

Verificación y Análisis de Sistemas Basados en Reglas Activas

Tesista: Lorena Chavarría Báez
Departamento de Computación
CINVESTAV-IPN

email: lchavarría@computacion.cs.cinvestav.mx

Director: Dra. Xiaou Li
Departamento de Computación
CINVESTAV-IPN

1 Motivación

Los sistemas activos permiten la ejecución de tareas de forma automática, gracias a que cuentan con una base de conocimiento la cual contiene reglas acerca de un problema específico. Estas reglas generalmente constan de tres partes: evento, condición y acción, por ello son llamadas reglas ECA. El evento describe una situación para la cual la regla es capaz de responder, la condición examina el contexto en el cual el evento tomó lugar y la acción describe la tarea que se debe ejecutar si, cuando sucedió el evento, la condición se evaluó como verdadera.

En la construcción de una base de reglas el conocimiento de uno (varios) experto(s) se transfiere a un sistema de software. Durante este proceso pueden suceder algunos problemas: primero, el conocimiento transferido puede ser incompleto, inconsistente o parcialmente erróneo. Esto sucede especialmente cuando son varios los expertos involucrados, ya que pueden tener diferentes ideas acerca del mismo problema. Segundo, el conocimiento puede no ser fielmente representado en el sistema debido a problemas de comunicación entre el experto y el diseñador de las reglas. Tercero, hay que considerar que se pueden cometer errores de programación. Debido a esto, la verificación de la base de reglas es una de las herramientas necesarias para construir un sistema activo confiable. Por lo tanto, el desarrollo de técnicas para verificar su correctitud es esencial.

Otro aspecto importante de las reglas ECA es su mantenibilidad, la cual se ve influenciada por tres factores: entendimiento, análisis y modificación. Debido a la interacción y dependencia que puede tener una regla con otra(s), cuando se intenta entender, analizar o modificar una de ellas es difícil no sólo prever sus efectos, sino además, comprobar cuáles fueron las causas que provocaron su disparo. Por esta razón, es importante realizar una evaluación de su interacción.

Asimismo, es importante analizar características propias de las reglas ECA tales como terminación y confluencia.

2 Trabajo Relacionado

Los trabajos de verificación de bases de conocimiento existentes utilizan reglas de producción. Una regla de producción consta de una premisa y una conclusión. Los errores más comunes en bases de reglas de producción son: redundancia, inconsistencia, incompletitud y la ocurrencia de ciclos [4]. Debido a restricciones de espacio no mostramos su definición. Esta se puede consultar en [4].

Usualmente la verificación de reglas de producción estaba basada en la comparación de las mismas, sin embargo, en propuestas recientes han empleado técnicas gráficas tales como grafos dirigidos, Redes de Petri (PN) e hipergrafos dirigidos [1], [3]. En esos trabajos, los nodos se utilizan para representar cláusulas simples de una regla y los arcos dirigidos para representar relaciones causales. En [3] se propone el uso de una PN especial, denominada ϖ -net, y en [1] los autores emplean una PN Coloreada. Posteriormente utilizan el árbol de alcanzabilidad para detectar reglas erróneas. En estos trabajos es posible la detección de errores sólo proporcionando al sistema una marca inicial para la PN. En [2], los autores proponen un método de análisis estático usando una red Predicado/Transición. La detección de errores se hace sobre un método de reconocimiento de patrones y es independiente de la marca inicial de la PN.

El mantenimiento de las reglas ECA ha sido abordado en [8]. En ese trabajo, los autores se enfocan en bases de datos activas y proponen la utilización de las métricas: número de anclas, distancia y potencial de disparo, como auxiliar en el mantenimiento de la base de reglas. Para representar a la misma, hacen uso de la gráfica de disparo. Con esta representación, los autores sólo consideran reglas disparadas por eventos primitivos. En [7] se retoman las métricas propuestas en [8] y se emplea un modelo basado en Red de Petri (PN) para representar a las reglas. Esto permite considerar reglas con eventos primitivos y compuestos. El método de análisis está basado en la matriz de incidencia de la PN.

En cuanto al análisis de terminación y confluencia, en [6] los autores traducen una base de reglas activas a cláusulas lógicas para posteriormente aplicar los resultados de terminación y confluencia conocidos para este tipo de reglas. Su principal aportación es incorporar la semántica

de ejecución.

3 Objetivo

Realizar un estudio de las características las reglas activas, para desarrollar herramientas de análisis que permitan la detección de errores tales como redundancia, inconsistencia e incompletitud. Además, realizar un análisis de las propiedades de terminación y confluencia y, finalmente, proponer parámetros que permitan analizar la interacción de las reglas.

