

Tesis Doctoral



# Modelado de Sistemas Híbridos Usando el Formalismo DEVS Paralelo y el Lenguaje Modelica

Resumen y Conclusiones

Victorino Sanz Prat  
**Ingeniero en Informática**

Departamento de Informática y Automática  
Escuela Técnica Superior de Ingeniería Informática  
Universidad Nacional de Educación a Distancia  
**Madrid, 2010**



**Departamento** Informática y Automática  
E.T.S. de Ingeniería Informática

**Título** Modelado de Sistemas Híbridos  
Usando el Formalismo DEVS Paralelo  
y el Lenguaje Modelica

**Autor** Victorino Sanz Prat

**Titulación** Ingeniero en Informática  
Facultad de Informática  
Universidad Politécnica de Madrid

**Directores** Alfonso Urquía Moraleda  
Sebastián Dormido Bencomo



# Índice general

<b>1</b>	<b>Resumen</b>	<b>1</b>
1.1.	Introducción . . . . .	1
1.2.	Objetivos . . . . .	4
1.3.	Estructura del Documento . . . . .	8
1.4.	Publicaciones . . . . .	12
1.5.	Proyectos de Investigación . . . . .	13
<b>2</b>	<b>Conclusiones y Trabajos Futuros</b>	<b>15</b>
2.1.	Conclusiones . . . . .	15
2.2.	Futuras Líneas de Investigación . . . . .	19
	<b>Bibliografía</b>	<b>21</b>



# 1

## Resumen

### 1.1 Introducción

---

El paradigma de modelado físico permite describir sistemas de manera modular. Un sistema es dividido en subsistemas, y cada subsistema es descrito usando una interfaz, ecuaciones de balance de masa, energía y momento, y materiales. La interfaz se usa para describir relaciones entre subsistemas. El modelo es considerado como una restricción entre variables [Åström et al., 1998].

La metodología de modelado orientado a objetos facilita la descripción de modelos usando ecuaciones acausales, las cuales posibilitan el paradigma de modelado físico. En el modelado orientado a objetos, los entornos de modelado se encargan de realizar automáticamente las manipulaciones matemáticas requeridas para simular el modelo en un ordenador (tales como la asignación de causalidad, la evaluación de lazos algebraicos o la reducción de índice del modelo). Esto representa una considerable ventaja cuando se compara con el paradigma de modelado orientado a bloques, en el cual los modelos deben ser manipulados manualmente antes de poder simularlos [Åström et al., 1998]. Esta metodología también facilita el diseño, programación, reutilización y mantenimiento de los modelos [Cellier, 1996].

Modelica es un lenguaje de modelado de propósito general, distribuido bajo su propia licencia libre, que permite la descripción de modelos siguiendo la

metodología de la orientación a objetos. De manera similar al esfuerzo realizado en los años 60 para describir el estándar CSSL [Augustin et al., 1967], Modelica constituye un esfuerzo internacional para estandarizar la descripción de modelos usando dicha metodología [Mattsson et al., 1998], en comparación con la multitud de lenguajes existentes en el momento. El uso de un lenguaje estándar facilita el intercambio de modelos entre usuarios y herramientas. Modelica incluye características de diferentes lenguajes, tales como ALLAN [Jeandel et al., 1997], Dymola [Elmqvist, 1978], NMF [Sahlin et al., 1996], ObjectMath [Fritzson et al., 1995], Omola [Andersson, 1989], SIDOPS+ [Breuneuse and Broenink, 1997] y Smile [Kloas et al., 1995].

La primera versión de Modelica apareció en Septiembre de 1997. Desde entonces, el uso de Modelica para describir modelos se ha ido incrementando. La industria y los investigadores han aceptado ampliamente el uso del lenguaje, tal y como demuestra el incremento en el número de contribuciones y participantes de la Conferencia Internacional de Modelica [Modelica, 2010]. De manera similar, también se ha incrementado el número de librerías de desarrolladas en Modelica (comerciales y de libre distribución). Incluso el propio lenguaje ha estado en continuo desarrollo, siendo la 3.1 su última versión [Modelica Association, 2009].

Las funcionalidades proporcionadas por el lenguaje Modelica se ven incrementadas mediante a desarrollo de librerías de componentes. Las librerías de Modelica representan agrupaciones de componentes que son diseñados conjuntamente para facilitar la descripción de modelos usando diferentes formalismos o en diferentes dominios. En la actualidad, Modelica soporta la descripción de modelos en múltiples dominios (e.g. eléctrico, mecánico, termodinámico y químico) por medio de diferentes librerías [Modelica Libraries, 2010]. Algunos de los formalismos soportados son ODE, DAE, Bond Graphs [Cellier and Nebot, 2005] y System Dynamics [Cellier, 2008].

Modelica facilita la descripción de modelos de eventos discretos e híbridos por medio de funcionalidades para gestionar eventos en el tiempo y en el estado [Otter et al., 1999; Mattsson et al., 1999]. Gracias a estas funcionalidades para el modelado híbrido, Modelica soporta varios formalismos de modelado orientado



a eventos, tales como Petri Nets [Mosterman et al., 1998], StateGraphs [Otter et al., 2005] y StateCharts [Ferreira and de Oliveira, 1999].

