
INTERFAZ GRAFICA PARA MODELACION DE REDES DE TRANSPORTE

Claudio Briceño F., Ubaldo Taladriz T. y Eduardo Valenzuela F.

EXE Ingeniería & Software Ltda.

Seminario 143 Of. 1002 Fono-Fax 2227581

RESUMEN

En el ámbito de la modelación de redes de transporte, diversas herramientas computacionales han sido desarrolladas para apoyar la labor del especialista, tanto a nivel urbano como interurbano. En Chile, entre las herramientas más utilizadas es posible mencionar a SATURN, TRANSYT, SIDRA, PICADY y OSCADY para redes preferentemente urbanas; MARTED y SERC, para redes interurbanas. Si bien estas herramientas poseen diversas características que las hacen, en algunos casos, ser sustitutivas; todas poseen una insuficiencia común: no tienen una interfaz gráfica que permita construir visualmente la red a modelar y que entregue resultados gráficos de las etapas de calibración o ajuste, en un ambiente que incorpore conceptos de manipulación directa usando computación gráfica interactiva, entendiéndose por esto el manejo de representaciones concretas de objetos a través de acciones inmediatas, crecientes y reversibles (Shneiderman, 1987).

En este contexto, el presente trabajo describe el diseño, desarrollo e implementación de una herramienta computacional que permite construir gráficamente la red de modelación, en términos de nodos, arcos y sus características, y desplegar las variables de servicio que el analista desea conocer de manera gráfica y numérica.

Se concluye que la interfaz desarrollada permite "ver lo que se desea simular" y "ver el estado de la simulación" en términos de red de modelación y de variables de servicio; características que optimizan considerablemente las etapas de calibración y modelación.

Este trabajo debe ser entendido como una primera etapa en la construcción de una plataforma de integración con cualquiera de las actuales herramientas referidas o en un futuro cercano como el principio de un desarrollo planificado de modelos genéricos de fácil localización.



1. INTRODUCCION

1.1. Justificación

En el ámbito de la modelación de redes de transporte, diversas herramientas computacionales han sido desarrolladas para apoyar la labor del especialista, tanto a nivel urbano como interurbano. En Chile, entre las herramientas más utilizadas es posible mencionar a SATURN, TRANSYT, SIDRA, PICADY y OSCADY para redes preferentemente urbanas; MARTED y SERC, para redes interurbanas. Si bien estas herramientas poseen diversas características que las hacen, en algunos casos, ser sustitutivas; todas poseen una insuficiencia común: no tienen una interfaz gráfica que permita construir visualmente la red a modelar y que entregue resultados gráficos de las etapas de calibración, ajuste o simulación.

El presente trabajo de investigación equivale a una primera etapa en la construcción de una plataforma de integración con cualquiera de las actuales herramientas referidas o en un futuro cercano como el principio de un desarrollo planificado de modelos genéricos de fácil localización.

El documento ha sido estructurado en cuatro capítulos de los cuales el presente es el primero. El capítulo N°2 presenta los requerimientos básicos necesarios para modelar redes con TRANSYT. El capítulo N°3 presenta el análisis y diseño de la Interfaz Humano-Computador (IHC, en adelante) denominada INGRATA (INterfaz GRAfica para TrAnsyt). Por último, el capítulo N°4 presenta las principales conclusiones del trabajo y las líneas de desarrollo futuro.

1.2. Conceptos de Interfaces.

Los conceptos de IHC han sido desarrollados desde la década de los 80, masificándose a través de las interfaces de los sistemas Macintosh de Apple. Por razones comerciales, otras arquitecturas y sistemas operativos hoy en día tienen una mayor base instalada. Claramente hablamos de los procesadores Intel y del sistema operativo DOS. Sobre este último, se comenzaron a desarrollar interfaces al estilo Macintosh, la versión estable y que hoy es el standard de facto en nuestros computadores personales es Windows 3.1.

Durante el tiempo que escribimos este documento, Windows 95, la esperada siguiente versión de Windows, no ha sido liberada y su ambicioso conjunto de características han hecho demorar su llegada a nuestros computadores. Probablemente en el momento de la exposición de este trabajo, Windows 95 ya estará disponible. El sistema operativo Windows 95 proporciona un significativo aumento con respecto a la tradicional interfaz de Windows. Las aplicaciones ahora serán 32 bits y sus interfaces podrán incorporar numerosos elementos soportados por Windows 95 como manipulación directa, orientación a objetos y Object Linking and Embedding, Versión 2.0 (OLE 2.0), entre otros.

Las aplicaciones de software entregan una interfaz al usuario que es función del sistema operativo que las soporta. En este sentido, las aplicaciones de la especialidad de transporte han sido deficientes



en los aspectos IHC, principalmente debido a las limitaciones del propio sistema operativo en que operan.

En consecuencia, este trabajo de investigación tiene como objetivo principal incorporar todo el conocimiento actual de las IHC al área de la Ingeniería de Transporte. En este contexto, se intenta desarrollar una IHC para ingenieros que construyen y modelan Sistemas de Transporte, usando una herramienta que nos permite fundir las principales tendencias de software, (como orientación a objetos y esquemas Cliente/Servidor) en una aplicación que funcionará en Windows y Windows 95, explotando al máximo las facilidades de cada entorno.

