

RESUMEN DE TESIS DOCTORAL

Animación de Especificaciones OASIS Mediante Redes de Petri Orientadas a Objeto

Tesista: Pedro Sánchez Palma
Universidad Politécnica de Cartagena
Campus Muralla del Mar s/n 30202
Cartagena, España
e-mail: Pedro.Sanchez@upct.es

Asesor: Isidro Ramos Salavert
Universidad Politécnica de Valencia, España
e-mail: iramos@dsic.upv.es

Resumen

La validación mediante animación permite el desarrollo sistemático de especificaciones software claras, concisas, precisas y sin ambigüedad, aspectos presentes en los métodos formales de la ingeniería del software pero ausentes en el prototipado tradicional.

Además, la calidad de los modelos conceptuales depende en gran medida del grado de integración de los usuarios en el proceso de desarrollo. Este artículo describe los objetivos y metas conseguidas en esta línea de trabajo de la Ingeniería del Software y como resultado de un trabajo de Tesis Doctoral.

Palabras Clave: Requerimientos de Ingeniería, Validación por Animación y Métodos Formales.

1 Introducción

Los sistemas hardware y software cambian continuamente y de manera inevitable tanto en tamaño como en funcionalidad. Debido al incremento en la complejidad, la posibilidad de cometer errores va también en aumento. Gran parte de dichos errores suponen un coste económico en tiempo, e incluso hasta en vidas humanas.

Uno de los mayores objetivos de la Ingeniería del Software es el poder facilitar a los desarrolladores la construcción de sistemas que operen con cierto grado de confianza, independientemente de la complejidad asociada. Una forma de conseguir este objetivo es usando métodos formales en forma de lenguajes, técnicas y herramientas con base matemática que permitan tanto la especificación como la verificación de dichos sistemas. A priori, el uso de métodos formales no garantiza la corrección aunque puede incrementar de manera considerable el conocimiento del sistema al detectar inconsistencias, incompletitud, ambigüedad, etc.

La Ingeniería de Requisitos (Wieringa, 1996) tiene por objetivo obtener una especificación de los requisitos del sistema que se desea construir a cierto nivel de abstracción y cumpliendo ciertas propiedades. Así, el modelo conceptual representa un enlace entre el espacio del problema y el espacio de la solución al expresar, haciendo uso de un lenguaje de especificación, los requisitos funcionales del sistema. De esta forma, el modelo conceptual constituye la base inicial para las tareas que siguen en el desarrollo de un sistema.

Tradicionalmente, las especificaciones de los requisitos se han venido expresando en notaciones semi-formales, aunque cada vez son más las que se escriben usando lenguajes de especificación formal. Las especificaciones formales poseen ciertas ventajas adicionales como consecuencia de tener una sintaxis y semántica bien definidas. Una especificación formal permite explicitar todos los detalles minimizando o eliminando las omisiones, ambigüedades o inconsistencias en la especificación. Además, es posible "razonar" acerca de las propiedades de una especificación formal, en particular, verificarla y validarla con respecto de los requisitos iniciales.

Si además el lenguaje de especificación es ejecutable entonces se obtiene una ventaja adicional: la especificación representa no sólo el modelo conceptual sino también el modelo de ejecución del software que se va a implementar.

El desarrollo tradicional del software presenta el problema de que la implementación final del sistema es la primera versión precisa del mismo que puede ser ejecutada. De esta forma, validar el sistema sólo es posible cuando se han tomado ya decisiones de diseño, lo que implica gran cantidad de aspectos de implementación. Como consecuencia, cuando se detectan errores se necesita rehacer el diseño.

Si las especificaciones son ejecutables entonces es posible la validación en un nivel de abstracción elevado incrementándose la corrección y confianza en el software y reduciendo considerablemente los tiempos y costes de desarrollo. Con la validación en las primeras fases del ciclo de vida de desarrollo de software, no sólo se consigue detectar errores de manera temprana sino que, aquellos requisitos que no están claros pueden ser concretados y completados a partir de la interacción con la ejecución de la especificación.

La construcción del modelo conceptual es pues un proceso de descubrimiento tanto para el analista como para el usuario. La idea es construir el modelo conceptual de manera iterativa, con refinamientos sucesivos y con la participación del usuario. En cada iteración el analista verifica el modelo y lo valida con el usuario. Una vez se alcanza el acuerdo entre las partes involucradas en el proceso, el modelo conceptual se utiliza como contrato y entrada a las fases que siguen en el desarrollo. De esta forma se reduce el coste asociado a modificaciones posteriores o a requisitos parcialmente especificados.

