

Grado en Ingeniería Informática
Computación

Trabajo de Fin de Grado

**Transpilador de un lenguaje de modelado
personalizado de sistemas de simulación
dinámicos discretos a Python**

Autor

Baldwin David Rodríguez Ponce

2022

BORRADOR

Grado en Ingeniería Informática
Computación

Trabajo de Fin de Grado

**Transpilador de un lenguaje de modelado
personalizado de sistemas de simulación
dinámicos discretos a Python**

Autor

Baldwin David Rodríguez Ponce

Directora

María Begoña Losada Pereda

BORRADOR

Tabla de contenidos

1. Introducción	1
1.1. Motivación y planteamiento del proyecto	2
1.2. Alcance	3
1.3. Objetivos	3
1.4. Antecedentes	5
1.4.1. TFG del estudiante de la ETSIIT de la UGR (generador de sistemas de simulación, nombre de sección provisional)	5
1.4.2. Lenguajes de simulación	5
1.4.3. Software de simulación	5
2. Simulación	7
2.1. Conceptos básicos	7
2.1.1. Modelo	7
2.1.2. Sistema	7
2.2. Ventajas e inconvenientes	8
2.3. Diseño basado en eventos	8
2.4. Sistemas dinámicos discretos	8
3. Compilación	9
3.1. Conceptos básicos	9
3.1.1. Analizador léxico	9
3.1.2. Analizador sintáctico	9
3.1.3. Analizador semántico	10
3.2. Herramientas	10
Bibliografía	11

Anexos

A. Artefactos relacionados a la planificación del proyecto	15
A.1. Esquema de Descomposición de Trabajo	15

1.CAPÍTULO

Introducción

El modelado matemático o modelado analítico se aprovecha de las características del problema cuya respuesta se desea obtener para llegar a la mejor conclusión. Sin embargo, muchas veces nos encontraremos con problemas que no se pueden modelar analíticamente debido a su complejidad en tiempo o espacio. En estas situaciones, debemos hacer uso de técnicas de aproximación de resultados para dar con soluciones que, aunque no se pueden garantizar óptimas, sí que se pueden considerar lo suficientemente buenas. Una de estas técnicas vendría a ser la simulación de sistemas, la cual a través de modelado simbólico/lógico intenta reproducir el comportamiento de un determinado sistema con el fin de analizar una serie de resultados a escoger.

Nosotros hablaremos de una definición más exacta de “Simulación de Sistemas” y “Modelado” más adelante, pero por ahora podemos citar la definición encontrada en Banks, 2010:

Una simulación es la imitación del comportamiento de un proceso o sistema del mundo real en el tiempo. **Simulation involves the generation of an artificial history of the system, and the observation of that artificial history to draw inferences concerning the operating characteristics of the real system that is represented.**

Actualmente hay muchos tipos distintos de sistemas de simulación, como por ejemplo los “Modelos de Montecarlo” y “Modelos en ecuaciones diferenciales”. No obstante, el principal objeto de estudio de este proyecto será una categoría muy importante a la que se conoce como “Modelos de simulación dinámicos discretos”, específicamente aquellos

Toca traducir lo que queda. Además asegúrate de que la cita sea correcta, Jerry Banks aparece dos veces en tus referencias

que se pueden modelar a través de un diseño orientado a eventos.

Sin embargo, es necesario recalcar que existe ya una multitud de aplicaciones orientadas totalmente a la simulación de sistemas: Arena, AutoMod, Extend, entre otros. El hecho de que exista software específico para esto nos da un indicativo de lo importante que es esta área para el estudio de resultados y el modelado de procesos. De hecho, el tipo específico de modelos del cual hablaremos es muy utilizado en sistemas de salud, de transporte, control de inventarios, líneas de montaje o ensamblaje, entre otros.

Tienes que hacer una cita aquí, esto no lo estás sacando de la nada.

No olvides citar esta lista de aplicaciones.

Has dicho software, pero en realidad sí que hay lenguajes dedicados a esto

Como mencionábamos antes, ya hay lenguajes de programación enfocados a la simulación de sistemas. Lo cual nos indica la unión de este campo con el de compilación y diseño de procesadores de lenguajes formales. Ahora mismo parece ser el momento más idóneo para usar un procesador de este tipo ya que debemos seguir una serie de tokens y una gramática fija en estos lenguajes.

Se puede ver, por tanto, que la unión entre los campos de compilación y simulación de sistemas puede generar herramientas diseñadas específicamente para agilizar el proceso de creación e implementación de modelos sin preocuparse mucho por detalles irrelevantes para el programador de la simulación.

Sabiendo todo esto, procederemos hablar en este capítulo sobre la propuesta de nuestro proyecto, sus objetivos y una serie de conceptos básicos necesarios para la simulación de sistemas y la compilación en general.

