

UNIVERSIDAD AUTONOMA DE NUEVO LEON

FACULTAD DE CIENCIAS FISICO MATEMATICAS



"ESTANDARIZACION ENTRE SISTEMAS
CAD/CAM/CAE"

T E S I S

QUE PARA OBTENER EL TITULO DE
LICENCIADO EN CIENCIAS COMPUTACIONALES

PRESENTA
JOSE IGNACIO TIJERINA ACOSTA

MONTERREY, N. L.

DICIEMBRE DE 1987.

TL
TS155
.6
.T55
1987
c.1

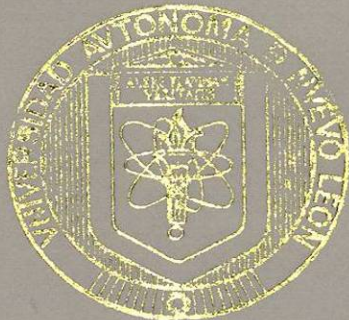


1080171543

UNIVERSIDAD AUTONOMA DE NUEVO LEON

FACULTAD DE CIENCIAS FISICO MATEMATICAS

FACULTAD DE CIENCIAS FISICO-MATEMATICAS



ESTANDARIZACION ENTRE SISTEMAS
CAD/CAM/CAE

TESIS
T E S I S

QUE PARA OBTENER EL TITULO DE
LICENCIADO EN CIENCIAS COMPUTACIONALES

LICENCIADO EN CIENCIAS COMPUTACIONALES
PRESENTA
JOSE IGNACIO TIJERINA ACOSTA

JOSE IGNACIO TIJERINA ACOSTA

MONTERREY, N. L.

DICIEMBRE DE 1987



UNIVERSIDAD AUTONOMA DE NUEVO LEON

FACULTAD DE CIENCIAS FISICO-MATEMATICAS

"Estandarización entre sistemas CAD/CAM/CAE"

TESIS

que para obtener el título de

LICENCIADO EN CIENCIAS COMPUTACIONALES

presenta

JOSE IGNACIO TIJERINA ACOSTA

Monterrey, N.L.

Diciembre de 1987

A mi padre
Lic. José Ignacio Tijerina Jaime

a quien le agradezco su incansable
labor de guía y compañero

A mi madre
Profra. Bertha Acosta de Tijerina

por su ánimo, voluntad y apoyo
constante

**A mi hermana
Profra. Bertha Nylda Tijerina Acosta**

**por saber representar el
ejemplo familiar**

**A mi hermano
Ing. Pedro Tijerina Acosta**

**por su capacidad y esfuerzo
demostrados en las grandes tareas**

**A mi hermana
Lic. Elsa Mireya Tijerina Acosta**

**por saber reconocer el valor de
la tenacidad y constancia**

A mi cuñado

Lic. Jorge Alberto Tijerina Ramos

**por su colaboración en mi
carrera profesional**

**Con todo cariño
a mis sobrinos**

**Paola Mélaney y
Jorge Alberto**

**A mi asesor de tesis
Ing. Oscar Recio Cantú**

**por su esfuerzo, colaboración
y consejos**

**A todos mis maestros,
compañeros y amigos de la
Facultad de Ciencias Físico-Matemáticas**

**Finalmente, un profundo agradecimiento
a todos mis compañeros de VITRO TEC
por la oportunidad brindada**

Contenido

1. Introducción.
 - 1.1 El origen de las gráficas computacionales.
2. Tecnología CAD/CAM/CAE.
 - 2.1 Computadoras, núcleo básico de los sistemas CAD/CAM/CAE.
3. Estandarización en sistemas CAD/CAM/CAE.
 - 3.1 El valor de los estándares gráficos.
 - 3.2 Estandarización en tres categorías funcionales.
4. Interface con programas de aplicación (API).
 - 4.1 GKS (Graphical Kernel System).
 - 4.1.1 Language Bindings.
 - 4.2 PHIGS (Programmer's Hierarchical Interactive Graphics Standard).
 - 4.2.1 Organización jerárquica del modelo.
 - 4.3 Relación GKS-PHIGS.
5. Interface con dispositivos virtuales (VDI).
 - 5.1 CGI (Computer Graphics Interface).
6. Intercambio de información entre sistemas.
 - 6.1 Formatos neutrales.
 - 6.2 Alternativas actuales.
 - 6.3 IGES (Initial Graphics Exchange Specification).
 - 6.3.1 Formato de archivo IGES.
 - 6.3.2 Situación actual de IGES.
 - 6.4 SET (Standard D'Exchange et De Transfert).
 - 6.5 CGM (Computer Graphics Metafile).
 - 6.6 Formatos de base de datos externos.
 - 6.7 Alternativas futuras.
7. Caso práctico.
 - 7.1 Transferencia de información I-DEAS ---- MEDS.
8. Conclusiones.
9. Bibliografía.

Apéndice A Ejemplo archivo IGES.

Apéndice B Ejemplo formatos de base de datos externos.

1. Introducción.

1.1 El origen de las gráficas computacionales.

La evolución de la computación en el último cuarto de siglo ha sido notable. Años de investigación y desarrollo han sido necesarios para alcanzar estos avances.

En 1950 una terminal gobernada por una computadora conectada al sistema Whirlwind I de MIT fue utilizada para generar trazos simples. Este dispositivo gráfico hacía uso de un tubo de rayos catódicos (CRT) similar al utilizado en los televisores, fueron éstos los primeros pasos dados en gráficas computacionales. Algunos años antes, ya se había iniciado la utilización de los CRT para el despliegue de información alfanumérica.

Durante los años 50's las gráficas generadas por computadora tuvieron poco progreso, esto debido a que los equipos de cómputo fueron utilizados, principalmente, para resolver problemas de cálculo numérico.

