDE, M@RT, y Computación Autónoma. Por consiguiente, consideramos necesario trabajar con estudiantes de ingeniería desde el **segundo año** con el fin de que con buen tiempo puedan prepararse en estas áreas. Durante el proceso, los estudiantes de pregrado envueltos en el proyecto podrán ir avanzando en la complejidad de sus entregables (e.g. publicaciones y prototipos) hasta llegar a su proyecto de fin de carrera.

En el nivel de maestría, se espera que los estudiantes envueltos en este proyecto trabajen tiempo completo desde su primer semestre en algún área de interés del mismo.

Tanto en estudiantes de pregrado como de posgrado, se buscarán estudiantes con gusto por la programación de computadoras y por la Ingeniería de Software.

Además, el proyecto se nutrirá con: 1) los proyectos de fin de carrera de estudiantes no becarios, pero que quieren hacer su investigación en algún área de interés en el proyecto; y 2) estudiantes de servicio social. **Estos estudiantes no serán pagos por el proyecto y por consiguiente no se incluyen en el presupuesto de la Sección 6.**

En total, el proyecto requerirá de los siguientes recursos humanos **en el primer semestre de ejecución**. Se espera que al ir avanzando en el proyecto, se puedan conseguir los recursos financieros para contratar más personal:

- 1 estudiante de la Maestría en Ciencias Computacionales.
- 2 estudiantes de Ingeniería en Sistemas Computacionales.
- 1 estudiante de Ingeniería Electrónica.

4.4.2. Recursos Financieros

Al inicio del proyecto, se espera que los recursos sean internos (Departamento de Investigación y Facultad de Ingeniería y Tecnología de la Universidad de Morelos). No obstante, se someterá este proyecto a una próxima convocatoria del CONACYT a mediados de 2014 con el fin de recibir recursos externos. El presupuesto se describe en detalle en la Sección 6.

4.4.3. Recursos Técnicos

El estudiante de la Maestría en Ciencias Computacionales deberá estar en la línea de Ingeniería de Software y contar con excelentes conocimientos en programación y modelado de software.

Los estudiantes de Ingeniería en Sistemas Computacionales deberán contar con excelentes conocimientos en programación y de esperarse también en Ingeniería de Software.

El estudiante de Ingeniería Electrónica deberá contar con excelentes conocimientos en programación y ensamblado de microcontroladores.

4.4.4. Recursos Materiales

Los recursos materiales necesarios se dividen en la adecuación del espacio de trabajo y las computadoras que utilizarán los estudiantes.

Con respecto al espacio de trabajo, se requieren los siguientes recursos:

- 4 escritorios, uno para cada estudiante **(los proveerá la FIT. Por lo tanto, no se incluyen en el presupuesto de la Sección 6).**
- 4 sillas, una para cada estudiante **(las proveerá la FIT. Por lo tanto, no se incluyen en el presupuesto de la Sección 6).**
- 1 tablero grande.

El espacio del grupo de trabajo estará en la oficina de los profesores de la Academia de Software (edificio de la FIT). En esta área se está adecuando el espacio para que funcione el **Laboratorio de Ciencias Computacionales.**

Se requieren los siguientes recursos para la creación de: 1) la **infraestructura software** que permita evolucionar los modelos que representan (y son causales con) el sistema en tiempo de ejecución, en donde se aplicarán las técnicas para la gestión de la incertidumbre; y 2) las **herramientas de software para probar y validar** de manera exhaustiva que los cambios realizados son coherentes con los nuevos requisitos del sistema, garantizando al menos que el sistema opere de manera 'razonable':

- 4 desktops completas (monitor, torre, teclado, ratón y regulador de corriente), una para cada estudiante **(las proveerá la FIT. Por lo tanto, no se incluyen en el presupuesto de la Sección 6).**
- 1 impresora.

En el proyecto se crearán prototipos en donde se probarán las aproximaciones propuestas. Tal como se describe en la Sección 4.2, se requiere la construcción de cuatro prototipos en cuatro escenarios: 1) composición de servicios Web de tamaño real; 2) maqueta de una ciudad inteligente; 3) maqueta de un edificio inteligente; y 4) robot de rescate. La composición de servicios Web no requiere de hardware y por lo tanto no se requieren recursos materiales para su construcción. No obstante, los otros tres prototipos requieren recursos físicos para su construcción. Específicamente, para construir las maquetas de la ciudad inteligente y del edificio inteligente se requiere el siguiente hardware.