Cuando el lenguaje de especificación que se usa no es directamente ejecutable la validación del modelo requiere un esfuerzo adicional. Así pues, la validación de las especificaciones de requisitos por animación de estos y usando técnicas de prototipado es el punto de partida y la motivación de este trabajo.

2 Marco de Trabajo

Este trabajo de Tesis (Sánchez Palma, 2000) se ha generado en el grupo de investigación "Modelado Conceptual Orientado a Objetos", perteneciente al Departamento de Sistemas Informáticos y Computación de la Universidad Politécnica de Valencia. Una de las principales actividades llevadas a cabo en los últimos años por este grupo ha sido el desarrollo de OASIS Open and Active Specification of Information Systems (Pastor, 1995, Letelier, 1998), un modelo formal para la construcción de modelos conceptuales que sigue el paradigma orientado a objeto. Desde la perspectiva de OASIS, un sistema de información es una sociedad de objetos autónomos y concurrentes que interactúan entre sí. Dentro de esta línea, el propósito de este trabajo de Tesis es analizar y realizar la generación automática de código para animar especificaciones OASIS. Se han utilizado redes de Petri como formalismo destino en dicha generación, dadas las buenas propiedades de las mismas en cuanto a la posibilidad de simulación, demostración de propiedades y disponibilidad de herramientas.

Los trabajos que han posibilitado la consecución de esta Tesis se enmarcan en los dos proyectos siguientes de la Comisión Interministerial de Ciencia y Tecnología (CICYT): OASIS (1994-1997, TIC94-0557) y MENHIR (Participantes en el Proyecto Menhir, 1998) (1997-2000, TIC97-0593-C05-01).

3 Objetivos

El objetivo del trabajo ha sido establecer las bases teóricas y prácticas necesarias para la validación automática de especificaciones OASIS haciendo uso de las redes de Petri.

De forma más específica se pueden considerar logrados los siguientes objetivos:

- Extensión y compleción el lenguaje OASIS. El trabajo asociado se ha enmarcado dentro del desarrollo de la versión \$3.0\\$ del lenguaje (Letelier, 1998). El punto de partida fue la versión \$2.2\\$ (Pastor, 1995) en la que se llevaron a cabo diversas modificaciones y ampliaciones. Algunas mejoras conseguidas son: la utilización uniforme de la Lógica Dinámica para expresar las propiedades de una clase OASIS; la incorporación de la perspectiva cliente; la redefinición del concepto de proceso (incluyendo operaciones y protocolos) y el enriquecimiento de los mecanismos usados para definir clases complejas agregación y especialización).
- Definición un modelo abstracto de ejecución para el lenguaje OASIS que constituye la semántica operacional del mismo.
- Establecimiento de las pautas de traducción necesarias para pasar de OASIS redes de Petri. Se ha implementado un programa traductor que genera automáticamente una red de Petri equivalente partiendo de una especificación OASIS.
- Construcción de una versión inicial de un entorno para la animación automática de especificaciones OASIS enmarcado dentro del proceso de ingeniería de requisitos que incluya la animación de modelos conceptuales de forma automática y considerando el uso de escenarios.

Si estudiamos las soluciones existentes observamos las siguientes deficiencias:

1. La funcionalidad e integración de los módulos desarrollados para validación es escasa. En la mayoría de los casos se trata de versiones preliminares.
2. No se proporciona en ningún caso la semántica operacional (modelo de ejecución) de los distintos entornos.
3. La implementación está hecha en lenguajes imperativos (el lenguaje C++ suele ser habitual en estos casos) con lo que la comprobación de que la implementación es fiel a la semántica del modelo conceptual de partida se ve dificultada.
4. No se producen trazas como resultado de la animación con lo que el usuario debe deducir las interacciones habidas entre los objetos.

El trabajo realizado sobre animación de especificaciones OASIS se sitúa ventajosamente, con respecto de los trabajos anteriores, por los siguientes motivos:

1. OASIS tiene asociado una semántica declarativa basada en Lógica Dinámica (Meyer, 1998) y una semántica operacional que constituye el modelo de ejecución del lenguaje. Para todos los conceptos del lenguaje OASIS se presenta un procedimiento de traducción a esta lógica (a excepción de transacciones, agregación y comunicación sincrónica). Tareas como verificación, validación y generación automática de código se ven favorecidas, en consecuencia, dado el marco formal que las integra.
2. La animación viene dada por la ejecución de una implementación concurrente (en redes de Petri) de la semántica del modelo OASIS, evitando la necesidad de monitores artificiales encargados de "repartir" el hilo de ejecución del sistema.
3. La interacción con el animador se realiza mediante técnicas de escenarios lo que posibilita la comunicación usuario-analista. En los trabajos mencionados anteriormente, el resultado de la animación se presenta para cada objeto, lo que conlleva un esfuerzo adicional al tener que componer la interacción global entre todos los objetos participantes.
4. Las redes de Petri son un formalismo que posibilita la demostración de la equivalencia entre una especificación y su correspondiente implementación.
5. Es posible particionar un problema en dos partes: una

orientada al modelado conceptual con OASIS y otra representada haciendo uso directamente de las redes de Petri. De esta forma, los aspectos de concurrencia que se modelen más fácilmente con las redes de Petri pueden representarse directamente.

6. El traductor implementado tiene como punto de partida el conjunto de fórmulas en Lógica Dinámica de una especificación. Esto posibilita la evolución futura de OASIS (en su sintaxis y/o semántica asociada) minimizando o eliminando la necesidad de modificar el código de dicho traductor. Esto es, la Lógica Dinámica aparece como punto de referencia para posibles implementaciones futuras.

4 Resumen del Trabajo

El marco de trabajo se resume en los siguientes puntos:

Modelo. OASIS es el enfoque considerado para representar el modelo conceptual correspondiente a los requisitos del usuario.

Animación. Se obtiene de forma automática un prototipo orientado a la validación de requisitos funcionales especificados usando el lenguaje OASIS. La implementación se realiza utilizando redes de Petri y el entorno de animación elegido para estas redes es la herramienta CODESIGN¹

Escenarios. Se considera el uso de escenarios para representar parte de los requisitos del usuario y para mostrar las trazas obtenidas tras la animación de la especificación.

