Modelamiento y Verificación de Sistemas Utilizando Cálculos de Procesos

Alejandro Arbeláez, Andrés Aristizábal, Julian Gutiérrez, Hugo A. López, Carlos Olarte,
Jorge Andrés Pérez y Camilo Rueda
Pontificia Universidad Javeriana - Cali, Colombia.

Frank D. Valencia
CNRS y LIX - École Polytechnique de París, Francia.

Motivación

Los *cálculos de procesos* son lenguajes formales diseñados para el estudio de sistemas de cómputo concurrentes. Pueden considerarse como lenguajes *abstractos* de programación, donde las ideas de *proceso* e *interacción* son centrales en la especificación de sistemas. Están definidos en términos de un conjunto reducido de operaciones (*primitivas*) que establecen el tipo de interacciones posibles entre los procesos, así como su evolución en el tiempo. Intuitivamente, el tipo de primitivas reflejan las características y propiedades de los sistemas de interés. Por las características de las primitivas, el modelamiento de sistemas utilizando cálculos de procesos es inherentemente *composicional*: la especificación de un sistema tiene lugar por la composición de procesos simples que representan los subsistemas que lo conforman. Los cálculos de procesos pueden considerarse así como *instrumentos de abstracción* de sistemas dinámicos complejos: cada cálculo está concebido de forma tal que la especificación de sistemas es gobernada por algún criterio de abstracción particular. En otras palabras, el modelamiento se concentra en ciertos aspectos claves del comportamiento del sistema; otros aspectos, menos importantes de acuerdo con algún criterio de abstracción particular, son idealizados. Una ventaja inmediata de este acercamiento de especificación es que facilita enormemente la *verificación* rigurosa de propiedades esenciales de los sistemas modelados. De esta forma, los cálculos de procesos constituyen una metodología concreta de diseño de sistemas complejos.

Estas características de los cálculos de procesos han constituido una fuerte motivación para su uso en una amplia gama de contextos, entre los que se pueden mencionar: sistemas distribuidos [7], biología sistémica [11], lenguajes visuales/orientados a objetos [12], y sistemas reactivos [13]. El interés de la comunidad científica en estos formalismos se evidencia también en la extensión de cálculos existentes con conceptos como tiempo [14], movilidad [7] y comportamiento probabilístico/estocástico [6, 10].

Programación Concurrente por Restricciones (CCP)

Un tipo particularmente interesante de cálculos de procesos son los basados en el modelo de programación concurrente por restricciones (CCP, por sus siglas en inglés [13]). En este tipo de cálculos, se propone la abstracción de *restricción como información parcial*, en contraposición al concepto convencional de *valuación* utilizado en los lenguajes de programación imperativos. De esta forma, las restricciones en CCP denotan los *posibles valores* que una variable puede tomar, en lugar de definir un único valor. En el modelo CCP, los procesos interactúan entre sí *adicionando* y *consultando* restricciones en un medio compartido denominado *almacén*. El almacén constituye así el medio único de comunicación y sincronización entre los procesos. La estructura del almacén está dada en términos de un *sistema de restricciones*, que define el tipo de restricciones y las interdependencias entre ellas. Este sistema de restricciones es ortogonal al modelo, y a partir de él pueden derivarse capacidades de inferencia sobre los elementos del almacén.

Una de las características más atractivas de CCP es que la perspectiva *operacional* de los procesos se combina, en un solo formalismo, con una perspectiva *declarativa* fuertemente ligada a la lógica. Esto significa que los términos de procesos pueden verse al mismo tiempo como agentes de cómputo y fórmulas lógicas. Esta dualidad permite al modelo beneficiarse de técnicas y resultados tanto de la teoría de la concurrencia como de la lógica.

En este contexto, es posible pensar en los cálculos basados en restricciones como lenguajes apropiados para el modelamiento de sistemas en diversas áreas. Es posible identificar algunas ventajas concretas:

- La abstracción de restricción como información parcial permite definir sistemas utilizando la información disponible (posiblemente incompleta), por lo que los modelos derivados son naturalmente extensibles. Adicionalmente, esto facilita el estudio de sistemas a diversos niveles de detalle, por ejemplo, adicionando restricciones relativas a los subsistemas de interés particular.
- La naturaleza de los sistemas de restricciones subyacentes a CCP hace que los modelos basados en restricciones sean independientes de los tipos de datos necesarios para representarlos. Es posible definir sistemas de restricciones sobre diversos tipos de datos, lo que redunda en una mayor fidelidad de los modelos. De esta forma por ejemplo, la inclusión en los modelos de información cuantitativa proveniente de experimentos reales es posible.
- Los cálculos de procesos basados en restricciones poseen una agradable reciprocidad entre teoría y práctica, en forma de lenguajes de programación y simuladores que permiten validar en la práctica modelos formulados en la teoría. A su vez, las herramientas construidas cuentan con el soporte y propiedades derivadas de la teoría, lo que permite dar cuenta de su correctitud.

