

42
34



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

FACULTAD DE INGENIERIA

**COMUNICACION ENTRE SISTEMAS
CAD**

TESIS PROFESIONAL
QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A :
MA. DEL ROSARIO NICOLAU BARRAINCUA

DIRECTOR:
M. EN C. GABRIEL MAURICIO ALVAREZ MEDINA



MEXICO, D. F.

1990





UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

INDICE

	Página
CAPITULO I INTRODUCCION	
1.1 Sistemas CAD	1
1.2 Organización de la Tesis	2
1.3 Objetivos	3
CAPITULO II EL SOFTWARE GRAFICO DE UN SISTEMA CAD	
II.1 Estandares Gráficos	4
II.2 Interfaces Gráficas	5
II.2.1 CORE	6
II.2.2 GKS	7
II.2.3 PHIGS	11
II.2.4 CGM	13
II.2.5 CGI	15
II.2.6 NAPLPS	17
II.3 IGES (Initial Graphics Exchange Specification)	18
CAPITULO III EL SOFTWARE DE APLICACIONES DE UN SISTEMA CAD	
III.1 Aplicaciones CAD	22
III.2 Estudio de Casos Especificos de Software CAD	
III.2.1 I-DEAS CAEDS (SDRC)	24
III.2.2 CADAM (IBM CADAM INC)	30
CAPITULO IV INTEGRACION DE SISTEMAS CAD CON BASES DE DATOS	
IV.1 Importancia de esta integración	33
IV.2 Conceptos de Bases de Datos	34
IV.2.1 Arquitectura de un SMDB	35
IV.2.1.1 ANSI/X3/SPARC	35
IV.2.1.2 CODASYL 71	35
IV.2.1.3 CODASYL 78	37
IV.2.2 Modelos de Datos	37
IV.2.2.1 Modelo Relacional	38
IV.2.2.2 Modelo Jerárquico	38
IV.2.2.3 Modelo Reticular	39
IV.2.2.4 Comparación entre los Modelos	40
IV.3 Bases de Datos CAD	40
IV.3.1 Características de un SMDB CAD	42
CAPITULO V TRANSFERENCIA DE INFORMACION GRAFICA CADAM-CAEDS (TGCAD)	
V.1 Planteamiento General	46
V.2 Estructura de TGCAD	47
V.3 Ejemplificación	51
CONCLUSIONES	57
GLOSARIO	59
REFERENCIAS	62

CAPITULO I

INTRODUCCION

I.1 Sistemas CAD.

Los sistemas de Diseño Asistido por Computadora, sistemas DAC o sistemas CAD ("Computer Aided Design"), como se les denomina mundialmente, representan una tendencia tecnológica que integra, bajo un ambiente de computación, diferentes disciplinas de las áreas físico-matemáticas con objeto de acelerar algunas fases del proceso de diseño susceptibles de ser automatizadas. Este proceso, bajo un esquema general, comprende la síntesis, el análisis, la optimización, documentación y producción de un producto.

Para la ingeniería, CAD puede considerarse como una herramienta auxiliar en áreas como la mecánica, la civil, la eléctrica, la electrónica o la aeronáutica, por mencionar tan sólo algunas, permitiendo al ingeniero centrar su atención en la actividad creativa del diseño, editando, visualizando, analizando y documentando a éste en forma interactiva.

Historicamente, CAD se inició en la década de los 60's como una tecnología de dibujo por computadora, muy limitada por el hardware existente y pobre en las estructuras de datos y algoritmos de procesamiento de información. En los 70's creció el interés de científicos y analistas, generando gran parte del trabajo teórico que asienta las bases del CAD actual, ubicándose ya para los 80's como una tecnología de vanguardia con un mercado en desarrollo bien localizado y acrecentando su influencia en los sistemas clásicos de diseño y producción.

I.2 Organización de la Tesis.

El software de un sistema CAD puede dividirse en tres categorías según su función: el Sistema Operativo, el Sistema de las Aplicaciones y el Sistema Gráfico [BESABO]. El Sistema Operativo se encarga de la organización y uso de los recursos de la computadora. Como los sistemas CAD no tienen una dependencia directa con algún tipo de sistema operativo en particular, en el presente trabajo se estudian los otros dos sistemas.

El objetivo del Sistema Gráfico es proporcionar las funciones gráficas requeridas por los submódulos de aplicación para la captura, procesamiento y despliegue de información. El Capítulo II trata de la importancia de los estándares gráficos en este Sistema, su interrelación e introduce a las principales características de cada uno de ellos.

El Sistema de Aplicaciones (Capítulo III) está constituido por programas que a través del Sistema Operativo y el Sistema Gráfico, realizan los procesos de dibujo, diseño, análisis y documentación del objeto de interés. En este Capítulo se introduce a este software y dos paquetes comerciales internacionalmente conocidos: CAEDS de SDRC y a CADAM de IBM CADAM-INC., que formaron la base de desarrollo en esta tesis.

El Capítulo IV trata el problema del crecimiento y variedad en los sistemas CAD y de los beneficios que se pueden obtener al integrarlos con una base de datos. Se describen las características más importantes que deben cubrir la tecnología de las Bases de Datos para soportar Aplicaciones CAD.

La realización de una interface unidireccional entre los productos CADAM-CAEDS es descrita en el Capítulo V. Esta comunicación se realiza a través de "Geometry Interface" (Interface Geométrica) de CADAM e I-DEAS PEARL, la Base de Datos de CAEDS, utilizando los lenguajes FORTRAN H-EXTENDED e IDEAL que es un lenguaje interactivo de CAEDS.

Para finalizar se presentan las conclusiones de esta tesis.

1.3 Objetivos.

Transferir información geométrica entre dos Sistemas CAD que posee la Universidad, CADAM y CAEDS, de Diseño e Ingeniería Asistidos por Computadora. Estos dos productos son considerados líderes en el mercado internacional y actualmente entre ellos, no existe un medio de transferencia de información.

Analizar los estándares gráficos como mecanismos de comunicación entre Sistemas CAD.

Analizar la tendencia actual en ambientes CAD, de integrarlos a través de un Sistema Manejador de Bases de Datos.

CAPITULO II

EL SOFTWARE GRAFICO DE UN SISTEMA CAD

II.1 Estándares Gráficos.

Debido a la gran importancia que tiene la componente gráfica en un sistema CAD, en el presente Capítulo se hace énfasis especial a los esfuerzos de estandarización gráfica dentro de la comunidad CAD, describiendo cada estándar gráfico y su interrelación.

Su finalidad es lograr la independencia entre el software de aplicación y los dispositivos gráficos, esto permitiría que el software existente no requiera ser modificado ante un cambio de equipo; que los programas sean transportables entre instalaciones distintas y que el programador de aplicaciones no tenga que recibir capacitación diferente conforme al equipo con el que vaya a trabajar [NEWMS1] [HAGEB1]. Están estructurados por secciones, siendo la primera una descripción funcional, donde se expone la semántica estandarizada, tratando que sea lo más independiente posible del hardware, del lenguaje y del sistema operativo. Las secciones restantes son especificaciones de la sintaxis, que se refieren a la forma como se organizarán los parámetros y los caracteres que constituyen un comando para ser ejecutado.

La independencia indicada en el párrafo anterior, se pretende lograr a través de interfaces entre el programa de aplicación y el dispositivo gráfico, proporcionando un conjunto de funciones gráficas independientes de las características del dispositivo.

La estandarización de los componentes de un Sistema CAD, ha seguido dos vertientes [ENCAB84], la primera, a través de la definición de funciones gráficas; el traslado y almacenamiento de información gráfica y la segunda vertiente se refiere, a los datos que definen un objeto producido por un sistema CAD para ser transferidos a otro sistema CAD.

II.2 Interfaces Gráficas.

Las interfaces gráficas estandares influyen en la concepción de un sistema CAD, al forzar la separación de cada componente y fijar las funciones que cada una debe realizar.

En la Figura 2.1 se muestran las interfaces gráficas estandares y su relación entre ellas.

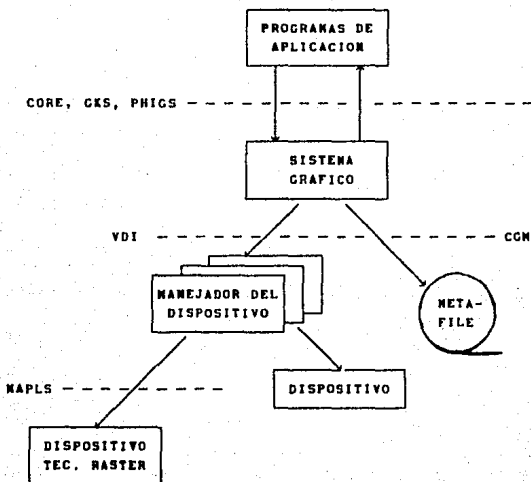


Figura 2.1 Interfaces Gráficas

2.2.1 CORE.

CORE es una propuesta de estándar para un paquete de subrutinas gráficas. Fue desarrollado por la ACM SIGGRAPH Graphics Standards Planning Committee (GSPC) y se publicó en 1977 para tecnología vector y en 1978 se presentó una revisión para tecnología raster.

La metodología en el diseño de CORE se orientó a la estructura de los programas que lo iban a soportar, debido a que esta, es el factor más importante para la transportabilidad de los mismos. Con este objetivo, se separaron las funciones gráficas esenciales (núcleo o core) para la generación de dibujos, de otras funciones como las de modelado, para que el programador de aplicaciones tenga un conjunto de funciones gráficas ya desarrolladas, independientes del dispositivo físico que las utilice [MICH81b].

CORE genera objetos gráficos en 2D o 3D por medio de funciones que generan primitivas gráficas (ej. una línea o una cadena de caracteres) acompañadas de parámetros (ej. coordenadas), atributos de las primitivas (ej. color, ancho de línea, intensidad) y la transformación de vista que es la descripción de la posición visual de un objeto, al escoger la región gráfica en la que será desplegada y reproducida en la superficie lógica de salida [MICH81a].

En CORE se mantienen dos tipos de estructura de datos gráfica según el nivel al que pertenezcan: la no estructurada y la estructurada por segmento. La primera permite solo dos operaciones sobre la estructura de datos; adicionarla o borrarla. En la segunda se define una imagen fragmentándola en segmentos que contienen primitivas que pueden ser creadas, cambiadas, borradas o reemplazadas. Sólo un segmento puede ser abierto a la vez y al cerrarse no se le puede hacer ninguna modificación [MICH81b].

Existen cuatro niveles de implantación en CORE debido a que fue diseñado para soportar un sistema según las diferentes necesidades gráficas que se requieran para cada caso.

Para funciones que no hayan sido contempladas durante la creación de este estándar, existe una subrutina de escape para manejarlas, esto ocasiona que se consideren características propias de un dispositivo y se pierda la transportabilidad de las aplicaciones [BRUN88].

Por último, CORE no posee un estándar con la interface de un lenguaje, lo cual provoca una serie de modificaciones en los programas de aplicación por cada instalación que lo soporte.

II.2.2 GKS.

Al igual que CORE, GKS -Kernel del Sistema Gráfico- ("Graphical Kernel System") es una interface funcional entre el programa de aplicación y un Sistema Gráfico. Fue desarrollado por ISO ("International Standard Organization") y por ANSI ("American National Standard Institute") para graficación en dos dimensiones y aceptado como estándar en 1985.

La Figura 2.2 muestra el papel que desempeña GKS en un Sistema Gráfico, donde cada capa, puede evocar a las funciones de las capas adyacentes por la parte inferior. De esta forma el programa de aplicación puede tener acceso a la capa orientada a la aplicación, a la capa dependiente del lenguaje y a los recursos del sistema operativo [ENCA83]. La interface de la capa dependiente del lenguaje es la que integra a GKS con un lenguaje, proporcionando las convenciones de lenguaje necesarias como los nombres de las funciones, los tipos de datos o la secuencia de las llamadas.

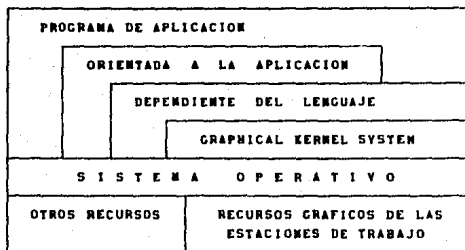


Figura 2.2 Modelo por capas de GKS

GKS utiliza el concepto de "estación de trabajo" ("workstation") para obtener la independencia de dispositivo. Una estación de trabajo, es una abstracción de dispositivos físicos donde sólo existe un área de desplegado o ninguna y una o más áreas de dispositivos de entrada [ENDE84].

Con respecto a la generación de imágenes, esta se logra a través de elementos básicos de graficación llamados primitivas. Cada primitiva se define mediante un sistema de coordenadas absolutas, diferenciándose de CORE, en que este último emplea un sistema de coordenadas relativas.

A continuación se definen las primitivas de GKS:

- POLYLINE.** Forma líneas rectas conectadas a través de una sucesión de puntos.
- POLYMARKER.** Genera símbolos que son utilizados como señales para identificar localizaciones.
- TEXT.** Genera una cadena de caracteres.
- FILL AREA.** Define los límites que van a ser rellenados por un modelo de dibujo patrón.
- CELL ARRAY.** Se crea un arreglo de celdas o "píxeles" cada una con su propio valor e intensidad.