Tal y como se ha mencionado anteriormente, el uso de una metodología o formalismo matemático para describir modelos facilita su desarrollo, mantenimiento, reutilización y adaptación a diferentes experimentos y situaciones. El formalismo DEVS Paralelo (P-DEVS) [Chow, 1996] y el paradigma de modelado orientado al proceso son dos metodologías ampliamente usadas para describir sistemas de eventos discretos. A su vez, el empleo de estas metodologías para la descripción de modelos dinámicos híbridos ha sido discutida en múltiples contribuciones [Vangheluwe, 2000; Giambiasi and Carmona, 2006; Kelton et al., 2007].

El formalismo DEVS Clásico fue descrito por Zeigler [1976]. P-DEVS es una extensión de DEVS Clásico que permite la ocurrencia de eventos simultáneos, eliminando la restricción que supone su tratamiento secuencial. DEVS ha sido considerado como el formalismo de modelado de sistemas de eventos discretos equivalente a las ecuaciones diferenciales [Zeigler, 1989]. Otros formalismos y metodologías de modelado de sistemas de eventos discretos pueden ser descritos usando DEVS [Vangheluwe, 2000].

El paradigma de modelado orientado al proceso es uno de las tres “aproximaciones para describir el mundo” [Kiviat, 1969] comúnmente usadas para describir sistemas de eventos discretos. Este paradigma proporciona un mecanismo intuitivo para describir un sistema como una serie de procesos interconectados, en lugar de múltiples ocurrencias de eventos. Los sistemas modelados siguiendo la orientación al proceso son descritos como un flujo de entidades que transcurren a través de los procesos del sistema usando los recursos disponibles [Law, 2007]. Arena, cuyas componentes son internamente representados usando el lenguaje SIMAN [Pegden et al., 1995], es un entorno de simulación ampliamente usado en las universidades y la industria para describir sistemas de eventos discretos siguiendo la aproximación de modelado orientado al proceso [Kelton et al., 2007].

La viabilidad de describir modelos usando el formalismo DEVS Clásico en Modelica fue demostrada en Fritzson [2003]. A su vez, una librería en Modelica llamada ModelicaDEVS [Beltrame and Cellier, 2006] fue desarrollada para

modelar sistemas de tiempo continuo usando el formalismo DEVS Clásico y los algoritmos de integración QSS [Kofman, 2004]. En estas implementaciones de DEVS en Modelica la comunicación entre modelos es descrita mediante el cambio de valor en variables booleanas, y usando las funcionalidades de modelado híbrido de Modelica previamente mencionadas para describir la gestión de los eventos. Sin embargo, tal y como será discutido, la descripción de modelos usando el formalismo DEVS Paralelo y la aproximación de modelado orientado al proceso requiere de mecanismos adicionales que actualmente no están presentes en Modelica. La descripción del mecanismo de comunicación entre modelos P-DEVS en Modelica no es trivial. Ninguna de las librerías de Modelica disponibles actualmente soportan el formalismo DEVS Paralelo, ni la aproximación de modelado orientado al proceso para describir modelos de sistemas de eventos discretos en Modelica. El uso de P-DEVS para describir formalmente sistemas de eventos discretos en Modelica facilitaría la construcción, mantenimiento y reutilización de modelos así como la introducción de otros formalismos, tales como la aproximación de modelado orientado al proceso, en Modelica.

## 1.2 Objetivos

---

El principal objetivo de la presente tesis doctoral es incluir en el lenguaje Modelica nuevas funcionalidades para describir la parte de eventos discretos de modelos híbridos usando el formalismo DEVS Paralelo y el paradigma de modelado orientado al proceso. El uso de estas metodologías combinadas con la metodología de modelado orientado a objetos, soportada por Modelica, facilitará la descripción de modelos dinámicos híbridos. Para alcanzar este objetivo, se propone la realización de las siguientes tareas:

1. La identificación y análisis de los requisitos necesarios para describir modelos en Modelica siguiendo el formalismo DEVS Paralelo y la aproximación de modelado orientado al proceso. Esta tarea constituye la base del trabajo presentado en esta tesis.

2. Se diseñará y desarrollará una nueva librería en Modelica, llamada DEVS-Lib, para facilitar la descripción de modelos de eventos discretos usando el formalismo DEVS Paralelo. La definición de nuevos modelos atómicos y acoplados usando esta librería será lo más similar posible a la especificación formal de modelos en P-DEVS.
3. La comunicación entre modelos P-DEVS sigue una estructura de paso de mensajes. El mecanismo actual de Modelica para describir la comunicación entre modelos se basa en la conexión de variables de modelos, por lo que no facilita la transmisión de información estructurada entre modelos. Se propondrá un mecanismo de comunicación por medio de paso de mensajes en Modelica, con el objetivo de facilitar la descripción de modelos P-DEVS. Este mecanismo se podrá utilizar para comunicar información estructurada entre modelos usando diversas configuraciones (1:1, 1:N, N:1). Por tanto:
  - a) Se propondrá un mecanismo de paso de mensajes en Modelica, junto con las modificaciones del lenguaje necesarias para facilitar la comunicación de modelos usando el mecanismo propuesto.
  - b) Usando las funcionalidades que proporciona Modelica actualmente, se programará un mecanismo de paso de mensajes con un comportamiento lo más similar posible al mecanismo propuesto anteriormente. Este mecanismo será usado en el desarrollo de la librería DEVSLib.
4. El formalismo DEVS Paralelo será utilizado para describir el comportamiento de algunos de los componentes del lenguaje SIMAN. Esta especificación formal facilitará la programación de dichos componentes en Modelica, usando la nueva librería DEVSLib. De esta manera, el formalismo DEVS Paralelo servirá para soportar el modelado orientado al proceso en Modelica.
5. Se diseñarán y programarán en Modelica dos nuevas librerías, llamadas SIMANLib y ARENALib, que soportarán la descripción de modelos de eventos discretos siguiendo el paradigma de orientación al proceso. Estas

