

Propuesta de proyecto del Trabajo Fin de Máster (TFM)

Máster Universitario:

- Investigación e Innovación en Inteligencia Computacional y Sistemas Interactivos (MU I2-ICSI)

Título: Análisis de sistemas basado en Líneas de Producto de Redes de Petri

Estudiante: GARCÍA-BERMEJO MAZORRA, Víctor

Director: GÓMEZ MARTÍNEZ, María Elena

Empresa o Institución del director: Grupo MISO, Universidad Autónoma de Madrid

Correo electrónico institucional: mariaelena.gomez@uam.es

Curso: 2020-21

Fecha: 26 / 03 / 2021

1. Introducción

Un sistema se puede definir como un conjunto de elementos que están relacionados entre sí en base a unas normas y leyes. Partiendo de esta línea, existen infinidad de adjetivos para los sistemas dependiendo de su función o contexto, como por ejemplo crítico –aquel en el cual un fallo puede propiciar consecuencias fatales- o concurrente –sistema el cual contiene varios elementos que pueden hacer su trabajo en paralelo- pero su principal elemento común es que todos ellos pueden ser representados como el conjunto de las representaciones individuales de sus elementos.

Una de las actividades cruciales de los sistemas concurrentes y críticos es la evaluación, validación y verificación de la seguridad. Un sistema debe cumplir su misión en cualquier problema de seguridad (viéndola como *safety* en inglés), teniendo la capacidad de recuperarse en caso de fallo. Además, la verificación de las propiedades de seguridad durante el desarrollo de un sistema supone un gran aumento en sus costes. Para intentar asegurar que se cumple esta actividad, una forma de analizar y modelar sistemas es utilizando redes de Petri.

Acotando el problema, en determinadas ocasiones y con el fin de abaratar costes de producción manteniendo calidad, se generan un conjunto de productos software que tienen un conjunto de elementos comunes, pero que sin embargo, en el resto de los elementos pueden diferir de forma variable. Este conjunto de sistemas similares es denominado Línea de Productos Software (SLP). Un ejemplo claro de uso de SLP es la industria automotriz donde se genera un software muy semejante pero con matices para cada coche [1].

La evaluación de la seguridad de una familia SPL se realiza de dos formas. En la primera mediante la generación o el análisis de todos los productos de la SPL. Este caso supone un gran problema a nivel computacional porque habría que hacer un análisis individual sobre cada miembro de la familia. En la segunda forma se analizarían las propiedades del SPL a la vez. Para ello, se utilizan distintas técnicas, como SAT solvers [2], Constraint Satisfaction Problems [3], y model checking [4]. Sin embargo, con estos métodos, no se tiene en cuenta las propiedades dinámicas de una Línea de Productos de Redes de Petri (PNPL) [5].

En las aproximaciones basadas en Desarrollo Dirigido por Modelos, para la especificación de la seguridad de los sistemas software, actualmente se utiliza el denominado lenguaje *Object Constraint Language* (OCL) [6] [7]. Mediante este lenguaje se expresan en el Lenguaje de Modelado Unificado (UML) [8] restricciones o contratos que cumplirán el modelo diseñado. Un contrato sirve para añadir pre- o post-condiciones a los diversos elementos de un modelo y sus relaciones [9] [10]. Con ello, se podrá efectuar una validación y una verificación temprana de un sistema crítico, pudiendo detectar así, los problemas potenciales durante la fase de diseño. Paralelamente a esta mejora en la fase de diseño, la industria del software está empezando a utilizar métodos formales de la ingeniería del software como el model checking [4]. Estas técnicas son utilizadas en sistemas software que pueden ser analizados y modelados mediante una red de Petri [11].

2. Objetivos y planteamiento

Este Trabajo de Fin de Máster propone la generación de una serie de módulos y *plug-ins* para el análisis basado en Líneas de Producto de Redes de Petri. Este proyecto se dividirá en dos fases. En la primera de ellas, se cumplirán una serie de objetivos orientados a la investigación para comprender de forma correcta el ámbito de la herramienta:

- Estudio y análisis de sistemas con variabilidad. Inicialmente este estudio se centrará en la búsqueda y el entendimiento de trabajos centrados en el UML con variabilidad. El principal motivo por el cual se debe aprender, es porque los diagramas por los cuales está compuesto un sistema software, se pretenden que sean considerados los recursos de entrada que manejará este proyecto puesto que sirven para la representación de gran parte de los sistemas software. Adicionalmente se estudiará Líneas de Productos de *Business Process Management and Notation* (BPMN) que es una forma de representar procesos en sistemas de negocios.
- Comprensión del estado del arte de los programas cuya finalidad es el diseño de redes de Petri. Puesto que la salida del proyecto será una serie de redes de Petri a analizar, éstas deberán ser compatibles con los programas que tienen esa capacidad.