Por razones de espacio, el escenario de este documento está basado en TRANSYT Versión N°8.0, una de las muchas herramientas que hoy en día ocupamos y que carecen de una IHC adecuada. La interfaz diseñada considera diversos tópicos que potencian considerablemente el producto final, como por ejemplo: Manipulación Directa, What you see is what you Get, Tiempo de Respuesta, Monitoreabilidad y Consistencia (Briceño, 1994).

A continuación se describen algunos de estos conceptos:

Manipulación Directa. Una de las expresiones de manipulación directa es la capacidad de Drag&Drop, es decir, la posibilidad de manipular directamente objetos arrastrándolos desde una posición a otra. Este es un punto clave de la interfaz diseñada. El resultado de una operación de Drag&Drop depende del destino del objeto. Por ejemplo si un nodo es arrastrado a una nueva posición claramente es porque se desea moverlo a esa ubicación y por lo tanto esas serán sus nuevas coordenadas. Lo que acabamos de describir se llama Default Drag&Drop. Sin embargo, arrastar un objeto hacia una nueva posición puede tener varias interpretaciones. Por ejemplo, al arrastar un nodo, quizás sólo se desea trasladar propiedades y no el objeto, o quizás se desea copiar el objeto, en vez de moverlo. Estas operaciones llamadas non-default, son provistas al usuario si arrastra el objeto con el botón derecho del mouse, a través de un menú pop-up que aparece cuando el usuario suelta el objeto. Entonces el usuario puede seleccionar que operación desea aplicar. Nótese que siempre el conjunto de operaciones disponibles estará determinado por la nueva posición o destino del objeto.

Orientación a Objetos. La definición más simple de orientación a objetos es: agrupar las propiedades de algo y las operaciones que se pueden hacer sobre ese algo en una unidad que llamaremos objeto. La idea es ser capaz de definir jerarquías de objetos, de manera de lograr herencias de propiedades y operaciones. Otro punto fundamental es la capacidad de definir operaciones (que llevan el mismo nombre) aplicables a toda la jerarquía de objetos y que cada objeto implementa a su manera. Hemos descrito Encapsulación, Herencia y Polimorfismo, respectivamente.

En la interfaz todos los objetos responden al botón derecho del mouse, incluso si no son arrastrados hacia otro objeto o posición, con un menú pop-up que despliega las operaciones disponibles. Estos menús frecuentemente son llamados contextuales, ya que muestran sólo las operaciones aplicables al objeto en su contexto.

En la interfaz la operación más común alcanzable con el botón derecho es la inspección de propiedades. Las propiedades son mostradas en una ventana secundaria que lista nombres y valores



1. INTRODUCCION

1.1. Justificación

En el ámbito de la modelación de redes de transporte, diversas herramientas computacionales han sido desarrolladas para apoyar la labor del especialista, tanto a nivel urbano como interurbano. En Chile, entre las herramientas más utilizadas es posible mencionar a SATURN, TRANSYT, SIDRA, PICADY y OSCADY para redes preferentemente urbanas; MARTED y SERC, para redes interurbanas. Si bien estas herramientas poseen diversas características que las hacen, en algunos casos, ser sustitutivas; todas poseen una insuficiencia común: no tienen una interfaz gráfica que permita construir visualmente la red a modelar y que entregue resultados gráficos de las etapas de calibración, ajuste o simulación.

El presente trabajo de investigación equivale a una primera etapa en la construcción de una plataforma de integración con cualquiera de las actuales herramientas referidas o en un futuro cercano como el principio de un desarrollo planificado de modelos genéricos de fácil localización.

El documento ha sido estructurado en cuatro capítulos de los cuales el presente es el primero. El capítulo N°2 presenta los requerimientos básicos necesarios para modelar redes con TRANSYT. El capítulo N°3 presenta el análisis y diseño de la Interfaz Humano-Computador (IHC, en adelante) denominada INGRATA (INterfaz GRAfica para TrAnsyt). Por último, el capítulo N°4 presenta las principales conclusiones del trabajo y las líneas de desarrollo futuro.

1.2. Conceptos de Interfaces.

Los conceptos de IHC han sido desarrollados desde la década de los 80, masificándose a través de las interfaces de los sistemas Macintosh de Apple. Por razones comerciales, otras arquitecturas y sistemas operativos hoy en día tienen una mayor base instalada. Claramente hablamos de los procesadores Intel y del sistema operativo DOS. Sobre este último, se comenzaron a desarrollar interfaces al estilo Macintosh, la versión estable y que hoy es el standard de facto en nuestros computadores personales es Windows 3.1.

Durante el tiempo que escribimos este documento, Windows 95, la esperada siguiente versión de Windows, no ha sido liberada y su ambicioso conjunto de características han hecho demorar su llegada a nuestros computadores. Probablemente en el momento de la exposición de este trabajo, Windows 95 ya estará disponible. El sistema operativo Windows 95 proporciona un significativo aumento con respecto a la tradicional interfaz de Windows. Las aplicaciones ahora serán 32 bits y sus interfaces podrán incorporar numerosos elementos soportados por Windows 95 como manipulación directa, orientación a objetos y Object Linking and Embedding, Versión 2.0 (OLE 2.0), entre otros.

Las aplicaciones de software entregan una interfaz al usuario que es función del sistema operativo que las soporta. En este sentido, las aplicaciones de la especialidad de transporte han sido deficientes



en los aspectos IHC, principalmente debido a las limitaciones del propio sistema operativo en que operan.