El evento que vino a promover la importancia de las computadoras gráficas interactivas como un nuevo campo de aplicación lo fue la publicación de una brillante tesis presentada por Ivan E. Sutherland para obtener el grado de Ph. D. en el MIT. Su tesis titulada 'Sketchpad: A Man-Machine Graphical Communication System', vino a probar a los lectores que el área de la computación era un campo motivante, útil y factible para enfocar esfuerzos de investigación.

A mediados de los años 60's, ambiciosos proyectos de investigación estaban en marcha en el MIT, General Motors, Bell Telephone Laboratories, Lockheed Aircraft, Brigham Young University, etc. La edad de oro de la computación gráfica había iniciado.

Si los 60's representaron años de trabajo intenso en investigación, la década de los 70's ha sido una etapa en la que se inició la recolección de los frutos que ésta podía generar.

Las gráficas computacionales interactivas, rápidamente evolucionan y pasan a ser el principal medio de comunicación hombre-computadora. Los equipos de computación gráfica se encuentran ahora en muchos países, con aplicaciones muy variadas, y son, ya, utilizados con propósitos educativos. La instantánea atracción de los usuarios, de todas edades, hacia las gráficas computacionales ha ayudado bastante a ampliar las aplicaciones, lo que indudablemente garantiza su continuo crecimiento y popularidad.

2. Tecnología CAD/CAM/CAE.

Computer-Aided Design (CAD), Computer-Aided Manufacturing (CAM) y Computer-Aided Engineering (CAE) son elementos de lo que es conocido como Computer-Integrated Manufacturing (CIM), todos estos términos, y algunos otros que se manejan, señalan el uso de los sistemas computacionales, acompañados de terminales gráficas y de algún otro hardware y software especializado, para asistir en el desarrollo de las actividades de ingeniería relacionadas con diseño y manufactura. Estos sistemas CAD/CAM/CAE eliminan gran parte del trabajo tedioso y repetitivo que rodea a la ingeniería, incrementando la productividad y liberando tiempo al personal para actividades mas creativas.

Una década y media de experiencia ha probado que los sistemas trabajan, que ahorran dinero a las compañías que los utilizan al reducir el porcentaje de errores en diseño y manufactura de productos.

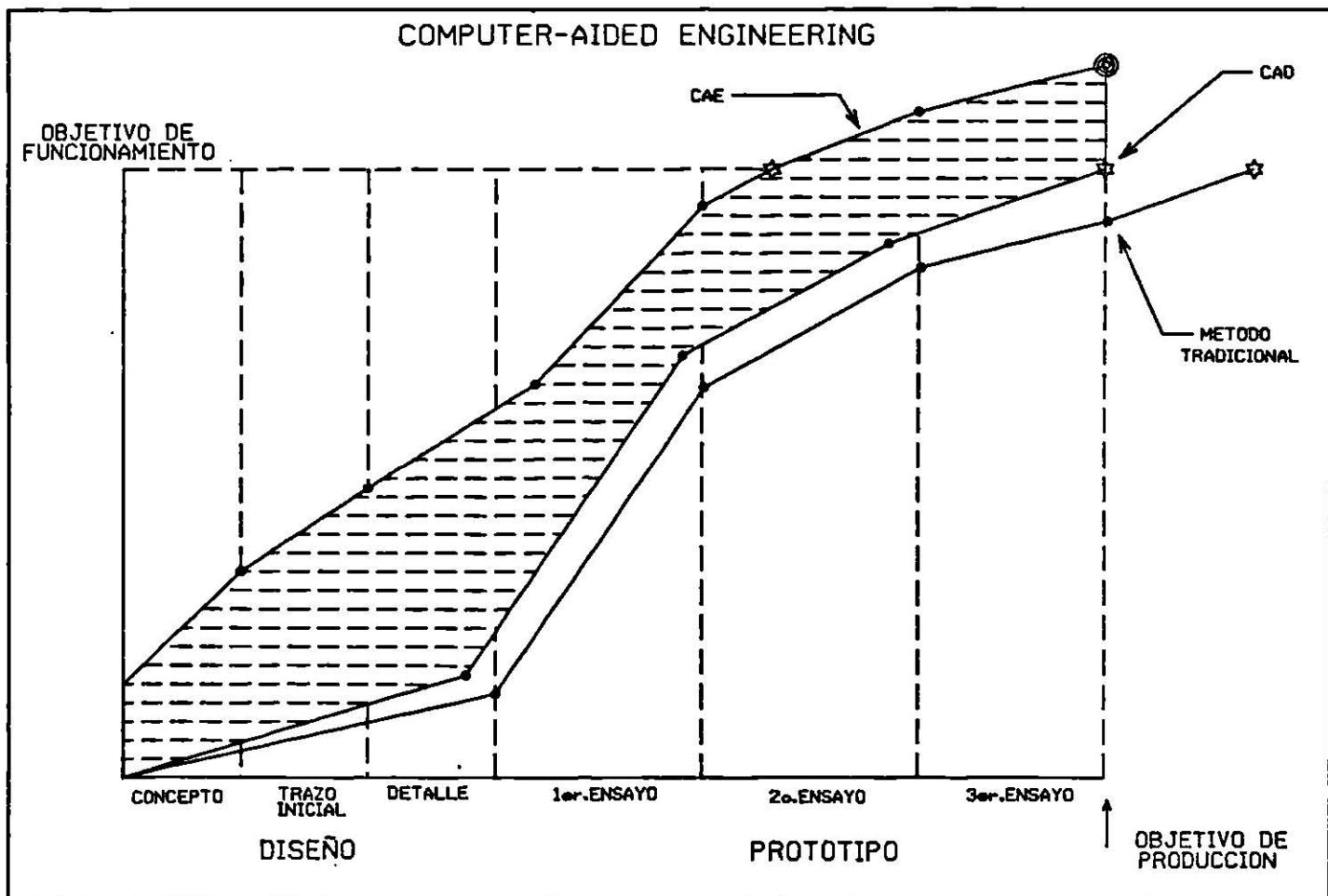


Fig. 2.1 Gráfica comparativa entre sistemas CAD/CAM/CAE vs. métodos manuales.