CCP en Biología y Seguridad: Algunos Resultados

Con base en estas ventajas, el grupo de Investigación AVISPA inició la exploración del uso de cálculos basados en CCP para el modelamiento y verificación de sistemas concurrentes en dos áreas de interés creciente en la actualidad: Biología Sistémica [5] y Análisis de Protocolos de Seguridad. El cálculo utilizado en el estudio de la primer área fue `ntcc` [8], un formalismo que generaliza CCP con conceptos de tiempo discreto, asincronía y no determinismo. Una característica importante de `ntcc` es que cuenta con un *sistema de prueba* que permite verificar propiedades de procesos `ntcc` expresadas como fórmulas de lógica temporal lineal (o LTL). Por su parte, el cálculo elegido para el estudio de protocolos de seguridad fue SPL [3], que si bien no es un cálculo basado en restricciones, posee similitudes importantes conceptuales con este modelo, como el concepto de almacén *persistente* de mensajes. SPL cuenta con una semántica operacional basada en eventos que, entre otras cosas, permite derivar una serie de *principios de prueba* útiles para la verificación de propiedades.

Nuestros resultados en el modelamiento de sistemas biológicos son muy prometedores. La aplicabilidad de `ntcc` en el estudio de estos sistemas fue demostrada modelando un sistema celular conocido como la *bomba de sodio-potasio* y verificando una propiedad no trivial sobre dicho modelo [4]. El modelo en `ntcc` resulta más completo que un modelo similar propuesto en el cálculo π [2], pues tanto el sistema de *transporte activo* que ejecuta la bomba, como un sistema complementario de *transporte pasivo* son considerados e integrados. Además de su completitud, el modelo es intuitivo y extensible.

Adicional a lo anterior, en el área de Biología Sistémica podemos destacar la construcción de un simulador para procesos `ntcc`. Este simulador aprovecha las capacidades de satisfacción de restricciones incluidas en Mozart, un lenguaje de programación multiparadigma que implementa algunas de las nociones del modelo CCP. La sintaxis de los procesos en este simulador es sencilla, lo que puede facilitar la interacción con usuarios no expertos. Capacidades de graficación de los resultados de la simulación son fácilmente integrables en el prototipo actual.

En el campo de Análisis de Protocolos de Seguridad, se destaca el uso de SPL para el modelamiento y verificación de propiedades de protocolos de comunicación para sistemas *Peer-to-Peer* (P2P) [1]. Estos sistemas son una tendencia reciente en computación que buscan aprovechar recursos usualmente dispersos en redes abiertas (e.g., Internet) como tiempo de procesamiento y capacidades de almacenamiento. Se caracterizan por tener una topología altamente dinámica y por enfocarse en la consecución de propósitos bien definidos. Ejemplos comunes de sistemas P2P son las aplicaciones de mensajería instantánea, trabajo colaborativo y comercio electrónico.

Por su misma naturaleza, las necesidades de seguridad en la comunicación en sistemas P2P son diferentes a las necesidades de los sistemas convencionales de comunicación. Particularmente importante resulta el aseguramiento de *anonimidad*, o el garantizar que el origen de las comunicaciones no puede ser descubierto por usuarios maliciosos dentro y fuera de la red P2P. Por ejemplo, en los sistemas de distribución de música digital, los usuarios están interesados en que una autoridad reguladora no descubra quién y cómo descarga archivos.

Un protocolo para sistemas P2P que persigue este objetivo es MUTE. MUTE puede abstraerse como un protocolo de *búsqueda* en sistemas P2P basado en palabras claves: un usuario inicia una búsqueda con algunas de éstas palabras y comunica su necesidad a los *peers* con los que tiene conexión directa (denominados *vecinos*). Estos vecinos verifican si pueden satisfacer la búsqueda: en caso afirmativo, le responden al peer que la origino. En caso negativo, reenvían las palabras claves de entrada a sus propios vecinos. El proceso continua hasta que la búsqueda se ha extendido entre todos los peers.

Uno de nuestros resultados es un modelo de MUTE usando SPL. Este modelo se caracteriza por ser conciso, considerando explícitamente los múltiples roles de los agentes asociados al sistema P2P. A partir de este modelo, y utilizando los mencionados principios de prueba de SPL, fue posible probar una propiedad relativa al poder de los atacantes internos (*insiders*). Dicha propiedad garantiza anonimidad para los mensajes transmitidos en el protocolo, siempre y cuando las llaves (públicas) entre cada agente del sistema permanezcan ocultas a dichos insiders.

