



DEPARTAMENTO  
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

# Localización y modelado simultáneos mediante generación y actualización automática de controladores discretos

6 de diciembre de 2015

Propuesta de tesis

**Director: Nicolas D'Ippolito**

Mail: [srdipi@gmail.com](mailto:srdipi@gmail.com)

Integrante	LU	Correo electrónico
Pasquini, Ivan	141/09	<a href="mailto:pasquiniivan@gmail.com">pasquiniivan@gmail.com</a>



**Facultad de Ciencias Exactas y Naturales**  
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

## 1. Introducción

### 1.1. La síntesis de controladores

El modelo del mundo y la máquina de Michael Jackson establece un marco desde el cual aproximarse a la ingeniería de requerimientos. En este modelo, los requerimientos  $R$  son declaraciones prescriptivas sobre el mundo expresadas en términos de fenómenos sobre la interfaz entre la máquina que queremos construir y el mundo en el cual viven los problemas que queremos resolver. Dichos problemas son capturados como declaraciones prescriptivas expresadas en término de fenómenos del mundo llamados objetivos  $G$ , y declaraciones descriptivas sobre lo que nosotros asumimos que es verdad en el mundo, suposiciones de dominio  $D$ .

La tarea clave de la ingeniería de requerimientos es comprender y documentar los objetivos y las características del entorno. Teniendo estos modelos se pueden formular un conjunto de requerimientos para la máquina de forma que en conjunto con el entorno satisfagan los objetivos. Más formalmente  $R, D \models G$ .

Esto puede ser formulado como un problema de síntesis. Dados un conjunto de suposiciones del dominio y un conjunto de objetivos del sistema, construir automáticamente un modelo operacional de la máquina tal que compuesto con el modelo del entorno, los objetivos sean alcanzados. Para cumplir los objetivos, el modelo restringe la ocurrencia de eventos controlables basándose en la observación de los eventos que ya ocurrieron. Este problema se conoce como el problema de síntesis de controladores y está siendo estudiado exhaustivamente en varios aspectos de la ingeniería de los requerimientos.

Los Labelled Transition Systems (LTSs) son ampliamente utilizados para modelar y analizar el comportamiento de sistemas concurrentes y distribuidos. Son un sistema de transición de estados en el cual las transiciones son etiquetadas con acciones. El conjunto de acciones de un LTS se conoce como alfabeto comunicacional y constituye las interacciones que el sistema modelado puede tener con el entorno.

Los fluents nos permiten especificar propiedades basadas en estados en modelos basados en eventos. FLTL es una lógica temporal lineal que nos permite razonar sobre fluents. Un fluent está compuesto por un conjunto de acciones que lo activa, un conjunto de acciones que lo terminan y un valor booleano que indica si comienza activo o no.

Podemos describir la síntesis de controladores de la siguiente forma. Dados un LTS que describe el comportamiento del entorno, un conjunto de acciones controlables, un conjunto de fórmulas FLTL que representan las suposiciones del dominio y un conjunto de fórmulas FLTL que representan los objetivos del sistema, el problema de control de LTS es encontrar un LTS que solo restrinja la ocurrencia de acciones controlables y garantice que la composición en paralelo entre el ambiente y el LTS es deadlock free y que si las suposiciones de dominio se satisfacen, entonces los objetivos del sistema también son satisfechos.

### 1.2. La exploración en robótica

En el área de robótica, el problema de la exploración consiste en recorrer mediante un vehículo autónomo una zona desconocida para obtener el conocimiento total de este o en caso de no poder reconocerlo en su totalidad, maximizar el espacio explorado. La utilización de robots es muy importante para la cartografía o búsqueda y rescate en lugares que son peligrosos o inaccesibles para las personas. En el caso de la utilización de robots autónomos, se utiliza la técnica de localización y modelado simultáneos, para construir un mapa de un entorno desconocido en el que se encuentra el robot, a la vez que estima su trayectoria al desplazarse dentro del entorno.

Nuestro modelo matemático para atacar dichos problemas consiste en los MTS (Modal Transi-

tion System), que son nociones abstractas de los LTSs. Extienden a los LTSs ya que las transiciones que los MTSs poseen pueden denotar eventos requeridos o bien posibles. Debido a esta particularidad de los MTSs, es que podemos modelar mediante este formalismo información parcial del mundo.

Hay una relación de refinamiento entre los MTSs y los LTSs. Un LTS se puede ver como un MTS en donde la función de transición de las acciones posibles es igual que la función de transición de las acciones requeridas. Los LTSs que refinan un MTS son descripciones completas del comportamiento del sistema y se llaman implementaciones.

Cuando comenzamos la exploración, la información que tenemos sobre el mundo es parcial. Podemos codificarla en un MTS en donde la información que tenemos son transiciones requeridas y la información que podemos adquirir mediante exploración son transiciones posibles. El mundo puede representarse como un LTS que refina nuestro MTS inicial.