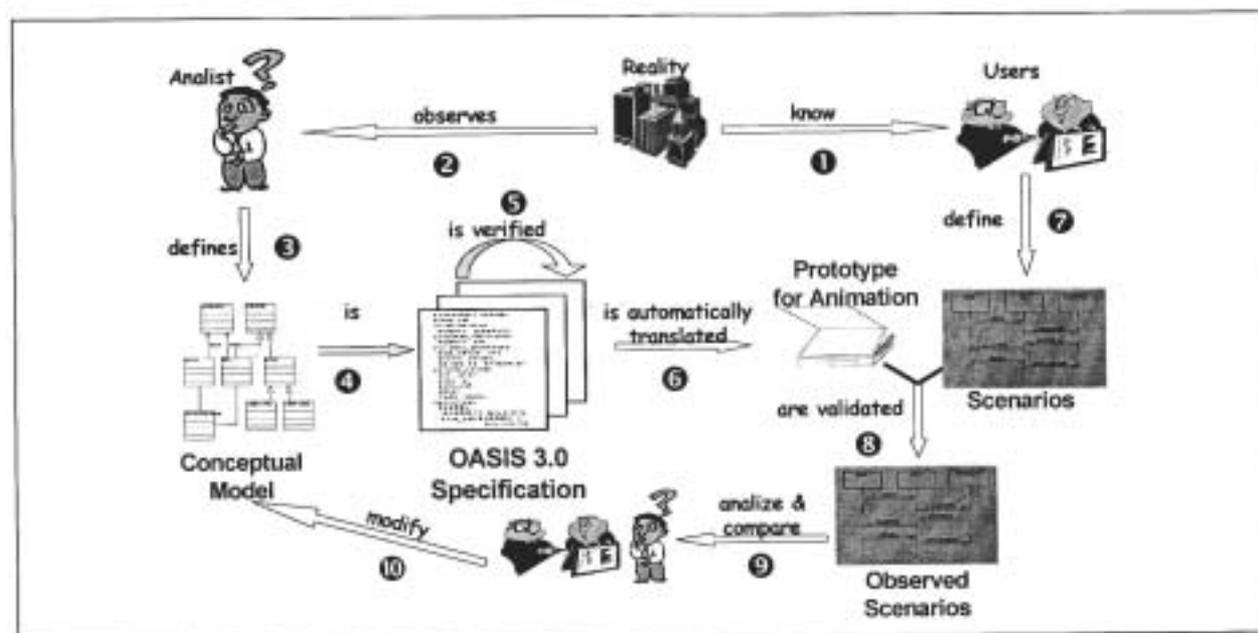


Figura: 1 Modelado Conceptual basado en OASIS

¹ <http://www.nik.ee.ethz.ch/~naedele/moses/Distribution250.html>

La calidad de un modelo conceptual depende en gran medida del grado de Integración del usuario en el proceso asociado a la captación de requisitos. La Figura 1 muestra el modelo de proceso considerado para OASIS cuando se sigue el enfoque planteado. El punto de partida son los usuarios que conocen la realidad (el sistema) que se desea modelar. La tarea del analista es llevar a cabo la captura de requisitos que permita obtener el modelo conceptual del sistema (probablemente con la intervención de los usuarios). Dicho modelo conceptual viene representado en alguna notación gráfica (por ejemplo, UML para aquellos aspectos de la notación que tengan una correspondencia clara con los conceptos OASIS). Un proceso de traducción lleva de la notación gráfica a la especificación OASIS para el sistema modelado. Las distintas tareas de verificación comentadas en las secciones anteriores son aplicables en este punto. Cuando la especificación OASIS está libre de errores sintácticos y/o semánticos entonces se procede a obtener automáticamente el prototipo para animación. Previamente, los usuarios definen un conjunto de escenarios (casos de prueba) que consideran representativos y que describen parcial o totalmente el comportamiento del sistema. La ejecución del prototipo (usando como datos de entrada los escenarios) da lugar a resultados representados en la misma notación. Usuarios y analistas estudian las diferencias obtenidas lo que da lugar a posibles modificaciones en el modelo conceptual de partida. El ciclo continúa hasta que se obtiene la conformidad de los usuarios pasando el modelo conceptual a estar verificado y validado. Las siguientes consideraciones aportan un mayor detalle al proceso presentado:

- Por simplificar la figura, no se ha representado el que los escenarios (confeccionados por usuarios y analistas) pueden ser también punto de entrada para la obtención del modelo conceptual.
- Los escenarios confeccionados por los usuarios pueden presentar información parcial del sistema que aparezca después, como resultado de la animación, de manera interrelacionada. Este aspecto justifica, en parte, la presencia de los usuarios en el proceso de validación. Si lo único que espera el analista es obtener los escenarios que los usuarios han confeccionado entonces dichos usuarios son prescindibles en el proceso. Dicho de otra forma, puede haber resultados de la animación que se presenten como indicativos de error cuando en realidad son situaciones particulares que los usuarios no reflejaron en los escenarios o bien representaron de forma parcial.
- Los escenarios deben incluir alguna representación del estado que facilite al analista el proceso de comparación. Por simplificar, en cada paso de la vida de cada objeto puede mostrarse valor antiguo y nuevo para cada atributo modificado.

- La validación puede empezar con un subconjunto del modelo total. Es más, es posible particionar el modelo de forma que se ejecuten prototipos para cada una de las particiones calculadas. Es obvio que la única condición que se impone es la consistencia del modelo a partir del cual se va a generar el prototipo.
- El prototipo se construye en un entorno de implementación concurrente lo que permite una correspondencia más natural de los conceptos de OASIS con los elementos del prototipo. Cuando el entorno no es concurrente se hace preciso un monitor que serialice el paralelismo inherente a una especificación OASIS.

5 Conclusiones

Este trabajo apoya el argumento de que las especificaciones formales dan soporte al proceso de especificación de requisitos al permitir que las propiedades recogidas puedan ser examinadas dentro de un marco preciso de trabajo basado en la validación mediante animación. De esta forma se resaltan los puntos fuertes de los métodos formales y el prototipado, al tiempo que se mitigan los problemas clásicos.

Podemos resumir el aporte en los siguientes items:

1. Se ha desarrollado la semántica declarativa de OASIS incorporando la noción de paso como conjunto consistente de acciones llevadas a cabo concurrentemente que determinan el cambio de estado.
2. Se ha establecido un modelo de ejecución de OASIS (Letellier, 1999) que es punto de referencia para la animación de sociedades de objetos OASIS. Este modelo de proceso es concurrente diferenciándose así de otras propuestas industriales con orientación secuencial y monitorizada.
3. Se ha establecido una correspondencia entre los conceptos de clase OASIS y la Lógica Dinámica utilizada (Meyer, 1998). Se han considerado de OASIS las evaluaciones, precondiciones, disparos, protocolos, operaciones, atributos derivados y restricciones de integridad (estáticas y dinámicas). Al así hacerlo, se consigue integrar en un marco homogéneo los distintos aspectos del lenguaje y se proporciona un camino para la demostración de propiedades.
4. Se han sentado las bases teóricas para incorporar en el proceso de traducción a Lógica Dinámica las relaciones de especialización entre clases (particiones estáticas y dinámicas).
5. Se ha definido una arquitectura para la implementación de los conceptos incluidos en una clase OASIS. Con este enfoque, se ha construido un traductor de especificaciones OASIS (vistas como fórmulas en Lógica Dinámica) a redes de

Petri con lo que se consigue la ejecutabilidad de dichas especificaciones. El prototipo que se obtiene está orientado a validación por lo que aquellos requisitos no funcionales o bien relativos al interfaz con el usuario quedan fuera del proceso de animación.