dos nuevas librerías proporcionarán funcionalidades de análisis y descripción de modelos similares a las del entorno de simulación Arena y el lenguaje SIMAN. La librería SIMANLib reproducirá las funcionalidades de los bloques Create, Dispose, Queue, Seize, Delay, Release, Branch, Count, Tally y Assign, del lenguaje SIMAN. Estos bloques han sido seleccionados debido a que constituyen los componentes básicos para describir la mayoría de los procesos encontrados en diferentes sistemas logísticos. La librería ARENALib reproducirá las funcionalidades de los módulos Create, Dispose, Process, Record, Decide y Assign, pertenecientes al panel BasicProcess de Arena. Estos componentes, de manera análoga a Arena, serán construidos usando componentes de la librería SIMANLib.

6. Algunos sistemas de eventos discretos son de naturaleza estocástica. Otra de las tareas propuestas en la presente tesis doctoral será proporcionar al lenguaje Modelica de funcionalidades para el modelado estocástico. Ni el lenguaje Modelica ni la Librería Estándar de Modelica incluyen funcionalidades para la generación de números aleatorios. Se desarrollará una nueva librería en Modelica, llamada RandomLib, para facilitar la generación de números aleatorios uniformes y muestras aleatorias. Las funcionalidades de RandomLib reproducirán las funcionalidades para el modelado estocástico incluidas en el entorno Arena, con el fin de facilitar la validación de los modelos desarrollados.
7. Se desarrollarán e incluirán en las nuevas librerías, construidas en esta tesis, modelos de interfaz para hacer dichas librerías compatibles con el resto de librerías de Modelica. Es decir, para permitir la conexión de modelos de eventos discretos construidos usando DEVSLib, SIMANLib y ARENALib con modelos construidos usando otras librerías de Modelica. Estos modelos de interfaz se encargarán de traducir los mensajes generados por los modelos P-DEVS en señales de tiempo discreto, y las señales de tiempo discreto y continuo en mensajes. También se estudiará la interacción entre componentes de modelos orientados al proceso y modelos de tiempo continuo. Se

extenderán las funcionalidades de algunos de los componentes orientados al proceso para que interactúen con modelos de tiempo continuo, y así poder describir modelos híbridos orientados al proceso.

8. También se estudiará la descripción de sistemas de control híbridos. En este caso, los controladores de eventos discretos son descritos usando el formalismo DEVS Paralelo y la planta de tiempo continuo es descrita usando otras librerías de Modelica. Ambas partes se conectarán usando los modelos de interfaz descritos anteriormente.
9. Por último, se programará un conjunto de modelos con el fin de ilustrar el uso de las nuevas librerías desarrolladas. Estos modelos incluirán tanto modelos estocásticos de eventos discretos como modelos híbridos deterministas. Se usarán dos modelos, uno de un cajero automático y otro del modelo presa-depredador descrito por Lotka y Volterra, para presentar la construcción de modelos de eventos discretos usando el formalismo DEVS Paralelo en Modelica, por medio de la librería DEVSLib. La descripción de sistemas dinámicos híbridos usando DEVSLib se presentará por medio de dos modelos, uno de un sistema de dos tanques y otro de un sistema de comunicación opto-electrónico. La aplicación del formalismo DEVS Paralelo a la descripción de sistemas de control híbridos se discutirá usando un modelo del sistema de refrigeración de un supermercado y otro de una grúa manejada por un controlador discreto empotrado. Se utilizarán modelos de un cajero de banco, de un restaurante y de una fabrica de componentes electrónicos para ilustrar la construcción de modelos orientados al proceso usando SIMANLib y ARENALib. Para finalizar, el desarrollo de modelos híbridos orientados al proceso usando las funcionalidades incluidas en SIMANLib y ARENALib se describirá por medio de los siguientes modelos: una enlatadora de zumo de naranja, un controlador del nivel de un tanque y un sistema de horno de fosa.