- 2 Bare Conductive - Glowing House Set - Voltage Village (para Arduino). Uno para cada una de las maquetas.
- 2 Arduino Starter Kit. Uno para cada una de las maquetas.
- 2 Arduino UNO. Uno para cada una de las maquetas.

El robot de rescate se construirá con los equipos marca Lego de la FIT. **Por lo tanto su costo no se especifica en el presupuesto de la Sección 6.**

4.5. Aportación del Proyecto

4.5.1. Avances en la Ciencia y/o la Tecnología que se Prevén Alcanzar

La finalidad del proyecto es proporcionar métodos, técnicas y herramientas en el ámbito de la MDE para desarrollar de manera sistemática y productiva software auto-adaptativo en un mundo inteligente. En concreto, se proporcionarán soluciones que permitan abordar algunos de los retos tecnológicos e ingenieriles que introducen los Sistemas Auto-Adaptativos. Retos como la gestión de la incertidumbre, la evolución de este tipo de sistemas, además de proporcionar un soporte metodológico al proceso de desarrollo de estos sistemas.

El proyecto se desarrolla en el ámbito de la MDE haciendo hincapié en el uso de los M@RT como mecanismo para resolver algunos de los retos presentados. Siguiendo esta visión se propone abordar los retos propuestos de la siguiente manera:

1. Gestionar y Analizar la Incertidumbre en Sistemas Auto-Adaptativos: En este contexto, pretendemos introducir la noción de incertidumbre como un elemento de primer nivel en el diseño, desarrollo y evolución de sistemas auto-adaptativos. Para ello, revisaremos las estrategias de desarrollo dirigido por modelos aplicadas a este tipo de sistemas, y extenderemos nuestras propuestas previas que gestionan la auto-adaptatividad para: 1)

extender los lenguajes y modelos específicos de dominio para introducir la noción de incertidumbre; 2) definir técnicas de análisis del impacto de la incertidumbre sobre las adaptaciones posibles de estos sistemas; y 3) construir una infraestructura que en tiempo de ejecución sea capaz de analizar la información de contexto en busca de situaciones excepcionales y poder reaccionar de manera autónoma y proactiva teniendo en cuenta la incertidumbre.

2. Tratar la Evolución del Software Auto-Adaptativo: En este reto, pretendemos iniciar el camino hacia la auto-evolución de los sistemas mediante combinación de técnicas MDE, M@RT y Computación Autónoma para conseguir sistemas auto-adaptativos capaces de evolucionar. Con este fin, usaremos técnicas basadas en la gestión de la incertidumbre para dotar al sistema de la capacidad de razonar ante situaciones inesperadas y que pueda ofrecer ciertas garantías de operación sobre el concepto de “funcionamiento aceptable”. En este sentido, se definirán mecanismos que puedan observar el contexto y adquirir o aprender de manera autónoma nuevo conocimiento que incorporarán a los modelos del sistema, reduciendo la incertidumbre específica con relación a algún objetivo o comportamiento sistémico.

3. Diseñar un Marco Metodológico y Arquitectónico basado en MDE y en el uso de los M@RT para concebir y diseñar Sistemas Auto-Adaptativos: Para llevar a cabo este reto es necesario abordar algunos temas de investigación específicos que nos permitan construir una propuesta metodológica y arquitectónica innovadora:

A) Aproximaciones BOTTOM UP vs. TOP DOWN: el enfoque TOP-DOWN puede no ser adecuado para diseñar sistemas auto-adaptativos de gran tamaño, considerando la naturaleza dinámica, la imposibilidad de anticiparse a requisitos cambiantes y componentes o subsistemas software que se incorporan o desaparecen del sistema en tiempo de ejecución. En la IF donde el software deberá construirse integrando servicios y cosas ya desarrolladas e implantadas (normalmente en infraestructuras en la nube) las aproximaciones BOTTOM UP encajan con mayor facilidad. Ambas aproximaciones pueden coexistir en escenarios futuros, la primera para diseñar sistemas locales o de tamaño manejable, y la segunda aplicada a grandes sistemas. Conseguir integrar y compatibilizar ambas aproximaciones será un reto interesante en el ámbito del proyecto.