6. Se ha presentado una propuesta de ambiente para la captura y validación de requisitos de manera incremental. Esta validación viene soportada por la ejecución de las redes de Petri correspondientes a una especificación OASIS. La interacción con los usuarios se basa en escenarios expresados como Diagramas de Secuencia. Este es el mecanismo que permite a los usuarios validar que se está construyendo el "producto correcto".

6 Trabajo Futuro

El trabajo como continuación del presente tiene una doble vertiente. Desde un punto de vista teórico, los siguientes son aspectos susceptibles de consideración futura:

1. Ampliar el modelo de ejecución (Letelier, 1999) para incorporar la comunicación síncrona.

2. Justificar la equivalencia entre el modelo OASIS y la implementación en redes de Petri. Este trabajo no es trivial dadas las dimensiones del grafo de alcanzabilidad asociado a las redes obtenidas en la traducción y la distancia conceptual que existe entre los formalismos presentados. En (Sánchez, 1997) se esboza un camino para demostrar este aspecto (se necesita una Función de Transformación que establezca la equivalencia entre la representación del estado y el marcado de la red).

Desde un punto de vista aplicado surgen las siguientes tareas:

1. Integrar los traductores implementados en un entorno de especificación gráfico para OASIS.

2. Construir un traductor de los distintos repositorios del lenguaje OASIS (en cada una de sus versiones) a Lógica Dinámica. Así, el animador puede ser útil para cualquier versión.

3. Construir un módulo de comparación de escenarios que automatice la detección de diferencias entre lo esperado y lo resultante.

4. Modificar el traductor de Lógica Dinámica a redes de Petri de forma que incorpore las fórmulas de derivación y las restricciones de integridad (estáticas y dinámicas).

5. Enlazar la aplicación CODESIGN con algún simulador externo de forma que pueda realizarse la validación de forma remota y puedan recogerse estadísticas de disparos, marcados, etc.

Referencias

Letelier P., Ramos I., Sánchez P., Pastor O. OASIS versión 3.0: Un Enfoque Formal para el Modelado Conceptual Orientado a Objeto. Servicio de Publicaciones de la Universidad Politécnica de Valencia, SPUPV-98.4011, 1998. www.dsic.upv.es/users/oom/oasis.html

Letelier P., Sánchez P., Ramos I. Un Modelo Abstracto de Ejecución para Animar Modelos Conceptuales. VI Congreso Internacional de Investigación en Ciencias Computacionales, páginas 80-91, Cancún México, Septiembre de 1999.

Participantes del Proyecto Menhir. MENHIR: Modelos, Entornos y Nuevas Herramientas para la Ingeniería de Requisitos. Actas de las III Jornadas de Investigación y Docencia en Bases de Datos, JIDBD'98, Valencia, 1998.

Meyer J.-J.Ch. A Different Approach to Deontic Logic: Deontic Logic viewed as a Variant of Dynamic Logic. In *Notre Dame Journal of Formal Logic*, vol. 29, pages 109-136, 1988.

Pastor O., Ramos I. OASIS versión 2 (2.2): A Class-Definition Language to Model Information Systems Using an Object-Oriented Approach. Servicio de Publicaciones Universidad Politécnica de Valencia, SPUPV-95.788, 1995.

Sánchez Palma, Pedro. Animación de Especificaciones OASIS mediante redes de Petri Orientadas a Objeto. Tesis Doctoral. Director: Isidro Ramos Salavert, Universidad Politécnica de Valencia. Publicación digital en CD con ISBN 84-699-2899-6. Año 2000. www.dsic.upv.es/users/oom/docs/tesis-ppalma.pdf

Sánchez P., Letelier P., Ramos I. Representación de Objetos en Teorías y su Traducción a un Modelo de Concurrencia. *Revista Computación y Sistemas*, vol. 1, No. 2, páginas 101-111, 1997.

Wieringa R.J. Requirements Engineering: Frameworks for understanding. John Wiley & Sons, 1996.