## 1.3 Estructura del Documento

---

La presente tesis doctoral está organizada en los capítulos y apéndices indicados a continuación:

- Capítulo 2. En este capítulo se presenta la evolución y la situación actual de las diferentes técnicas de modelado que pueden ser aplicadas a la descripción de sistemas híbridos. Se describe la evolución de las metodologías de modelado de tiempo continuo y de eventos discretos, detallando las características del lenguaje Modelica, el formalismo DEVS Paralelo y el entorno de simulación Arena. La evaluación de las metodologías descritas se ha utilizado como inicio para el desarrollo de la presente tesis.
- Capítulo 3. En este capítulo se identifican y discuten los requisitos necesarios para describir modelos P-DEVS en Modelica. La identificación de las diferencias conceptuales entre Modelica y P-DEVS supone el primer paso para el desarrollo de esta tesis. El resto del trabajo presentado radica en las aproximaciones adoptadas y desarrolladas para solucionar dichas diferencias, y facilitar la descripción de modelos P-DEVS en Modelica.
- Capítulo 4. En este capítulo se describe el diseño y la implementación del mecanismo de paso de mensajes en Modelica. Este ha sido el reto más importante encontrado durante el desarrollo de esta tesis, y representa la piedra angular de los trabajos realizados. Se discuten las aproximaciones estudiadas y programadas durante el desarrollo del mecanismo. Se detalla la aproximación seleccionadas, su uso y los puertos definidos para establecer comunicaciones entre modelos.
- Capítulo 5. En este capítulo se describe la librería DEVSLib, que soporta el formalismo DEVS Paralelo en Modelica. Se describe la arquitectura general de la librería y sus componentes. El mecanismo de paso de mensajes previamente desarrollado se usa para describir la comunicación entre modelos en DE-

VSLib. La presentación de la librería se realiza desde el punto de vista del desarrollador, describiendo los detalles de su programación.

Capítulo 6. En este capítulo se describe el uso de la librería DEVSLib para la construcción de modelos de eventos discretos. Se discuten dos casos de estudio descritos usando DEVSLib. El modelo de un cajero automático construido con DEVSLib es descrito como ejemplo de sistema de eventos discretos puro. También se discute el desarrollo de un modelo de eventos discretos del sistema presa-depredador de Lotka-Volterra, el cual está definido mediante ecuaciones diferenciales. Este modelo se ha construido usando los métodos de integración QSS incluidos en DEVSLib, para demostrar que la librería puede usarse para describir múltiples tipos de modelos basados en el formalismo DEVS.

Capítulo 7. En este capítulo se presenta la descripción de modelos híbridos usando la librería DEVSLib. Se detalla el comportamiento de los modelos de interfaz incluidos en la librería, para poder combinar modelos P-DEVS con modelos de otras librerías de Modelica. Se presenta el uso de dichos modelos de interfaz para describir sistemas híbridos por medio de dos casos de estudio. El primer caso de estudio representa un sistema de dos tanques controlado por un controlador de eventos discretos. Este caso de estudio se ha incluido para ejemplificar la interactividad entre la parte de tiempo continuo del modelo (i.e., tanques y válvulas) y la parte de eventos discretos (i.e., el controlador), que es descrita de manera algorítmica. El segundo caso de estudio representa un sistema de comunicación opto-eléctrico, donde la parte eléctrica se ha modelado usando modelos de tiempo continuo en Modelica y la parte óptica se ha modelado usando DEVSLib. Este caso de estudio se ha incluido para mostrar la versatilidad del uso de DEVSLib y Modelica para describir sistemas híbridos multi-dominio.

Capítulo 8. En este capítulo se presenta la aplicación de las funcionalidades de modelado incluidas en DEVSLib para la descripción de sistemas de control híbridos. Dichas funcionalidades son descritas mediante dos casos de estudio:

un sistema de grúa con un controlador discreto empotrado y un sistema de refrigeración de un supermercado. Ambos modelos se usan para presentar la viabilidad de describir sistemas de control híbridos usando DEVSLib en combinación con otros modelos en Modelica. Se discute también la evaluación de diferentes aproximaciones (usando Modelica, modelos atómicos en DEVSLib o modelos acoplados en DEVSLib) para describir los controladores.

Capítulo 9. En este capítulo se analizan las funcionalidades necesarias para describir modelos en Modelica siguiendo la aproximación de modelado orientado al proceso. Los modelos orientados al proceso se comunican usando un mecanismo equivalente al de los modelos P-DEVS, y por tanto, el mecanismo de paso de mensajes desarrollado puede utilizarse para facilitar su descripción. Sin embargo, se requieren algunas funcionalidades adicionales para poder describir modelos orientados al proceso. Dichas funcionalidades incluyen la gestión de la información que describe las entidades y su flujo a través del sistema, así como la gestión de estructuras de datos de tamaño variable usadas para almacenar información definida por el usuario e indicadores estadísticos.

Capítulo 10. En este capítulo se presenta una nueva librería en Modelica, llamada SIMANLib, que incluye funcionalidades de bajo nivel para la descripción de modelos orientados al proceso. Se detalla la arquitectura de la librería, su diseño, componentes y su programación. Los componentes incluidos en SIMANLib son equivalentes a los bloques Create, Dispose, Queue, Seize, Delay, Release, Branch, Count Tally y Assign de SIMAN. La descripción de los componentes de SIMANLib se ha realizado usando el formalismo P-DEVS, y su desarrollo se ha realizado usando la librería DEVSLib. Estos componentes pueden ser usados para modelar la mayoría de los procesos encontrados en sistemas logísticos, y a su vez son usados para describir los componentes de ARENALib. Como caso de estudio se describe el modelo de un restaurante construido mediante SIMANLib.