B) COMPOSICIÓN Y REUSO: En este tipo de sistemas las piezas deberán componerse, y proporcionar la información necesaria (semántica e inteligencia) en base a modelos, de manera que se facilite la construcción de sistemas complejos adaptativos a partir de “pequeños” servicios y/o cosas (*smart objects*) que sean “*adaptive ready*”. Con el término ADAPTIVE READY planteamos la idea de poder construir sistemas adaptativos a partir de piezas “preparadas” para ser integradas en este tipo de sistemas.

C) MARCO CONCEPTUAL Y ARQUITECTÓNICO: Se plantea definir un marco conceptual genérico y una infraestructura que, apoyándose en técnicas de M@RT, proporcione un patrón a seguir para dar soporte a los retos abordados en el proyecto sobre un amplio abanico de

tecnologías y dominios. El objetivo es facilitar la correcta implementación de los Sistemas Auto-Adaptativos Dinámicos y la transformación de sistemas convencionales en Sistemas Auto-Adaptativos.

4.5.2. Beneficios e Impacto Social, Económico y Ambiental en la Región, el País y el Mundo

La Sección 3.4 del presente documento lista los beneficios o contribuciones científico-técnicas del presente proyecto. El mundo abierto es considerado como un “hot topic” a nivel mundial. Por lo tanto, es factible decir que el presente proyecto tendrá una aplicación e impacto positivo global. Con respecto al impacto a nivel nacional, el presente proyecto se enmarca en apoyar al Programa Especial de Ciencia, Tecnología e Innovación del CONACYT. Más aún, ya que la IF, la Computación en la Nube, y el avance de los dispositivos móviles requieren de nuevas aplicaciones, creemos que este proyecto podrá impulsar una nueva generación de empresas de carácter tecnológico en México. Por consiguiente, el tema a desarrollar podrá ser de especial interés para PROSOFT. Por último, el proyecto no generará desechos de ningún tipo.

4.5.3. Productos de Investigación Esperados: Artículos, Libros, Tesis, Patentes, etc.

El plan de difusión de resultados está orientado a poder materializar los objetivos de esta convocatoria. Para lograr una investigación de calidad, evidenciada por la internacionalización de las actividades y la publicación de sus resultados en foros de alto impacto científico y tecnológico, proponemos:

1. **Publicación en Revistas y Congresos:** Nuestra intención es publicar en revistas científicas con buen factor de impacto (1 o superior) y en congresos internacionales comparables (ocupando posiciones relevantes en el Science Citation Index) prestando especial atención a los más relevantes en las áreas de interés del proyecto como son SASO, SEAMS, SPLC, UBICOMP, BPM, CoopIS, Mobiquitous, PERCOM, Mobile-HCI, ICWS, ICSE, UMAP (anterior AH), ICAC, CHI, VL/HCC e ICSOC, entre otros. Consideramos importante mantener la presencia en conferencias relacionadas con el Desarrollo Dirigido por Modelos y los Métodos de Producción de Software como MODELS, CAiSE, ER, ICMT y ECMDA. En cuanto a revistas, nuestra intención es publicar en revistas como: IEEE Software, IEEE Computer, IEEE Pervasive Computing, IEEE Transactions on Consumer Electronics, ACM Transactions on Autonomous and Adaptive Systems (TAAS), Information Systems Journal, Data and Knowledge Engineering, ACM e IEEE Transactions on Software Engineering, ACM Transactions on the Web (TWEB) e Internet Technology (TOIT), Information and Software Technology, SoSyM, PUC, MTAP, etc.
2. **Proyectos de Fin de Carrera:** El proyecto se nutrirá con proyectos de fin de carrera. La clave consistirá en capturar talentos jóvenes en los primeros semestres y así

nutrir el proyecto con sus ideas y conocimiento. Al momento de comenzar el proyecto, se hará pública la convocatoria de varios proyectos de fin de carrera en tres áreas de acuerdo a los objetivos del proyecto: 1) gestión de la incertidumbre en sistemas auto-adaptativos; 2) definición de técnicas para abordar la evolución de sistemas auto-adaptativos; y 3) definición de un marco metodológico y arquitectónico para el desarrollo de sistemas auto-adaptativos.