En consecuencia, este trabajo de investigación tiene como objetivo principal incorporar todo el conocimiento actual de las IHC al área de la Ingeniería de Transporte. En este contexto, se intenta desarrollar una IHC para ingenieros que construyen y modelan Sistemas de Transporte, usando una herramienta que nos permite fundir las principales tendencias de software, (como orientación a objetos y esquemas Cliente/Servidor) en una aplicación que funcionará en Windows y Windows 95, explotando al máximo las facilidades de cada entorno.

Por razones de espacio, el escenario de este documento está basado en TRANSYT Versión Nº8.0, una de las muchas herramientas que hoy en día ocupamos y que carecen de una IHC adecuada. La interfaz diseñada considera diversos tópicos que potencian considerablemente el producto final, como por ejemplo: Manipulación Directa, What you see is what you Get, Tiempo de Respuesta, Monitoreabilidad y Consistencia (Briceño, 1994).

A continuación se describen algunos de estos conceptos:

Manipulación Directa. Una de las expresiones de manipulación directa es la capacidad de Drag&Drop, es decir, la posibilidad de manipular directamente objetos arrastrándolos desde una posición a otra. Este es un punto clave de la interfaz diseñada. El resultado de una operación de Drag&Drop depende del destino del objeto. Por ejemplo si un nodo es arrastrado a una nueva posición claramente es porque se desea moverlo a esa ubicación y por lo tanto esas serán sus nuevas coordenadas. Lo que acabamos de describir se llama Default Drag&Drop. Sin embargo, arrastar un objeto hacia una nueva posición puede tener varias interpretaciones. Por ejemplo, al arrastar un nodo, quizás sólo se desea trasladar propiedades y no el objeto, o quizás se desea copiar el objeto, en vez de moverlo. Estas operaciones llamadas non-default, son provistas al usuario si arrastra el objeto con el botón derecho del mouse, a través de un menú pop-up que aparece cuando el usuario suelta el objeto. Entonces el usuario puede seleccionar que operación desea aplicar. Nótese que siempre el conjunto de operaciones disponibles estará determinado por la nueva posición o destino del objeto.

Orientación a Objetos. La definición más simple de orientación a objetos es: agrupar las propiedades de algo y las operaciones que se pueden hacer sobre ese algo en una unidad que llamaremos objeto. La idea es ser capaz de definir jerarquías de objetos, de manera de lograr herencias de propiedades y operaciones. Otro punto fundamental es la capacidad de definir operaciones (que llevan el mismo nombre) aplicables a toda la jerarquía de objetos y que cada objeto implementa a su manera. Hemos descrito Encapsulación, Herencia y Polimorfismo, respectivamente.

En la interfaz todos los objetos responden al botón derecho del mouse, incluso si no son arrastrados hacia otro objeto o posición, con un menú pop-up que despliega las operaciones disponibles. Estos menus frecuentemente son llamados contextuales, ya que muestran sólo las operaciones aplicables al objeto en su contexto.

En la interfaz la operación más común alcanzable con el botón derecho es la inspección de propiedades. Las propiedades son mostradas en una ventana secundaria que lista nombres y valores



actuales de propiedades relativas a la red de modelación, como por ejemplo tiempo de ciclo y repartos para los nodos y longitud y velocidad para los arcos.

Pop-Ups accesados mediante el botón derecho del mouse entregan un acceso rápido a los comandos contextuales. En otras palabras, nuestra interfaz está orientada al dato, es decir la información que está presente en la pantalla, a través de representaciones gráficas. Además, la interfaz es astuta, en el sentido de responder que operaciones son posibles, es decir; ella misma entrega la correcta funcionalidad para ser manipulado a medida que el usuario interactúa. El usuario no tendrá que decir: ¿cómo puedo saber que es posible de hacer con esto?.

1.3. Metodología.

La metodología adoptada para diseñar e implementar la interfaz aquí referida, consideró las siguientes etapas o actividades: Recopilación de Antecedentes, Estudio del Marco Conceptual, Análisis Global del Proyecto, Análisis de la Interfaz, Diseño de la Interfaz, Modelo Conceptual del Usuario, Prototipos, Evaluación de la Interfaz, Discusión y Conclusiones. Por razones de espacio, en lo que sigue del documento sólo se describen las principales etapas y sus interrelaciones.

La primera tarea fue realizar un repaso de los conceptos involucrados en el tema, como: modelos existentes y las interfaces de sus implementaciones. Hecho lo anterior, se realizó el análisis de TRANSIT, con lo que se definieron en su totalidad los requerimientos de la interfaz. Los objetivos se lograron haciendo un estudio simultáneo del usuario y de su problema. De los varios tipos de usuarios potenciales a considerar, se dio más importancia al usuario simulador de sistemas de transporte, y no a otras categorías como la de usuarios investigadores. Básicamente, el problema del usuario estudiado era dibujar la red de transporte deseada, asociar antecedentes específicos a los objetos dibujados, generar los archivos de entrada para TRANSYT a partir de dichos antecedentes, correr TRANSYT y evaluar gráficamente los resultados de la simulación. El resultado del análisis es el Modelo Conceptual del Usuario, es decir, el conjunto de objetos, acciones, y control del usuario sobre dichos objetos y acciones en la solución del problema.