Trabajo Actual y Futuro

Nuestros resultados recientes abren una amplia gama de posibilidades de profundización. En particular, la idea de constituir un marco de trabajo *declarativo* inspirado en CCP para el estudio de problemas relevantes, nos resulta muy interesante en la actualidad. Tal marco estaría compuesto de resultados teóricos que propicien la construcción de herramientas de software. Así, sería posible derivar un cálculo "Secure CCP" que se ajuste a los requerimientos del análisis de protocolos de seguridad, o un "Bio CCP" que sea su análogo en la Biología Sistémica. A continuación describimos algunos pasos iniciales en esta dirección.

Un resultado reciente es la definición de un cálculo de procesos *persistente* [9]: un cálculo donde tanto las acciones de entrada como las de salida están *replicadas* (es decir, se ejecutan por siempre). La intuición detrás de las primeras es representar la posibilidad que tiene un agente de cómputo de comprometerse en múltiples (posiblemente infinitas) ejecuciones de protocolos; las segundas acciones buscan modelar el hecho de que todos los mensajes transmitidos en la ejecución de un protocolo pueden ser recuperados por un agente malicioso. Este cálculo pretende ser la base de un CCP seguro: mientras que las salidas infinitas se poseen por los principios básicos de CCP, restaría por incluir en el modelo la capacidad que tienen los agentes de comprometerse en diversos protocolos.

En el campo biológico, actualmente estamos trabajando en el modelamiento de sistemas que incluyen información probabilística. En particular, nos interesa analizar las características y el comportamiento de las redes de regulación genética. Hasta el momento hemos obtenido algunos resultados preliminares en este campo, en forma de modelos y pruebas sobre ellos. Una extensión probabilistica para `ntcc`, que asigna una probabilidad a cada proceso involucrado en una escogencia y restringe el no determinismo a la composición paralela, está orientada en esta dirección. Otros retos inmediatos en esta área incluyen la inclusión de mecanismos para razonar sobre tiempo *continuo*, de forma que la inclusión de ecuaciones diferenciales en modelos biológicos sea posible.

References

[1] Andres Aristizabal, Hugo Lopez, Camilo Rueda, and Frank Valencia. Formally Reasoning About Security Issues in P2P Protocols: A Case Study. In *Proceedings of TFIT 2006*, INRIA Technical Reports, March 2006.

[2] G. Ciobanu, V. Ciubotariu, and B.Tanasa. A pi-calculus model of the Na pump. In *Genome Informatics 2002*, pages 469–472. Universal Academy Press.

[3] F. Crazzolara and G. Winskel. Events in security protocols. In Pierangela Samarati, editor, *Proceedings of the 8th ACM Conference on Computer and Communications Security*, pages 96–105, Philadelphia, PA, USA, November 2001. ACM Press.

[4] Julian Gutierrez, Jorge A. Perez, Camilo Rueda, and Frank D. Valencia. A Timed Process Calculus for Modeling and Verifying Biological Systems. Submitted for Publication., 2006.

[5] Julian Gutierrez, Jorge Andrés Pérez, and Camilo Rueda. Modelamiento de Sistemas Biológicos usando Cálculos de Procesos Concurrentes. *Epiciclos*, 4(1). Available at `http://epiciclos.puj.edu.co`.

[6] O. M. Herescu and C. Palamidessi. Probabilistic asynchronous π-calculus. In *Proceedings of FOSSACS 2000 (Part of ETAPS 2000)*, volume 1784 of *LNCS*, pages 146–160. Springer, 2000.

[7] R. Milner. *Communicating and Mobile Systems: The π-Calculus.* Cambridge University Press, 1999.

[8] Mogens Nielsen, Catuscia Palamidessi, and Frank Valencia. Temporal Concurrent Constraint Programming: Denotation, Logic and Applications. *Nordic Journal of Computing*, 9:145–188, 2002.

[9] C. Palamidessi, V. Saraswat, F. Valencia, and B. Victor. On the Expressiveness of Linearity vs Persistence in the Asynchronous Pi-Calculus. Submitted for Publication.

[10] Corrado Priami. Stochastic π-calculus. *The Computer Journal*, 38(6):578–589, 1995.

[11] A. Regev and E. Shapiro. *Modelling in Molecular Biology*, chapter The π-calculus as an abstraction for biomolecular systems, pages 219–266. Natural Computing Series. Springer, 2004.

[12] Camilo Rueda, Gloria Alvarez, Luis O. Quesada, Gabriel Tamura, Frank D. Valencia, Juan Francisco Díaz, and Gerard Assayag. Integrating constraints and concurrent objects in musical applications: A calculus and its visual language. *Constraints*, 6(1):21–52, 2001.

[13] V. Saraswat, M. Rinard, and P. Panangaden. The semantic foundations of concurrent constraint programming. In *POPL '91*, pages 333–352, Jan 1991.

[14] Vijay Saraswat, Radha Jagadeesan, and Vineet Gupta. Foundations of timed concurrent constraint programming. In *Proceedings, Ninth Annual IEEE Symposium on Logic in Computer Science, Paris, France, 4–7 July, 1994*, pages 71–80, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1994. IEEE.