CIM es el término utilizado para describir la completa automatización de la industria, con todos los procesos trabajando bajo el control de computadores, los cuales tienen acceso a una base de datos centralizada.

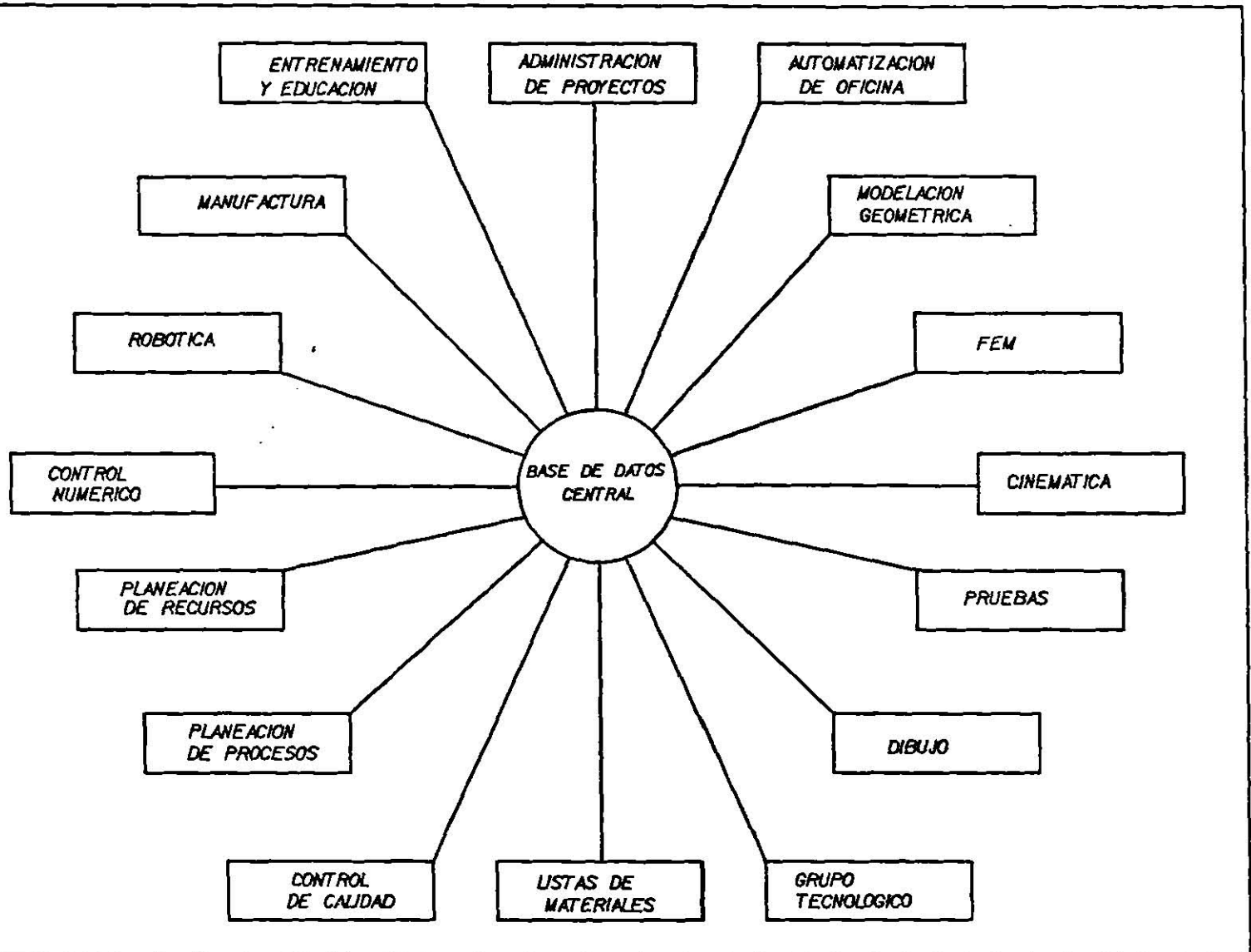


Fig. 2.2 Computer-Integrated Manufacturing.

Sin embargo, los sistemas CAE actuales sólo satisfacen las necesidades locales de algunos grupos dentro de las empresas, y es así, como aparecen las llamadas "islas de automatización".