Las consideraciones fundamentales en el diseño de interfaces, que se tomaron en cuenta como parte de la metodología de trabajo en descripción, y que se encuentran en la mayoría de la literatura (Foley, 1990; Shneiderman, 1987), fueron la Manipulación Directa; Orientación a Objetos; Aspectos Compositivos; Presentación y Distribución de la Información en la Pantalla; Aspectos Tipográficos; Estilo de Texto; Tipos de Líneas; Uso del Color; Armonía; Contrastes; y Balance.



2. MODELO DE SIMULACION TRANSYT.

2.1. Conceptos Básicos

Los principios del modelo de simulación TRANSYT han sido descritos en Robertson (1969). Posteriormente, la experiencia de los usuarios de la herramienta computacional ha sugerido diversas mejoras las cuales han sido incorporadas en versiones posteriores del programa original, llegando en la actualidad a la novena versión. Sin embargo, la versión considerada en el presente trabajo es la octava. El software permite simular y optimizar una red vial urbana vía modificación de la programación de los semáforos de dicha red. Es decir, permite programar semáforos de tal forma que éstos se encuentren coordinados y minimicen el consumo de recursos. En este aspecto, TRANSYT minimiza una función denominada INDICE de RENDIMIENTO (Performance Index) o IR, el cual es una combinación lineal del tipo:

$$IR = \sum_{i=1}^N (W \cdot w_i d_i + \frac{K}{100} \cdot k_i s_i) \quad (1)$$

donde:

- N : es el número de arcos de la red;
- W : Costo de las Demoras, en \$/ADE-Hrs;
- K : Costo de cada 100 Detenciones, en \$/ADE-Hrs;
- w_i : peso de la demora en arco i-ésimo;
- d_i : demora en arco i-ésimo;
- k_i : peso de las detenciones en arco i-ésimo;
- s_i : número de detenciones en arco i-ésimo.

TRANSYT considera los siguientes supuestos acerca de la situación del tráfico:

Las principales intersecciones de la red a simular están semaforizadas;
 Todas las intersecciones semaforizadas poseen el mismo ciclo de semáforo o la mitad de éste;
 Cada flujo vehicular de cada movimiento simulado durante cierto período de tiempo, se supone constante (es decir; no existe reasignación de flujo).

2.2. Representación de la Red Vial

La red vial es representada por nodos interconectados por arcos. Cada intersección señalizada (semáforo, disco pare, ceda el paso) es representada por un nodo; los arcos pueden representar movimientos específicos (que requieren ser modelados por separado) o tipos de vehículos (normalmente transporte público y transporte privado).

Los objetos que normalmente se utilizan para definir los tipos de arcos y nodos se bosquejan a continuación.



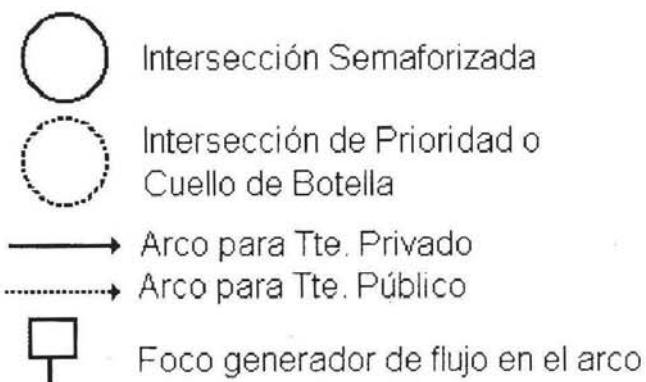


Figura N°1: Definición de Objetos Gráficos.

Con esta definición de objetos es posible construir redes de transporte, cuyas principales características son:

- Redes planas y poco densas, es decir el número de arcos es mucho menor que el cuadrado del número de nodos (TRANSYT acepta máximo 50 nodos).
- Normalmente el número de arcos está relacionado con el número de nodos mediante un factor (alrededor de cuatro en redes sin transporte público, al rededor de 8 en redes con transporte público).
- Son redes conexas, es decir para cada par de nodos cualquiera de la red, siempre existe un camino o ruta posible para ir de uno a otro nodo.

2.3. Formato del Archivo de Datos.

Los datos son abastecidos al programa mediante un archivo único de entrada, en formato ASCII. Cada línea del archivo de entrada se denomina tarjeta, existen diversos tipos de tarjetas, cada una es definida mediante un número. La idea es abastecer al programa con antecedentes acerca del Tiempo de ciclo; control del proceso de optimización; flujo horario y características de cada arco de la red y repartos de cada nodo semaforizado.

En todos los casos, los datos son ingresados en las tarjetas como números enteros, sin decimales fracciones o caracteres alfanuméricos. Los números son justificados a la derecha en columnas de 5 espacios (bloques). Si cierto bloque no posee dato, el programa asume cero. El número de cada tarjeta es ingresado siempre en el primer bloque. El bloque máximo es el 16 (columna 80).

Las tarjetas deben ser ingresadas siempre en orden ascendente, la única excepción es con las tarjetas 11 a 23 las que pueden ser ingresadas en cualquier orden (sólo entre ellas mismas).



