Un enfoque declarativo para modelar el comportamiento en sistemas reactivos

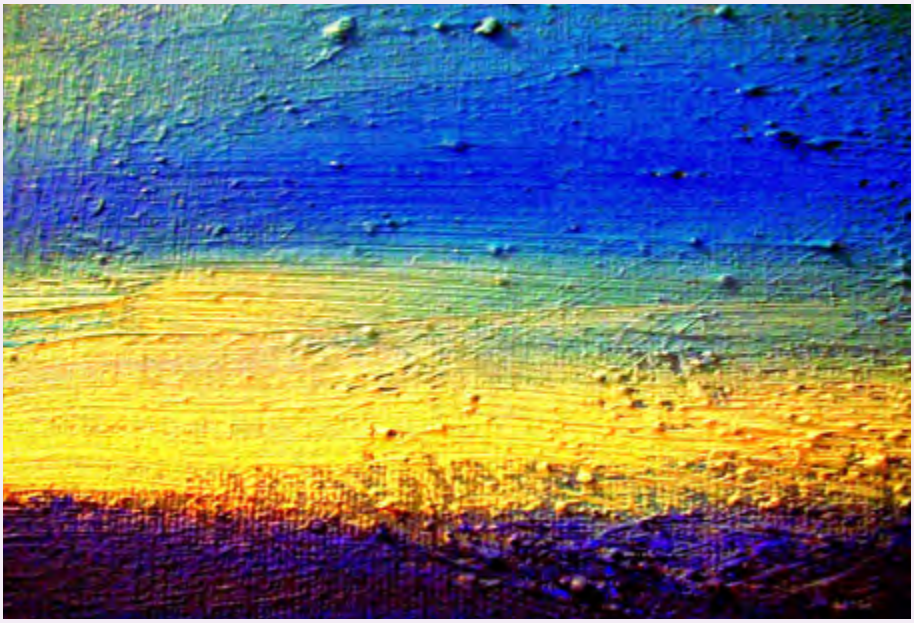
Tesis Doctoral / Universidad Nacional del Sur, 2013

Fernando Asteasuain
Ingeniería en Informática, Universidad Nacional de Avellaneda

Directores:
Dr. Pablo Fillottrani, Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur, Bahía Blanca
Dr. Víctor Braberman, Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires.

Jurado
Dr. Diego Martínez, Universidad Nacional del Sur, Bahía Blanca
Dr. Nazareno Aguirre. Universidad Nacional de Río Cuarto, Córdoba.

Fecha de defensa de la tesis: 20/12/2013



Resumen

Existe consenso en la comunidad académica y de ingenieros de *software* que es fundamental entender, modelar y describir el comportamiento del *software* complejo desde etapas tempranas del desarrollo. El paradigma de descripción declarativa, basado en el modelado de las propiedades y objetivos esenciales de los objetos y agentes, posee características especialmente prometedoras para este tipo de desafíos. Sin embargo, las alternativas existentes en

este paradigma son lógicas temporales que poseen limitaciones prácticas y teóricas. Asimismo, la verificación formal de propiedades sigue siendo en la actualidad uno de los mayores desafíos para la transferencia de tecnología de verificación de *software* como *model checking*. Los usuarios de estas técnicas deben enfrentar el desafío de expresar propiedades en el lenguaje formal usado en la herramienta de especificación. Dos de las aproximaciones más utilizadas son lógicas temporales como LTL, y notaciones operacionales basadas en autómatas. Ambas aproximaciones requieren usuarios "expertos" o con conocimientos avanzados para poder expresar, describir y validar la propiedad de interés. Todo esto indica la necesidad de contar con un lenguaje formal declarativo para expresar propiedades, que sea lo suficientemente expresivo y que permita realizar tareas de validación de manera simple e intuitiva.

En esto contexto, el objetivo global de esta tesis fue elaborar un enfoque de modelado declarativo, capaz de manejar distintos niveles de abstracción, con semántica precisa y clara, para modelar el comportamiento de sistemas reactivos. El objetivo específico fue el desarrollo de un lenguaje de modelado declarativo, basado en notaciones gráficas (escenarios), capaz de modelar y describir el comportamiento de sistemas reactivos. El lenguaje posee una semántica y sintaxis clara y precisa, con la posibilidad de realizar razonamiento automático, modelado incremental, y validación intuitiva de propiedades.

