

RELSOFT S.A.

SOFTWARE PARA INGENIERIA

Corrientes 1455 - Piso 3° - Of. 13 - C1042AAA Buenos Aires - ARGENTINA - Telefax (5411) 4375 0169

Internet: www.e-plant.com

Buenos Aires 31.05.2000

RS11600

Ref: Evaluación del sistema PDS Piping de Intergraph

Tipo: Documento Interno

1. OBJETO

Este documento tiene como objetivo hacer una evaluación del módulo de Piping del sistema PDS versión 6.3.1 de Intergraph, aprovechando el trabajo de conversión del modelo de la Estación Polo Arara /Petrobras desde EPLANT (ver RS11400) y el trabajo de asesoría para el Proyecto Hawiyah/Aramco (ver RS14400).

A continuación se analiza la Arquitectura y Funcionalidad de PDS y se destacan características positivas y negativas de su funcionamiento. No es una evaluación detallada de todos sus módulos, sólomente de los puntos más significativos del módulo de piping.

2. CARACTERISTICAS DEL SISTEMA

2.1 DESCRIPCION

PDS de Intergraph Corporation es un sistema para el diseño electrónico de plantas de proceso. Es un sistema reconocido y utilizado mundialmente desde el año 1982. Tiene el 75 % del mercado mundial a nivel de facturación (datos Daratech 1997).

El sistema utiliza una interfase gráfica basada en Microstation y guarda datos de referencia en una base de datos propietaria. Parte de los datos relativos a los materiales utilizados en el proyecto son guardados en una base de datos relacional.

2.2 ARQUITECTURA DEL SISTEMA

Es la misma que cuando se desarrolló el sistema en el año 1982: un sistema gráfico y una base de datos asociada, vinculando los datos por la aplicación.

Los únicos cambios no son significativos: el sistema gráfico pasó del IGDS (Interactive Graphic Design System) a Microstation (que funcionan practicamente igual) y el sistema de base de datos DMRS propietario de Intergraph que pasó a ser una mezcla entre un sistema de base de datos propietario y un sistema relacional.

Es una arquitectura antigua donde la consistencia de los datos es asegurada únicamente por el correcto funcionamiento de la aplicación. Hay muchos escenarios operativos que fácilmente generan la corrupción de los datos.

2.3 INTERFASE GRAFICA

Si bien en los años 80 el sistema Microstation era sin duda el mejor sistema CAD disponible, la falta de desarrollo en los últimos diez años ha revertido esta situación.

No aprovecha la memoria de la máquina (todo lo escribe continuamente a disco), los comandos son confusos, requiere de muchos pasos para realizar operaciones simples, la interfase es obsoleta.

Como si esto fuera poco, PDS trabaja con un subconjunto de comandos muy limitados de Microstation, para evitar que el usuario corrompa la integridad de los datos. No existen por ejemplo comandos para copiar y renombrar una línea entera de cañería, etc. En cambio sigue requiriendo doble pantalla por la proliferación de menús que un tiempo se invocaban desde una mesa digitalizadora y ahora están en pantalla.

2.4 BASE DE DATOS PROPIETARIA

Toda la información de referencia del sistema es guardada en una base de datos propietaria a la cual se accede con un formato texto (la interfase es el Notepad).

A este respecto, veinte años de evolución en bases de datos han pasado sin que Intergraph se haya dado cuenta.

Es guardada en esta forma, información fundamental como:

- Tablas de dimensiones de componentes.
- Definición paramétrica de componentes.
- Definición de las Especificaciones de Cañería.

2.5 BASE DE DATOS DE MATERIALES DEL PROYECTO

Los elementos que se dan de alta gráficamente en un proyecto tienen asociadas varias tablas donde el sistema guarda información alfanumérica de vario tipo.

PDS utiliza un sistema de base de datos relacional (por ejemplo Oracle) para estas operaciones.

Un estudio pormemorizado de la estructura de datos empleada muestra claramente que la arquitectura original jerárquica ha sido respetada y adaptada de alguna forma a una estructura relacional, dando origen a una proliferación de tablas inusitada.

En este esquema, a cada modelo (archivo de Microstation) corresponden varias tablas por separado. Por ejemplo cada modelo de equipos tiene asociadas 8 tablas nuevas, cada modelo de cañería 4 tablas nuevas, etc.

Si bien el usuario en principio no necesita conocer estos detalles, una estructura de datos de este tipo muestra deficiencias de desarrollo serios y genera la imposibilidad de acceder directamente a los datos del proyecto via consultas a la base de datos, que es a los efectos prácticos totalmente

crítica y no contiene todos los datos necesarios: algunos son guardados únicamente en los archivos gráficos de Microstation, por ejemplo la línea asociada a cada componente de cañería.