GENERALIZED DRAWING PRIMITIVES. Permite manejar funciones geométricas especiales que generan curvas, arcos circulares y arcos elípticos.

Las primitivas que forman un dibujo pueden agruparse por partes llamadas segmentos. Los segmentos se pueden crear, borrar, copiar, insertar, modificar y transferirse entre estaciones de trabajo.

En el ambiente de GKS, se definen tres sistemas coordenados:

a).Coordenadas de Mundo (CM). Es el sistema de coordenadas utilizado por el programador de aplicaciones para definir su entrada y su salida gráfica.

b).Coordenadas Normalizadas del Dispositivo (CND). Es un sistema coordenado para información gráfica normalizado en el rango de (0,0) a (1,1) de números reales.

c).Coordenadas del Dispositivo (CD). Sistema coordenado por cada estación de trabajo basados en las dimensiones de la superficie de desplegado.

Las primitivas y sus atributos, que se comentan en el siguiente párrafo, son mapeados de CM a CND y viceversa por transformaciones de normalización, de CND a CND por transformaciones de segmento y de CND a CD y viceversa por transformaciones de estación de trabajo como se muestra en la Figura 2.3.

Un atributo en GKS es una propiedad particular que controla la apariencia de un dibujo. Los atributos de las primitivas comprenden características como el color, tipo de línea, tamaño de carácter, espaciado, etc. Los atributos de segmento son visibilidad, detectabilidad, luminosidad, prioridad (precedencia para entrada y salida) y transformación.

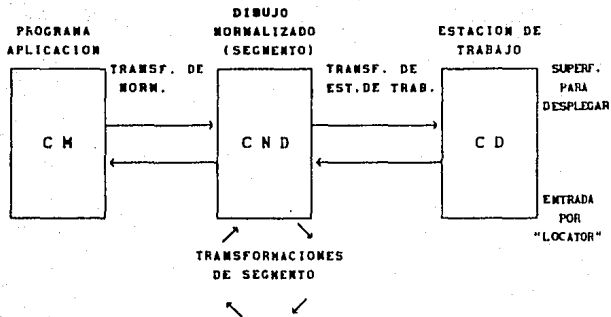


Figura 2.3 Modelo Gráfico en GKS

Las características de los dispositivos físicos de entrada son mapeadas a dispositivos lógicos de entrada. GKS clasifica a estos últimos en las siguientes seis clases, de tal forma que sean equivalentes según la función que realicen:

- LOCATOR. Proporciona una posición en Coordenadas de Mundo.
- STROKE. Proporciona una secuencia de posiciones en CM.
- VALUATOR. Provee un número real.
- CHOICE. Provee un número entero de un conjunto a seleccionar.
- PICK. Proporciona el nombre de un segmento con un identificador asociado con alguna primitiva en particular.
- STRING. Proporciona una cadena de caracteres.

Para implementación, GKS maneja nueve niveles que cubren las diferentes necesidades de los sistemas gráficos. Después de haber sido aprobado como estándar, se pensó en la incorporación de extensiones a 3D que fuesen completamente compatibles a lo desarrollado anteriormente.

GKS-3D maneja las mismas primitivas que GKS pero en 3D y maneja una nueva primitiva, la "Fill Area Set" que permite especificar el llenado de diferentes áreas y controlar su apariencia en los bordes, introduciendo atributos relacionados a

esta primitiva como el tipo de borde, factor de escala en el ancho del borde y un índice del color del borde [PUK_86]. En el diagrama de transformaciones se define un nuevo tipo de transformación, la de vista, la cual, a partir del dibujo normalizado, permite la orientación de una vista con sus coordenadas.

11.2.3 PHIGS.

PHIGS ("Programmers Hierarchical Interface Graphics System") es un estándar diseñado conjuntamente por ANSI e ISO que soporta una estructura gráfica jerárquica en dos y tres dimensiones.

Maneja, al igual que GKS, el concepto de estación de trabajo y tiene las mismas primitivas y atributos de éste, diferenciándose más bien por su organización y manipulación de datos gráficos.

PHIGS se organiza por estructuras de datos llamadas "estructuras" constituidas por primitivas, atributos, datos de los programas de aplicación, transformaciones de modelado, vistas y referencias a otras estructuras (ver Figura 2.4).

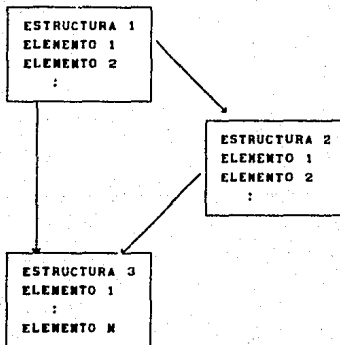


Figura 2.4 Organización de los datos en PHIGS

Las estructuras pueden ser modificadas con un editor gráfico mediante el cual es posible cambiar las primitivas, los atributos o modificar la jerarquía. A través de los "set names" -agrupación de clasificaciones de definición bajo un nombre- PHIGS puede controlar la intensidad de la luz, la visibilidad y la detectabilidad de un grupo de primitivas aunque estén en diferentes estructuras, a través de dos tipos de filtros (inclusión y exclusión), facilitando estas tareas de despliegado en el diseño interactivo.

La modificación del modelo gráfico en PHIGS se muestra en la Figura 2.5 y sus definiciones fueron resumidas del artículo de David Shuey [SHUE86].

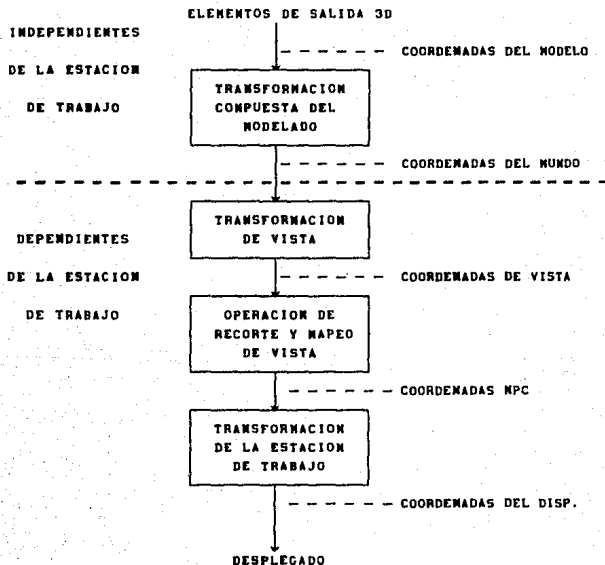


Figura 2.5 Modelo Gráfico en PHIGS

Las transformaciones del modelo se usan para localizar y orientar objetos en el espacio y mantener su relación con otros objetos.

Las Transformaciones de vista son las que describen la orientación del punto de vista de la escena.

El Recorte y Mapeo de Vista, son transformaciones a las entidades gráficas del sistema coordinado de la vista, hacia espacio de las coordenadas de proyección normalizada (Normalized Projection Coordinate) NPC, generando como producto final únicamente la información gráfica de interés.

Las transformaciones de la estación de trabajo mapean las coordenadas del espacio NPC a las coordenadas del dispositivo de despliegue de la estación de trabajo.

II.2.4 CGM (Computer Graphics Metafile).

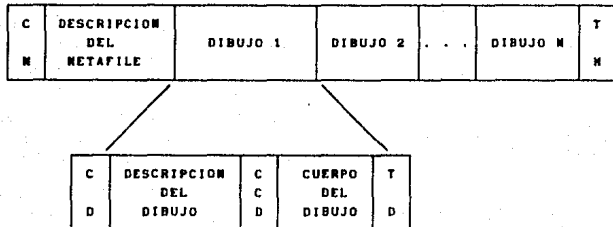
CGM es un archivo secuencial que contiene la descripción gráfica de un dibujo con el objeto de servir como medio de almacenamiento y transferencia de información gráfica independiente del dispositivo gráfico. En 1980 se empezó a trabajar para el desarrollo de este estándar bajo el nombre de Virtual Device Metafile, en 1984 adquirió el nombre actual y en 1986 fue aceptado como estándar.

CGM estandariza la semántica y la sintaxis de un conjunto de elementos para la definición de dibujos independientes del dispositivo [HEND86].

El estándar esta organizado en cuatro partes: la primera es una especificación funcional donde se identifica cada elemento, sus parámetros y su significado. Las otras tres partes presentan las diferentes formas de codificación, por caracteres ASCII (2a. Parte), binaria (3a. Parte) y texto claro (4a. Parte) referente a

la fácil comprensión por un lector humano. El estándar no especifica el formato físico de los registros en ninguna forma de codificación.

La estructura del Metafile esta formada por una descripción del Metafile y una secuencia de dibujos independientes, cada uno con su propia descripción (Ver Figura 2.6).



Delimitadores:

CM = Comienza Metafile TM = Termina Metafile CD = Comienza dibujo
 CD = Comienza cuerpo del dibujo TD = Termina Dibujo.

Figura 2.6 Estructura del Archivo Metafile

Los elementos que componen el metafile se definen en un espacio coordenado cartesiano bidimensional llamado Coordenadas del Dispositivo Virtual (CDV), que puede ser entero o real y de precisión variable. La especificación del atributo de color se logra mediante un mecanismo indexado a una tabla de colores o en forma directa, mediante una combinación de los tres colores primarios luz (rojo, verde y azul).

La "descripción del metafile", contiene las declaraciones que son válidas para todo el metafile, como su versión y descripción, el tipo de coordenadas (VDC), su precisión, y valores default para los atributos de los dibujos. En la "descripción del dibujo", se especifican los elementos que aplican a ese dibujo en especial, como la selección del modo de color (indexado o directo), el color del fondo, el ancho de la línea, etc.

El cuerpo del dibujo esta formado por primitivas y los atributos asociados a estas (ver [HEND86] para una referencia más completa).

Los metafiles se utilizan para almacenamiento de datos generados por sistemas gráficos en cintas, ya sea como respaldo o como transferencia a otro sistema gráfico. Otra aplicación es la transferencia de información del programa a un dispositivo gráfico, utilizando el formato de CCM para el sistema de SPOOL ("Simultaneous Peripheral Operations Online") que es el sistema que controla la utilización de los recursos periféricos.

11.2.5 CGI.

CGI ("Computer Graphics Interface") es un estándar que especifica las funciones y la sintaxis en el intercambio de datos y su control entre el software gráfico independiente del dispositivo y uno o más manejadores de dispositivo, dependientes del dispositivo gráfico. Este estándar esta siendo examinado por el comité de ANSI, X3H3.3 el cual decidió que aceptará únicamente datos en 2D, reservando a los sistemas gráficos la tarea de convertir los datos 3D a proyecciones en 2D.

El software CGI esta formado por dos partes: el intérprete de los comandos, que no requiere ningún cambio según se adicionen nuevos dispositivos periféricos, y los manejadores de dispositivo, responsables de producir la imagen deseada para un dispositivo en particular. Esto permite que no sea necesario modificar el software gráfico conforme aparezcan nuevos dispositivos, sino más bien, escribir un manejador de dispositivo para la nueva unidad.

COMPONENTES DE CGI.

CGI esta definida por 5 conjuntos de funciones (POWERB):

- a). Control, Negociación y Errores
- b). Salida y Atributos
- c). Segmentación
- d). Entrada
- e). Raster

Las funciones de control del dispositivo son las que intervienen en la inicialización y terminación de una sesión de CGI (atributos y estados default de las funciones), en la implantación del buffer de salida para la actualización periódica de la imagen desplegada y las funciones de comienzo y terminación de imagen.

Negociación, se refiere a la forma en que CGI puede soportar las variaciones en las características físicas del hardware que maneje, por medio de funciones de consulta que son las que regresan información acerca de las limitaciones y características del dispositivo, para poder decidir las funciones gráficas que se pueden usar y cuales se pueden simular, generando un software más flexible para soportar hardware nuevo.

En caso de existir esta comunicación, se define un conjunto mínimo de funciones y capacidades disponibles en todo el rango de dispositivos que van a ser soportados. Ahora bien, con la evolución tecnológica del hardware gráfico, debe pensarse en el aprovechamiento de sus innovaciones. Pensando en lo anterior, CGI controla el problema de las variaciones en el hardware, distribuyendo las funciones relacionadas en grupos llamados "option sets", de tal forma que controlen un aspecto de la interface. Cada "option set" a su vez esta subdividido en funciones requeridas y no requeridas, obteniéndose una guía para los implementadores con la que pueden escoger subconjuntos de funciones de CGI que cubran las necesidades de sus productos.

El manejo de errores se simplifica proporcionando sólo orientaciones a errores más o menos serios con objeto de que el implementador pueda escoger el procedimiento más conveniente y económico para la detección y respuesta a los mismos.

Las primitivas y atributos que soporta CGI son las mismas que CGM (ver que están al mismo nivel en la Figura 2.1) y también están expresadas en el mismo espacio de coordenadas de dispositivo virtual (VDC). A diferencia de CGM, CGI tiene la capacidad de llenado de áreas en figuras no contiguas formando una figura cerrada. Esta función depende fuertemente del dispositivo, por lo cual es muy importante que CGI la soporte.

La segmentación es similar a la de GKS, pero esta limitada, no permite edición por elemento de segmento y un segmento solamente puede ser renombrado, borrado, reabierto para adición de elementos o para ser copiado en otro segmento.