- Capítulo 11. En este capítulo se presenta otra nueva librería en Modelica, llamada ARENALib, que incluye funcionalidades de alto nivel para la descripción de modelos orientados al proceso. Dichas funcionalidades son de alto nivel en comparación con las funcionalidades de bajo nivel incluidas en SIMANLib, que describen acciones y procesos más simples. Se detalla la arquitectura de la librería, su diseño, componentes y su programación. Los componentes incluidos en ARENALib son equivalentes a los módulos Create, Dispose, Process, Record, Decide y Assign del panel BasicProcess de Arena. Como caso de estudio se describe el modelo de una fábrica de componentes electrónicos desarrollado usando ARENALib.
- Capítulo 12. En este capítulo se describen las funcionalidades incluidas en SIMANLib y ARENALib para el desarrollo de sistemas híbridos siguiendo la aproximación de modelado orientado al proceso. El uso de estas funcionalidades para describir sistemas se muestra por medio de tres casos de estudio: una fábrica de zumo de naranja enlatado, un sistema de control del nivel de un tanque y un horno de foso. Estos modelos incluyen diferentes características modeladas usando las funcionalidades de modelado híbrido orientado al proceso descritas previamente.
- Capítulo 13. En este capítulo se describe el desarrollo de la librería RandomLib. Esta nueva librería puede usarse, en combinación con el resto de librerías desarrolladas, para describir sistemas estocásticos de eventos discretos. La librería incluye un generador de números aleatorios y múltiples funciones para la generación de muestras aleatorias.
- Capítulo 14. En este capítulo se detallan las conclusiones de esta tesis doctoral, así como algunas ideas sobre futuras líneas de investigación.
- Apéndice A. Este apéndice contiene la descripción del modelo de semáforos desarrollado usando Modelica. Este modelo se diseñó y programó durante el desarrollo del mecanismo de paso de mensajes como método de sincronización en la comunicación de modelos P-DEVS. Sin embargo, debido al escaso rendimiento

obtenido se descartó su uso en el desarrollo del mecanismo de comunicación y por esta razón su descripción no está incluida en el Capítulo 4.

## 1.4 Publicaciones

---

Los trabajos de investigación realizados en la presente tesis doctoral han dado lugar a las siguientes publicaciones:

1. **V. Sanz**, A. Urquia and S. Dormido. ARENALib: A Modelica Library for Discrete-Event System Simulation. In *Proceedings of the 5<sup>th</sup> International Modelica Conference*, Vienna, Austria, 2006, pp 539–548.
2. **V. Sanz**, A. Urquia and S. Dormido. DEVS Specification and Implementation of SIMAN Blocks Using the Modelica Language. In *Proceedings of the Winter Simulation Conference 2007*, Washington, D.C., USA, 2007, p 2374.
3. **V. Sanz**, A. Urquia and S. Dormido. Introducing Messages in Modelica for Facilitating Discrete-Event System Modeling. In *Proceedings of the 2<sup>nd</sup> International Workshop on Equation-Based Object-Oriented Languages and Tools*, Paphos, Cyprus, 2008, pp 83–94.
4. **V. Sanz**, A. Urquia and S. Dormido. Introducing Messages in Modelica for Facilitating Discrete-Event System Modeling. *Simulation News Europe*, 18(2), 2008, pp 42–53.
5. **V. Sanz**, S. Jafer, G. Wainer, G. Nicolescu, A. Urquia and S. Dormido. Hybrid Modeling of Opto-Electrical Interfaces Using DEVS and Modelica. In *Proceedings of the DEVS Integrative M&S Symposium, Spring Simulation Multiconference*, San Diego, CA, USA, 2009.
6. **V. Sanz**, F.E. Cellier, A. Urquia and S. Dormido. Modeling of the ARGES-IM “Crane and Embedded Controller” System using the DEVSLib Modelica library. In *Proceedings of the 3<sup>rd</sup> IFAC Conference on Analysis and Design of Hybrid Systems (ADHS’09)*, Zaragoza, Spain, 2009.

7. F.E. Cellier and **V. Sanz**. Mixed Quantitative and Qualitative Simulation in Modelica. In *Proceedings of the 7<sup>th</sup> International Modelica Conference*, Como, Italy, 2009, pp 86–95.
8. **V. Sanz**, A. Urquia and S. Dormido. Parallel DEVS and Process-Oriented Modeling in Modelica. In *Proceedings of the 7<sup>th</sup> International Modelica Conference*, Como, Italy, 2009, pp 96–107.
9. **V. Sanz**, A. Urquia and S. Dormido. Integrating Parallel DEVS and Equation-Based Object-Oriented Modeling. In *Proceedings of the DEVS Integrative M&S Symposium, Spring Simulation Multiconference*, Orlando, FL, USA, 2010.