Una vez cumplidos estos objetivos, se propondrá un desarrollo de la herramienta basado en el ciclo de desarrollo incremental iterativo, donde se pretende ir implementando y acoplando en cada iteración nuevos subsistemas que se obtengan de las investigaciones anteriores (UML y BPMN). Previamente a este desarrollo, se hará una estimación y planificación del mismo obteniendo como resultado un diagrama de Gantt donde se verán representadas las iteraciones y se harán estimaciones de tiempo.

- Fases de análisis y diseño de la herramienta. Mediante metodologías de la ingeniería del software, se obtendrán los requisitos funcionales y no funcionales que se tendrán en cuenta a lo largo del desarrollo del proyecto y se hará un prototipo con los subsistemas de transformación a red de Petri resultantes de los prerrequisitos y el análisis. Este diseño base se irá haciendo más complejo conforme se adhieran nuevos subsistemas en las sucesivas iteraciones.
- Fase de implementación. Mediante los conocimientos obtenidos en la asignatura de Desarrollo de Software Dirigido por Modelos, se implementarán en cada iteración las transformaciones consecuentes para obtener las redes de Petri correspondientes a los diagramas de entrada. Destacar que se trabajará con el entorno de programación Eclipse y con la extensión llamada Eclipse Modeling Framework (EMF).
- Fase de pruebas. Para finalizar cada iteración, se harán pruebas para la comprobación de que los subsistemas y módulos transforman de forma correcta a Líneas de Productos de Redes de Petri. Estas redes de Petri resultantes, se deberán poder analizar mediante algún programa externo al proyecto para poder verificar la seguridad del sistema crítico.

El resultado final de todos estos objetivos, es obtener una aplicación de transformación de los diversos diagramas UML que componen el diseño de un sistema crítico, en redes de Petri o Líneas de Producto de redes de Petri para su posterior análisis en una herramienta externa como podrían ser GreatSPN o WoPeD.

3. Metodología y plan de trabajo

Tal y como se ha propuesto en el apartado anterior, se va a seguir un ciclo de desarrollo iterativo incremental. Por ello, se debe tener en cuenta que de la tarea T4 hasta la tarea T7, no se efectúan en orden cronológico. En la siguiente tabla se describen las tareas:

Tareas y subtareas	Horas
T1. Estudio y análisis de sistemas con variabilidad	
T1.1 UML	20
T1.2 UML con variabilidad (Líneas de Producto UML)	15
T1.3 BPMN con variabilidad	10
T2. Estudio y análisis de redes de Petri para modelado de la variabilidad	15
T3. Actividades de pre-desarrollo	
T3.1 Generación de diagrama de Gantt	20
T4. Análisis de la herramienta	
T4.1 Generación de documento de especificación de requisitos	25
T5. Diseño de la herramienta	
T5.1 Subsistemas de herramientas generales de PNML	10
T5.2 Subsistemas de transformación de UML	25
T5.3 Subsistemas de transformación de BPMN	5
T6. Implementación de la herramienta	
T6.1 Subsistemas de herramientas generales de PNML	30
T6.2 Subsistemas de transformación de UML	50
T6.3 Subsistemas de transformación de BPMN	15
T7. Pruebas de la herramienta	
T7.1 Comprobación de que las redes de Petri resultantes son correctas	
T7.1.1 Subsistemas de herramientas generales de PNML	10
T7.1.1 Subsistemas de transformación de UML	10
T7.1.3 Subsistemas de transformación de BPMN	5
T7.2 Pruebas de análisis para verificar y validar las redes de Petri en programas externos	
T7.2.1 Subsistemas de herramientas generales de PNML	5
T7.2.2 Subsistemas de transformación de UML	5
T7.2.3 Subsistemas de transformación de BPMN	5
T8 Generación memoria final	20
TOTAL DE HORAS / TOTAL HOURS	300