La única forma para obtener reportes de materiales del proyecto es utilizando el sistema de reportes propio de PDS, que acceden también a los archivos de Microstation. La única salida disponible es a un archivo de texto. El formato de los reportes se define mediante templates en archivos de texto.

2.6 ESPECIFICACIONES DE CAÑERÍA

Se puede destacar una sola característica positiva: las clases son definibles por rango de diámetro y no es necesario que los componentes estén realmente disponibles en el catálogo antes de ser cargados en la clase.

Por el resto, su estructura es realmente caótica. Veamos porqué.

La definición de las Clases de Especificaciones utiliza una mezcla de códigos (que son mantenidos en las Standard Notes, o sea tablas de validación y traducción de códigos) y parámetros.

Si bien el acceso a la generación de los componentes utiliza directamente los parámetros y códigos de la clase, inexplicablemente uno de los parámetros principales (el Model Code), no tiene asociada ninguna Standard Note, lo cual impide realizar listados de materiales intelegibles, traduciendo este código a un equivalente descriptivo.

Esto obliga a utilizar un código denominado Commodity Code, que tiene asociadas las características descriptivas del componente INDEPENDIENTEMENTE de los efectivos parámetros utilizados por el componente mismo.

Es una falencia muy grave que genera un trabajo adicional innecesario en la generación de estos códigos y sus descripciones asociadas, cualquier modificación en la clase genera la necesidad de modificar estos códigos y su descripción. Facilita la generación de errores en los cómputos al duplicar información.

Es un buen ejemplo de como una mala arquitectura de datos origine complicaciones inútiles al usuario e inseguridad en los cómputos.

PDS no es un sistema abierto: varios parámetros (por ejemplo los códigos de extremo) son agrupados según categorías fijadas a nivel de código del programa, si bien algunos seteos se hacen a nivel de las Standard Notes.

Los Códigos de Extremo son utilizados también para identificar caras bridadas que se unen con bulones pasantes, cuando el uso de un bulón pasante es una característica propia del componente y no de la cara.

La selección de los códigos (Model Code, End Preparation, AABCC code, etc.) utiliza reglas arbitrarias organizadas caóticamente y accesibles desde sólo dos lugares: los manuales y la tradición oral de los instructores. Lo mismo pasa con las reglas de Formación de Nombres de Tablas de Dimensiones.

Estas incoherencias a nivel de definición de propiedades y falta de criterios generan muchos problemas en la redacción de las clases y en la interpretación de los MTO.

2.7 INCOMPATIBILIDAD ENTRE PROYECTOS

Cada proyecto utiliza únicamente la información almacenada en el directorio de proyecto.

Los modelos gráficos, las definiciones paramétricas de los componentes, las tablas de dimensiones, las especificaciones, las tablas de traducción de códigos (las Standard Notes), etc. están relacionados a cada proyecto específico.

Cuando se modifica la forma de un componente, se agregan tablas de dimensiones o peso o se realiza una modificación cualquiera, ésta únicamente se incorpora al proyecto en ejecución. Para incorporar estas modificaciones a cualquier otro proyecto es necesario transferirlas manualmente via archivos de texto.

Si bien este esquema simplifica el backup del proyecto, complica compartir información entre proyectos distintos.

La transferencia de partes del proyecto, por ejemplo de modelos enteros de un proyecto a otro, en teoría se puede hacer via el sistema de BackUp denominado Archival. En teoría, porqué en la práctica es un proceso vulnerable a cambios de versiones y a la complejidad de los parámetros a setear.

No existe un catálogo consultable automáticamente con las piezas disponibles. Esto se debe en parte al formato texto empleado para guardar los datos y en parte a la forma caótica de organización de la información.

2.8 COMANDOS INEFICIENTES

Los comandos gráficos son por lo general confusos y requieren pasos innecesarios que podrían ser automatizados. A continuación se dan algunos ejemplos:

- No existe un comando de Alta de Línea de Cañería, por lo cual cuando se define una línea de ruta nueva no hay ninguna forma de saber si se está dando de alta realmente una línea nueva o, por error se está utilizando una línea que ya existe en el modelo. Menos hay algún recaudo para saber si esa línea ya está definida en otro modelo. La única forma para saber que líneas están definidas en un modelo es generando un reporte a un archivo de texto, desde el módulo de reportes.
- La secuencia en la colocación de componentes necesita de tres pasos: posicionamiento del Punto Activo (una referencia gráfica en la pantalla), una selección genérica del componente que se quiere generar, la selección del punto de colocación. Por ejemplo al colocar una brida hay que especificar por que punto se la inserta, cuando en realidad el sistema tiene, en la mayoría de los casos, todos los elementos para decidirlo solo.