## 1.5 Proyectos de Investigación

---

La mayoría de los resultados obtenidos en esta tesis doctoral han sido obtenidos en el marco de diferentes proyectos de investigación:

1. “Herramientas interactivas para el modelado, visualización, simulación y control de sistemas dinámicos”, *CICYT, DPI2004-01804*, Enero 2004 – Diciembre 2006, Investigador principal: Prof. Dr. Sebastián Dormido Bencomo.
2. “Control de sistemas complejos en la logística y producción de bienes y servicios. Acrónimo: COSICOLOGI-CM”, *IV PRICIT 2005–2008, Plan Regional de Ciencia y Tecnología de la Comunidad de Madrid, Ref. S-0505/DPI/0391*, Enero 2005 – Diciembre 2008, Investigador principal: Prof. Dr. Sebastián Dormido Bencomo.
3. “Modelado, simulación y control basado en eventos”, *CICYT, DPI2007-61068*, Octubre 2007 – Septiembre 2012, Investigador principal: Prof. Dr. Sebastián Dormido Bencomo.



# Conclusiones y Trabajos Futuros

En este capítulo se presentan las conclusiones obtenidas tras el desarrollo de la presente tesis doctoral, así como algunas ideas para futuras investigaciones.

## 2.1 Conclusiones

---

Las conclusiones obtenidas tras el desarrollo de esta tesis doctoral son:

- El uso del formalismo P-DEVS y la aproximación de modelado orientado al proceso conjuntamente con la metodología de modelado orientado a objetos facilita la descripción de sistemas dinámicos híbridos. El uso de un formalismo matemático para describir el comportamiento de la parte de eventos discretos de un modelo híbrido facilita su desarrollo, mantenimiento, reutilización y les proporciona solidez.
- Se han analizado e identificado los requisitos necesarios para describir modelos P-DEVS y orientados al proceso usando lenguajes EOO, y particularmente el lenguaje Modelica. Dichos requisitos incluyen la descripción del comportamiento orientado a eventos del modelo, la descripción de las comunicaciones entre modelos siguiendo un mecanismo de paso de mensajes, y la descripción de interfaces para combinar modelos de eventos discretos con modelos de otras librerías de Modelica.

- Se han analizado las diferencias existentes respecto del mecanismo de comunicación entre modelos en P-DEVS y en los lenguajes EOO, para poder describir un mecanismo de comunicación basado en paso de mensajes. Dicho mecanismo basado en paso de mensajes ha sido propuesto, especificado y diseñado para ser utilizado en lenguajes EOO.

Se ha programado parcialmente el mecanismo de comunicación propuesto, usando las funcionalidades actualmente ofrecidas por Modelica. Se han evaluado y programado múltiples alternativas para desarrollar dicho mecanismo de comunicación. Durante este estudio, se desarrolló en Modelica un modelo del mecanismo de sincronización de procesos mediante semáforos. La aproximación finalmente seleccionada para el desarrollo se basa en el uso de memoria dinámica para transmitir mensajes entre modelos. Usando esta aproximación, se definió y programó un tipo básico de mensaje y las operaciones necesarias para realizar la comunicación entre modelos usando mensajes en Modelica.

El mecanismo programado se ha usado para describir la comunicación de modelos P-DEVS en Modelica. Esto facilitará la descripción de modelos P-DEVS y modelos siguiendo la orientación al proceso.

- Se ha diseñado y desarrollado una nueva librería en Modelica, llamada DEVSLib, para soportar el formalismo P-DEVS. Esta librería incluye funcionalidades para describir modelos P-DEVS atómicos y acoplados, así como para combinar dichos modelos con modelos desarrollados usando otras librerías de Modelica. Por tanto, DEVSLib se puede utilizar para describir modelos de eventos discretos, de tiempo continuo (usando los métodos de integración QSS) e híbridos.

La descripción de modelos usando DEVSLib es análoga a su especificación formal en P-DEVS, es decir, se realiza mediante la descripción de los elementos de la tupla. DEVSLib también soporta una extensión a la especificación de modelos atómicos P-DEVS que permite el uso de señales de tiempo continuo como entradas a las funciones de transición. La comunicación entre

modelos DEVSLib se ha realizado usando el mecanismo de paso de mensajes desarrollado.

- Se ha analizado la aplicación de las funcionalidades incluidas en DEVSLib para la descripción de modelos de control híbridos. DEVSLib puede utilizarse para describir sensores y actuadores basados en eventos. La librería también puede utilizarse para la descripción de controladores basados en tiempo discreto y en eventos, representándolos mediante modelos atómicos o acoplados. Estas funcionalidades, combinadas con las funcionalidades de Modelica para la descripción de modelos de tiempo continuo, facilitan la descripción de sistemas de control híbridos.
- Se han analizado los requisitos necesarios para describir modelos en Modelica siguiendo la aproximación de modelado orientado al proceso. La comunicación entre modelos orientados al proceso se corresponde con la comunicación entre modelos P-DEVS, por tanto, el mecanismo de paso de mensajes desarrollado se podrá utilizar para su descripción. Sin embargo, también se requieren algunas funcionalidades adicionales, tales como la descripción de las entidades del sistema y la gestión de estructuras de datos de tamaño variable, que son utilizadas para almacenar información definida por el usuario e indicadores estadísticos. Estas funcionalidades adicionales se han programado en Modelica por medio de dos librerías externas adicionales, llamadas “entities.c” y “objects.c”. La primera se ha desarrollado para gestionar la información necesaria para describir las entidades y su flujo a través del sistema. La segunda se ha desarrollado para gestionar objetos dinámicos, usados para describir matrices y listas de tamaño variable en memoria dinámica.
- Usando las funcionalidades desarrolladas, se han desarrollado dos nuevas librerías en Modelica, llamadas SIMANLib y ARENALib, para soportar la descripción de modelos siguiendo la aproximación de modelado orientado al proceso. Estas nuevas librerías reproducen algunas de las funcionalidades proporcionadas por el lenguaje SIMAN y el entorno de simulación Arena.