Problema y marco teórico

El modelado declarativo resulta un enfoque atractivo y natural para capturar requerimientos sobre comportamiento. Sin embargo, en muchos casos, la descripción y validación de propiedades es una tarea compleja, aun para personas con conocimiento en el tema. En este contexto la notación en escenarios representa una alternativa. Su representación gráfica facilita el entendimiento con usuarios y clientes, el comportamiento y casos de tests son más simples de confirmar o refutar, y complejos flujos de control e interacción pueden describirse fácilmente. Sin embargo, las notaciones de escenarios no han sido concebidas como lenguajes de especificación declarativos. Respecto de la especificación de propiedades, un importante problema en la utilización de lenguajes formales como LTL, o notaciones basadas en autómatas es el grado avanzado de conocimiento que se requiere para poder expresar las propiedades de interés y, a su vez, validar que la propiedad está realmente describiendo la propiedad que el usuario tiene en mente. Este problema aún persiste cuando se utilizan patrones de especificación. La evidencia indica que para realizar tareas de validación no alcanza con analizar la descripción en lenguaje natural del patrón elegido, sino que debe analizarse su traducción a un lenguaje formal. Esto sugiere que el lenguaje de especificación debe ser fácil de usar, y lo suficientemente expresivo para permitir a usuarios expertos y no expertos usarlo apropiadamente. Lenguajes formales como LTL o notaciones basadas en autómatas no logran satisfacer por completo la validación de propiedades. Este contexto ilustra la importancia y necesidad de contar con un lenguaje declarativo formal para expresar propiedades.

Metodología y marco de trabajo

Como primera instancia se definieron atributos de calidad deseables en un lenguaje formal para una correcta especificación y validación de propiedades. Dichos atributos son:

Sucinto, Facilidad de Verificación, y Modificabilidad. Sucinto se refiere a qué tan conciso puede ser expresar una propiedad. Este atributo es esencial para poder hacer más sencilla la validación de propiedades. La facilidad de verificación se subdivide a su vez en dos subatributos: facilidad de comparación, y de complemento. El primero establece que las propiedades deben ser fáciles de comparar, de distinguir, y de entender la relación entre ellas. La segunda se refiere a que debe ser sencillo entender las distintas situaciones que llevan a la violación de la propiedad. Esta información es de gran utilidad a la hora de validar una propiedad. Finalmente, modificabilidad se refiere a la habilidad para poder manipular una propiedad para poder adaptarla a nuevas contextos de aplicación.

Teniendo en cuentas estas características, el paso siguiente fue el desarrollo de un lenguaje declarativo. Dicho lenguaje se denominó FVS (*FeatherWeight Visual Scenarios*). FVS, a pesar de su simpleza, es lo suficientemente poderoso como para describir todos los patrones de especificación. Por un lado, su naturaleza gráfica y visual ayuda al usuario a concentrarse únicamente en las propiedades y no tratar con las complicaciones de su formalización. Los escenarios son construidos usando una cantidad mínima de operadores simples, obteniendo así especificaciones concisas y compactas. Relaciones lógicas y semánticas pueden fácilmente deducirse directamente de los escenarios, aumentando la posibilidad de razonar sobre las propiedades. El lenguaje cuenta con la posibilidad de construir automáticamente anti-escenarios, los cuales ayudan al usuario en la especificación de propiedades examinando el comportamiento que lleva a una violación de una propiedad. Por último, la especificación de propiedades es flexible, y puede adaptarse fácilmente a diferentes contextos de aplicación.

Para validar la aplicabilidad y usabilidad de FVS respecto de los atributos definidos, se tomó como caso de estudio los patrones de especificación. Se compararon las especificaciones de todos los patrones de especificación de FVS contra las fórmulas lógicas LTL propuestas para dichos patrones y contra notaciones basadas en autómatas. Como conclusión de dicha comparación se pudo establecer que las especificaciones en FVS son más concisas, más simples de comparar, analizar y modificar. Esencialmente, FVS logra manejar apropiadamente todas las tareas que involucra la validación de propiedades.

Para incorporar nociones avanzadas de modularización, se exploró la posibilidad de definir a FVS como un lenguaje de modelado orientado a aspectos. En los últimos años, la orientación a aspectos ha surgido como un enfoque interesante para tratar con la complejidad en la descripción de entidades de *software*. Sin embargo, algunos autores han señalado dificultades para aplicar la filosofía orientada a aspectos en notaciones operacionales. Muchas aproximaciones orientadas a aspectos terminan recayendo en mecanismos de composición (*weaving*) sintácticos, sin una semántica clara. FVS ataca estos problemas brindando una mayor flexibilidad para desacoplar la interacción entre los aspectos y el sistema bajo análisis.

El último paso de la investigación llevada a cabo en la tesis consistió en aumentar el poder expresivo de FVS para poder denotar lenguajes omega-regulares, una característica distintiva en este tipo de aproximaciones. Para tal fin se introdujeron tipos de eventos abstractos que permiten razonar en un nuevo nivel de abstracción. Se implementó una traducción de escenarios FVS a autómatas de Buchi lo cual permitió integrar FVS con otras herramientas basadas en otros tipos de especificaciones. Los resultados permitieron comprobar que los autómatas generados por la traducción propuesta son comparables en tamaño a autómatas conocidos en la literatura, y fueron útiles para detectar errores relevantes en la especificación de protocolos de comunicación.