Los arcos pueden ser representados por números enteros entre 1 y 9999. Excepcionalmente pueden ser usados números negativos (-1 a -9999). Todo arco representado con un número negativo es tratado de la forma estándar, sin embargo; sus datos de demora y detenciones son omitidos en el cálculo y presentación del IR. Se recomienda que los arcos sean numerados de tal forma que los primeros dos dígitos equivalgan al nodo donde llegan y que los siguientes dígitos sean un contador según el sentido del reloj, de 1 a 9 para arcos de transporte privado y de 10 a 99 para arcos de transporte público. Ejemplo: arco 102 y arco 1020. Ambos arcos llegan al nodo 10. El primero (102) es el segundo arco de transporte privado que llega al nodo 10. En cambio, el arco 1020 es el segundo arco de transporte público que llega al nodo 10. Mayores detalles pueden ser encontrados en Robertson et. al. (1980).

3. DISEÑO DE INGRATA

3.1. Estudio y Definición del Usuario

El análisis y diseño de la interfaz tuvo como objetivo lograr la máxima flexibilidad en la manipulación directa del ambiente de desarrollo, usando computación gráfica interactiva. La manipulación directa consiste en el manejo de representaciones concretas de objetos, a través de acciones inmediatas, crecientes y reversibles (Shneiderman, 1987). Esta comunicación Humano-Computador se logra mediante las representaciones, estáticas y dinámicas, de la información que la computación gráfica proporciona.

Una Interfaz pensada para Windows95 debe tener un diseño centrado en los datos que se manipularán. En este sentido, los objetos son una metáfora natural ya que nos premiten acercarnos a nuestras tareas del mundo real. Sin embargo, un diseño orientado a objetos no garantiza una buena Interfaz, sólo entrega el potencial. Un buen diseño debe también preocuparse del usuario y de sus tareas, a través de prototipos y pruebas de uso (Trower, 1995).

La tarea de análisis y diseño se desarrolló mediante un estudio simultáneo del usuario y su problema, de manera de definir el modelo conceptual del usuario, es decir, definir los objetos, acciones y control que sobre éstos tiene el usuario durante su interacción.

La primera etapa del estudio de los usuarios fue la identificación de los objetos, de acuerdo con la figura N°1 y la nomenclatura utilizada por TRANSYT. En segundo lugar, se identificaron los requerimientos de la Interfaz a partir de los requerimientos de TRANSYT. Al respecto, los principales requerimientos son el poder dibujar una red de transporte de acuerdo a los objetos ya definidos, con todas las potencialidades que ofrecería dibujar la red a mano alzada sobre un papel (borrar, mover, cortar, etc); asociar características gráficas (tamaño, color,etc.) y físicas (velocidad, flujos, etc.) a dichos objetos, y a partir de ésta red, generar el archivo de entrada para TRANSYT, correr TRANSYT, analizar los resultados de la modelación ya sea observando el archivo ASCII tradicional o los antecedentes de nivel gráfico, modificar características específicas de algún objeto, modificar la red (generando alternativas) y volver a correr TRANSYT, evaluando cada simulación. Dicho estudio, se realizó teniendo en cuenta los principales objetivos de diseño de una interfaz. Estos son, proporcionar retroalimentación y consistencia, minimizar la memorización, entregar el control al usuario y manejar errores permitiendo acciones reversibles y evitando conducción hacia ellos.



(Foley, 1984; Shneiderman, 1987). Por último, se consideró que el usuario posee conocimientos básicos del ambiente Windows y conocimientos avanzados del modelo TRANSYT, sólo en cuanto a sus requerimientos para poder ejecutarlo.

3.2. Definición de Objetos y Acciones

Como se mencionó, el problema del usuario es la construcción de redes de transporte y la aplicación del modelo TRANSYT sobre la red construida. La solución debe ser un ambiente que permita manipular los elementos de una red de transporte. Por manipular se debe entender la capacidad de relacionarlos, asignarles propiedades y valores a esas propiedades usando el mouse. Este ambiente debe incorporar un modo de consulta y/o ejecución que permita la aplicación de modelos sobre la red, de sus correspondientes procesos de ajuste (con la posibilidad de modificar elementos "on the fly" de manera visual). Por último, la solución debe ser capaz de mostrar todas las componentes que influyen en el proceso al mismo tiempo.

De acuerdo a esta definición, se identificaron los subproblemas siguientes:

- Construcción de la Red
- Interacción con TRANSYT
 - Aplicación del Modelo
 - Procesos de Ajuste
 - Obtención de Resultados
 - Presentación de Resultados

A continuación se describen los objetos y operaciones sobre los objetos involucrados en la problemática a resolver.

Objeto : Proyecto

Crear	:	Crear un proyecto nuevo con nombre Nombre_Proyecto
Abrir	:	Abrir un proyecto existente llamado Nombre_Proyecto
Grabar	:	Grabar el proyecto activo
Grabar como	:	Grabar el proyecto activo como Nombre_Proyecto_1

Objeto : Componentes de un proyecto

Alternativa	:	Define la red de la alternativa del proyecto activo
Nuevo	:	Abre ventana nueva para confeccionar red nueva para la alternativa activa
Abrir	:	Carga y despliega red ya existente del proyecto activo de la alternativa activa
Inspeccionar	:	Inspecciona características gráficas y/o físicas del objeto seleccionado o de la red del proyecto activo de la alternativa activa
Eliminar	:	Elimina la red de la alternativa activa del proyecto activo



Aplicar	:	Ejecuta TRANSYT para la red de la alternativa activa del proyecto activo
Vista	:	Despliega resultado de la aplicación de TRANSYT para red de la alternativa activa del proyecto activo