El desarrollo de SIMANLib se ha realizado usando la librería DEVSLib, ya que sus componentes han sido definidos como modelos P-DEVS atómicos. De manera similar al desarrollo de Arena, cuyos componentes están descritos mediante el lenguaje SIMAN, el desarrollo de ARENALib ha sido realizado usando los componentes de SIMANLib. Por tanto, los componentes de SIMANLib y ARENALib pueden ser ordenados de manera jerárquica y combinados para construir modelos usando diferentes niveles de abstracción.

- Las funcionalidades incluidas en SIMANLib y ARENALib se han extendido para facilitar la descripción de modelos híbridos orientados al proceso. Los bloques Create y ExternalAssign de SIMANLib, y los módulos Assign y ExternalProcess de ARENALib pueden ser utilizados para combinar modelos orientados al proceso con otros modelos de Modelica.
- Se ha desarrollado también una nueva librería en Modelica, llamada RandomLib, para facilitar la generación de números y muestras aleatorias. RandomLib, conjuntamente con DEVSLib, SIMANLib y ARENALib, puede ser utilizada para describir modelos estocásticos de sistemas logísticos. RandomLib incluye la programación del generador de números aleatorios CM-RG, también incluido en Arena, para facilitar la validación de los modelos desarrollados por medio de comparaciones con modelos equivalentes desarrollados usando SIMAN o Arena. RandomLib incluye también funciones para generar números aleatorios uniformes y muestras aleatorias de múltiples distribuciones de probabilidad, continuas y discretas.
- Finalmente, se han desarrollado varios casos de estudio con el fin de presentar las funcionalidades y uso de las librerías desarrolladas. Se han descrito modelos de un cajero automático, un sistema de presa-depredador, un sistema de comunicación opto-electrónica, un sistema de refrigeración de un supermercado, un sistema de grúa con controlador empotrado, un restaurante, una fábrica de componentes electrónicos, una fábrica de zumo de



naranja enlatado, un sistema de control del nivel de un tanque y un sistema de horno de fosa.

## 2.2 Futuras Líneas de Investigación

---

Algunas ideas de futuros trabajos de investigación podrían ser:

- La mejora del rendimiento de la simulación de modelos de eventos discretos puros e híbridos en Modelica, siendo descritos usando el formalismo P-DEVS.
- El análisis y definición de un posible lenguaje estándar para la descripción de modelos P-DEVS, la combinación de dicho lenguaje con lenguajes EOO, y la traducción de sus modelos a modelos DEVSLib que puedan ser simulados usando Dymola. Dicho lenguaje se podría utilizar para compartir modelos entre herramientas que soporten P-DEVS.
- La extensión de las funcionalidades incluidas en SIMANLib y ARENALib, por medio del desarrollo de componentes adicionales.
- La representación gráfica de la ejecución de los modelos orientados al proceso en Modelica, para permitir una mejor comprensión del comportamiento del modelo durante la simulación



# Bibliografía

- Andersson, M. [1989], Omola - an object-oriented language for model representation, Technical report, TFRT 7417, Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- Åström, K. J., Elmqvist, H. and Mattsson, S. E. [1998], Evolution of continuous-time modeling and simulation, *in* 'Proceedings of the 12<sup>th</sup> European Simulation Multiconference (ESM'98)', Manchester, UK, pp. 9–18.
- Augustin, D. C., Fineberg, M. S., Johnson, B. B., Linebarger, R.Ñ., Sansom, F. J. and Strauss, J. C. [1967], 'The SCi continuous system simulation language (CSSL)', *Simulation* **9**, 281–303.
- Beltrame, T. and Cellier, F. E. [2006], Quantised state system simulation in Dymola/Modelica using the DEVS formalism, *in* 'Proceedings of the 5<sup>th</sup> International Modelica Conference', Vienna, Austria, pp. 73–82.
- Breunese, A. P. J. and Broenink, J. F. [1997], 'Modeling mechatronic systems using the SIDOPS+ language', *Simulation Series* **29**(1), 301–306.
- Cellier, F. E. [1996], Object-oriented modeling: Means for dealing with system complexity, *in* 'Proceedings of the 15<sup>th</sup> Benelux Meeting on Systems and Control', Mierly, The Netherlands, pp. 53–64.