Las funciones de entrada están diseñadas para que soporten los modelos de entrada de otros estándares. Las operaciones Raster sirven para manejar imágenes o partes de estas anteriormente generadas, compuestas por elementos llamados "píxeles" que se encuentran organizados en forma rectangular en una memoria de refresco para formar un "bitmap" o mapa de bits, de donde son trasladados a la pantalla.

II.2.6 NAPLPS.

NAPLPS ("North American Presentation Level Protocol Syntax") es una interface gráfica estándar del código que define, almacena y transmite información gráfica. A diferencia de estándares como CGI, NAPLPS explota una tecnología en particular, la de formato RASTER, con objeto de obtener un mayor provecho de esta tecnología tan difundida. Fue diseñado para la transmisión de texto e imagen en un código de formato muy compacto para el sistema Videotext.

Los datos y comandos en NAPLPS son enviados por caracteres ASCII que representan 256 piezas de datos. Para superar esta limitante, una secuencia de comandos indica una de seis tablas que se superponen a los valores de ASCII [PANAB4]. La primera tabla esta compuesta por caracteres ASCII asociados a valores ASCII. La segunda es un complemento de caracteres de la primera, como simbolos y algunas letras griegas. La tercera tabla esta formada por comandos para la creación de dibujos. La cuarta, contiene un conjunto de "mosaicos" e incluye los caracteres del sistema europeo Prestel. La quinta tabla es un conjunto de macros de comandos que asignan una etiqueta a una serie de comandos de dibujo. La última tabla contiene comandos definidos por el programa aplicación. Para lograrlo, el programa primero debe enviar su definición de los comandos a una terminal NAPLPS.

II.3 IGES

IGES ("Initial Graphics Exchange Specification") es un estándar en la definición y formato de información, creada y almacenada por sistemas CAD, publicado por la National Bureau of Standards en 1980.

Utilizando IGES, la transferencia de datos entre dos sistemas CAD requiere que cada uno de ellos tenga dos procesadores; un procesador que construya la descripción solicitada y un postprocesador que la pueda reconstruir, como se ilustra en la Figura 2.7.

La idea principal, más que un intercambio de modelos, es una descripción común del modelo donde cada sistema CAD debe ser capaz de generarla e interpretarla en su representación interna propia [ENCAB4].

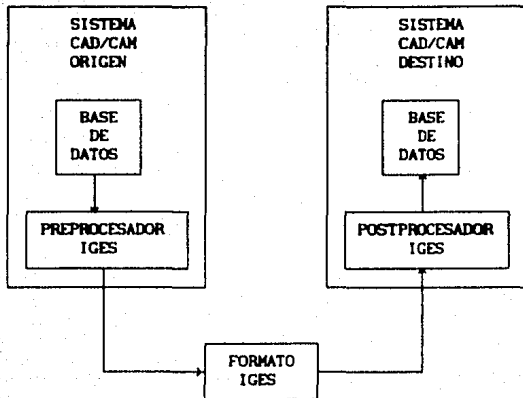


Figura 2.7 Comunicación entre dos Sistemas CAD con IGES.

La información en un archivo IGES está formada por unidades de información llamadas entidades. Estas pueden ser de cuatro tipos: las que describen una geometría (como líneas y curvas); las de anotación (como medidas y notas en el dibujo); las que definen una estructura (como grupos y asociaciones) y por último las que no están incluidas en las tres anteriores (como elementos finitos o superficies) [BESAB0].

El formato de un archivo IGES está formado por cinco secciones [IBM_86]:

1. Sección de Inicio. Es el prólogo del archivo.
2. Sección Global. Contiene la información que describe el preprocesador y proporciona la información necesaria para que el postprocesador maneje el archivo.
3. Sección de Entrada de Directorio. Contiene para cada entidad una entrada de directorio que proporciona un índice para el archivo e información de sus atributos.

4. Sección de Parámetros. Contiene todos los datos de los parámetros asociados a cada identidad.
5. Sección Final. Esta formada por un registro que contiene el número total de registros en cada una de las secciones anteriores.

Las relaciones en el mapeo de entidades entre IGES y un sistema CAD pueden ser de cinco formas:

- Uno a uno. El sistema transmisor envía un equivalente exacto al receptor.
- Uno a muchos. Una entidad en el sistema transmisor debe ser almacenado en más de una entidad del sistema receptor.
- Muchos a uno. Varias entidades del sistema origen son convertidos a una sola entidad en el sistema destino.
- Aproximación. Una entidad definida en el sistema transmisor carece de un equivalente exacto en el receptor.
- No mapeable. Una entidad en sistema transmisor no puede ser aproximado a un equivalente.

Scowen [SCOW83] hace una crítica hacia IGES, identificando los defectos en el diseño de este estándar, para que no se repitan en un posible sucesor. Las fallas principales detectadas por él son:

- * Los archivos IGES son muy grandes, cada entidad requiere por lo menos de de tres registros (240 bytes).

- * Existen diferentes formas para definir un objeto, p.e. un rectángulo puede ser una curva compuesta (102), o un plano limitado (108) o una subfigura.

- * Si el propósito de este estándar es la transferencia de información entre dos sistemas diferentes, no existe razón para que puedan ser leídos por un humano, por lo que sugiere Scowen que los valores numéricos estén en hexadecimal para evitar

errores e imexactitudes en las conversiones de binario a decimal y viceversa.

* El procesador de IGES no contempla la posibilidad de errores o inconsistencia y la detección de estos por transmisión esta pobremente diseñada.

* Finalmente se sugiere que los parámetros para una entidad IGES puedan ser especificados por posición o por nombre, existiendo valores default que sean declarados en la Sección Global.

CAPITULO III

EL SOFTWARE DE APLICACIONES DE UN SISTEMA CAD

III.1 Aplicaciones CAD.

Las Aplicaciones CAD son programas que cubren tareas específicas en el proceso del Diseño. En sus comienzos estas aplicaciones fueron creadas para facilitar la actividad del dibujante en dos dimensiones y aumentar su productividad. Con el tiempo, se añadieron nuevas funciones especializadas, tales como aumentar de tamaño porciones del dibujo, textos, especificaciones, soportar modelos en tres dimensiones y diferentes tipos de análisis. Un modelo es el objeto o la forma que uno se propone y desarrolla para la ejecución de un diseño.

Las Aplicaciones CAD se relacionan con diversas tecnologías cuyos acrónimos y descripciones son los siguientes:

CAE. - Ingeniería Asistida por Computadora - ("Computer Aided Engineering") es un ambiente que integra las tareas de modelado, análisis, simulación, modificación y documentación de un producto en un ambiente de Ingeniería.

CAM. - Manufactura Asistida por Computadora - ("Computer Aided Manufacturing"). Se refiere a la aplicación de la computadora en un proceso de manufactura y es la etapa que continua a un proceso típico de diseño de producto.

CAD/CAM. Trata los procesos de diseño, documentación y fabricación por computadora, utilizando una estación de trabajo gráfica interactiva y sistemas automatizados de producción.

CIM. - Manufactura Asistida por Computadora - ("Computer Integrated Manufacturing"). Integra todas las funciones del diseño y fabricación de un producto utilizando la computadora, extendiéndose inclusive a la administración del proyecto, su planeación, estudio de mercado así como su distribución.

La primera fase en un ambiente CAD se refiere al modelado o síntesis de un objeto. El Modelado ha evolucionado en las siguientes ramas:

- Modelo por Figura de Alambre en 2D o 3D.
- Modelo de Superficie.
- Modelo Volumétrico.
- Modelo Híbrido.

El Modelado por Figura de Alambre considera una definición topológica de vértices unidos por segmentos de curva pero donde no exista información de relación entre los elementos.

El Modelado de Superficie integra definiciones topológicas, basado en funciones que no tienen una representación matemática bien conocida (p.e. cuadráticas), que definen cuerpos abiertos, donde la integración total se logra por continuidades en los límites pero de la cual es posible obtener una mayor definición física.

El Modelado Volumétrico se basa en una representación lo más real posible de objetos, los cuales son susceptibles de ser "consultados" por cualquier pregunta de carácter físico. Los objetos que se modelan están típicamente limitados por superficies cuadráticas y su construcción puede realizarse principalmente por tres procedimientos:

a). El Booleano. Consiste en combinaciones booleanas (unión, intersección o sustracción) de primitivas de sólidos limitados por superficies cuadráticas.

b). Realizado ("Extrusión"). Parte de una figura cerrada en dos dimensiones, resaltándola con líneas o curvas para formar un objeto en 3D, esta misma figura también puede ser rotada alrededor de un eje.

c). Representación por Fronteras (B-rep). Este proceso se basa en la definición de un objeto a través de las fronteras que lo limitan, las cuales mantienen relaciones topológicas determinadas de antemano.

El Modelado Híbrido es la tendencia de los últimos años bajo la cual se integran diversas tecnologías de modelado en torno a un esquema de representación común.

Borgerson hace una comparación entre los tres primeros tipos de modelado según la tarea que se realice en las diferentes actividades de diseño de la Ingeniería (Diseño, Análisis, Control, Documentación e Ingeniería de Manufactura), para la implantación de un Sistema CAE. Este autor concluye que los modelos volumétricos ofrecen las mayores ventajas (este estudio no considero los modelos híbridos surgidos a mediados de los 80's); son los más eficientes al crear una representación lo más exacta y completa posible en la geometría del objeto (aunque esta demostrado que existe una gran cantidad de problemas no resueltos y por tanto estos modelos tendrán que evolucionar [ALV80]), además de ser la forma más productiva en el proceso de la creación del diseño al requerir menos pasos para definir una parte geométrica, aunque son los que requieren más recursos de la computadora [BOR80].

III.2 Estudio de Casos Específicos de Software CAD.

III.2.1 I-DEAS (Integrated Design Engineering Analysis System).

I-DEAS es un paquete de software CAE, desarrollado por SDRC que integra el modelado, diseño, análisis, pruebas y dibujo, a través de una base de datos común [SDRC86].

La Figura 3.1 muestra el sistema I-DEAS en forma modular.

Modulo de Bases de Datos

La base de datos es un conjunto de archivos en un formato estándar que permite la comunicación de los diferentes módulos del Sistema. Cada modelo es almacenado en tres archivos con diferente extensión según su contenido:

MF1 - Contiene la información no gráfica del modelo.

MF2 - Este archivo guarda información gráfica unicamente para acceso aleatorio.

MF3 - Reside información gráfica.

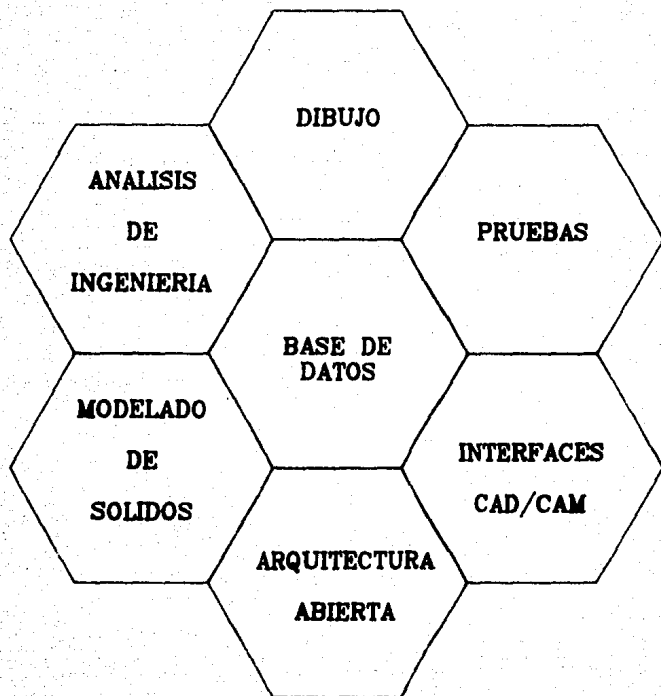


Figura 3.1 Módulos que conforman al Sistema I-DEAS

I-DEAS separa los modelos en diferentes archivos para manejar en una forma más eficiente los datos gráficos y no gráficos del mismo.

Modulo de Modelado de Sólidos (GEOMOD)

I-DEAS GEOMOD genera la geometría y el diseño para un sistema mecánico. Tiene la siguiente configuración.

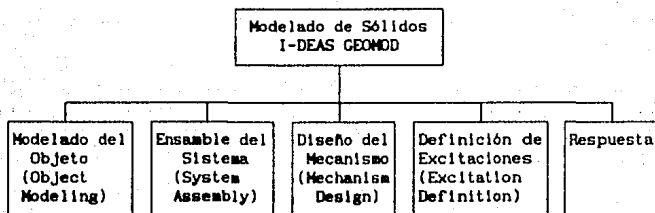


Figura 3.2 Configuración de I-DEAS GEOMOD

o Modelado del Objeto.

Este módulo genera objetos a partir de primitivas 3D y perfiles en 5 formas:

1. Objetos Básicos (Primitivas). Son objetos sólidos generados por funciones cuadráticas tales como cubos, esferas, cilindros, etc.