La acción *Nuevo* del objeto *Alternativa* brinda la posibilidad de dibujar, sobre una ventana que semeja un papel en blanco, círculos y flechas de un sólo sentido. Dado que se está confeccionando una red nueva, los atributos gráficos consideran el color rojo y línea continua delgada para ambos objetos. De ésta forma, al finalizar la tarea de dibujar círculos y flechas (dibujo definido como *Red Elemental*), el usuario posee una red en color rojo y línea continua delgada. Esta representación indica que la red topológica existe pero no posee los datos necesarios para modelar con TRANSYT. A medida que se ingresan datos específicos para modelar TRANSYT, la red elemental cambia su fisonomía. Al introducir datos a los objetos, éstos cambian de color (del rojo al azul) y de grosor de línea (de simple a doble continua o doble segmentada según corresponda a nodo de prioridad o arco de transporte público, respectivamente). Con esto, cuando la red elemental es completamente azul, estamos en presencia de una red vial para modelar con TRANSYT. Entonces, si la red posee objetos de color rojo y objetos de color azul, significa que se encuentra en un paso intermedio de confección.

La acción *Inspeccionar* del objeto *Alternativa* considera tres opciones. La primera, en el caso de no haber seleccionado ningún objeto en particular de la red, se despliegan los antecedentes que son comunes a toda la red vial, de acuerdo al formato explicitado en Figura N°2. La segunda opción, si se ha seleccionado un círculo (o nodo), se despliegan los antecedentes de acuerdo al formato explicitado en figura N°3. Por último, si se ha seleccionado una flecha (o arco), se despliegan los antecedentes según el formato explicitado en figura N°4.

INGRATA sólo admite como objetos a los nodos (círculos) y los arcos (flechas). En el caso de las líneas de parada, éstas se dibujan automáticamente cuando el usuario especifica si el arco posee línea de parada compartida. De igual forma, en el caso de existir un foco generador de flujo (edificio de estacionamiento, por ejemplo), el dibujo aparece automáticamente para el arco involucrado.

Las operaciones comunes sobre los objetos descritos son:

Agregar	:	Agregar elementos a una base en construcción o ya definida
Inspeccionar	:	Editar las propiedades del objeto seleccionado
Eliminar	:	Eliminar el elemento
Mover	:	Mover el elemento
Deshacer	:	Retroceder a estados anteriores, deshaciendo operaciones previas.

Respecto de la acción *Vista* del objeto *Alternativa*, mediante esta opción es posible editar el archivo de salida en formato ASCII de TRANSYT o desplegar gráficamente para cada arco, los antecedentes relativos a grado de saturación, flujo en el arco, demoras y detenciones.

De esta forma, la interacción con TRANSYT en lo referente al subproblema Procesos de Ajuste, se efectúa mediante iteraciones entre las acciones *Inspeccionar*, *Aplicar* y *Vista*.



3.3. Interacción con TRANSYT

La interacción entre aplicaciones Windows se logra a través de conversaciones DDE (Dynamic Data Exchange) donde las aplicaciones entran en diálogo para intercambio de datos o mediante OLE, mecanismo que permite compartir objetos entre aplicaciones. El hecho de compartir objetos significa que un dato va acompañado de las operaciones que se pueden aplicar sobre él.

Dado que TRANSYT es una aplicación DOS, ninguna de estas alternativas de alto nivel es posible. En esta primera versión, INGRATA genera los archivos de entrada necesarios para TRANSYT, corre el modelo y luego recoge los resultados para presentarlos de manera adecuada.

En términos de la interfaz, un usuario podrá construir la red de transporte y al mismo tiempo viendo dinámicamente el archivo de entrada para INGRATA. El mayor logro es proporcionar un acceso transparente y consistente por ambos caminos, es decir, el usuario puede modificar la red visualmente o modificar el archivo ASCII y los cambios serán refrescados automáticamente en ambas vistas. Esta metodología llamada Two ways Tools, se encuentra en diversas aplicaciones hoy en día.

3.4. Implementación

Muchas de las herramientas disponibles hoy en día permiten a diseñadores no programadores, generar gran parte del código de la interfaz de usuario. Estas herramientas, han llevado a toda la industria de software a preferir la metodología de diseño basada en prototipos como su método de desarrollo (Rosenberg, 1993), existiendo tres variedades de prototipos posibles dentro del proceso de desarrollo de software (Weiser, 1982):

- Prototipos para evaluar desempeño de computadores (memoria, velocidad)
- Prototipos para evaluar funcionalidad (algoritmos, estructuras de datos)
- Prototipos para evaluar interacción Humano-Computador.

En este trabajo, se usaron prototipos para evaluar interacción Humano-Computador, con el objetivo de refinar y validar el diseño de la interfaz de usuario. Los prototipos fueron adoptados como una herramienta de comunicación, es decir, sólo subconjuntos de la funcionalidad de la aplicación fueron implementados. Prototipos cuyo objetivo es probar y refinar un concepto en particular, requieren en rigor de una implementación mucho más completa de la funcionalidad en prueba. La decisión de usar el primer enfoque se debe a que de lo contrario, se pierde la idea de prototipos rápidos, ya que la implementación de funcionalidades completas toma mucho tiempo y porque en la mayoría de los casos es innecesario probar el sistema completo, si sólo se desea revisar una parte. Sin embargo, a medida que el prototipo adquirió más funcionalidad y refinamiento en cada iteración, se convirtió en la aplicación final.