- Cellier, F. E. [2008], World3 in Modelica: Creating System Dynamics models in the Modelica framework, *in* ‘Proceedings of the 6<sup>th</sup> International Modelica Conference’, Bielefeld, Germany, pp. 393–400.
- Cellier, F. E. and Nebot, A. [2005], The Modelica bond graph library, *in* ‘Proceedings of the 4<sup>th</sup> International Modelica Conference’, Vol. 1, Hamburg, Germany, pp. 57–65.
- Chow, A. C. H. [1996], ‘Parallel DEVS: A parallel, hierarchical, modular modeling formalism and its distributed simulator’, *Transactions of the Society for Computer Simulation International* **13**(2), 55–67.
- Elmqvist, H. [1978], A Structured Model Language for Large Continuous Systems, PhD thesis, Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- Ferreira, J. and de Oliveira, J. E. [1999], Modelling hybrid systems using State-Charts and Modelica, *in* ‘Proceedings of the 7<sup>th</sup> IEEE International Conference on Emerging Technologies and Factory Automation’, pp. 1063–1069.
- Fritzson, P. [2003], *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*, Wiley-IEEE Computer Society Pr.
- Fritzson, P., Viklund, L., Fritzson, D. and Herber, J. [1995], ‘High-level mathematical modelling and programming’, *IEEE Software* **12**(4), 77–87.
- Giambiasi, N. and Carmona, J. C. [2006], ‘Generalized discrete event abstraction of continuous systems: GDEVS formalism’, *Simulation Modelling Practice and Theory* **14**(1), 47 – 70.
- Jeandel, A., Boudaud., F. and Larivière, E. [1997], *ALLAN Simulation release 3.1 description*, M.DÉGIMA.GSA1887. GAZ DE FRANCE, DR, Saint Denis La plaine, France.
- Kelton, W. D., Sadowski, R. P. and Sturrock, D. T. [2007], *Simulation with Arena*, 4<sup>th</sup> edn, McGraw-Hill, Inc., New York, NY, USA.

- Kiviat, P. J. [1969], Digital computer simulation: Computer programming languages, Technical report, RAND Memo RM-5883-PR. RAND Corporation. Santa Monica, California.
- Kloas, M., Friesen, V. and Simons, M. [1995], ‘Smile - a simulation environment for energy systems’, *System Analysis Modelling Simulation* **18–19**, 503–506.
- Kofman, E. [2004], ‘Discrete event simulation of hybrid systems’, *SIAM Journal on Scientific Computing* **25**(5), 1771–1797.
- Law, A. M. [2007], *Simulation Modelling and Analysis*, 4<sup>th</sup> edn, McGraw-Hill, New York, NY, USA.
- Mattsson, S. E., Elmqvist, H. and Broenink, J. [1998], ‘Modelica – an international effort to design the next generation modeling language’, *Journal A, Benelux Quarterly Journal on Automatic Control* **38**(3), 16–19.
- Mattsson, S. E., Otter, M. and Elmqvist, H. [1999], Modelica hybrid modeling and efficient simulation, in ‘Proceedings of the 38<sup>th</sup> IEEE Conference on Decision and Control’, pp. 3502–3507.
- Modelica [2010], ‘Modelica web site’.  
**URL:** <http://www.modelica.org>
- Modelica Association [2009], ‘Modelica - a unified object-oriented language for physical systems modeling. language specification (v. 3.1)’.  
**URL:** <http://www.modelica.org/documents>
- Modelica Libraries [2010], ‘Modelica free and comercial libraries’.  
**URL:** <http://www.modelica.org/libraries>
- Mosterman, P. J., Otter, M. and Elmqvist, H. [1998], Modelling Petri Nets as local constraint equations for hybrid systems using Modelica, in ‘Proceedings of the Summer Computer Simulation Conference’, pp. 314–319.
- Otter, M., Årzén, K.-E. and Dressler, I. [2005], StateGraph - a Modelica library for hierarchical state machines, in ‘Proceedings of the 4<sup>th</sup> International Modelica Conference’, Hamburg, Germany, pp. 569–578.

- Otter, M., Elmqvist, H. and Mattsson, S. E. [1999], Hybrid modeling in Modelica based on the synchronous data flow principle, *in* 'Proceedings of the 10<sup>th</sup> IEEE International Symposium on Computer Aided Control System Design', pp. 151–157.
- Pegden, C. D., Sadowski, R. P. and Shannon, R. E. [1995], *Introduction to Simulation Using SIMAN*, McGraw-Hill, Inc., New York, NY, USA.
- Sahlin, P., Brign, A. and Sowell, E. F. [1996], The neutral model format for building simulation (v. 3.02), Technical report, Dept. of Building Sciences, The Royal Institute of Technology, Stockholm, Sweden.
- Vangheluwe, H. L. M. [2000], DEVS as a common denominator for multi-formalism hybrid systems modelling, *in* 'Proceedings of the IEEE International Symposium on Computer-Aided Control System Design', IEEE Computer Society Press, pp. 129–134.
- Zeigler, B. P. [1976], *Theory of Modelling and Simulation*, John Wiley & Sons, Inc.
- Zeigler, B. P. [1989], 'DEVS representation of dynamical systems: Event-based intelligent control', *Proceedings of the IEEE* **77**(1), 72–80.