3. **Tesinas de Maestría:** El proyecto se nutrirá del trabajo de los estudiantes de la Maestría en Ciencias Computacionales de la FIT. Específicamente, los proyectos finales de los siguientes cursos se direccionarán para fortalecer este proyecto: 1) Ingeniería de Software; 2) Metodologías Orientadas a Objetos; y 3) Arquitecturas de Software. La sinergia entre el presente proyecto y estos cursos es posible gracias a que estos cursos se ajustan perfectamente al proyecto. Además, se invitará a estudiantes desde el primer semestre de la maestría a llevar materias de ingeniería de software y a trabajar en el presente proyecto. Con lo anterior se espera contar con tesinas basadas en los resultados del proyecto.
4. **Herramientas:** Se comercializarán las siguientes herramientas software: 1) la **infraestructura software** que permita evolucionar los modelos que representan (y son causales con) el sistema en tiempo de ejecución, en dónde se aplicarán las técnicas para la gestión de la incertidumbre; y 2) las **herramientas de software para probar y validar** de manera exhaustiva que los cambios realizados son coherentes con los nuevos requisitos del sistema.

5. Plan Detallado del Proyecto

5.1. Generalidades

En este estudio se proponen métodos, técnicas y herramientas que permitan el desarrollo sistemático y productivo de software auto-adaptativo para el ámbito de la IF, donde múltiples dispositivos y servicios heterogéneos deben adaptarse para alcanzar una interacción coordinada y proporcionar a los usuarios servicios globalizados que se ajusten a sus necesidades.

Las soluciones propuestas se darán desde el ámbito de la MDE y los M@RT. Específicamente, se proporcionarán soluciones que permitan abordar la evolución de estos eco-sistemas en un contexto de incertidumbre en cuanto a servicios, dispositivos y cambios en el contexto. Además, se ofrecerá soporte metodológico al proceso de desarrollo de este tipo de sistemas.

5.2. Planificación de la Secuencia del Proyecto

La planificación del proyecto se presenta en términos de los entregables en cada uno de los objetivos del proyecto.

OBJETIVO 1: Definir Técnicas para Gestionar la Incertidumbre en Sistemas Auto-Adaptativos.

Los entregables que se obtendrán para este objetivo son los siguientes:

E1: Publicación científica de las técnicas existentes para la especificación de la incertidumbre.

E2: Publicación científica de DSL genéricos para la definición de incertidumbre.

E3: Ejemplos de uso del DSL en diferentes escenarios de la IF.

E4: Prototipos realizados para probar la propuesta.

OBJETIVO 2: Definir un Marco de Trabajo y Herramientas para Abordar la Evolución de Sistemas Auto-Adaptativos.

Los entregables que se obtendrán para este objetivo son los siguientes:

E5: Artículo científico sobre el estado del arte en evolución de software y sistemas adaptativos.

E6: Artículo científico acerca de la especificación de un marco de trabajo para describir los aspectos evolutivos de un sistema que sea procesable por el propio sistema software en tiempo de ejecución. Este marco estará apoyado por una infraestructura software y por herramientas para validar y probar posibles configuraciones de sistemas auto-adaptativos (ver E9).

E7: Artículo científico con una propuesta de diseño arquitectónico de componentes de software decorados con elementos de procesado y evolución de los modelos de E6.

E8: Ejemplos de aplicación del marco de trabajo de E6 y del diseño arquitectónico de E7 en diferentes escenarios de la IF.

E9: Dos productos software: 1) infraestructura software que apoye la evolución de sistemas; y 2) herramienta para validar y probar posibles configuraciones de sistemas auto-adaptativos.

OBJETIVO 3: Definir un Marco Metodológico y Arquitectónico para el Desarrollo de Sistemas Auto-adaptativos.

Los entregables que se obtendrán para este objetivo son los siguientes:

E10. Artículo científico con la especificación de un método para el desarrollo y evolución de sistemas auto-adaptativos que se adecue a las particularidades de este tipo de sistemas.