- La diagnostica de errores es un problema grave: salvo pocos casos, no dá la mínima clave de lo que está realmente pasando y es muchas veces contradictoria. Por ejemplo al colocar un componente puede aparecer un error señalando la falta de ese componente en la clase, cuando el problema es otro (por ejemplo una inconsistente definición de un extremo). Aparenta ser un sistema sofisticado por los controles que hace, pero sin ayuda al usuario para corregir los errores que comete.
- Errores aleatorios: de vez en cuando un error sin explicación (ver típica pantalla en esta hoja) impide la generación de un componente. En algunos casos, la misma secuencia de comandos en el mismo lugar del modelo permite generar el componente. Que pasó ?



- El comando de colocación automática de accesorios y caños (Auto Placement) es de uso relativo. En muchas oportunidades no es recomendable su uso: puede generar componentes que no existen (tramos de caño), puede eliminar componentes que sí existen (cuando por errores de tolerancia no hay espacio físico para un componente, ejemplo una tee), tiene serios problemas con las cañerías inclinadas.
- Todo lo que se hace es grabado a disco: no existe un comando de Undo, común a cualquier sistema desde hace muchos años. Esto hace bastante lenta la generación de los modelos, por las precauciones que hay que tomar: no es un sistema permisivo contra errores.
- No se pueden visualizar líneas por separado, lo cual obliga a no poner más de 20 o 30 líneas por modelo para evitar confusión.

2.9 GENERACION DE ISOMETRICOS DE CAÑERIA

Se realiza utilizando el programa Isogen, desarrollado en los años 70. De esa época mantiene las características de funcionamiento: configuración via archivo de texto (palabras clave seguidas de parámetros, que revela el original input via tarjetas) y generación del gráfico sin inteligencia alguna (se sustituyó la escritura directa a un plotter a un archivo gráfico). Está disponible una interfase para ayuda en la generación de este archivo. La posibilidad de personalizar los gráficos es muy limitada.

Si bien la calidad gráfica no es mala genera archivos sin inteligencia, con lo cual, cualquier dato que no aparece automáticamente hay que agregarlo manualmente en Microstation.

2.10 MTO

Generalmente es obtenido por reportes del Isogen, que aseguran mejor consistencia que los reportes de maquetas. Por lo menos esta es la postura de la mayoría de los usuarios. Esto se debe básicamente a que el trabajo en los modelos 3D es dificultoso, proclive a tolerar errores que no son detectados hasta la extracción del isométrico.

El reporte es obtenido en formato texto. No hay ningún módulo que permita la generación automática de Requerimientos de Materiales y su seguimiento.

El esquema de las Clases de Cañería con su dependencia absoluta de los Commodity Codes genera mucha incertidumbre en la consistencia de los cómputos en un proyecto, cuando el cambio de las clases hasta último momento es una práctica habitual. El resultado concreto son trabajos adicionales de chequeo manuales.

2.11 REGENERACION DEL GRAFICO

El comando que permite la regeneración de los componentes para adecuarlos a cambios en las clases de especificaciones no es automático y es de uso problemático, con lo cual se lo trata de utilizar al máximo una sola vez en todo el proyecto y al final. El resultado es que los cómputos automáticos desde maquetas durante un proyecto no son obtenibles con el estado de avance de los modelos.

2.12 DESIGN REVIEW

Es el sistema para realizar un WalkThrough por la planta virtual. Permite integrar más de un proyecto de PDS. Si bien la interfase es obsoleta, es un producto usable, pero tiene un muy serio problema: no es compatible 100% con los archivos DGN de Microstation. En otras palabras hay muchas circunstancias en las cuales directamente no visualiza todos los objetos generados con PDS y esto en forma aparentemente aleatoria. De esta forma obliga a chequeos y pérdida de tiempo adicionales, que no tienen ninguna justificación.

3. RESUMEN FINAL

Hay un punto a favor:

- Es un sistema difundido a nivel mundial y es requerido por algunas empresas, por lo menos como formato final de entrega de los modelos electrónicos. Este es por lo general el UNICO MOTIVO DE USO: cuando el cliente exige y paga por él. En este sentido el lobby de Intergraph ha sido fuerte, sobre todo en generar requerimientos en empresas que contratan grandes obras de ingeniería de montaje.

Los puntos negativos principales son:

- Una mala arquitectura de datos, que obliga a un trabajo adicional manual en la redacción de las Clases de Canería, facilitando errores en los cómputos y generando un overhead en controles manuales injustificados en un sistema de este tipo.
- Operatoria gráfica ineficiente y complicada, con baja productividad.
- Un sistema innecesariamente complicado, con altos costos de capacitación y mantenimiento.
- Aparece un desarrollo casi nulo en los últimos diez años, abriendo una fuerte incertidumbre sobre su uso futuro.