4. DISCUSIÓN Y CONCLUSIONES

Los ambientes gráficos son el presente y el futuro en cuanto a desarrollo de aplicaciones, vasta observar el explosivo crecimiento del multimedia. Debido a esto, cada vez se observa que con mayor frecuencia se desarrollan y perfeccionan herramientas para construcción de aplicaciones en esos ambientes, cuya principal característica es ser Multi-Plataforma y dar todo el soporte para el desarrollo de aplicaciones escalables a Cliente/Servidor o directamente Cliente/Servidor.

Por otra parte, estamos viendo como el clásico pegado de trozos de código, se está convirtiendo en manipulación de objetos ya construidos (componentes de interfaz más funcionalidad) que se pegan sobre la aplicación en desarrollo haciendo drag&drop.

Con respecto a los objetivos de este trabajo, en términos de Interfaz Humano-Computador, se acotó la implementación de las componentes finales de toda interfaz, éstas son, la ayuda en línea y la documentación. Las futuras direcciones para Interfaces Humano-Computador, no son un misterio. Actualmente se han incorporado los conceptos de tutores y expertos, estos ayudantes interactivos, pueden enseñar al usuario como hacer el trabajo (incluso con sus propios datos) o directamente hacer el trabajo por él. El siguiente paso es llenar de multimedia cada característica del software. Lo más ambicioso, será finalmente, insertar realidad virtual a las aplicaciones.

El principal aporte de este trabajo es entregar una herramienta que nos facilitará enormemente una de las tareas más "tediosas", aún cuando es la más importante, dentro de la modelación de redes de transportes, esto es; su construcción. En resumen, en este trabajo se implementó una interfaz para una aplicación de ingeniería, como primera etapa de un proyecto genérico de construcción de redes, en el ambiente gráfico standard de computadores personales, con las características que se encuentran en la mayoría de las aplicaciones comerciales existentes hoy en día en Windows 3.1 y con un diseño pensado para Windows 95.

Como línea de investigación y desarrollo, los autores del presente trabajo trabajan en la implementación de una interfaz gráfica que permita construir y modelar una red de transporte bajo diversas aplicaciones, de acuerdo a las definiciones de objetos gráficos que cada aplicación posea y abierta para ser conectada a cualquier modelo que proporcione comunicación, proyecto que lleva por nombre LIBERTAD.

REFERENCIAS

- Briceño, C. (1992) Diseño e Implementación de una Interfaz Gráfica para un Shell de Sistemas Expertos, **Memoria de Título**, Departamento de Computación, Universidad de Chile. No publicada.
- Foley, J.D. y van Dam, A. (1984) **Fundamentals of Interactive Computer Graphics**. Addison-Wesley, Nueva York.
- Foley, J.D., van Dam,A., Feiner, S.K. y Hughes, J.F. (1990) **Computer Graphics**. Addison-Wesley, Nueva York.



Robertson, D.I. (1969). TRANSYT: a traffic networks study tool. **Ministry of Transport, RRL, Report LR 253**. Crowthorne, 1969 (Road Research Laboratory).

Robertson, D.I., Lucas, C.F. y Baker, R.T. (1980). Coordinating traffic signals to reduce fuel consumption. **Department of the Environment Department of Transport, TTRL, Report LR 934**, Crowthorne, 1980 (Transport and Road Research Laboratory).

Rosenberg, D. (1993) User Interface Prototyping for GUI Products. **Borland International Conference**, San Diego, Mayo 1993

Shneiderman, B. (1987) **Designing The User Interface**. Addison-Wesley, Nueva York.

Trower, T. (1995) Designing for Windows95. **Borland International Conference**, San Diego, Agosto 1995.

Weiser, M. (1982) Scale Modes and Rapid Prototyping. **ACM SIGSOFT Software Engineering Notes**