E11. Herramientas que permitan dar soporte al método especificado.

| | Año 1 (trimestres) | | | | Año 2 (trimestres) | | | |
|----------|--------------------|----|----|----|--------------------|----|----|----|
| Objetivo | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | E3 | E4 | E1 | E2 | | | | |
| 2 | | | | E5 | E6 | E7 | E8 | E9 |
| 3 | | | | | | | | |

| | Año 3 (trimestres) | | | |
|----------|--------------------|-----|----|----|
| Objetivo | 9 | 10 | 11 | 12 |
| 1 | | | | |
| 2 | | | | |
| 3 | E11 | E10 | | |

5.3. Estructura Organizacional y Personal Participante

El proyecto será desarrollado en el **Laboratorio de Ciencias Computacionales**. La organización de este proyecto se describe a continuación:

Dr. Germán Harvey Alférez Salinas (*Líder del Proyecto*)

|

Mtro. Alejandro Garrido (*Coordinador Técnico de Hardware*) – **Mtro. Ignacio Cruz** (*Asesor Técnico de Software*) – **Mtra. Mary Adame** (*Elaboración de Reportes y Administración de Recursos Financieros*) – **Mtro. Raúl Rodríguez** (*Coordinador del Análisis Estadístico*)

6. Presupuesto

En el siguiente presupuesto se contemplan cuatro becarios para empezar a ejecutar el proyecto. Se espera que el número de becarios incremente a medida que el proyecto avance y se consigan más recursos financieros. Se presenta el presupuesto por el **primer semestre de ejecución**.

Para empezar la ejecución del proyecto, en su primer semestre se solicita que el Departamento de Investigación de la Universidad de Montemorelos aporte \$35,400.00. **Esta cantidad se le devolverá al Departamento de Investigación de la Universidad de Montemorelos cuando el proyecto haya sido aceptado en alguna convocatoria del CONACYT.** En la Sección 4.4.4 se especifican los rubros que cubrirá la FIT y que **no** se especifican a continuación.

| Rubro | Justificación | Financiador | Costo |
|---|-----------------------|--|------------------------------------|
| 1 estudiante de la Maestría en Ciencias Computacionales | Recursos humanos | Departamento de Investigación, Universidad de Montemorelos | \$7,000.00 |
| 2 estudiantes de Ingeniería en Sistemas Computacionales | Recursos humanos | Departamento de Investigación, Universidad de Montemorelos | \$7,000 x 2 estudiantes = \$14,000 |
| 1 estudiante de Ingeniería Electrónica | Recursos humanos | Departamento de Investigación, Universidad de Montemorelos | \$7,000.00 |
| 1 tablero grande | Recursos físicos | Departamento de Investigación, Universidad de Montemorelos | \$1,000.00 |
| 1 impresora para imprimir informes y artículos científicos | Recursos tecnológicos | Departamento de Investigación, Universidad de Montemorelos | \$800 Impresora Laser Brother |
| 2 Easy Weareable Kit para la creación de maquetas de la ciudad inteligente y del edificio inteligente | Recursos tecnológicos | Departamento de Investigación, Universidad de Montemorelos | \$450 x 2 = \$900 |
| 2 Glowing House Set - Voltage Village (para Arduino) para la creación de maquetas de la ciudad inteligente y del edificio inteligente | Recursos tecnológicos | Departamento de Investigación, Universidad de Montemorelos | \$450 x 2 = \$900 |
| 2 Arduino Starter Kit para la creación de maquetas de la ciudad inteligente y del edificio inteligente | Recursos tecnológicos | Departamento de Investigación, Universidad de Montemorelos | \$1,500 x 2 = \$3,000 |
| 2 Arduino UNO para la creación de maquetas de la ciudad inteligente y del edificio inteligente | Recursos tecnológicos | Departamento de Investigación, Universidad de Montemorelos | \$400 x 2 = 800 |
| Total por el primer semestre del proyecto | | | \$35,400.00 |

7. Control del Programa de Trabajo del Proyecto

7.1. Identificación de Riesgos y Puntos Críticos

El mayor riesgo consiste en que alguno de los estudiantes deje el grupo de investigación. Este hecho generaría un riesgo ya que el estudiante marcharía con el conocimiento adquirido que es clave para la consecución del proyecto. En este caso, se buscará reemplazar el recurso tan pronto como sea posible.

Otro riesgo consiste en la adecuada ejecución del cronograma del proyecto. Con el fin de enfrentar este riesgo, el líder del proyecto estará observando semanalmente el avance de los hitos y entregables del cronograma por cada estudiante.

