

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. Antecedentes	3
1.2.1. TFG del estudiante de la ETSIIT de la UGR (generador de sistemas de simulación, nombre de sección provisional)	3
1.2.2. Lenguajes de simulación	3
1.2.3. Software de simulación	3
2. Gestión del proyecto	5
2.1. Alcance	5
2.1.1. Objetivos	6
2.1.2. Requisitos	7
2.1.3. Entregables	8
2.1.4. Exclusiones	8
2.1.5. Supuestos	8
2.1.6. EDT	8
2.2. Metodología	9
2.2.1. Significado de las columnas/tablas/listas	9
2.2.2. Significado de etiquetas	9
2.2.3. Formato de las cartas	9
2.3. Tareas y estimación de dedicaciones	9
2.3.1. Descripción de tareas a realizar	10
2.3.2. Dependencias entre tareas	10
2.3.3. Periodo de desarrollo de tareas	10
2.3.4. Estimación de dedicación a cada una de las tareas	10
2.3.5. Hitos de desarrollado	10

2.4. Análisis de riesgos y viabilidad	10
2.5. Caracterización del sistema de información y del sistema de comunicaciones	10
2.5.1. Sistema de Información	10
2.5.2. Comunicaciones	11
3. Simulación	13
3.1. Conceptos básicos	13
3.1.1. Modelo	13
3.1.2. Sistema	13
3.2. Ventajas e inconvenientes	14
3.3. Diseño basado en eventos	14
3.4. Sistemas dinámicos discretos	14
Bibliografía	15
Anexos	
A. Artefactos relacionados a la planificación del proyecto	19
A.1. Esquema de Descomposición de Trabajo	19

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. Antecedentes

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

1.2.2. 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

1.2.3. Software de simulación

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

2.CAPÍTULO

Gestión del proyecto

2.1. Alcance

El alcance de este proyecto incluye el trabajo necesario para diseñar, implementar y documentar un framework de modelado y desarrollo de simulación de sistemas dinámicos discretos basados en eventos. Dicho framework se dividirá en tres módulos principales:

- Lenguaje: Un lenguaje de modelado específico pensado para ser usado por usuarios que no tengan mucha experiencia en programación. Se hará uso de las herramientas de desarrollo de compiladores Flex y Bison para generar un transpilador que traduzca ficheros de este lenguaje a código Python. A través de él se plantea:
 - Permitir la rápida implementación de este tipo de modelos a través de grafos de sucesos.
 - 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 condiciones lógicas.
 - Inclusión de código adicional escrito directamente en Python en caso de

Cambiar redacción

ser necesario.

- Núcleo: Una serie de módulos que implementarán un microframework de simulación de este tipo de sistemas en específico para Python, pensado para ser usado por programadores y para acotar la traducción del nuevo lenguaje. A través de él se plantea:
 - 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.
- CLI: Una interfaz de comandos por terminal que se usará para gestionar, configurar y ejecutar los proyectos desarrollados con este framework.

2.1.1. Objetivos

Objetivo general

Generar un lenguaje de modelado de simulación de sistemas dinámicos discretos basados en eventos junto con un transpilador que lo traduzca a Python, un microframework para acotar la traducción y un CLI para gestionar proyectos desarrollados con este producto.

Objetivos específicos

1. Diseñar un nuevo lenguaje de modelado y simulación de sistemas dinámicos discretos basados en eventos usando las herramientas de desarrollo de procesadores de lenguaje Flex y Bison.

2. Diseñar, implementar y verificar una serie de módulos y funcionalidades realizadas en Python con el fin de generar un microframework para simular estos mismos sistemas.
3. Implementar y verificar un transpilador que traduzca el lenguaje del producto a Python usando las características del objetivo anterior.
4. Diseñar, implementar y verificar una serie de operaciones accesibles desde una CLI de cara a ser usadas para la gestión, configuración y ejecución parametrizada de simulaciones realizadas con el producto.
5. Implementar distintas técnicas de análisis de salidas, experimentación y optimización de modelos dentro del núcleo del framework.
6. Desarrollar un manual de usuario que contendrá toda la documentación necesaria para hacer uso del framework.

2.1.2. Requisitos

El requisito base principal del proyecto consiste en cumplir con un tiempo de dedicación total máximo de 300 horas. Sin embargo, se pueden listar otros requisitos específicos:

Caracterización de la memoria

Debe ser bien citado y referenciado para evitar el plagio Se debe utilizar una metodología de citas. Apartado de calidad

Caracterización del framework

Un lenguaje verificado que no pete a la primera El resultado son métricas en un dataframe (o varios si harás lo de validación) Cumplir con la línea base de calidad definida en el apartado (...)