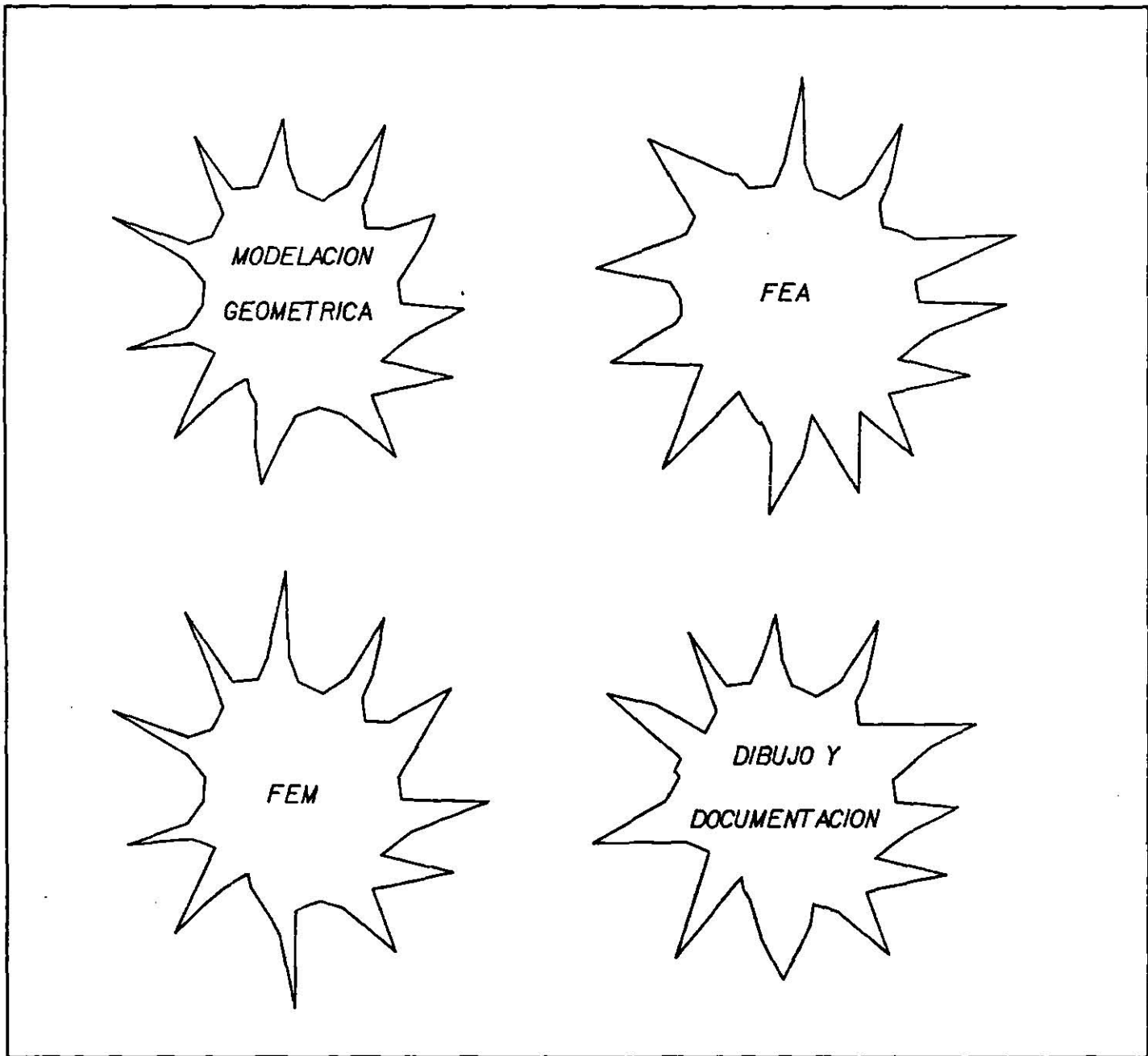


Fig. 2.3 Islas de automatización en CAE.

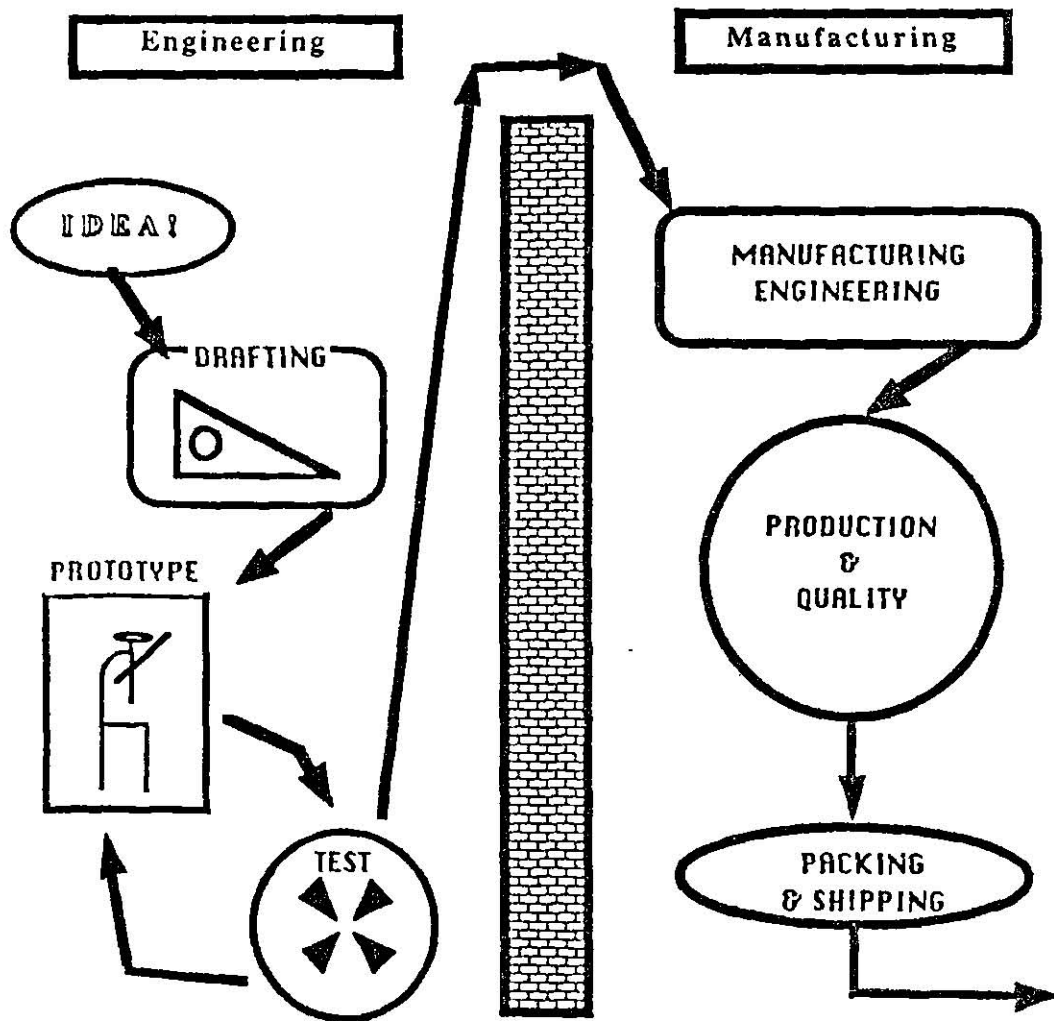


Fig. 2.4 Etapas que constituyen los procesos independientes de ingeniería y manufactura en una compañía.