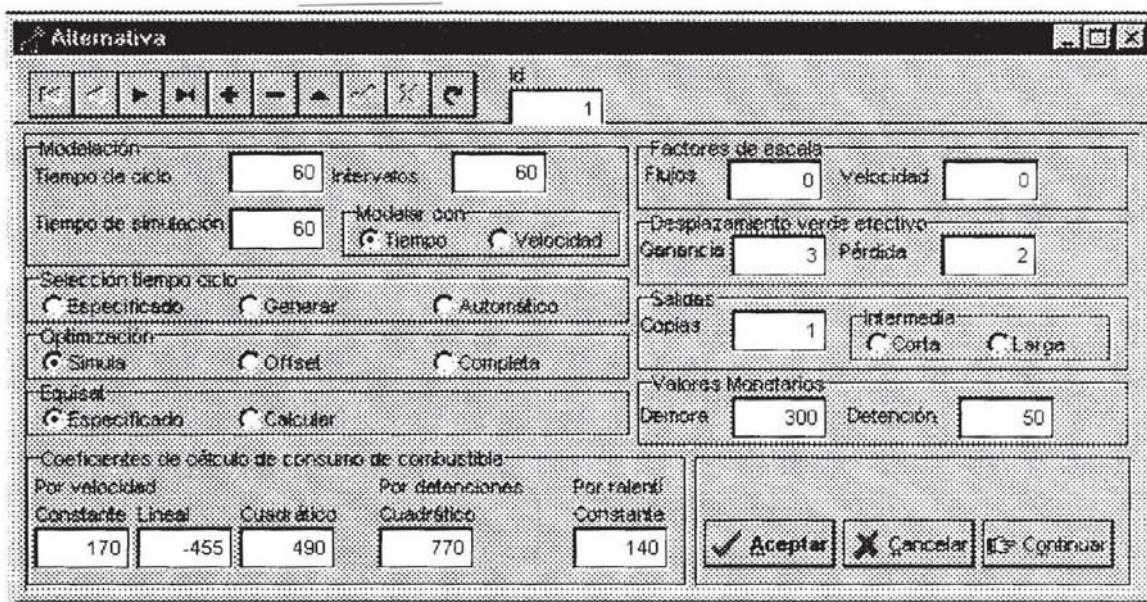


Figura N°2: Antecedentes Red

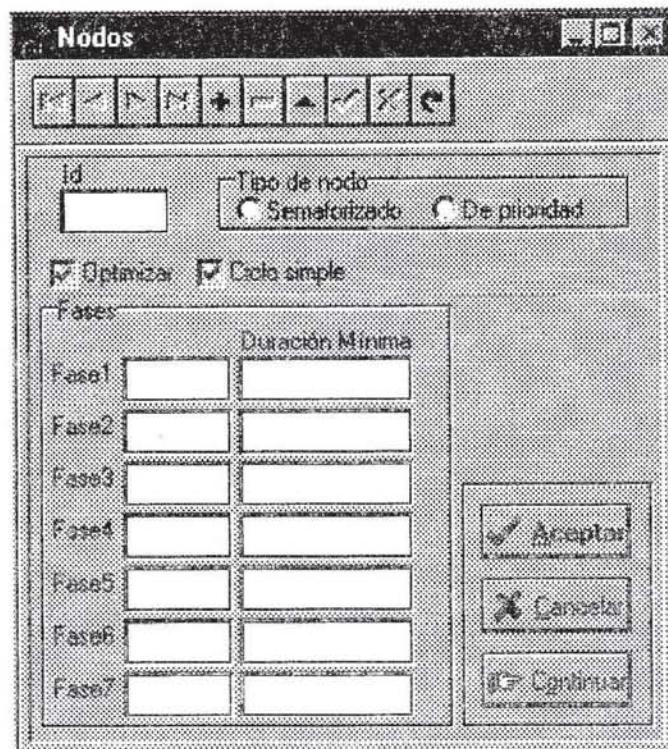


Figura N°3: Antecedentes Nodos

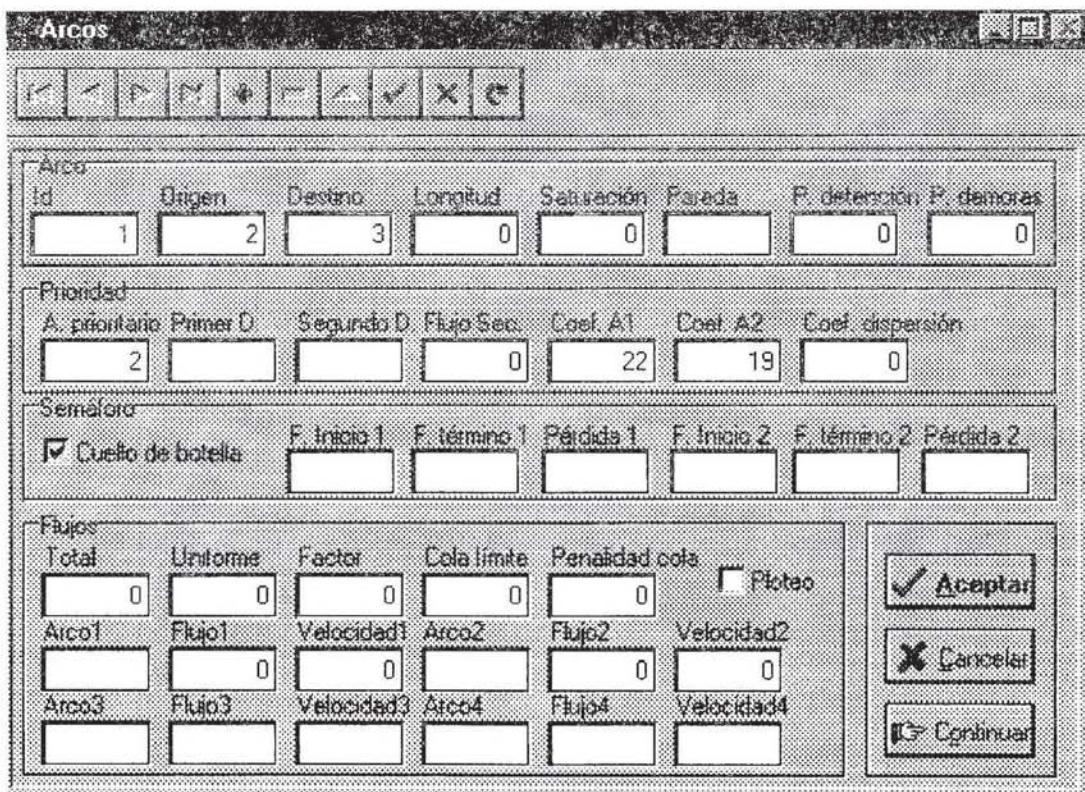


Figura N°4: Antecedentes Arcos