El problema de la síntesis de controladores para MTSs consiste en ver si todas, alguna o ninguna de sus implementaciones pueden ser controladas por un controlador LTS. Al responder esta pregunta, también estamos respondiendo la pregunta de si el problema es realizable, ya que si la respuesta es ninguno el problema no es realizable.

Cuando exploramos, lo hacemos para poder realizar un objetivo. Si intentamos sintetizar una máquina tal que al componerla con el MTS que representa nuestro conocimiento del mundo, para todas las implementaciones posibles, el objetivo es alcanzable, sabemos que podemos cumplir el objetivo, ya que el mundo puede representarse como un LTS que refina nuestro modelo. Cuando no es posible sintetizar un controlador que pueda garantizar que los objetivos se alcancen para ninguna de las implementaciones de nuestro modelo, podemos afirmar que el objetivo no es realizable en el mundo. Pero si para algunas de las implementaciones existe un controlador, y para otras no, no podemos afirmar nada sobre nuestro objetivo.

El trabajo de esta tesis consistirá en:

- Construir una herramienta que nos permita explorar para refinar nuestro conocimiento inicial.
- Identificar que condiciones se deben cumplir para poder afirmar si nuestro objetivo es realizable o no.
- Dar una respuesta en caso de que las condiciones se cumplan.
- Analizar cuanto fue necesario refinar para poder dar la respuesta.

## 2. Objetivos

Nuestro primer objetivo es identificar los posibles inputs y outputs. Necesitamos saber que información es necesario proporcionarle al robot para que pueda explorar, y en que forma vamos a proporcionarle dicha información. También es importante definir que esperamos obtener de la exploración y como vamos a mostrarlo.

Lo siguiente es identificar que condiciones debemos proveer para garantizar que hay una solución de exploración. Tenemos que averiguar si podemos garantizarlo, si requerimos cumplir algunas condiciones para que esto ocurra, y en que casos serían necesarias estas condiciones.

El próximo paso es definir formalmente el problema de exploración. Necesitamos un modelo sobre el cual demostrar nuestros teoremas sobre la exploración.

Vamos a elaborar la implementación que resuelva computacionalmente el problema definido. Lo haremos sobre la herramienta MTSA. Necesitamos una herramienta que nos permita investigar

la exploración en detalle.

Por ultimo vamos a probar casos de uso relevantes a la solución propuesta. Serán casos que consideremos útiles para ver las particularidades de la exploración, y casos de estudio de la bibliografía.

### 3. Plan de trabajo

Presentamos un plan de trabajo cascada simplificado que tiene como objetivo mostrar las tareas que se llevan la mayor parte del esfuerzo en cada momento. En la práctica, se utiliza un proceso iterativo. Representaremos a continuación de una manera abstracta las siguientes tareas candidatas:

- Formalización del problema identificando distintos escenarios posibles [Formalización]
- Diseño, prueba e implementación de algoritmos correctos para el problema en cada escenario [Implementación]
- Validación de la técnica en casos de estudio [Validación]
- Aplicaciones motivadoras [Aplicaciones]

	Meses							
Actividad	1	2	3	4	5	6	7	8
Implementación	light-gray	light-gray	light-gray	light-gray	light-gray	light-gray	light-gray	light-gray
Formalización				light-gray	light-gray	light-gray	light-gray	light-gray
Validación							light-gray	light-gray
Aplicaciones								

## Referencias

- [1] R. Bloem, B. Jobstmann, N. Piterman, A. Pnueli, and Y. Saar. Synthesis of reactive(1) designs. *J. Comput. Syst. Sci.*, 78(3):911–938, May 2012.
- [2] N. D’ Ippolito, D. Fischbein, M. Chechik, and S. Uchitel. Mtsa: The modal transition system analyser. In *Automated Software Engineering, 2008. ASE 2008. 23rd IEEE/ACM International Conference on*, pages 475–476, Sept 2008.
- [3] N. R. D’ Ippolito, V. Braberman, N. Piterman, and S. Uchitel. Synthesis of live behaviour models. In *Proceedings of the Eighteenth ACM SIGSOFT International Symposium on Foundations of Software Engineering, FSE ’10*, pages 77–86, New York, NY, USA, 2010. ACM.
- [4] N. D’ippolito, V. Braberman, N. Piterman, and S. Uchitel. The modal transition system control problem. In *19th International Symposium on Formal Methods*, volume 7436, pages 155–170, Paris, France, 2012.
- [5] D. Giannakopoulou and J. Magee. Fluent model checking for event-based systems. In *Proceedings of the 9th European Software Engineering Conference Held Jointly with 11th ACM SIGSOFT International Symposium on Foundations of Software Engineering, ESEC/FSE-11*, pages 257–266, New York, NY, USA, 2003. ACM.
- [6] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata Logics, and Infinite Games: A Guide to Current Research*. Springer-Verlag New York, Inc., New York, NY, USA, 2002.
- [7] J. Magee and J. Kramer. *Concurrency: State Models & Java Programs*. John Wiley & Sons, Inc., New York, NY, USA, 1999.