Para realizar la interface entre estas "islas" se requiere de redes de comunicación que les permitan a las computadoras hablar entre sí, así como estándares en el software de interface que permita ejecutar, eficientemente, diferentes programas de aplicación compartiendo la misma base de datos.

Dentro de un futuro cercano, las compañías que no utilicen estas herramientas estarán en seria desventaja de competencia, con respecto a otras compañías que si los utilicen. Sin embargo sería igualmente perjudicial el comprar o rentar un sistema de este tipo antes de estar preparados para usarlos eficientemente, estos sistemas no solucionan sólo todos los problemas, requieren de la participación multidisciplinaria de personal capacitado, son así, una herramienta mas para la ingeniería. El manejo de la información tanto gráfica como no gráfica es parte central de las actividades de ingeniería que utilizan la computadora como herramienta de diseño.

Basándonos en las actividades que involucra cada área, en posteriores referencias, incluidas dentro de este trabajo, podremos hablar de sistemas CAD cuando sean sistemas exclusivamente para dibujo, sistemas CAD/CAE o CAE cuando los sistemas de dibujo incluyan actividades de análisis ingenieril, y sistemas CAD/CAM o CAD/CAM/CAE cuando estén incluidas actividades de manufactura.

Las figuras 2.5 y 2.6 muestran la relación que existe entre CAE, CAD y CAM en el proceso de desarrollo de un producto.

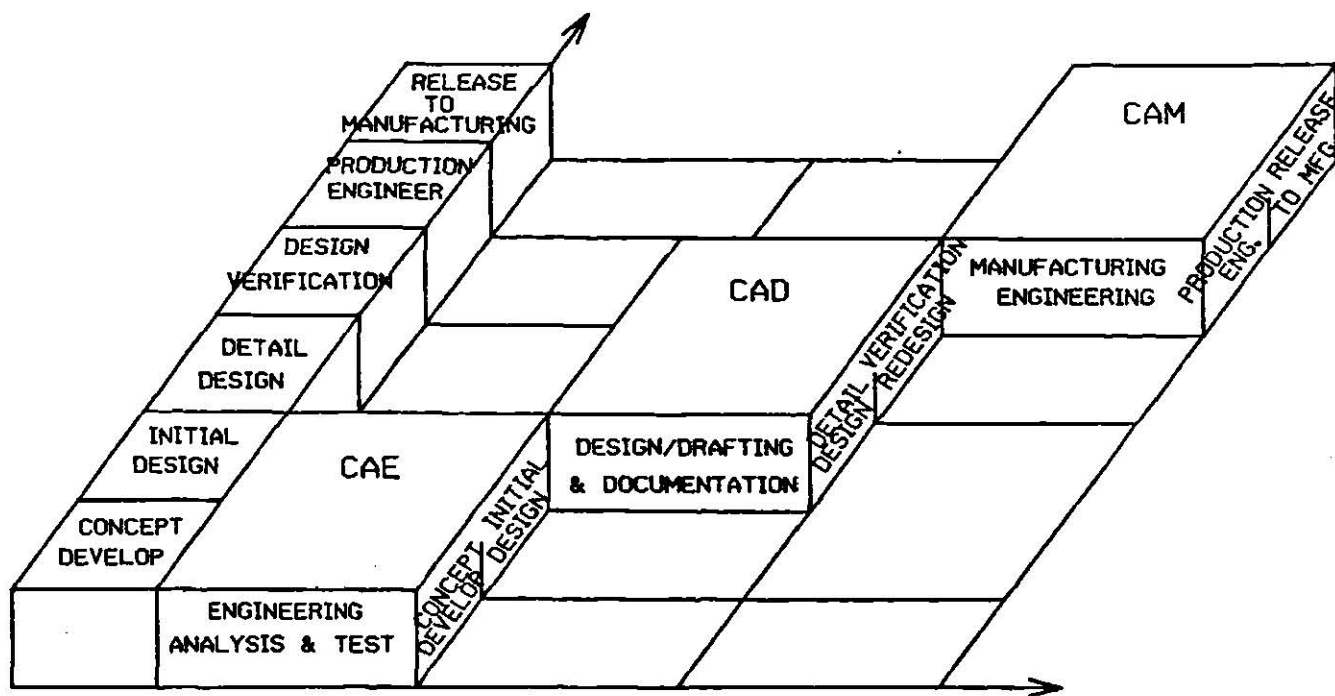


Fig 2.5 Relación entre CAD, CAM y CAE.