4 Metodología

Nuestra metodología está dividida en tres fases:

Fase I. En esta etapa se desarrolla un estudio acerca de los errores que pueden ocurrir durante la construcción de bases de reglas de producción. Esto con el fin de mapear tales errores a las bases de reglas activas y así obtener definiciones para ellos en este contexto. En nuestras definiciones consideramos el elemento adicional (evento) de las reglas activas. Una vez obtenidas estas definiciones, se desarrollan métodos de detección y se enuncian algunos resultados relevantes.

Fase II. Durante esta etapa se analizarán las características de terminación y confluencia de las reglas activas. Para esto es necesario evaluar las diferentes técnicas que se han propuesto en la literatura. Consideramos que incorporando nuestras definiciones de errores al análisis de terminación y confluencia, es factible determinar con mayor precisión si estos problemas pueden ocurrir.

Fase III. En esta etapa analizaremos las características de la interacción entre las reglas activas. Este análisis está enfocado a obtener una mejor base de reglas a través de la identificación de parámetros que permitan realizar la evaluación de su interacción. Ya que estos parámetros pueden ser vistos como métricas, es necesario demostrar su validez teórica y práctica. La primera nos dirá cómo y cuándo utilizar tales métricas. Para realizarla es necesario investigar acerca de marcos de trabajo que permitan realizar la validación de métricas de software. La segunda se realizará a través de experimentos y casos de estudio. Desde luego, los resultados de estas validaciones nos proporcionarán una retroalimentación para una mejor definición de nuestras métricas.

Previo a estas fases es necesario realizar un estudio de las características de las reglas activas.

5 Avances y Resultados

Actualmente contamos con las definiciones para reglas redundantes, incluidas, en conflicto, aisladas, de punto muerto e inalcanzables, las cuales caracterizan a los errores más comunes que ocurren durante el desarrollo de bases

de reglas. Hemos ampliado nuestro conjunto de definiciones para determinar no sólo cuando un tipo de error efectivamente sucede, sino también las condiciones bajo las cuales podría ocurrir. Por ejemplo, un conflicto entre dos reglas ocurre cuando, activadas por el mismo evento y verificando la misma condición, ejecutan acciones contradictorias. Sin embargo, si la condición de las reglas es diferente entonces el conflicto ocurrirá sólo cuando la condición de ambas reglas sea evaluada como verdadera al mismo tiempo. Una vez obtenidas estas definiciones, para realizar la detección de errores nos hemos auxiliado de una extensión de las Redes de Petri, denominada Red de Petri Coloreada Condicional (CCPN) [5]. La CCPN fue desarrollada específicamente para modelar reglas activas. Utilizando la matriz de incidencia de la CCPN como herramienta de análisis, es posible detectar todos los errores que hemos definido. A diferencia de otros enfoques de verificación de reglas de producción, nuestro método es independiente de la marca inicial de la red.

Referencias

- [1] Wu, Q., Zhou, C., Wu, J., Wang, C.: Study on Knowledge Base Verification Based on Petri Nets. International Conference on Control and Automation (ICCA2005) Budapest, Hungary, June 27-29 (2005)
- [2] Zhang, D., Nguyen, D.: PREPARE: A Tool for Knowledge Base Verification. IEEE Trans. on Knowledge and Data Engineering 6(6) (1994)
- [3] He, X., Chu, W., Yang, H.: A New Approach to Verify Rule-Based Systems Using Petri Nets. Information and Software Technology 45(10) (2003) 663 – 670
- [4] González, J. A., Dankel, D.: The Engineering of Knowledge-Based Systems. Prentice Hall (1993)
- [5] Li, X., Medina-Marín, J., Chapa S. V.: Applying Petri nets on active database systems IEEE Trans. on System, Man, and Cybernetics, Part C: Applications and Reviews accepted for publication (2005).
- [6] S. Comani, L. Tanca. Termination and Confluence by Rule Prioritization. IEEE Trans. on Knowledge and Data Engineering, Vol., 15, No. 2, 2003
- [7] Chavarría-Báez, L., Li X., "Measuring triggering-interactions complexity on active databases based on conditional colored Petri net model", 1st International Conference on Electrical and Electronics Engineering and X conference on Electrical Engineering (ICEEE/CIE), Acapulco, México, Sept. 8-10, 2004
- [8] Díaz O., Piattini M. and Calero C. Measuring Triggering-Interaction Complexity on Active Databases, *Information Systems*, vol. 26, 2001, pp 15-34