PROYECTO 2 – Assembler Robot Simulator (ARS)

202003975 – Diego André Mazariegos Barrientos

Resumen

El software denominado “Proyecto 2 ARS” el cual es una aplicación enfocada en apoyar en la tarea de simular el cálculo del tiempo óptimo para el ensamblaje de un determinado producto esto con el fin de apoyar en el área de producción a la empresa Digital Intelligence, S. A.

El software fue desarrollado mediante la implementación de Tipos de Datos Abstractos (TDA’s) para el almacenamiento de los datos y su manipulación, el uso del algoritmo de creación propia para el manejo de las líneas de ensamblaje para determinar el tiempo óptimo, manipulación de archivos con extensión XML para el ingreso y salida de datos, creación de reportes mediante la tecnología Graphviz y HTML. Además, la implementación de una interfaz gráfica mediante el binding de la biblioteca gráfica Tcl/Tk denominado Tkinter.

El proyecto solicitado cumple con todos los requerimientos propuestos por el cliente, se cumplieron y se probaron profundamente todas las opciones solicitadas.

Palabras clave

- TDA, HTML, Tkinter, Graphviz, memoria dinámica

Abstract

The software called "Project 2 ARS" is an application focused on supporting the task of simulating the calculation of the optimal time for the assembly of a certain product to support the Digital Intelligence S.A. company in the production area.

The software was developed through the implementation of Abstract Data Types (ADT's) for data storage and manipulation, the use of the algorithm of own creation for the management of the assembly lines to determine the optimal time, manipulation of files with XML extension for data input and output, report creation using Graphviz and HTML technology. In addition, the implementation of a graphical interface by binding the graphical library Tcl / Tk called Tkinter.

The requested project meets all the requirements proposed by the client, all the requested options were fully met and tested.

Keywords

- *TDA, HTML, Tkinter, Graphviz, dynamic memory.*

Introducción

El presente artículo tiene como finalidad dar un análisis profundo acerca del desarrollo del software "Proyecto 1 ARS".

El software diseñado se basa principalmente en el paradigma de la programación orientada a objetos, esto con el fin de darle una forma organizada al programa, para conservar la simpleza del mismo y si en algún punto se necesitase la reutilización del mismo.

Por medio de la programación orientada a objetos se hizo posible la implementación de los Tipos de Datos Abstractos (TDA's), los cuales fueron de utilidad para el guardado y manipulación de los datos entrantes y salientes del programa por medio de archivos de extensión XML.

Para el procesamiento de los datos se hizo uso de estructuras recursivas y mediante un algoritmo de creación propia se determinó el tiempo óptimo para el ensamblaje de cada producto, para la impresión de reportes se utilizó la tecnología Graphviz y también archivos HTML.

Desarrollo del tema

El contexto del proyecto se basa en la empresa Digital Intelligence, S. A. que ha desarrollado una máquina capaz de ensamblar las partes de cualquier producto. La máquina creada por Digital Intelligence, S.A. puede construir cualquier producto ensamblando automáticamente los componentes (partes) que lo conforman. En base a esto la empresa Digital Intelligence, S. A. se ha visto en la necesidad de solicitar un software el cual simule el funcionamiento de esta máquina con "n" líneas de ensamblaje y cada línea de ensamblaje con "m" posibles componentes a seleccionar de forma que pueda predecir el tiempo "óptimo" para elaborar cualquier producto que pueda ser ensamblado en la máquina. Para un mejor entendimiento del funcionamiento de la máquina y del comportamiento a simular en el software "ARS" ver la Figura A1.

El programa cuenta con un menú en el cual están disponibles 6 opciones como se muestra en la figura A2. Para la carga de archivos se realiza por medio de archivos con extensión XML que cuentan con una estructura previamente definida dependiendo de si se trata de si son datos para la máquina o datos para la simulación como se muestra en la Figura A3 y Figura A4. La lectura e interpretación de dicha estructura se realiza mediante la interfaz de programación de manipulación XML, ElementTree. La biblioteca ElementTree incluye herramientas para analizar XML usando API's basadas en eventos y documentos. Esta misma estructura y herramientas se utilizan para la generación de los archivos de salida. Para el almacenamiento de los datos se utilizan los TDA's y para la generación de los reportes la tecnología Graphviz y HTML. Procedimientos los cuales se detallarán con más profundidad a continuación.

a. Almacenamiento de los datos de entrada.
   Los datos de entrada que serían una determinada cantidad de productos se almacenan en una lista doblemente enlazada que guarda la información individualmente de cada producto en forma de objeto como en la Figura 1.

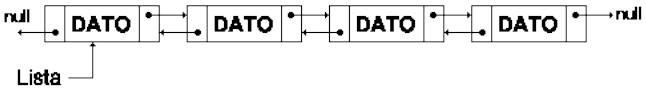

Figura 1. Lista doblemente enlazada.