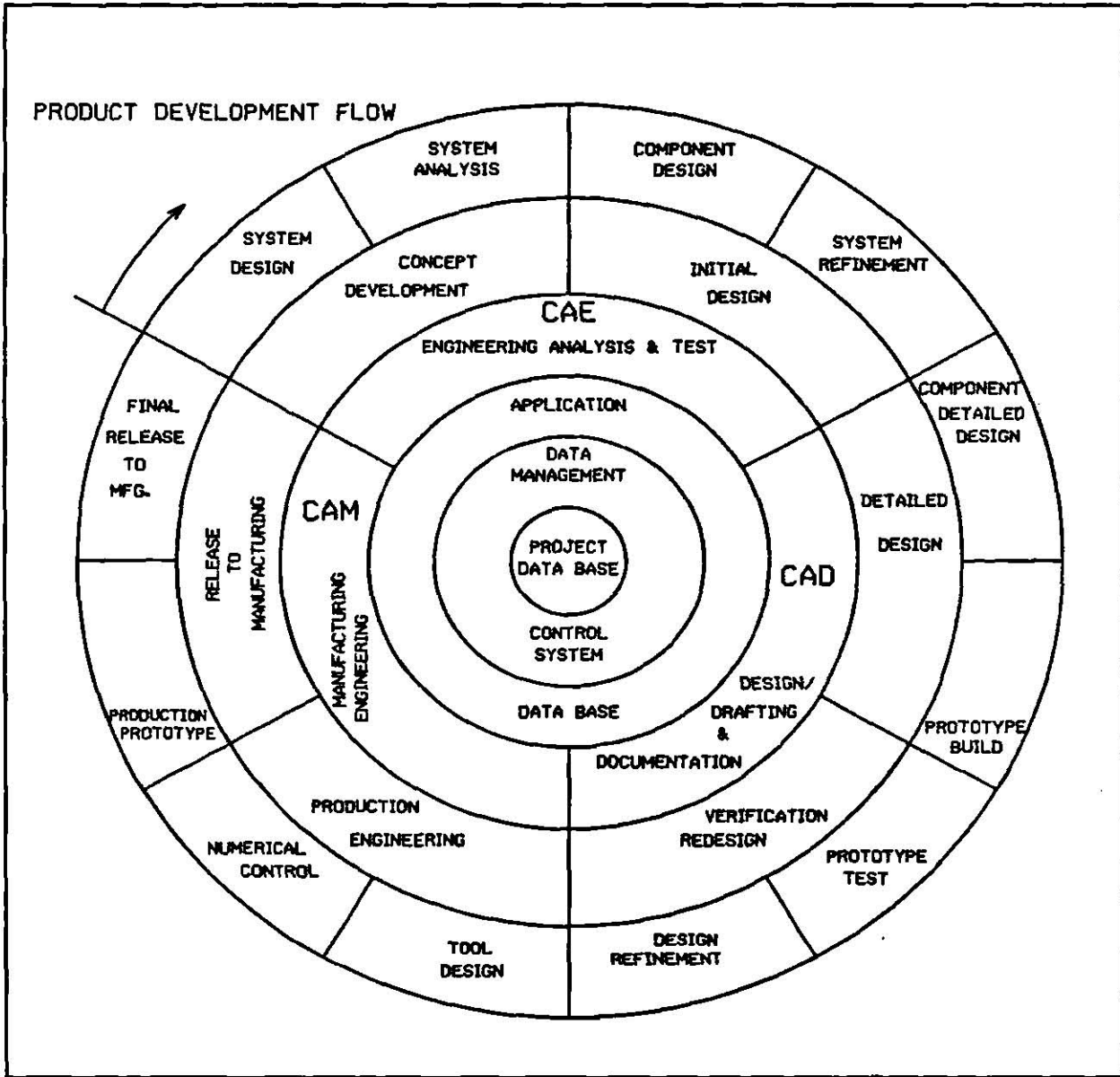


Fig. 2.6 Relación entre CAD, CAM y CAE.

2.1 Computadoras, nucleo básico de los sistemas CAD/CAM/CAE.

La parte central e ingrediente esencial en la tecnología CAE es la computadora digital, justamente como las computadoras han sido utilizadas en los negocios para procesar datos numéricos, los sistemas CAE almacenan, recuperan y despliegan información en forma gráfica.

Ahora bien, los elementos computacionales que integran un departamento de ingeniería CAD/CAM, al igual que un centro de cómputo con alguna otra aplicación, se pueden ver por separado en hardware y software.

Los componentes de hardware se encuentran en una gran variedad de tamaños, configuraciones y capacidades. Por consiguiente es posible configurar un sistema que se ajuste a las necesidades particulares para cada organización.

Grupos de ingeniería que no estén envueltos en producción pueden elegir un sistema exclusivamente para dibujo y diseño, por otro lado grupos de producción se inclinarían por una configuración que integre los requerimientos de manufactura (CAM).

Por supuesto que el hardware tendría un pequeño valor en utilidad si no es acompañado por el software adecuado a las necesidades que se deseen cubrir. La tendencia que se ha está tomando es que el usuario arme su propia configuración no restringiéndose, necesariamente, a adquirir tanto hardware como software de un sólo proveedor. Los sistemas CAD/CAM son totalmente modulares, el usuario puede armar la configuración que mejor se ajuste a sus necesidades.

Un sistema "llave en mano" (turnkey), es definido como cualesquier configuración, manejada con un sólo proveedor, que incluye todo hardware y software necesario para que el sistema pueda funcionar y sea capaz de producir resultados en una aplicación particular. La mayoría de los sistemas CAE que han sido vendidos han sido manejados con un sólo proveedor, tanto en hardware como en software. A medida que las aplicaciones han aumentado las especialidades de los proveedores se han identificado, haciendo crecer la importancia de analizar el software de terceros. Esto nos puede sugerir que el uso de esta tecnología requerirá algo mas que sólo sistemas "llave en mano", y al mismo tiempo los proveedores de los sistemas deberán ya estar considerando incluir el soporte y compatibilidad con software de aplicación en CAE desarrollado por otras compañías.

En estos momentos existen en el mercado proveedores que venden sistemas completos (turnkey systems), proveedores que manejan exclusivamente hardware, exclusivamente software, o exclusivamente equipo periférico. Es tarea del usuario la elección del camino a tomar, típicamente se ha estado observando que mientras mas especializado sea el proveedor en un area en particular, mejores son sus productos.

El uso de software CAE no es exclusivo de sistemas grandes, es posible utilizarlo en las computadoras personales en aplicaciones muy específicas. Los componentes en hardware de un sistema CAD/CAE personal pueden estar configurados como se muestra en la fig. 2.7.

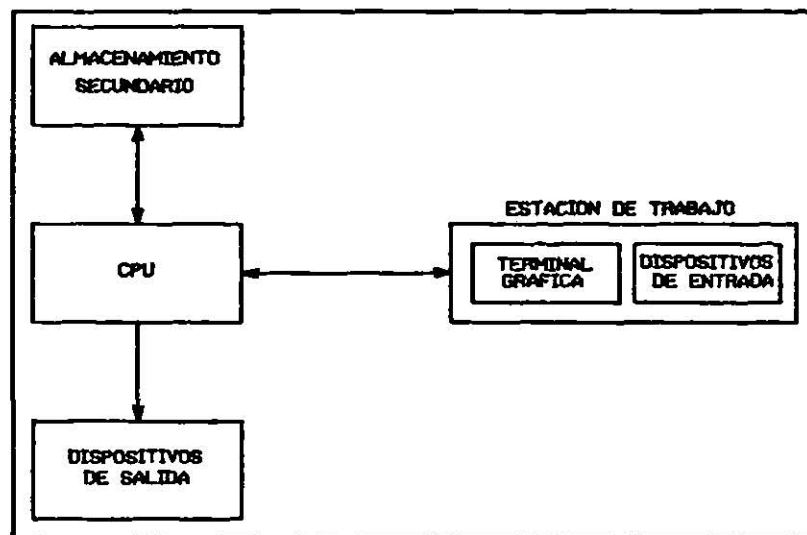


Fig. 2.7 Un sistema CAD/CAE personal.