7.2. Gestión de Cambios

El líder del proyecto estará en todo momento observando los cambios en los entornos/situaciones/dispositivos en los cuáles se pueda aplicar la solución propuesta. En caso de cambios repentinos que puedan afectar la ejecución del proyecto (e.g. nuevas metodologías o tecnologías), el líder del proyecto realizará los cambios necesarios para asegurar el éxito del proyecto.

7.3 Medición de Avances

Las tareas del proyecto se evaluarán mediante los entregables en el cronograma de la Sección 5.2. En caso de detectarse algún inconveniente o retraso, se procederá a actuar inmediatamente para solucionarlo.

8. Evaluación y Cierre del Proyecto

El desempeño del equipo se evaluará en publicaciones realizadas, prototipos construidos, herramientas construidas, proyectos de fin de carrera relacionadas con el proyecto finalizados, y tesinas de maestría relacionadas con el proyecto exitosas. Habiendo analizado los resultados obtenidos, y dadas por concluidas las actividades del proyecto, se ofrecerán conclusiones y recomendaciones de nuevas líneas de investigación relacionadas con la Ingeniería Dirigida por Modelos para el Desarrollo de Software Auto-Adaptativo en el Mundo Inteligente.

9. Bibliografía

- [1] Hong J., Suh E., Kim S. Context-aware systems: A literature review and classification. *Expert Systems with Applications* 36, 8509–8522, 2009
- [2] Domingue J., et al. The Future Internet - Future Internet Assembly: Achievements and Technological Promises. *LNCS* 6656, 2011

- [3] Cheng B., et al. Software Engineering for Self-Adaptive Systems: A Research Roadmap, Self-Adaptive Systems, LNCS 5525,1-26, 2009
- [4] Garlan D. A 10-year perspective on software engineering self-adaptive systems. Keynote SEAMS, 2013
- [5] Lemos R., Giese H., Müller H., Shaw M., et al. Software Engineering for Self-Adaptive Systems: A Second Research Roadmap. Software Engineering for Self-Adaptive Systems, 1-32, 2010
- [6] Lemos R., Giese H., Müller H., Shaw M. Software Engineering for Self-Adaptive Systems II - International Seminar, Dagstuhl Castle, Germany, LNCS, 2013
- [7] McKinley P., Sadjadi S., Kasten E., Cheng B. Composing adaptive software. Computer 37, 56-64, 2004
- [8] Baresi L., Guinea S. Self-supervising BPEL processes. IEEE Transactions on Software Engineering 37, 247-263, 2011
- [9] Cetina C., Giner P., Fons J., Pelechano V. Autonomic Computing through Reuse of Variability Models at Runtime: The Case of Smart Homes. IEEE Computer 42(10), 37-43, 2009
- [10] Charfi A., Mezini M. AO4BPEL: An aspect-oriented extension to BPEL. World Wide Web 10 (3), 309-344, 2007
- [11] Koning M., Sun C., Sinnema M., Avgeriou P. VxBPEL: Supporting variability for web services in BPEL. Information and Software Technology 51, 258-269, 2009
- [12] Acher M., Collet P., Lahire P., France R. Managing variability in workflow with feature model composition operators. In Proc. SC'10, 17-33, 2010
- [13] Bucchiarone A., Mezzina A., Pistore M. CAptLang: A language for context-aware and adaptable business processes. In Proc. VaMoS, 2013
- [14] Hallerbach A., Bauer T., Reichert M. Capturing variability in business process models: the Provop approach. Software Maintenance 22(6-7), 519-546, 2010
- [15] La Rosa M., van der Aalst W., Dumas M., ter Hofstede. A. Questionnaire-based variability modeling for system configuration. Software and System Modeling 8(2), 251-274, 2009
- [16] Nguyen T., Colman A., Han J. Modeling and managing variability in process-based service compositions. In Proc. ICSOC, 404-420, 2011