2. Perfiles ("Profiles"). Realizando o haciendo girar figuras en 2D alrededor de algún eje para obtener un objeto en 3D.

3. Cubriendo ("Skinning"). Unión de diferentes cortes (2D) mediante superficies de forma libre para formar un sólido.

4. Por partes elementales. Ensamblando un objeto como un conjunto de puntos, facetas, superficies y cuerpos. Esta técnica se conoce como Representación por Fronteras.

5. Modificando sólidos por:

- Operaciones booleanas, realizadas con otro sólido (p. e. unión, diferencia e intersección).
- Configuración (shaping) Deformando objetos para producir nuevas geometrías.
- Tweaking. Cambiando las medidas de un objeto al modificar su geometría: puntos, facetas y superficies.

o Ensamble del Sistema.

Es la unión de diferentes objetos para formar un sistema. Con este módulo se verifica si existe interferencia entre los volúmenes, se calculan sus propiedades de inercia y se puede cortar, mediante operadores booleanos, la geometría con un plano u otro objeto.

o Diseño del Mecanismo.

Este módulo analiza el funcionamiento cinemático y dinámico de un mecanismo, generando diferentes desplegados para mostrar el movimiento del cuerpo.

o Definición de Excitaciones.

Te permite definir movimientos y fuerzas para un mecanismo ya sea estableciendo un movimiento primario, funciones auxiliares o cargas gravitatorias.

o Respuesta.

Es la función de respuesta que genera el submódulo de análisis cinemático, proporcionándola por nodo (su posición, velocidad y aceleración).

Módulo de Análisis de Ingeniería

Esta formado por I-DEAS Supertab y por I-DEAS Systan. I-DEAS Supertab analiza por modelo de elemento finito, cargas y condiciones de frontera; también analiza su estática, dinámica, transferencia de calor de estructuras y componentes mecánicos. I-DEAS Systan no se encuentra instalado en DGSCA.

I-DEAS Supertab esta formado por:

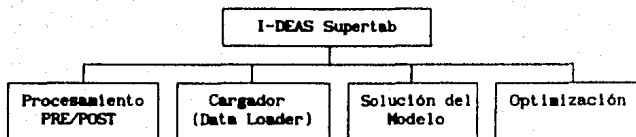


Figura 3.3 Configuración de I-DEAS Supertab

o PRE/POST Procesamiento.

Maneja tres formas para generar una malla (División de una superficie en elementos finitos para facilitar su análisis): manualmente, semiautomáticamente y "TRIQUAMESH", que es completamente automática. Genera y manipula nodos, elementos, tablas físicas y de materiales, además de analizarlos y verificarlos. Transfiere datos del archivo del modelo a la base de datos relacional PEARL y tiene un traductor de archivos ICES, Universal y de algunos de análisis (NASTRAN, ANSYS).

o Cargador de Datos.

Convierte los resultados de los paquetes de análisis en archivos universales de I-DEAS.

o Solución del Modelo.

Este módulo permite el análisis por elemento finito con condiciones de estática lineal, dinámica de modo normal, flujo potencial y respuesta en fuerzas y escribe los resultados en el archivo del modelo.

o Optimización.

Este módulo optimiza las propiedades físicas basándose en la definición de cargas y considerando fuerzas y/o restricciones en desplazamiento. Elimina las posibles redundancias de información en los modelos de elementos finitos.

Los módulos de Dibujo (I-DEAS Geodraw) y Pruebas (I-DEAS Tdas) no son descritos por no estar instalados.

Modulo de Interface CAD/CAM

Con I-DEAS se pueden recibir y transmitir diseños de o hacia otro sistema CAD/CAM a través de ICES. También soporta otras interfaces directas o paquetes de análisis.

Modulo de Arquitectura Abierta

A través del Sistema Manejador de Bases de Datos I-DEAS Pearl se puede obtener acceso a datos externos e integrarlos como una base de datos de I-DEAS, empleando el lenguaje de CAEDS, I-DEAL (Ver a continuación).

I-DEAS Pearl es un SMDB Relacional, que trabaja con cada módulo de I-DEAS en forma de menús, permitiendo realizar las siguientes funciones:

- a). Abrir y Cerrar una Base de Datos.
- b). Crear, copiar, renombrar y borrar tablas.
- c). Relacionar tablas.
- d). Analizar columnas para datos específicos.
- e). Despliegado de tablas con la opción de trazar gráficas XY.

o I-DEAL ("I-DEAS Language").

Es un lenguaje de programación interactivo que permite al usuario controlar la ejecución de un proceso. Con I-DEAL se pueden realizar operaciones con funciones matemáticas (LOG, SENO, EXP, etc.), concatenación de cadenas de caracteres y manejo de tres tipos de variables: las definidas por el usuario, las creadas por el programa para ciertas funciones (p.e. errores, análisis de tablas) y las de Pila que contiene los últimos valores listados de las variables. Los programas en I-DEAL pueden llamar a otros programas escritos en I-DEAL.

Archivos Externos.

o Archivos película.

Estos archivos permiten captar sólo las imágenes gráficas desplegadas en pantalla para que puedan ser proyectadas rápidamente sin necesidad de sintetizarlas nuevamente.

o Archivos de programas.

Contiene los comandos de los módulos de I-DEAS y/o instrucciones del lenguaje I-DEAL.

o Archivos Universales.

Son archivos que tienen un formato especial por cada elemento gráfico y no gráfico. Pueden ser leídos y generados por I-DEAS o por el usuario o programador de aplicaciones y se utilizan para hacer interfaces con otros programas o aplicaciones propias con el mismo CAEDS.

III.2.2 CADAM -Sistema de Diseño y Manufactura Incrementado por Computación Gráfica- ("Computer-graphics Augmented Design and Manufacturing System").

CADAM es un sistema gráfico distribuido por IBM para generar información gráfica en 2D, $2\frac{1}{2}$ D y 3D. En la DGSCA de la UNAM se tiene instalado CADAM-2D y se pueden construir diferentes entidades geométricas como puntos, líneas, arcos, círculos, cónicas y curvas libres.

Los archivos que contienen los dibujos están ordenados en tres formas:

- por grupo,
- subgrupo o usuario
- modelo.

Por grupo reúne toda la información de una organización particular y está formado por dos secciones: el índice maestro que contiene una lista de todos los usuarios y el índice por usuario, que contiene apuntadores a todos los dibujos asignados por el usuario. Subgrupo o usuario se refiere al conjunto de dibujos pertenecientes al usuario. Los modelos son los dibujos que son almacenados por separado en el archivo geométrico (ver Fig. 3.4).

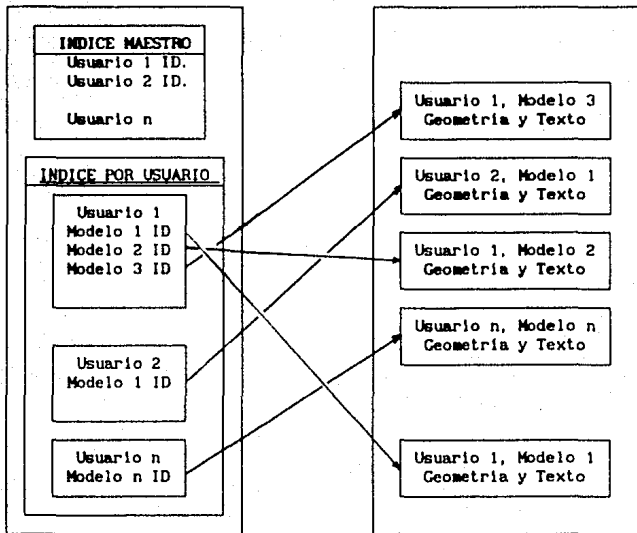
ARCHIVO INDICEARCHIVO GEOMETRICO

Figura 3.4 Estructura de Archivos del Sistema de Dibujo de CADAM

CADAM tiene un módulo de manejo de datos que permite hacer respaldo y recuperación de un archivo de dibujo completo o una parte de él. Actualiza el archivo de dibujo si se han hecho modificaciones como incrementar o borrar usuarios, cambiar el tamaño del archivo del dibujo, intercambiar modelos entre grupos diferentes y borrar modelos. Genera reportes del archivo de dibujo y de archivos respaldados y permite instalar seguridad por medio de una palabra clave para acceder el archivo de dibujo.

INTERFACE GEOMETRICA ó "GEOMETRY INTERFACE" es un programa que permite acceder y modificar la base de datos de CADAM. Esta dividido en cuatro partes según la función que realice:

o CADCD. Permite crear o modificar elementos geométricos en forma batch a los archivos de CADAM.

o CADET. Descompone un modelo de CADAM en sus diferentes componentes en forma batch.

o CADMACGM. Permite ejecutar en forma interactiva programas para generar geometría (CADCD) escritos por el usuario.

o SCLIB. Produce reportes de subgrupos y nombres de los dibujos contenidos en la base de datos.

"Geometry Interface" se empleo en este trabajo, para la extracción de información de CADAM, constituyendo la primer fase para la conversión a información aceptada por CAEDS con TGCAD (ver Capítulo V).

CAPITULO IV

Integración de Sistemas CAD con Bases de Datos

IV.1 Importancia de esta integración.

Ante el aumento en la cantidad y complejidad de los sistemas CAD y la necesidad de comunicarlos, se planteó, a principios de la década pasada la urgencia de integrarlos. En general estos sistemas se han desarrollado como paquetes de procesos aislados, con su propia estructura de información aún en aplicaciones idénticas, lo que ha provocado el tener almacenada la misma información para cada sistema bajo estructuras de datos distintas. Actualmente muchos usuarios encuentran dificultades, ya sea porque quieren cambiar de sistema CAD o cuentan con más de un sistema CAD y los diseños realizados ya no son soportados en el nuevo sistema, teniendo que realizar programas especiales que los comuniquen o hacer una transferencia manual de datos con las equivocaciones y pérdida de tiempo que esto implica.

La situación anterior motivo el establecimiento de estándares orientados al intercambio de información entre sistemas CAD, siendo uno de los primeros intentos IGES [IBM_86], que traduce de un formato de dato particular a un formato específico estándar, para que otro sistema pueda tomarlo. Este método, tiene en sí mismo sus propias ventajas y desventajas como se vió en el Capítulo II, además de no lograr eliminar la redundancia de datos, porque cada programa aplicación tiene sus propios archivos y los datos dependen del programa que lo ha generado.

Lo anterior podría ser atacado de una forma más adecuada a través de una base de datos común a todas las aplicaciones (Ver Figura 4.1). Esto lograría disminuir la redundancia y el espacio de almacenamiento requerido; los datos del diseño reflejarían una estructura más ordenada; se mantendría más fácilmente la consistencia y los diseñadores podrían trabajar directamente con

los datos, aumentando su productividad y reduciendo su costo general. Esta base de datos no debe pensarse sólo como un depósito de datos, sino más bien como una herramienta de diseño al ofrecer una forma más poderosa de manipulación de datos y lenguajes de programación de más alto nivel.

En este capítulo se definen los conceptos básicos de bases de datos para después hacer un análisis de las características deseables de un SMDB especial para cubrir las necesidades de un ambiente CAD.

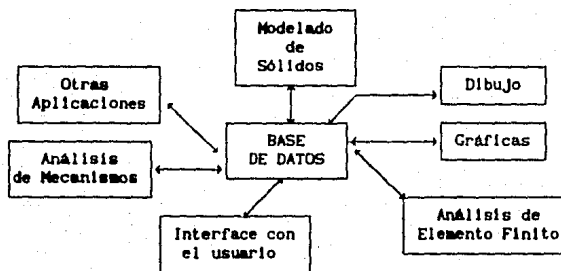


Figura 4.1 Sistema CAD con Bases de Datos Centralizada.

IV.2 Conceptos de Bases de Datos.

Una base de datos es una colección de datos interrelacionados que representan una porción de la realidad [BUCHR81]. El Sistema Manejador de Bases de Datos (SMDB) es el programa que se encarga de realizar el acceso a la base de datos de acuerdo a una estructura ya definida.

En general un SMDB esta conformado por un Lenguaje de Definición del Dato (LDD), un Lenguaje Manejador del Dato (LMD) y un conjunto de utilerías. El LDD permite definir la organización de los datos, el LMD provee los mecanismos para traer los registros de la estructura definida con el LDD y las utilerías se componen de un lenguaje de consulta a la base de datos, procedimientos de respaldo y recuperación, seguridad y generadores de reportes.

IV.2.1 Arquitectura de un SDBD.

La arquitectura de un SDBD es la configuración de sus componentes, siendo importante el grado de independencia de datos que esta provea. Independencia de datos se refiere a la facilidad para aislar los programas de aplicación de su estructura de almacenamiento y su estrategia de acceso [DATE81].

Las arquitecturas que veremos son la ANSI/X3/SPARC, CODASYL 71 y CODASYL 78 [KORI83].

IV.2.1.1 Arquitectura ANSI/X3/SPARC.

Esta dividida en tres niveles (Figura 4.2):

- Modelo Externo. Es la representación de los datos vista por los usuarios ó programas de aplicación (Esquema Externo).
- Modelo Conceptual. Son las estructuras lógicas que ayudan al usuario a describir su organización de datos (Esquema Conceptual).
- Modelo Interno. Es el que define la implementación física (estructura de almacenamiento y técnicas de acceso) del modelo conceptual (Esquema Interno).

Esta arquitectura no tiene modelo de dato.

IV.2.1.2 Arquitectura CODASYL 71.