Dependiendo de la calidad y capacidad de sistema personal que se desee tener, es la calidad de equipo que se selecciona para armar su configuración. Algunas opciones que se pueden encontrar para cada componente son:

- | | |
|----------------------------|--|
| Almacenamiento secundario: | cintas magnéticas, disco magnético, diskettes, etc. |
| CPU: | PC, PC + procesador matemático, estación de trabajo, etc. |
| Medios de salida: | graficador de plumas, graficador electrostático, impresora, etc. |
| Terminales gráficas: | monitor monocromático, monitor cromático de alta o baja resolución, etc. |
| Dispositivos de entrada: | light pen, joydisk, joystick, tableta, teclado, etc. |

La estación de trabajo en un sistema CAD/CAE representa la interface entre el sistema y el usuario, representa, así, un factor muy importante al determinar que tan conveniente sería, para un diseñador, el utilizar un sistema de este tipo.

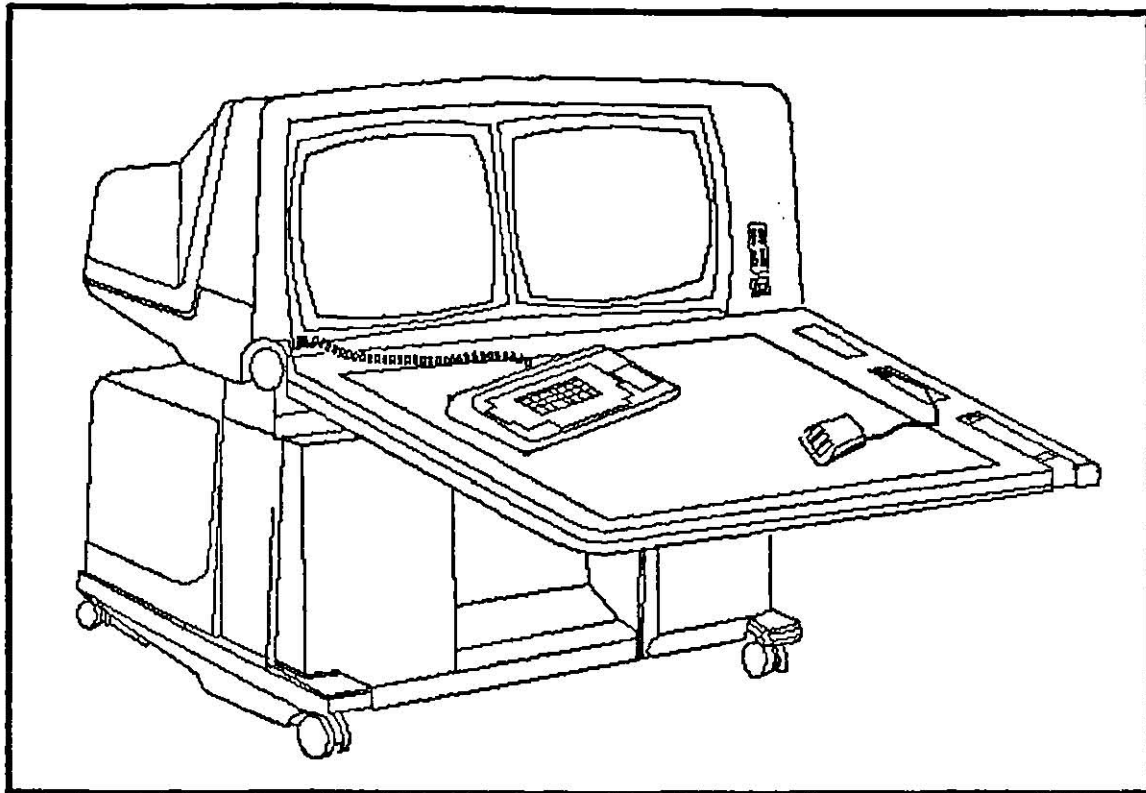


Fig. 2.8 Estación de trabajo gráfica interactiva, en ella se muestran dos dispositivos de entrada: teclado y tableta con mouse.

Un sistema CAD/CAE multiusuarios tiene una estructura similar al sistema personal, en cuanto a los componentes que lo integran, mas, como es de esperarse, las variantes se presentan en cantidades y capacidades. La palabra "mainframe" ha sido utilizada para denotar una unidad central de proceso o un sistema de cómputo centralizado que es capaz de soportar a mas de un usuario. Dependiendo de el tamaño de la computadora y de su estructura, algunos sistemas pueden soportar 30 o mas terminales de trabajo.