4. Bibliografía inicial

- [1] L. Northrop y P. Clements, «Software Product Lines: Practices and Patterns,» Addison-Wesley Longman Publishing, Boston, 2002.
- [2] D. Berre y A. Parrain, «The Sat4j library, release 2.2,» *JSAT*, vol. 7, nº 2-3, pp. 59-6, 2010.
- [3] K. N. Brown y I. Miguel, «Chapter 21 - uncertainty and change,» *Handbook of Constraint Programming*, ser. *Foundations of Artificial Intelligence*, vol. 2, pp. 731-760, 2006.
- [4] E. Clarke, «Model checking,» de *International Conference on Foundations of Software Technology and Theoretical Computer Science*, Berlin, 1997.
- [5] M. Rosa, W. v. d. Aalst, M. Dumas y F. Milani, «Business process variability modeling: A survey,» *ACM Comput Surv.*, vol. 50, nº 1, pp. 2:1-2:45, 2017.
- [6] «OMG,» Febrero 2014. [En línea]. Available: <https://www.omg.org/spec/OCL>.
- [7] J. Cabot y M. Gogolla, «Object Constraint Language (OCL): A Definitive Guide,» de *Formal Methods for Model-Driven Engineering*, Berlin, Springer, 2012.
- [8] OMG, «Welcome To UML Web Site!,» OMG, [En línea]. Available: <https://www.uml.org/>.
- [9] I. Porres y I. Rauf, «Generating class contracts from UML protocol statemachines,» *Proceedings of the 6th International Workshop on Model-Driven Engineering, Verification and Validation*, nº 8, pp. 1-10, 2009.
- [10] W. Damm, H. Hungar, B. Joske, T. Peikenkamp y I. Stierand, «Using contract-based component specifications for virtual integration testing and architecture design,» de *2011 Design, Automation & Test in Europe*, 2011.
- [11] E. Gómez-Martínez, R. J. Rodríguez, C. Benac-Earle, L. Etxeberria y M. Illarramendi, «A methodology for model-based verification of safety contracts and performance requirements,» *Journal of Risk and Reliability*, vol. Proceedings of the Institution of Mechanical Engineers, nº 232, pp. 227-247, 2018.

5. Propuesta de tribunal

Propuesta 1

Nombre y Apellidos: Juan de Lara Jaramillo

Entidad y Departamento: Escuela Politécnica Superior, Departamento de Ingeniería

Informática

Grupo de Investigación: Grupo de Modelado e Ingeniería del software

Correo electrónico institucional: juan.delara@uam.es

Propuesta 2

Nombre y Apellidos: Esther Guerra Sánchez

Entidad y Departamento: Escuela Politécnica Superior, Departamento de Ingeniería Informática

Grupo de Investigación: Grupo de Modelado e Ingeniería del software

Correo electrónico institucional: esther.guerra@uam.es

Propuesta 3

Nombre y Apellidos: Silvia Teresita Acuña Castillo

Entidad y Departamento: Escuela Politécnica Superior, Departamento de Ingeniería Informática

Grupo de Investigación: Grupo de Herramientas Interactivas Avanzadas

Correo electrónico institucional: silvia.acunna@uam.es

Propuesta 4

Nombre y Apellidos: Álvaro Manuel Ortigosa Juárez

Entidad y Departamento: Escuela Politécnica Superior, Departamento de Ingeniería Informática

Grupo de Investigación: Grupo de Herramientas Interactivas Avanzadas

Correo electrónico institucional: alvaro.ortigosa@uam.es

Propuesta 5

Nombre y Apellidos: Francisco Jurado Monroy

Entidad y Departamento: Escuela Politécnica Superior, Departamento de Ingeniería Informática

Grupo de Investigación: Grupo de Herramientas Interactivas Avanzadas

Correo electrónico institucional: francisco.jurado@uam.es

Propuesta 6

Nombre y Apellidos: Rosa Carro

Entidad y Departamento: Escuela Politécnica Superior, Departamento de Ingeniería Informática

Grupo de Investigación: Grupo de Herramientas Interactivas Avanzadas

Correo electrónico institucional: rosa.carro@uam.es

El estudiante

A handwritten signature in black ink on a light gray background. The signature is stylized and appears to read 'Víctor'.

Fdo.: García-Bermejo Mazorra, Víctor

Vº Bº del director o directores

Fdo.: Gómez Martínez, María Elena