Esta Arquitectura como se muestra en la Figura 4.3 esta formada por:

- Esquema. Es una descripción completa de la base de datos a través de un Lenguaje de Definición de Datos (LDD). Es una mezcla del modelo conceptual y el modelo interno de ANSI/X3/SPARC.
- Subesquema. Es un subconjunto del esquema con los registros y relaciones que el usuario desea ver. Sería el modelo externo de ANSI/X3/SPARC.

El modelo de dato de esta arquitectura es la red.

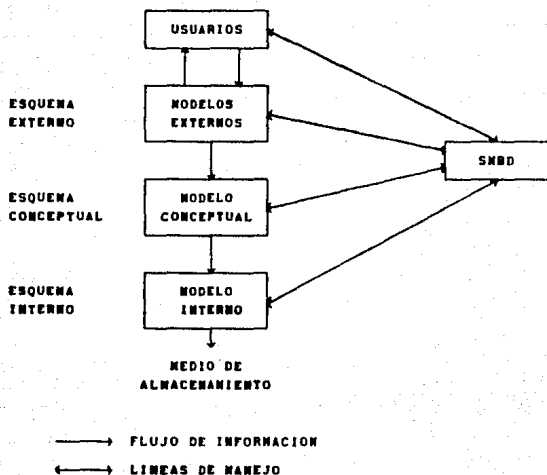


Figura 4.2 Arquitectura ANSI/X3/SPARC

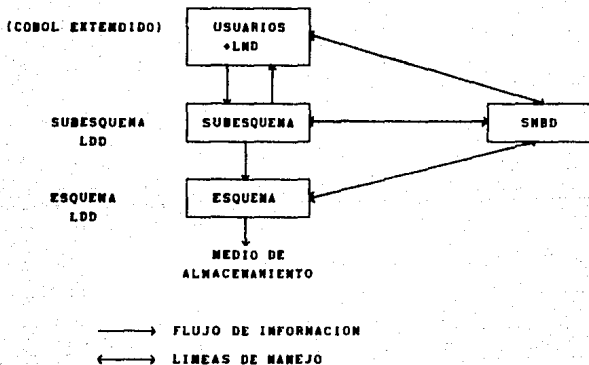


Figura 4.3 Arquitectura CODASYL 71.

IV.2.1.3 Arquitectura CODASYL 78.

En esta versión de CODASYL se extrajo del esquema lo referente a la implementación del almacenamiento para agruparlo en un nuevo esquema (del modelo interno) con su lenguaje propio (Lenguaje Descriptivo del Almacenamiento del Dato LDAD). La Figura 4.4 ilustra este cambio.

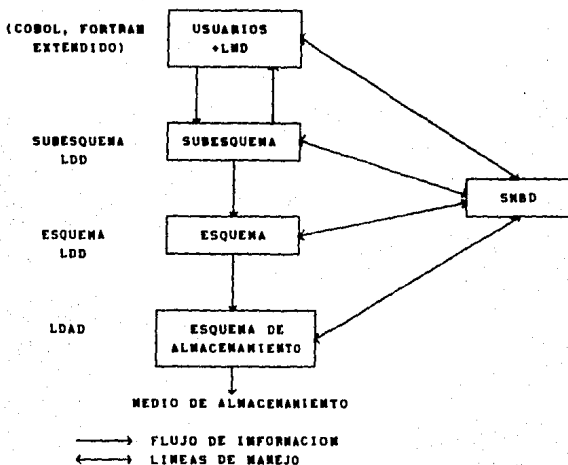


Figura 4.4 Arquitectura CODASYL 78

IV.2.2 Modelos de Datos.

Las bases de datos se identifican por el modelo de datos que soportan. Un modelo de datos permite un nivel más alto de abstracción para visualizar y representar mejor la realidad en las operaciones lógicas y físicas que se puedan realizar sobre la base de datos. Aquí solo veremos los tres modelos clásicos; el relacional, el jerárquico y el reticular [DATE81]. Estos modelos pueden comportarse de acuerdo a la teoría de gráficas como el jerárquico y el reticular o sustentarse en las operaciones de la teoría de conjuntos como el relacional.

IV.2.2.1 Modelo Relacional.

Una base de datos relacional esta formada por relaciones y se representan en forma de tablas de dos dimensiones, en las que las columnas corresponden a los dominios (atributos) y las filas serian las ocurrencias (ver Figura 4.5). Uno o más atributos forman una llave para acceder los otros atributos pertenecientes a una ocurrencia en particular [BUCH81].

PUNTOS	COORDX	COORDY
1	2.3	1.0
2	6.0	7.9
3	2.2	4.8
4	9.7	0.0
:	:	:

LINEAS	PORIG	PDEST	TIPOL
1	2	1	0
2	1	4	1
3	3	2	0
4	3	1	0
:	:	:	:

Figura 4.5 Modelo Relacional

Organizar la información en forma tabular hace que algunas veces se repitan los valores de algún atributo, lo que resulta en una redundancia de datos, este problema puede ser asminorado por el proceso de normalización. En un SDBD relacional los modelos conceptual y externos son representados por relaciones normalizadas.

Normalización es un proceso de descomposición de relaciones que permite llegar a un conjunto de relaciones que cumplen con las siguientes condiciones:

- Cada valor en una relación debe ser atómico.
- No deben presentarse anomalías de inserción y eliminación.
- Relaciones existentes no deben afectarse al crecer la Base de Datos.

IV.2.2.2 Modelo Jerárquico.

Los modelos jerárquicos tienen una estructura de árbol en la que los nodos o registros están conectados de forma que un hijo pueda estar conectado sólo a un padre (asociación uno a muchos) y

no pueden conectarse entre hijos.

Estas conexiones pueden interpretarse como apuntadores o por contigüidad física. Este modelo aunque eficiente para procesamiento, provoca redundancia y su modelo lógico se vuelve inflexible (ver Figura 4.6).

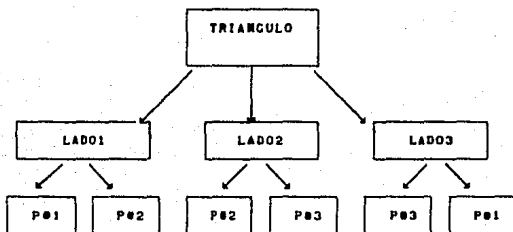


Figura 4.6 Modelo Jerárquico

IV.2.2.3 Modelo Reticular.

Este modelo, a diferencia del Jerárquico, sí permite que un padre tenga varios hijos y está restringido a que las asociaciones sean de dos tipos uno-a-muchos y muchos-a-uno.

Los accesos a los registros pueden ser por varios puntos y el borrado de un registro no implica necesariamente el borrado de los registros conectados.

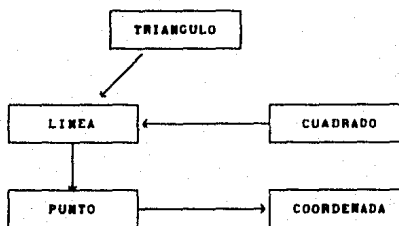


Figura 4.7 Modelo Reticular

IV.2.2.4 Comparación entre los Modelos de Datos.

Aunque los tres modelos son equivalentes, en cuanto a su flexibilidad para modelar seguirían este orden: el más flexible sería el Relacional seguido por el Reticular y después el Jerárquico. En cuanto a su eficiencia para procesar seguirían el orden opuesto. El modelo Relacional es completamente general ya que uno puede mapear cualquier estructura jerárquica o de redes en una estructura relacional.

IV.3 Bases de Datos CAD.

Conforme ha mejorado la ingeniería de diseño, aumenta en tamaño y complejidad la información que maneja. Dibujos, Catálogos de Productos, Libros de Especificaciones, Tablas, Programas, son datos que necesitan estar reunidos y organizados para manipularlos de una forma más eficiente. La tendencia actual en los sistemas CAD referente al manejo de información es incorporar un SMD que además de organizar la información, incluya rutinas de respaldo y recuperación en caso de fallas, genere reportes de su contenido y tenga algún lenguaje de consulta interactiva [CADR87, IDEAB6].

Los SMD actuales, por su orientación comercial, no cubren las necesidades en el manejo de la información de un sistema CAD. Los Sistemas Manejadores de Bases de datos CAD se encuentran en una etapa de investigación, teniendo mucha aceptación entre los diseñadores la Arquitectura ANSI/X3/SPARC con su configuración de tres esquemas, permitiendo una mayor independencia física de datos, a diferencia de CODASYL 71, que al ser el subesquema un subconjunto del esquema, el cual combina los modelos conceptual e interno, hace que tenga poca aplicación en sistemas CAD. CODASYL 78, ha tenido más éxito al separar del esquema un esquema de almacenamiento, además de que el lenguaje FORTRAN incluye las características de un LMD y se pueden definir estructuras de datos reticulares complejas.

Koriba propone un SMDO hipotético que tuviese lo mejor de cada aproximación. Este SMDO tendría una arquitectura ANSI/X3/SPARC con un modelo externo y conceptual basado en la aproximación relacional y con un modelo interno sustentado en la aproximación reticular [KORI83].

Ejemplos de estos intentos de SMDOs para CAD a nivel experimental son:

PHIDAS. Forma parte del sistema PHILIKON para diseñar 2 1/2 dimensiones, su arquitectura esta basada en ANSI/X3/SPARC teniendo un esquema conceptual y un subesquema externo basado en el modelo reticular de CODASYL [FISCH79].

TORNADO. Conjunto de subrutinas programadas en FORTRAN IV basado en CODASYL 78, maneja estructuras de datos de red muy complejas, maneja directamente asociaciones muchos-a-muchos, no maneja el registro corriente para simplificar la programación [ULFS81].

INTERGRAD. Escrito en FORTRAN y PASCAL, esta basado en el modelo relacional. Aunque maneja dibujo de dos dimensiones puede ser modificado para dibujos de tres dimensiones, maneja dos lenguajes el "Picture Definition Language (PDL)" y el "Picture Manipulation Language (PML)" [PATN82].

ARDBID. Base de datos relacional escrita en FORTRAN y PASCAL, descompone la base de datos en tres, una almacena la información pictórica en 2D, otra en 3D y otra que almacena la información no pictórica alfanumérica, con el objeto de reducir el tiempo de acceso. Cada usuario tiene un código de acceso para la protección en lectura y escritura. Mantiene un directorio de relaciones con las descripciones (nombre de atributo, tipo de dato, tamaño de campos, código de protección, etc) de cada relación [SHEN83].

La tendencia actual en los SMDOs para CAD tiene preferencia por el modelo Relacional al facilitar la representación de las asociaciones tan complejas entre los elementos de un Sistema CAD.

A nivel comercial gran parte de los proveedores ofrecen un SMDB relacional como una herramienta en la organización de los programas de análisis, para facilitar la realización de interfaces a otros programas de análisis, como es el caso de SUPERTAB de CAEDS (SDRC). La Compañía ATP esta trabajando en un SMDB que maneje toda la información incluyendo los elementos gráficos y pueda manejar la información generada con CADAM, CATIA y CAEDS. IBM también trabaja en un SMDB para CAD, utilizando el SMDB relacional DB2 que corre bajo el sistema operativo MVS (CADR87).

IV.3.1 Características de un SMDB CAD.

A continuación se presenta un resumen de las características que deben tener los SMDBs para CAD para que puedan manejar adecuadamente la información de diseño.

Manejo de objetos complejos

En la ingeniería de diseño, la información almacenada puede consistir desde unos pocos campos hasta centenas de ellos, formando asociaciones muy complejas. De esta complejidad se busca una forma de organizar los datos conceptualmente. Además de los modelos de datos tradicionales (Jerárquico, Reticular y Relacional) en bases de datos CAD se ha desarrollado el concepto de Jerarquía de abstracción, esto es, se descompone un sistema de ingeniería en subsistemas de tipos de datos abstractos, donde cada subsistema es descrito como una secuencia de registros que se definen aumentando el nivel de detalle, entendiéndose por tipos de datos abstractos la definición de la estructura de datos y las operaciones permitidas sobre estos datos.

La Jerarquía de abstracción permite que un elemento de diseño sea parte de más de un subsistema, manteniendo relaciones bien estructuradas que pueden utilizarse para comunicación y diseño [EAST81].

Control de Concurrencia para Transacciones Legras

En ambientes CAD o de ingeniería de diseño, una transacción puede ser un programa de aplicación en alguna tarea de diseño que puede durar horas o incluso días, a diferencia de las transacciones de bases de datos comerciales donde por poco tiempo se reserva para leer o escribir un registro o área de otros procesos. La duración de las transacciones en CAD hacen que el período de inconsistencia sea grande lo que provoca, la necesidad de crear un sistema diferente de control de concurrencia.

Buchmann propone una configuración de las bases de datos para CAD formada por una base de datos global del proyecto o BD/P, una base de datos del área de trabajo o BD/AT y bases de datos para control, diccionario, plantillas y símbolos, catálogos y restricciones. Con esta configuración se extraería parte de un diseño de la BD/P para colocarla en la BD/AT y una vez que se haya modificado la BD/AT se reintegra a la BD/P en base a la bitácora que lleva cada sistema manejador de las BDs/AT, no permitiendo que se reintegre otra BD/AT hasta que termine su propio proceso [BUCH86].