Un poco seco, cambia la redacción

1.1. Motivación y planteamiento del proyecto

En el área de simulación de sistemas, una familia de modelos son los “modelos dinámicos discretos”, siendo estos “dinámicos” porque su comportamiento cambia en el tiempo y “discretos” porque estos cambios ocurren en instantes determinados en vez de ocurrir continuamente. Dichos sistemas se suelen modelar usando un diseño “basado en eventos”, el cual nos indica que los cambios del modelo se deben considerar como “eventos” que pueden ocurrir y dar lugar a otros eventos. Para ello, es común generar un “grafo de sucesos” que represente todos los eventos posibles que pueden ocurrir en el sistema y las relaciones que tienen entre estos.

Siento que estás repitiéndote mucho, tienes que cambiar la calidad de este discurso

Podríamos decir que cada modelo tendrá sus peculiaridades y diferencias específicas a la hora de generar su implementación. Sin embargo, se da el hecho de que todos los modelos de esta familia comparten elementos en común independientemente del sistema a simular: el reloj de la simulación, el temporizador de eventos, la lista de sucesos, entre otros. Por

tanto, podemos abstraer el desarrollo de estos programas de tal forma que el desarrollador sólo deba encargarse de implementar todo aquello que sea único del modelo, quitándole así la responsabilidad de generar los elementos en común y agilizando el desarrollo en el proceso.

Tomando en cuenta que la principal herramienta de diseño de estos modelos serán los grafos de sucesos, nos encontraremos con el hecho de que éstos siguen una estructura representable a través de una gramática independiente de contexto. Por tanto, es posible generar una serie de analizadores léxico, sintáctico y semánticos que nos permitan procesar un lenguaje formal a otro código fuente. Por esta razón hemos considerado una buena opción el uso de herramientas de generación de compiladores como lo son Flex y Bison. Vemos que es posible crear estos analizadores de forma que este hipotético nuevo lenguaje sea traducido a código Python.

1.2. Alcance

El alcance de este proyecto incluye el trabajo necesario para diseñar, implementar y documentar los analizadores léxico, sintáctico y semántico de un nuevo lenguaje de programación enfocado en la agilización del desarrollo de modelos de simulación dinámicos discretos basados en eventos.

Dicho lenguaje será convertido a código Python, por lo que se incluirá una serie de funciones y módulos adicionales que funcionarán a modo de librería básica del programa una vez traducido éste. Por tanto, para poder generar este transpilador, haremos uso de las herramientas de desarrollo de compiladores Flex y Bison.

Por último, la gestión del proyecto se realizará siguiendo la metodología de desarrollo ágil Kanban para permitir construir de manera iterativa las distintas funcionalidades del proyecto.

Un poco crudo, ¿quizás puedes decir un poco más al respecto?

1.3. Objetivos

Conocido ya el alcance, podemos proceder a dar la lista de objetivos principales del proyecto:

- Hacer uso del método Kanban para el desarrollo y gestión del proyecto.
- Diseñar un nuevo lenguaje de programación inspirado en el léxico y sintaxis de

Así como lista no me convence mucho, revisa si puedes convertirlo a subsecciones con nombres relevantes (gestión, diseño, desarrollo...)

Flex, Bison y Python. Dicho lenguaje nos permitirá implementar simuladores de sistemas dinámicos discretos cuyos diseños estarán basados en eventos, pero se podrán implementar otros tipos de diseños a través de estos. Se debe permitir la rápida implementación de este tipo de modelos a través de grafos de sucesos.

- Hacer uso de las herramientas de generación de compiladores Flex y Bison con el fin de construir los analizadores léxico, sintáctico y semántico del nuevo lenguaje.
 - Permitir que la traducción del lenguaje incluya dentro del fichero generado las estructuras de datos, funciones y procedimientos que tienen en común todos los sistemas dinámicos discretos:
 - Generadores de datos aleatorios para distintos tipos de distribuciones.
 - Reloj y temporizador de simulación para ejecutar los eventos.
 - Estructura de datos para almacenar los sucesos según deben ocurrir en el tiempo.
 - Las respectivas implementaciones mínimas de los dos eventos que siempre formarán parte de todos los modelos: “Inicio” y “Fin”.
 - Generador de informes final que se ejecutará al finalizar la simulación y mostrará los resultados que se deseaban estudiar con ésta.
 - Permitir que el programador del lenguaje se encargue sólo de realizar las implementaciones pertinentes al sistema de simulación que desee desarrollar:
 - Especificación de las variables globales, variables de entrada, contadores estadísticos y medidas de rendimiento propias del modelo.
 - Inclusión de eventos adicionales y sus acciones correspondientes.
 - Creación y eliminación de eventos en función de tiempo y condicionales lógicas.
 - Inclusión de código adicional escrito directamente en Python en caso de ser necesario.
 - Permitir la configuración del comportamiento del programa traducido a través de parámetros de entrada.
-

1.4. Antecedentes

1.4.1. TFG del estudiante de la ETSIIT de la UGR (generador de sistemas de simulación, nombre de sección provisional)

1.4.2. Lenguajes de simulación

Mencionar que hay dos categorías de lenguajes de programación para desarrollar los simuladores: específicos y generales. El nuestro será específico.