Fuente: Capítulo 5 Listas doblemente enlazadas, conclase.net, 2021.

b. Almacenamiento de las posiciones de cada simulación/maquina.
   Cada objeto/simulación/maquina en su determinada lista cuenta con una lista

doblemente enlazada que sería la lista de posiciones en cada producto, se dice que esta lista doblemente enlazada es dinámica ya que no se define el tamaño de esta directamente en el programa, sino que su tamaño se acopla a la cantidad de posiciones ingresadas en cada archivo de entrada simulación/maquina en el tiempo de ejecución.

c. Algoritmo de creado.
Algoritmo fundamental en el proyecto que consiste en ir explorando la lista de cola de elaboraciones de cada producto el funcionamiento del dicho algoritmo se basa en recorrer la lista de cola de elaboraciones siempre que falte alguna instrucción (Elaboración), en otras palabras se recorre la lista de instrucciones hasta que todas se cumplan, el algoritmo descrito en palabras lo que hace es que al posicionarse en una determinada instrucción primero verifica si esta no se ha cumplido, si esta ya se hubiese cumplido simplemente crea un objeto acción con el atributo de "no hacer nada" y posteriormente agregar esta acción a una lista de acciones de ese producto, cosa que realizará siempre que continué con la instrucción siguiente, a menos que si esta completada la instrucción pero existan otras instrucciones en la misma línea de producción de esta, en ese caso no genera ninguna acción simplemente ignora dicha instrucción ya completada.

En caso de encontrar una instrucción no completada verificara si no existen otras instrucciones no completadas en la misma línea de producción antes que esta puesto que sí, si existen entonces se ignorará ya que seguramente esa instrucción anterior ya habrá generado la acción necesaria en la línea de producción de ese ciclo, en caso contrario se generará la acción que le sea correspondiente.

d. Archivos de salida.
El programa cuenta con una estructura ya definida para los archivos de salida, así como para los de entrada su extensión es de tipo XML su estructura sería la siguiente ver Figura 4.

```
SalidaSimulacion>
 <Nombre>
 <!--[Texto]--> *Nombre de la simulación que generó esta salida.
 </Nombre>
 <ListadoProductos>
 <Producto>
 <Nombre>
 <!--[Texto]-->
 </Nombre>
 <TiempoTotal><!--[NumeroEntero>0]--></TiempoTotal>
 <ElaboracionOptima>
 <Tiempo NoSegundo="[NumeroEntero]">
 <LineaEnsamblaje NoLinea="[NumeroEntero]">
 <!--[Texto]--> *Representa la acción en la línea (movimiento,
 no hacer nada ó ensamblar)
 </LineaEnsamblaje>
…
 </Tiempo>
 …
 </ElaboracionOptima>
 </Producto>
 …
 </ListadoProductos>
</SalidaSimulacion>
```

Figura 4. Estructura archivo de salida.

Fuente: elaboración propia.

e. Sistema de reportes de la cola de elaboración.
Para la generación de reportes y así poder visualizar de forma más agradable los terrenos se utiliza la tecnología Graphviz y se trabaja con archivos de salida de tipo DOT los cuales son legibles por medio de Graphviz que ofrece una interfaz para interpretar estos documentos en formato PDF dando como resultado los siguientes archivos de salida ver Figura 5.

Figura 5. Reporte de lista de cola de elaboración de cada producto generado mediante la herramienta Graphviz.

Fuente: elaboración propia.

f. Sistema de reportes de lista de acciones.
Para la generación de reportes y así poder visualizar de forma más agradable los terrenos se utiliza HTML y archivos en cascada como lo son CSS específicamente con la tecnología Bootstrap dando como resultado los siguientes archivos de salida ver Figura 6.

Figura 5. Reporte de lista de cola de elaboración de cada producto generado mediante la herramienta Graphviz.

Fuente: elaboración propia.

Para una mejor comprensión del programa y de los temas desarrollados ver Figura A5, donde se expone de una forma más clara como están relacionadas todas las funcionalidades del proyecto.

Conclusiones

El proyecto realizado es óptimo y cumple con las especificaciones requeridas por el cliente por lo que la aplicación fue completada con éxito.

Es posible y viable implementar el lenguaje de Python para dar una solución al planteamiento presentado.

Graphviz es una herramienta útil que permite la generación de reportes de una secuencia o lista de datos.

El Algoritmo de creado en el software es funcional y puede implementarse junto con TDA's y aplicar esto en la memoria dinámica como se demostró en este trabajo para la determinación de un tiempo óptimo en la elaboración de un producto.

Referencias bibliográficas

C. R. Severance, (2016). Python para todos. Recuperado en el año 2021. Disponible en https://uedi.ingenieria.usac.edu.gt/campus/pluginfile.php/145984/mod_resource/content/1/pythonlearn.pdf

M. Yassin, (2021). Python for beginners - Learn all the basics of python. Recuperado en el año 2021. Disponible en https://www.udemy.com/course/python-for-beginners-learn-all-the-basics-of-python/

R. P. Grimaldi, (1994). Matemática discreta y combinatoria: Una introducción con aplicaciones. (3ª ed.), Massachusetts, Addison-Wesley Publishing Company, Inc.