Control de Versiones y Propagación de Actualizaciones

Durante el diseño de un objeto en CAD aparecen diferentes alternativas como posibles soluciones a un problema de diseño. Se pretende que con una base de datos se lleve un mejor control sobre las versiones que se vayan generando, documentando al autor de la modificación así como el nombre del responsable y fecha en que la aprobo y facilitar la recuperación de las diversas versiones.

En el modelo de Buchmann la propagación de actualizaciones se referiría a la reintegración de la copia extraída BD/AT a la base de datos maestra BD/P, actualizando las bases de datos afectadas y notificando a los que se vieran afectados.

Independencia Física de Datos

Los sistemas CAD se encuentran en continuo crecimiento por lo que es de esperar que su estructura lógica cambie y el mantenerla separada de su implementación física hace que se reduzca el excesivo mantenimiento que se necesitara en un diseño complejo. El modelo relacional tiene esta característica [BITT83].

Extensión y Modificación Dinámica del Esquema

El esquema se refiere a la descripción del modelo de la base de datos, definiendo los objetos y las asociaciones entre ellos. Si se tuviese un esquema que cubriera todas las posibles combinaciones de diferentes tecnologías CAD, además de lo difícil de su realización, se volvería ineficiente tener presentes apuntadores y objetos que no fuesen necesarios y tendrían que tomarse en cuenta en los mantenimientos.

Staley escribe que debido a las necesidades de las Base de Datos CAD de manejar diferentes vistas de la misma, hacen necesario que los esquemas de CAD sean lo bastante flexibles para soportar modificación y extensión, siendo esencial que se haga en forma dinámica, sin requerir la recompilación de esquemas, ni descargar y cargar la base de datos [STAL86, LAFU79].

Procesamiento de Bases de Datos Distribuido

El diseño de Ingeniería se realiza en diferentes estaciones de trabajo con minicomputadoras o microcomputadoras. En una base de datos distribuida, la base de datos es almacenada en una red de computadores, de tal forma que cada usuario ve la base de datos como una base de datos central sin tener que estar enterado de la ubicación geográfica donde se encuentran las partes de la base de datos con las que se interactúa. Esta forma de procesamiento comunicaría diferentes estaciones de CAD, compartiendo una base de datos.

Las desventajas en esta forma de procesamiento son los problemas en la transferencia de la información pudiendo existir interferencia (ruido), que afectaría la integridad de los datos; el mantenimiento y costos serían mayores a una base de datos centralizada y aumentaría el número de Administradores de Bases de Datos por cada nodo de la red [KOR183].

Otras características importantes en un SMBD especial para CAD serían que los modelos de datos soportaran tipos de datos de ingeniería como vectores y matrices, que manejaran cadenas de caracteres de longitud diferente (notas, observaciones) y datos pictográficos que se almacenarían secuencialmente para que el proceso de despliegue sea más rápido. Esta base de datos deberá ser compartida por diferentes programas de aplicación en forma interactiva de acuerdo con el proceso del diseño, soportar una interfaz a lenguajes de programación de alto nivel y un lenguaje de consulta interactiva para consultarla.

También se incluirían facilidades de Bases de Datos comerciales como el respaldo y recuperación de la información, manejo de seguridad en el acceso a la base de datos, estadísticas de eficiencia para la afinación del SMBD y de las bases de datos y un diccionario de datos (documentación de la información almacenada en la base de datos).

CAPITULO V

TRANSFERENCIA GEOMETRICA CADAM-CAEDS (TGCAD)

V.1 Planteamiento General.

El intercambio de información entre diferentes sistemas CAD, es una de las principales actividades dentro de la comunidad de CAD y es así como se pensó en la estandarización de interfaces gráficas (GKS, PHIGS, ...), en la definición de datos gráficos en un formato estandar (IGES) y en programas especiales como el que ocupa este capítulo.

TGCAD es un conjunto de programas que convierten elementos geométricos básicos extraídos de un modelo realizado con CADAM en 2 y 2¹/₂D en un modelo bidimensional aceptado por CAEDS como se muestra en la Fig 5.1.

Fue desarrollado en el Laboratorio de CAD de Cómputo Académico en la UNAM, formado por cuatro estaciones de diseño de la familia IBM 5080 comunicadas a la computadora IBM 4381 via un controlador de comunicaciones, con la que se comparten los recursos del sistema. La 4381 mantiene la información de los tres productos de diseño instalados (CAEDS, CADAM y CBDS) en forma de una base de datos centralizada bajo el sistema operativo VM.

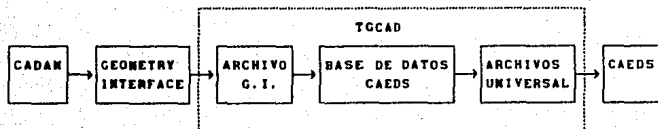


Figura 5.1 Proceso de Conversión CADAM-CAEDS.

Este programa pretende ser una ayuda en el diseño, debido a que es más fácil modelar bidimensionalmente en CADAM porque ofrece más facilidades que CAEDS y una vez que se tenga el modelo en 2D en CAEDS poderlo convertir a 3D por técnicas de realizado, rotación, combinación híbrida o por interpolación de superficies. De esta forma no se tiene que extraer datos de CADAM e introducirlos manualmente en CAEDS, no se generarían errores, reduciendo el tiempo y costo en la transferencia de la información.

TGCAD es fácil de usar, trabaja a base de menús, despliega mensajes de errores comunes y ofrece la opción de conservar una base de datos con la información del modelo. De esta forma el diseñador tiene la opción de consultar una base de datos relacional de uso sencillo, con información referente a su diseño en forma de tablas y si se desea utilizar herramientas del manejador como seleccionar cierta información de las tablas, realizar estadísticas con las columnas y desplegarlas en forma de barra o por curva, con textos. Con ayuda del lenguaje I-DEAL se pueden desarrollar otras aplicaciones en forma integrada con la Base de Datos, como reportes, que pueden ser de utilidad en la documentación de un proyecto o interfaces a otros programas.

V.2 Estructura de TGCAD.

TGCAD esta formado por los siguientes programas:

TGRESOLV.FORTRAN. Este programa realiza la extracción de modelos de CADAM a través de Geometry Interface, además de preprocesar y formatear esta información para que sirva como archivo de entrada al programa TGCAD.PRG.

TGCAD.EXEC. Es un procedimiento que ejecuta los programas TGDIB y TGRESOLV en una cuenta de CADAM, transfiriendolo a la cuenta de CAEDS que se le indique.

TGDIB.FORTRAN. Es un programa que pregunta por el nombre del modelo y al usuario de CADAM al que pertenece, para armar el archivo de entrada necesario para ejecutar TGRESOLV.

TGCAD.PRG. Es el programa principal y esta escrito en I-DEAL. Se encarga de leer el archivo generado por TGRESOLV cargandolo en una base de datos relacional, después la procesa para generar un archivo con la definición del dibujo en formato universal, para que CAEDS pueda leerlo y cargarlo como "working sets", manteniendo, si se desea, una base de datos del modelo. Esta formado por seis subprogramas:

- o TGMENU. Pregunta al usuario en forma de menús los datos necesarios para procesar el modelo y si se desea tiene un archivo de ayuda con una explicación breve de cada subprograma.

- o TGCREATA. Genera la base de datos donde se depositará la información.

- o TGLECTUR. Realiza la lectura de elementos geométricos del archivo generado por TGRESOLV y los introduce en la respectiva tabla de la base de datos.

- o TGPROCES. Procesa y ordena la información de las tablas para que estén listas para el siguiente programa.

- o TGESCRIB. Escribe con la información de la tablas, un archivo en formato universal.

- o TGLEEUNV. Lee el archivo universal y finaliza el programa.

El proceso de comunicación CADAM-CAEDS a través de los programas anteriores se muestra en la figura 5.2.

TGCAD trabaja en forma interactiva con el usuario, transfiriendo cada vista o detalle a un "working set" diferente de CAEDS. Realiza diferentes procesos en la conversión de las entidades geométricas de CADAM, por ejemplo los arcos circulares, arcos elípticos y elipses se simulan con curvas, calculando los puntos que conforman la figura. La distribución de los puntos para los arcos elípticos y las elipses, se calcularon en base a ecuaciones paramétricas [SMIT79] y el número de puntos fue según la excentricidad de las mismas. La entidad multipunto se simuló a través de líneas rectas y las curvas mayores de 25 puntos fueron particionadas para que las soporte CAEDS. Los siete tipos de líneas que soporta CADAM se convierten a las dos permitidas por CAEDS (discontinuas ó continuas). Las figuras son transferidas con el color original de CADAM.

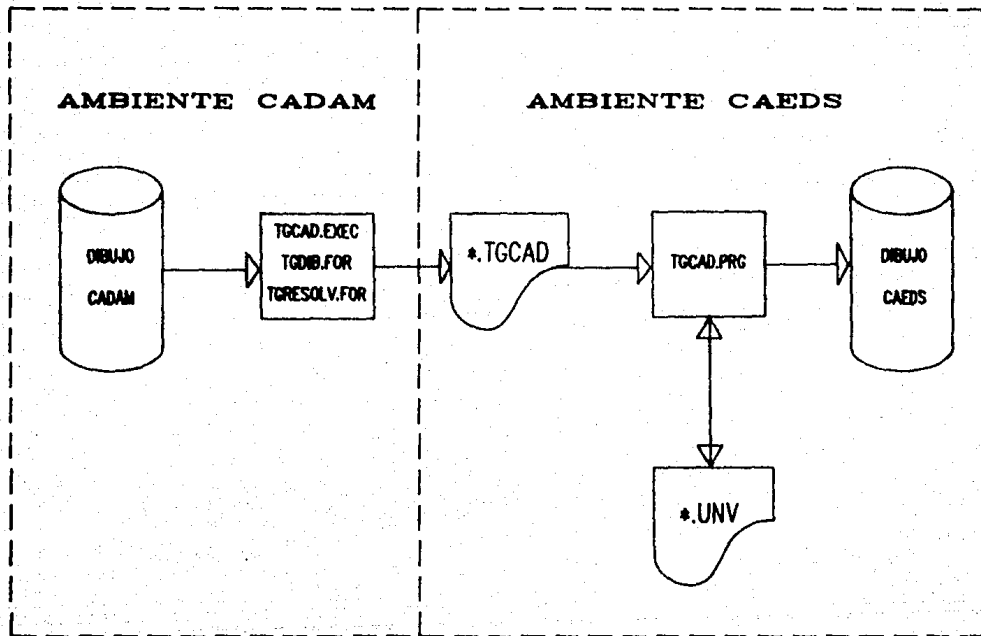


Figura 5.2 Secuencia de Ejecución de TGCAD.PRG

COPR 1989 CADAM FILES

VIEW PV 1.0 W 10.0

SELECT MENU

PN, FN= PLANO DE CU , CU

GROUP=CAD USER=CDUNAM

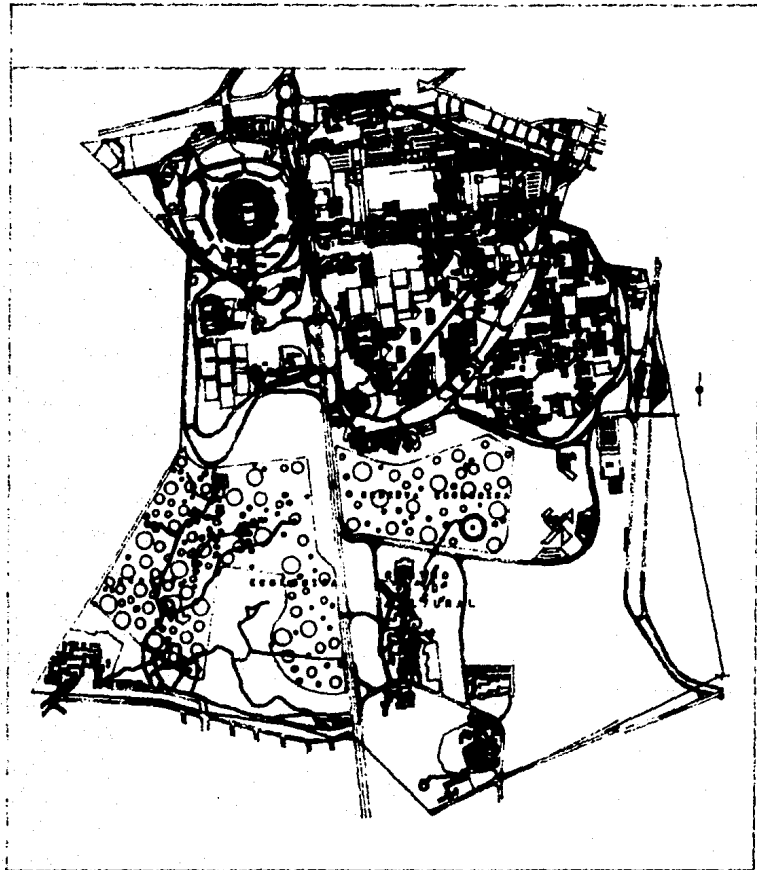


Figura 5.3 Plano de CU generado con CADAM.

V.3 Ejemplificación.

En esta sección se muestra un ejemplo de demostración del sistema TGCAD, asumiendo que el usuario ejecutor esta familiarizado con los productos CADAM, CAEDS y el sistema operativo VM.