Aquí mencionaré algunos lenguajes de programación creados con el fin de desarrollar simuladores

1.4.3. Software de simulación

Aquí mencionaré algunas aplicaciones creadas para desarrollar simulaciones (como Arena)

2.CAPÍTULO

Simulación

Definición de simulación

Explicación sobre simulación de sistemas.

¿Por qué y para qué la simulación de sistemas?

Puedes citar a uno de tus libros aquí.

2.1. Conceptos básicos

2.1.1. Modelo

Haz una cita a la definición de tu profesor de simulación de sistemas (igual no se puede)

Definición de modelo (Nos centraremos en modelos probabilísticos)

¿Clasificación de modelos? (los modelos de simulación deben ser simbólicos (no tienen una relación física o analógica con el sistema real, sino una relación lógica))

¿Qué significa ser dinámico?

¿Qué significa ser discreto?

2.1.2. Sistema

Definición de sistema

Partes y conceptos de un sistema de simulación (entorno del sistema, entidad, atributo, actividad, estado, suceso, término endógeno, término exógeno, contadores estadísticos, medidas de rendimiento)

2.2. Ventajas e inconvenientes

Ventajas e inconvenientes

2.3. Diseño basado en eventos

Grafo de sucesos

2.4. Sistemas dinámicos discretos

Partes fundamentales (en común) de los simuladores dinámicos discretos (lista de sucesos, temporizador de eventos, reloj...)

3.CAPÍTULO

Compilación

Habla aquí de los procesadores del lenguaje (formal)

¿Qué significa compilar?

Definición de Transpilador

3.1. Conceptos básicos

3.1.1. Analizador léxico

Lenguajes regulares

Expresiones regulares

Tokens

3.1.2. Analizador sintáctico

Lenguajes libres de contexto

Gramática libre de contexto

ETDS y acciones

3.1.3. Analizador semántico

Tabla de símbolos

Verificación estática

3.2. Herramientas

Compilador C++

Lex y Flex

Yacc y Bison

Bibliografía

- Buss, A. (1996). Modeling with event graphs. *Winter Simulation Conference*, 153-160.
<https://doi.org/10.1109/WSC.1996.873273>
- Maria, A. (1997). Introduction to modeling and simulation. *Proceedings of the 29th conference on Winter simulation - WSC '97*. <https://doi.org/10.1145/268437.268440>
- Perros, H. G. (2009). *Computer simulation techniques: the definitive introduction!*
- Banks, J. (2010). *Discrete-event system simulation*. Prentice Hall.
- Wainer, G. (2011). *Discrete-event modeling and simulation: theory and applications*. CRC Press.
- MATLAB. (2017). Understanding Discrete Event Simulation, Part 1: What Is Discrete Event Simulation.

Anexos

Artefactos relacionados a la planificación del proyecto

A.1. Esquema de Descomposición de Trabajo

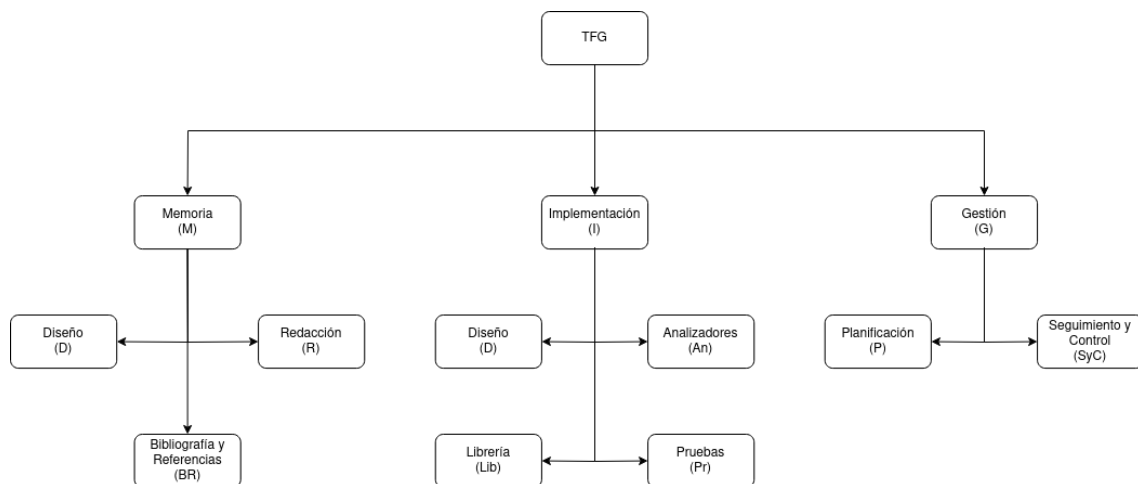


Figura A.1: Esquema de Descomposición de Trabajo del Proyecto