- [17] Blair G., Bencomo N., France R. Models@ run.time. IEEE Computer 42(10), 22-27, 2009
- [18] Bencomo N., Blair G., Götz S., Morin B., et al. Report on the 7th International workshop on models@run.time. ACM SIGSOFT Software Engineering Notes 38(1), 27-30, 2013
- [19] Ramirez A., Jensen A., Cheng B. A taxonomy of uncertainty for dynamically adaptive systems. In Proc. SEAMS, 99–108, 2012
- [20] Baresi, L., Di Nitto, E., Ghezzi, C. Toward open-world software: Issue and challenges. Computer 39, 36–43, 2006
- [21] Garlan D. Software engineering in an uncertain world. In Proc. FoSER, 125-128, 2010
- [22] Cheng B., Sawyer P., Bencomo N., Whittle J. A goal-based modeling approach to develop requirements of an adaptive system with environmental uncertainty. In Proc. MODELS, 468–483, 2009
- [23] Goldsby H., Cheng B. Automatically generating behavioral models of adaptive systems to address uncertainty. In Proc. MODELS, 568–583, 2008
- [24] Ranganathan A., Al-Muhtadi J., Campbell R. Reason- ing about uncertain contexts in pervasive computing environments. IEEE Pervasive Computing 3 (2), 62–70, 2004
- [25] Welsh K., Sawyer P., Bencomo N. Towards Requirements Aware Systems: Run-Time Resolution of Design-Time Assumptions. In Proc. ASE'11, 560-563, 2011
- [26] Whittle J., Sawyer P., Bencomo N., Cheng B., Bruel J. RELAX: a language to address uncertainty in self-adaptive systems requirement. Requirements Engineering 15 (2), 177–196, 2010
- [27] Bencomo N., Hallsteinsen S., Almeida E., Santana. A. View of the Dynamic Software Product Line Landscape, Computer 45(10), 36-41, 2012
- [28] Capilla R., Bosch J. The Promise and Challenge of Runtime Variability, Computer 44(12), 93-95, 2011
- [29] Garlan D., Schmerl B. Using architectural models at runtime: research challenges. Software Architecture 3047, 200– 205, 2004
- [30] Morin B., Barais O., Nain G., Jezequel J. Taming dynamically adaptive systems using models and aspects. In Proc. 31st Conference on Software Engineering, pp. 122–132, 2009
- [31] Vogel T., Giese H. Adaptation and abstract runtime models. In Proc. Workshop on Software Engineering for Adaptive and Self-managing Systems, 39–48, 2010

- [32] Schmid K., Eichelberger H., Goltz U., Lochau H. Evolving Adaptable Systems: Potential and Challenges. In Proc. 2nd Workshop Design For Future, 2010
- [33] Ayora C., Torres V., Weber B., Reichert M., Pelechano V.: Enhancing Modeling and Change Support for Process Families through Change Patterns. BMMDS/EMMSAD'13: 246-260, 2013
- [34] Alia M., Eliassen F., Hallsteinsen S., Stav S. MADAM: towards a flexible planning-based middleware. In Proc. SEAM 2006
- [35] Hallsteinsen S., Geihs K., Paspallis N., Eliassen F., et al. A development framework and methodology for self- adapting applications in ubiquitous computing environments. Systems and Software, 85(12), 2840-2859, 2012
- [36] Khan M. Unanticipated Dynamic Adaptation of Mobile Applications. PhD thesis, University of Kassel, Distributed Systems Group, Kassel, Germany, 2010
- [37] Epifani I., Ghezzi C., Mirandola R., Tamburrelli G. Model evolution by run-time parameter adaptation. In Proc. ICSE, 111-121, 2009
- [38] Weyns D., Andersson J. On the challenges of self-adaptation in systems of systems. In Proc. SESoS 2013, 47-51, 2013
- [39] Kephart J., Chess D. The vision of autonomic computing. Computer 36(1), 41-50, 2003
- [40] Zambonelli F., Viroli M. A survey on nature-inspired metaphors for pervasive service ecosystems. Pervasive Computing and Communications 7(3), 186-204, 2011
- [41] Andersson J., Baresi L., Bencomo N., Lemos N., et al. Software Engineering Processes for Self-Adaptive Systems. Software Engineering for Self-Adaptive Systems II, 51-75, 2010
- [42] Bures T., De Nicola T., Gerostathopoulos I., Hoch N., et al. A Life Cycle for the Development of Autonomic Systems: The e-Mobility Showcase. In Proc. 3rd Awareness Workshop @ IEEE SASO, 2013
- [43] Negri A., Trousse B., Senach B. Ideation of IoT services with citizen: coupling GenIoT and AloHa! Methods, 2012. URL <http://www.elliot-project.eu/sites/default/files/Ideation-of-IoT.pdf>
- [44] Sulistyo S.. Software Development Methods in the Internet of Things. ICT-EurAsia, 50-59, 2013