S. Bank, (2013). Graphviz. Recuperado en el año 2021. Disponible en https://graphviz.readthedocs.io/en/stable/index.html

Apéndices

Nombre Producto: SmartWatch
Cola de Prioridad: L1C2 -> L2C1 ->L2C2 ->L1C4
Tiempo de ensamblaje de Línea 1 y 2: 1s

Tiempo	Línea 1	Línea 2
1s	Mover brazo – componente 1	Mover brazo – Componente 1
2s	Mover brazo – componente 2	No hacer nada
3s	Ensamblar componente 2	No hacer nada
4s	No hacer nada	Ensamblar – Componente 1
5s	Mover brazo – Componente 3	Mover brazo – Componente 2
6s	No hacer nada	Ensamblar – Componente 2
7s	Mover brazo – Componente 4	No hacer nada
8s	Ensamblar componente 4	No hacer nada
Tiempo Óptimo de Ensamblaje del Producto "SmartWatch": 8 segundos		

Figura A1. Ejemplo de procedimiento de ensamblado de un producto cualquiera.

Fuente: IPC2 - Aclaraciones Proyecto 2, Universidad de San Carlos de Guatemala, 2021, Pág.1.

Figura A2. Menú principal de la aplicación.

Fuente: elaboración propia.

```
<Maquina>
 <CantidadLineasProduccion>
 <!--[0<NumeroEntero<1000]-->
 </CantidadLineasProduccion>
 <ListadoLineasProduccion>
 <LineaProduccion>
 <Numero>
 <!--[NumeroEntero>0]-->
 </Numero>
 <CantidadComponentes>
 <!--[0<NumeroEntero<1000]--> *Componentes de la línea
 </CantidadComponentes>
 <TiempoEnsamblaje>
 <!--[NumeroEntero>0]--> *Representa el tiempo en segundos
 que toma ensamblar en la línea
 </TiempoEnsamblaje>
 </LineaProduccion>
 …
 </ListadoLineasProduccion>
 <ListadoProductos>
 <Producto>
 <nombre>
 <!--[Texto]--> *Nombre del Producto
 </nombre>
 <elaboracion> *Corresponde a instrucciones para ensamblar el producto
 <!--[Texto con formato L1pC1p L2pC2p L3pC3p … LnpCnp]-->
 </elaboracion>
 </Producto>
 …
 </ListadoProductos>
</Maquina>
```

Figura A3. Estructura de los datos de entrada de Maquina.

Fuente: elaboración propia.

```
<Simulacion>
 <Nombre>
 <!--[Texto]-->
 </Nombre>
 <ListadoProductos>
 <Producto>
 <!--[Texto]--> *Nombre de producto configurado en la máquina
 </Producto>
 <Producto>
 <!--[Texto]2-->
 </Producto>
 …
 </ListadoProductos>
</Simulacion>
```

Figura A4. Estructura de los datos de entrada de Simulación.

Fuente: elaboración propia.

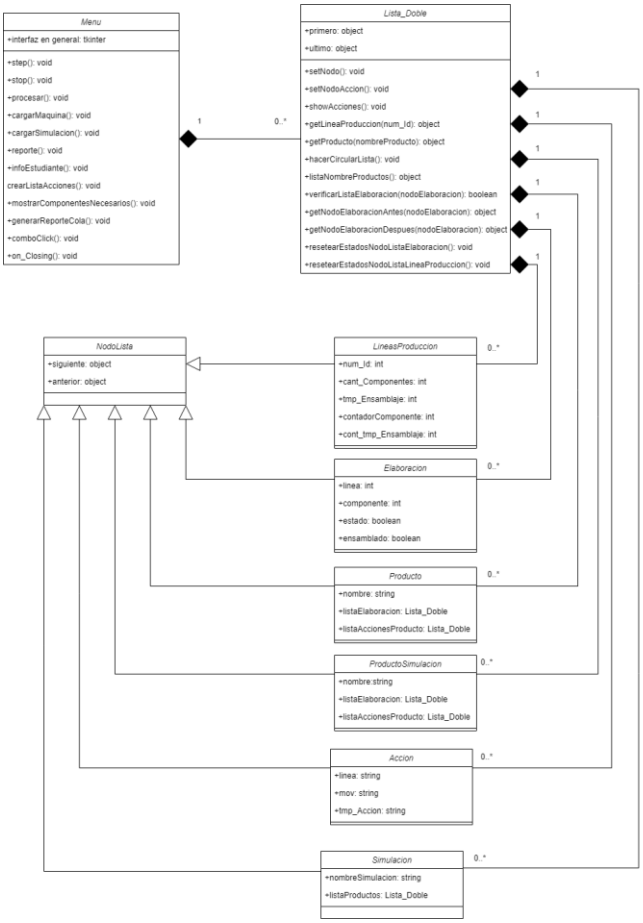

Figura A4. Diagrama de clases del proyecto.

Fuente: elaboración propia.