Para ejemplificar la transferencia geométrica de CADAM a CAEDS se seleccionó el plano de la Ciudad Universitaria (Figura 5.3) generado en CADAM. Durante la demostración de la secuencia de comandos necesarios para ejecutar TGCAD, el texto subrayado corresponde al introducido por el ejecutor del programa y la letra reducida corresponde a los mensajes generados por el sistema operativo y el programa TGCAD.

El primer paso es acceder CADAM y no ejecutar el "profile" (esto se logra con el comando "ACCESS (NOPROF" de VM), después se ejecuta el EXEC, TGCAD, este pregunta por el subgrupo donde esta el modelo, el nombre del mismo, su extensión, el nombre del archivo donde se pondra la definición del dibujo y por último la máquina de CAEDS a donde se enviará y donde se continuará el proceso (ver Figura 5.4).

```
TGCAD
R(194) R/O
EXECUTION BEGINS...
PROPORCIONE EL SUBGRUPO O USUARIO (EJ. TRAIN)
COUNAM
PROPORCIONE EL NOMBRE DEL DIBUJO (EJ. UNIVERSIDAD)
PLANOCU
PROPORCIONE LA EXTENSION DEL DIBUJO (EJ. G000)
DIB
EL NOMBRE DEL ARCHIVO DESTINO DEL DIBUJO
PLANOCU
EL NOMBRE DE LA CUENTA DE CAEDS QUE RECIBIRA EL ARCHIVO
CAEDS243
EXECUTION BEGINS...
CADSEG SEGMENT BOUNDARIES 00870000 - 00877777
CADAM SEGMENT NOW LOADED
EXECUTION BEGINS...
DASD 194 DETACHED
FILE PLANOCU TGCAD A1 SEND TO CAEDS243 AT UNAMVNI ON
02/08/80 14:38:24
READY; T=0.08/1.14 14:38:24
```

Figura 5.4 Ejemplo de despliegue de TGCAD desde CADAM

Una vez que el dibujo fue transferido a la cuenta de CAEDS, se desfirma de CADAM ("logoff") y se accede CAEDS sin ejecutar el "profile", se recibe el archivo del area de "READER" (con el comando RL, se posiciona en el archivo y se oprime la tecla PF9). Después, se ejecuta el "profile" ("PROFILE" y se da entrada), aparecen las pantallas de CAEDS (el nombre del modelo en CAEDS, descripción, unidades, etc), se conecta con el módulo de Modelado de Objetos en la pantalla de Modelado de Solidos. En esta pantalla se ejecuta el programa TGCAD.PRG con el siguiente comando:

/MF PR R TGCAD

Aquí, en forma de menús, solicita la información para ejecutarse, en general se puede elegir entre teclear la opción y dar entrada, escoger la opción default (la que esta entre paréntesis) o con la tableta digitalizadora posicionarse y hacer un "pick". La primera pantalla que aparece es la que se muestra en la Figura 5.5, esta ofrece dos opciones, DESCRIPCION explica los datos necesarios para la ejecución del programa (ver Fig. 5.6) y EJECUCION comienza el proceso.

TGCAD TRANSFERENCIA DE INFORMACION GRAFICA CADAM-CAEDS

U.N.A.M. 1990

D - DESCRIPCION

E - EJECUCION

I - BACKUP

S - ABORT

OPCIONES DEL PROGRAMA (EJECUCION) * E

Figura 5.5 Menu Principal de TGCAD.

PROGRAMA T G C A D

TCCAD ES UN PROGRAMA ESCRITO EN IDEAL QUE TRADUCE LOS DIBUJOS 2D GENERADOS EN CADAM A DIBUJOS 2D SOPORTADOS POR CAEDS.

LOS DATOS QUE PRECUNTA ESTE PROGRAMA SON:

- *EL NOMBRE DE LA BASE DE DATOS: SE REFIERE AL NOMBRE DE LA B.D. DONDE DEPOSITARA TODA LA INFORMACION DEL DIBUJO PARA PROCESARLA.
- *EL NOMBRE DEL ARCHIVO DE ENTRADA DE TCCAD: ES EL ARCHIVO QUE GENERO TCCAD EXEC Y CONTIENE LA INFORMACION DEL DIBUJO.
- *EL NOMBRE DEL ARCHIVO RESULTANTE: ES EL ARCHIVO EN FORMATO UNIVERSAL QUE GENERARA TCCAD Y CONTIENE EL DIBUJO SOPORTADO POR CAEDS.

LOS ARCHIVOS QUE GENERA ESTE PROGRAMA SON:

- *.UNY ARCHIVO DEL DIBUJO EN FORMATO UNIVERSAL.
- *.POB BASE DE DATOS QUE CONTIENE LA INF. DEL DIBUJO.

NOTA:

SI LOS DIBUJOS SON MUY GRANDES SE RECOMIENDA PROCESARLOS UNO POR UNO.
EN CASO DE QUE EXISTA UN PROBLEMA EN LA TRANSFERENCIA SE DEBERA REEJECUTAR TCCAD DESDE EL PRINCIPIO.

Figura 5.6 Pantalla que describe la forma en que corre TCCAD

Una vez introducida la opción de EJECUCION, comienza a solicitar el nombre de la base de datos, el archivo de entrada generado en CADAM, el nombre que tendra el archivo en formato universal y el sistema de unidades que tendra el modelo (Figura 5.7).

NOMBRE DE LA BASE DE DATOS(TGBASEDA) # PLANODB
 NOMBRE DEL ARCHIVO DE ENTRADA DE TCCAD # PLANOCU
 NOMBRE DEL ARCHIVO RESULTANTE # PLANOUNV

.....
 SI-METRIC_ABS_(SI)
 BG-BRITISH_GRAV
 NG-METRIC_GRAV
 BA-BRITISH_ABS
 MH-MODIFIED_SI_(MM)
 CH-MODIFIED_SI_(CM)
 IN-BRITISH_GRAV_(MOD)
 GN-METRIC_GRAV_(MOD)
 US-USER_DEFINED
 AU-AYUDA_UNIDADES

SISTEMA DE UNIDADES QUE DESEA PARA EL MODELO(METRIC_ABS_(SI))# SI

Figura 5.7 Secuencia para ejecutar TCCAD.

La siguiente etapa es la creación de tablas y la lectura de los elementos geométricos, al terminar esta ofrece un recuento por cada entidad (ver figura 5.8). Continúan los procesos de conversión de entidades y la escritura del archivo en formato universal desplegando mensajes referentes según el proceso que se este ejecutando. Para finalizar pregunta si se desea conservar la base de datos para consulta posterior. Una vez que termina la ejecución de TGCAD se procede a leer el archivo en formato universal con la siguiente secuencia de comandos:

```
/TASK Q  
MANAGE READ UNIVERSAL  
/TASK W  
/GET DIRECTORY
```

A partir de aquí se escoge el "working set" que se desea visualizar, si se desea ir juntando los diferentes "workings sets" se realiza por medio del comando "append working set" en vez de "get".

```
RESUMEN DE ELEMENTOS GEOMETRICOS EXTRAIDOS DEL ARCHIVO:  
PLANOUC  
E INTRODUCIDOS EN LA BASE DE DATOS  
PLANOUC  
NUMERO DE DIBUJOS LEIDOS: 1  
NUMERO DE ELEMENTOS :  
PUNTOS : 42  
LINEAS : 6106  
CIRCULOS : 50  
ARCOS : 38  
ELIPSES : 28  
CURVAS : 148
```

Figura 5.8 Despliegue del total de elementos leídos.

El modelo pasado finalmente a CAEDS es como el que se muestra en la figura 5.9. Una vez transferido el modelo, podemos por ejemplo, seleccionar una parte y realizar un realzado, para obtener una figura en 3D (Ver Figura 5.10).

CAEDS V2R1M1: Object Modeling

31-MAR-90 15:59:46

DATABASE: PLANO DE CU
VIEW: NO STORED VIEW
TASK: WORKING SET
PROFILE: NO CURRENT PROFILE EXISTS

UNITS - SI
DISPLAY: NO STORED OPTION
WORKSET: 3-PLANUS_1-9 (modified)
SKIN: NO CURRENT SKIN GROUP EXISTS

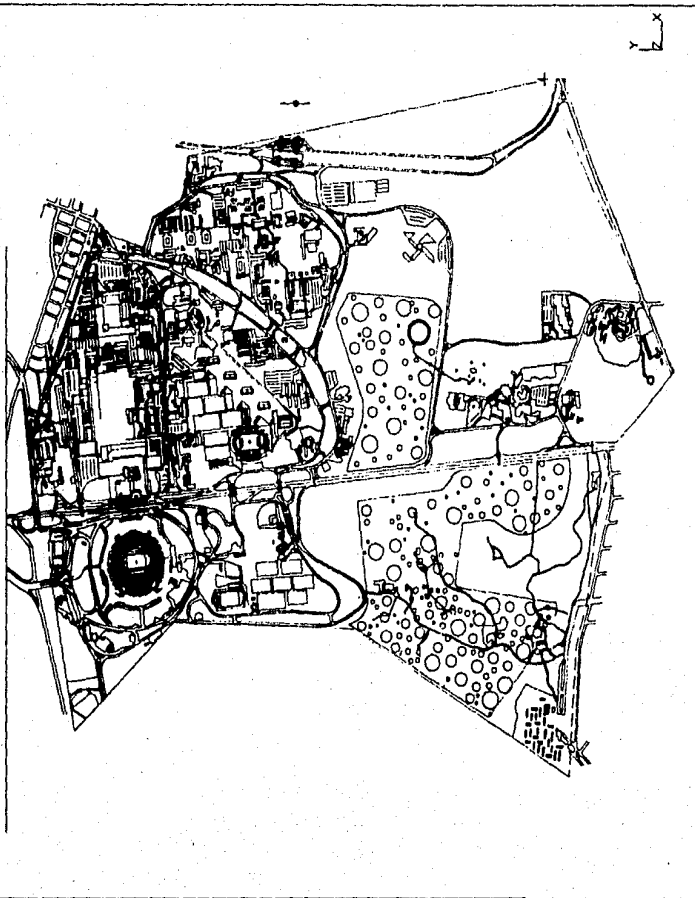


Figura 5.8 Modelo del Plano de CU transferido con TCCAD.

DATE: none, none, none
 USER: SUBJECT
 OBJECT: 11-OBJECT11

DISPLAY: none, none, none
 SIZE: 1-40170

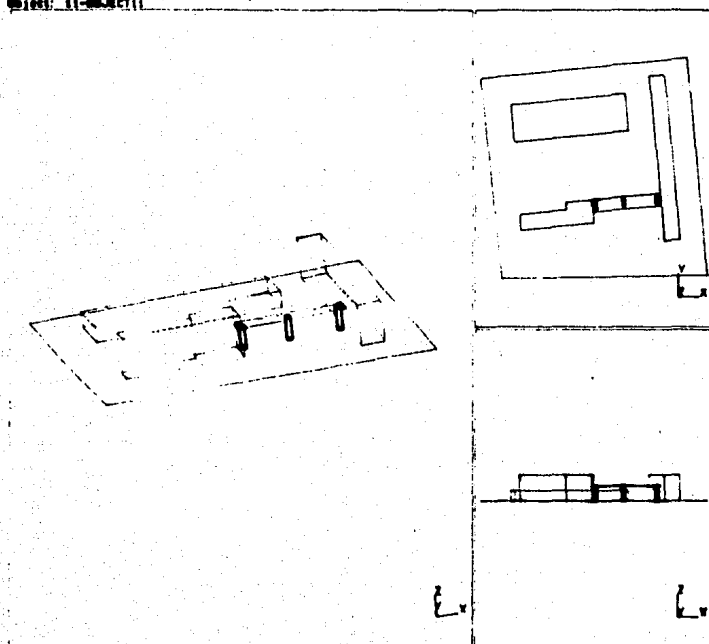


Figura 5.10 Realizado de una sección del Plano de CU

CONCLUSIONES

Por lo general una instalación de CAD, cuenta con diferentes sistemas CAD que son usados en etapas distintas del proceso de diseño, por lo que una comunicación entre estos sistemas es esencial. Esta comunicación se realiza via el intercambio del modelo del objeto y su principal problema es ocasionado por la diferente capacidad gráfica de cada sistema y por la incompatibilidad de las estructuras de datos que manejan.

Esta comunicación ha seguido tres vertientes, la estandarización de componentes de sistemas CAD (Capítulo II), su integración a través de una Base de Datos (Capítulo III) y procesadores especiales que cubran una necesidad especial de comunicación entre dos sistemas (Capítulo V).

Aunque solo algunas interfaces han alcanzado el status de estándar internacional, estas logran reducir la complejidad de un sistema CAD, facilitando el trabajo sobre sus módulos y ayudando a diseminar tecnología mediante la transferencia de modelos de generación, imágenes y códigos. En el Capítulo II se mostró el estado en las actividades de la estandarización, mostrando su interrelación y su influencia en la concepción completa de un sistema CAD.