Caracterización del manual

Utilizar referencias a recursos ajenas

Licencia del producto

2.1.3. Entregables

- Memoria:
- Framework:
- Manual:
- Planificación:
- Seguimiento y Control:
- Actas:

2.1.4. Exclusiones

Mi proyecto devolverá los datos en un sólo formato, no devolveré más tipos de ficheros para darle gusto al usuario.

2.1.5. Supuestos

2.1.6. EDT

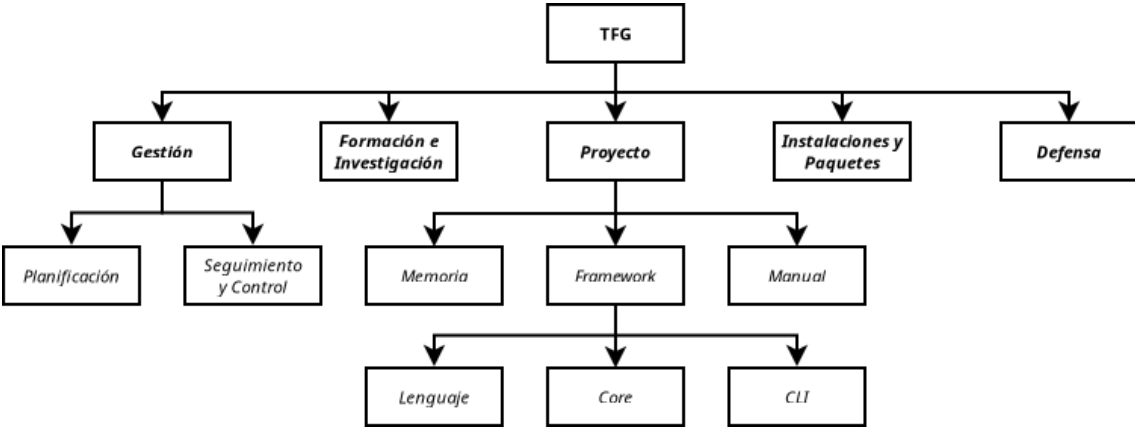


Figura 2.1: Esquema de Descomposición de Trabajo del Proyecto

2.2. Metodología

Explicar Kanban

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. Hacer uso del método Kanban para el desarrollo y gestión del proyecto.

2.2.1. Significado de las columnas/tablas/listas

- Backlog:
- To Do:
- Doing:
- Testing:
- Done:
- Approved:

2.2.2. Significado de etiquetas

- Bug:
- Bloqueado:
- Pendiente:
- No aceptada:

2.2.3. Formato de las cartas

2.3. Tareas y estimación de dedicaciones

No se puede predecir porque usamos Kanban. Sólo conocemos las fechas finales de entrega.

2.3.1. Descripción de tareas a realizar**2.3.2. Dependencias entre tareas****2.3.3. Periodo de desarrollo de tareas****2.3.4. Estimación de dedicación a cada una de las tareas****2.3.5. Hitos de desarrollado****2.4. Análisis de riesgos y viabilidad****2.5. Caracterización del sistema de información y del sistema de comunicaciones****2.5.1. Sistema de Información****Estructura**

Aquí debemos explicar la estructura

TFG: Directorio principal del proyecto

- TFG.Proyecto:
 - Memoria:
 - Framework:
 - Lenguaje:
 - Core:
 - CLI:
 - Manual:
 - TFG.Gestión:
 - Planificación:
 - Seguimiento:
 - Actas:
 - TFG.Defensa:
-

Formato

Denominación

Copias de seguridad

Usamos GitHub para control de versiones y copias de seguridad

2.5.2. Comunicaciones

Reuniones [Comunicaciones para responder dudas]

3.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.

3.1. Conceptos básicos

3.1.1. 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?

3.1.2. 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)

3.2. Ventajas e inconvenientes

Ventajas e inconvenientes

3.3. Diseño basado en eventos

Grafo de sucesos

3.4. Sistemas dinámicos discretos

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

Bibliografia

- 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. (2017a). Understanding Discrete Event Simulation, Part 1: What Is Discrete Event Simulation.
- MATLAB. (2017b). Understanding Discrete Event Simulation, Part 2: Why Use Discrete Event Simulation.
- MATLAB. (2017c). Understanding Discrete Event Simulation, Part 3: Leveraging Stochastic Processes.
- MATLAB. (2017d). Understanding Discrete Event Simulation, Part 4: Operations Research.
- MATLAB. (2017e). Understanding Discrete Event Simulation, Part 5: Communication Modeling.

Anexos

A.ANEXO

Artefactos relacionados a la planificación del proyecto

A.1. Esquema de Descomposición de Trabajo