El comunicar diferentes sistemas CAD a través de un Sistema de Bases de Datos se ha desarrollado solo en un nivel experimental. Entre los SMBDs experimentales de CAD ha tenido más aceptación el modelo relacional (INTERGRAD, ARDBID) por su independencia física de datos, porque permiten diferentes vistas lógicas y su organización es de fácil entendimiento. En el terreno comercial, algunos productos como CAEDS cuentan con un SMBD para utilizarlo como herramienta con los programas de análisis y otros como CADAM que incluyen facilidades de Bases de Datos como un sistema contra fallas y respaldo. Esto es debido a que los SMBDs son sistemas grandes y su tecnología no ha avanzado lo suficiente para cubrir las necesidades de un ambiente de diseño e ingeniería y prefieren que la investigación en este campo

progrese más. Se mostró un resumen de las características deseables en un SMD CAD que se ha presentado en la literatura con el objeto de exponer la dirección de la futura investigación al respecto.

La realización de procesadores especiales, como en el caso de TGCAD, que cubre las necesidades de comunicación entre sistemas específicos, aunque cumple con su objetivo, tiene varios inconvenientes, como el tener que conocer las peculiaridades de cada sistema, para lograr su comunicación, el tiempo en el desarrollo de su programación; además pueden existir problemas de incompatibilidad al cambiar el nivel de los productos comunicados y en general, el soporte y asesoría para estos productos tienen deficiencias, agravándose esta situación en países como el nuestro donde su difusión es mucho menor.

GLOSARIO

ANSI -"American National Standards Institute"-.

ASCII -"American Standard Code for Information Interchange"-
Código de datos estándar de ocho bits por carácter para aplicaciones en computadora.

Base de Datos Colección de Datos interrelacionados que se encuentran almacenados sobre alguna clase de dispositivo de almacenamientos de datos.

CAD -"Computer Aided Design"- Es un proceso de transformación de información basado en un sistema de computación y el staff asociado para asistir en la creación, modificación, análisis, despliegue y documentación de un diseño.

CADAM -"Computer-graphics Augmented Design and Manufacturing"- Sistema de CAD/CAM de alta precisión orientado a tareas en 2, $2\frac{1}{2}$ y 3D.

CAE -"Computer-Aided Engineering"- Aplicación de la tecnología CAD en ambientes de Ingeniería.

CBS -"Circuit Board Design System"- Paquete de graficación que se utiliza en el diseño, análisis y manufactura de circuitos impresos.

CORE. Propuesta de estándar en el diseño de una interface gráfica desarrollado por la ACM SIGGRAPH GSPC.

Entidad Geométrica Se refiere a una primitiva geométrica (bloque fundamental utilizado en la construcción de una gráfica, p.e. arco, círculo, punto) o un grupo de estas procesadas como una unidad independiente.

GKS -"Graphical Kernel System"- Interface gráfica estándar a nivel de programador desarrollada por ANSI e ISO.

Hardware Conjunto de piezas o dispositivos mecánicos, magnéticos y electrónicos de una computadora.

IGES -"Initial Graphics Exchange Specification"- IGES intenta estandarizar la comunicación en la información de dibujos y geometría entre dos sistemas de computo.

Integridad Mantenimiento en la consistencia de los datos en un sistema donde multiples usuarios tienen acceso y comparten la base de datos.

Interactivo Denota comunicación bilateral entre un sistema CAD o estación de trabajo y sus operadores.

ISO -"International Standards Organization"-.

NAPLPS -"North American Presentation Level Protocol Syntax"-.
Un estándar adoptado por AT&T para enviar texto de computadoras y gráficas sobre líneas de telecomunicación.

PHIGS -"Programmer's Hierarchical Interactive Graphics Standard"-.

Profile Entidades geométricas que en conjunto forman un ciclo y no mantienen intersecciones.

Programa de Aplicación. Software orientado a una tarea específica.

Raster (Tecnología) Tecnología que representa la información gráfica mediante elementos con identidad propia llamados pixeles.

Redundancia Existencia de campos de datos duplicados o repetición en la descripción de los contenidos de los campos de datos por cada entrada. La redundancia es obtenida cuando unidades de datos son expandidas para proporcionar más información que la estrictamente necesaria.

Sistema Operativo Es una colección de programas y datos organizados diseñados específicamente para manejar los recursos de un sistema de computadora y facilitar la creación de programas de computadora y controlar su ejecución en este sistema.

SMBD Sistema Manejador de Bases de Datos, es el software que organiza y controla el acceso a la base de datos en un sistema multiusuario.

Software Soporte lógico de la computadora en forma de programas, lenguajes, procedimientos, reglas y documentación asociada.

TRC (Tubo de Rayos Catódicos) Es el principal componente en una terminal de texto y/o gráfica. Estos tubos crean imágenes por medio de un rayo controlado de electrones incidiendo sobre una pantalla.

VDI -"Virtual Device Interface"-. Es una propuesta para estandarizar el camino en que el software gráfico se relaciona con los dispositivos gráficos. VDI consiste de un conjunto de comandos para un dispositivo abstracto, los cuales son traducidos por un driver para ser usado por cada dispositivo.

VDM -"Virtual Device Metafile"-. Propuesta de estándar para transportar la definición de un dibujo.

Vector (Tecnología) Tecnología basada en movimientos continuos punto a punto para la generación de entidades gráficas.

VM -"Virtual Machine"- Sistema Operativo de IBM formado por dos componentes: el Control Program (CP), también conocido como monitor de la máquina virtual, encargado de realizar las funciones de administración del procesador, la memoria y los dispositivos de I/O para crear máquinas virtuales; el segundo componente, el Conversational Monitor System (CMS) es un sistema operativo simple que realiza las funciones de procesamiento de comandos.

Working Set. Área geométrica bidimensional del sistema de modelado geométrico del producto CAEDS.

REFERENCIAS

[ALVA80] Alvarez Medina, G.M., *Un Sistema de Modelado Geometrico Hibrdo basado en Superficies Recortadas y Beta-Splines*, Tesis de Maestria, IIMAS Cd. México, 1990.

[BESA80] Besant, C.B., *Computer Aided Design and Manufacture*, John Wiley and Sons, New York, NY, USA (1980), 170 pag.

[BITT83] Bittner, J., *Data Independence in CAD/CAM databases*, Computer Applications in Production and Engineering, ed. E.A. Warman, North-Holland Pub. Co. 1983, pp. 573-587.

[BONO85] Bono, Peter R., *A survey of Graphics Standards and their Role in Information Interchange*, Computer, Vol. 18, No. 10, Oct. 1985, pp. 63-75.

[BORC80] Borgerson, B.R. y R.H. Johnson, *Beyond CAD to Computer Aided Engineering*, Information Processing 80, ed. S.H. Lavington, North-Holland Pub. Co. 1980, pp. 659-666.

[BRUN88] Bruns, R.L., *Making Graphics Software More Portable*, The S. Klein Computer Graphics Review, Jan-Feb 1988, pp. 67-83.

[BUCH81] Buchmann, A.P., *Databases for Process Plant Design*, CAD of Chemical process plant design, ed. M. Leesley, Gulf Publishing 1981, pp. 149-189.

[BUCH84] Buchmann, A.P., *Current Trends in CAD databases*, Computer Aided Desing, Vol. 16, No.3, May 1984, pp. 123-126.

[BUCH86] Buchmann, A.P., Carrera, R.M., *El uso de un sistema convencional de CAD versus un Manejador de Bases de datos especialmente disenado para manipular datos de CAD*, II Reunión Nacional de CAD/CAM, 17 y 18 de noviembre de 1986, Cuernavaca, Morelos, pp. 139-145.

[BUD83] Burger, W.F., *MLD: A language and database for Modeling*, Computer Applications in Production and Engineering, ed. E.A. Warman, North Holland Publishing Company 1983, pp. 559-571.

[CADR87] Computer Aided Design Report, *Data Base Management*, Vol.7, No.1, Jan. 1987.

[COW86] Cowan, D., *Survey of the CAD field*, CAD International Directory 1986, pp. 5-10.

[DATE81] Date, C.J., *An Introduction to Database Systems*, Addison-Wesley, 3a. Ed., 1981.

[EAST81] Eastman, C.M., *Database Facilities for Engineering Design*, Proceedings of the IEEE, Vol. 69, No. 10, Oct. 1981, pp.1249-1263.

[ENC83] Encarnacao, J., and Schlechtendahl, E.G., *Computer Aided Design: Fundamentals and System Architectures*, Springer-Verlag, New York, NY, USA (1983), 346p.

[ENC84] Encarnacao, J. et al, *A Reference Model for Components and Interfaces of a CAD System*, Integration of CAD/CAM, ed. D.Kochan, Elsevier Science Publishers B.V.1984 (North Holland), pp.135-153.

[ENDE84] Enderle G, Kansy, K y Pfaff G., *Computer Graphics Programming: GKS - The Graphics Standard*, Springer-Verlag, Berlin 1984.

[FISC79] Fischer, W.E., *PHIDAS A database management system for CAD/CAM application software*, Computer Aided Design, Vol.11, No.3, May 1979, pp. 146-149.

[GRO84] Groover, M.P. and Zimmers, Jr, E.W., *Computer Aided Design and Computer Aided Manufacturing*, Prentice-Hall, 1984.

[HAGE81] Hagen, P.J.W., *The impact of graphics standards on CAD/CAM software engineering*, CAD/CAM As a Basis for the Development of Technology in Developing Nations, ed. J.L.Encarnacao et al, North-Holland Pub. Co. 1981, pp. 59-66.

[HEND88] Henderson L., Journey M. y Osland C., *The Computer Graphics Metafile*, IEEE CG&A, Aug. 1986, pp.24-32.

[HOPG82] Hopgood, F.R.A. *The road to graphics standards*, Computer Aided Desing, Vol. 14, No.4, Jul. 1982, pp. 221-225.

[IBM88] IBM Corp., *Initial Graphics Exchange Specification (IGES) Processor, Program Reference and Operations (SH20-5630)*, White Plains,NY.: IBM Corp., 1986.

[KALAB3] Kalay, Y.E., *A relational database for non-manipulative representation of solid objects*, Computer Aided Design, Vol. 15, No. 5, Sept. 1983, pp. 271-276.

[KORIB3] Koriba,M., *Database Systems: their applications to CAD software design*, Computer Aided Design, Vol.15, No.5, Sept. 1983, pp 277-287.

[MICH81a] Michener, J.C.and A van Dam, *A functional Overview of the Core System with Glossary*, ACM Computing Surveys, Vol.10, No.4, Dec. 1981.

[MICH81b] Michener, J.C. y J.D. Foley, *Some Major Issues in the Design of the Core Graphics System*, ACM Computing Surveys, Vol.10, No. 4, Dec. 1981.

[MYER82] Myers,Ware, *CAD/CAM: The need for a broader focus*, Computer, Vol. 15, No. 1, Jan. 1982, pp. 105-117.

[NEWM81] Newman, W.H. and A. van Dam, *Recent Efforts toward Graphics Standardization*, ACM Computing Surveys, Vol. 10, No. 4, Dec. 1981.

[PAN84] Panazuk, C., *Software Standards will usher in the Age of Graphics*, Electronic Design, Jul. 12 1984, pp.92-106.

[PAT82] Patnaik, L.M. and Ramesh, N., *Implementation of an interactive relational graphics database*, Computer & Graphics, Vol. 6, No.3, 1982, pp. 93-96.

[POW86] Powers T., Frankel A., Arnold D., *The Computer Graphics Virtual Device Interface*, IEEE CG&A, Aug. 1986, pp.33-41.

[PUK_86] Puk, R.F., McConnell J.I., *GKS-3D: A Three-Dimensional Extension to the Graphical Kernel System*, IEEE CG & A, Aug. 1986, pp.42-49.

[ROGE78] Rogers, D.F., and J.A. Adams, *Mathematical Elements for Computer Graphics*, New York: Mc Graw Hill, 1976.

[SCOV83] Scowen, R.S., *IGES. A critical Review and Some Suggestions*, CAD Systems Framework, ed. K.Bo y F.M. Lillehagen, North Holland Publishing Company 1983, pp 47-55.

[SHEN83] Shenoy, R.S. and Patnaik, L.M., *Data Definition and Manipulation languages for a CAD database*, Computer Aided Design, Vol. 15, No. 3, May 1983, pp.131-134.

[SHUE86] Shuey, D., Bailey D., Morrisey T., *PHIGS: A Standard, Dynamic, Interactive Graphics Interface*, IEEE CG&A, Aug. 1986, pp. 50-57.

[SIE86] Siebers, C.R., *An introduction to Computer Graphics*, Computer Aided Design, Vol.18, No.3, May 1986, pp.132-138.

[SMIT79] Smith, L.B., *Drawing ellipses, hyperbolas or parabolas with a fixed number of points and maximum inscribed area*, Comput.J., Vol.14, p.81, 1979.

[STAL86] Staley, S.M. y Anderson, D.C., *Functional specification for CAD databases*, Computer Aided Design, Vol. 18, No. 3, May 1986, pp. 132-138.

[ULFS81] Ulfby S., Meen S., and Olan J., *TORNADO: A DBMS for CAD/CAM Systems*, Computer Aided Design, Vol. 13, No. 4, Jul. 1981, pp. 193-197.

[WILL77a] Williams, R., *A Survey of data structures for Computer Graphics Systems*, Data Structures, Computer Graphics and Pattern Recognition, eds. Klinger, A, Fu, Ks y Kunii, T.L., Academic Press (1977).

[WILL77b] -----, *On the application of relational data structures in computer graphics*

[WILL77c] -----, *Data Structures in Computer Graphics*