Una configuración típica de un sistema CAE puede ser la siguiente:

Hardware:

- a) Procesador central
- b) Estación(es) de trabajo o terminal(es) gráficas
- c) Sistema de almacenamiento en línea
- d) Consola de operación
- e) Unidad de cintas magnéticas
- f) Dispositivos de salida, tales como graficadores o impresoras
- g) Terminales alfanuméricas

Software:

- a) Sistema operativo
- b) Nucleo básico de graficación
- c) Productos de aplicación específicos

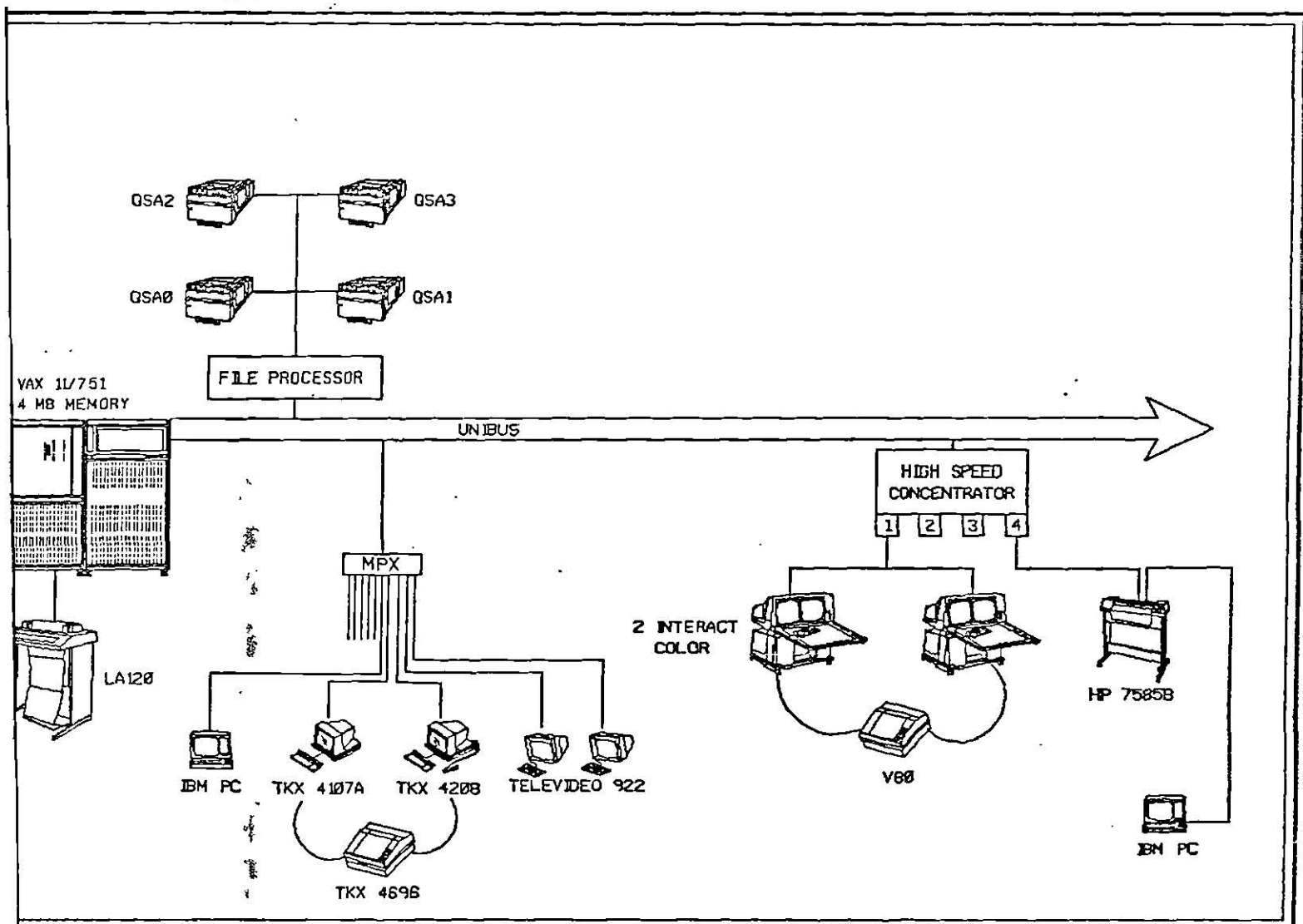


Fig. 2.9 Un sistema CAD/CAE multiusuarios.

3. Estandarización en sistemas CAD/CAM/CAE.

Durante la infancia de las gráficas computacionales en los años 60's, sólo unas cuantas organizaciones podían cumplir con los requerimientos computacionales que las gráficas computacionales interactivas requerían, sin embargo, la representación gráfica como medio de salida de procesos batch se encontraba mucho mas accesible. El primer estándar para salidas gráficas surgió de las necesidades de ese tiempo, creando salidas gráficas en diferentes dispositivos. La necesidad de transportar programas de computadora a computadora aún no era reconocida, pero la necesidad de independencia de dispositivos gráficos conectados a una misma computadora fue importante; el resultado de ésto fue el desarrollo de un paquete de software para permitir salida en un graficador de plumas, y es así como las "Rutinas Calcomp" vinieron a ser un 'estándar informal'. Esto fue una solución simple pero altamente efectiva.

Los requerimientos de las gráficas interactivas son mucho mas complejos, una gran variedad de paquetes de software gráficos fueron desarrollados durante los años 70's, los cuales soportaban, ya, medios de acceso interactivos.

Las gráficas computacionales tienen, dependiendo de diferentes puntos de vista, ventajas o desventajas al contar con una gran variedad de facilidades técnicas para la producción de gráficas. Graficadores de plumas, graficadores electrostáticos, impresoras, monitores, sistemas de proyección, etc. son sólo algunos de los medios de despliegue de información gráfica, con diferencias en funcionamiento y en rango de costo. Teniendo presente ésta gran diversidad, la independencia de dispositivos gráficos representa un gran objetivo en la estandarización.

3.1 El valor de los estándares gráficos.

El valor de algún estándar gráfico lo podemos conocer al evaluar sus ventajas contra sus desventajas. Para dar una idea mas clara, imaginemos por un momento el estándar gráfico ideal; la independencia perfecta de dispositivo que permitiera trabajar con cualquier medio de entrada y cualquier otro medio de salida, y la organización de información que ajustara en todas las aplicaciones gráficas interactivas. La utilización de un estándar de este tipo traería las siguientes consecuencias:

a) Un usuario que trabajara con un paquete de aplicación que utilizara el estándar, podría seleccionar el hardware gráfico de acuerdo a sus necesidades de costo, funcionamiento, tamaño o cualquier otro criterio, sin encontrar opciones limitadas. Así mismo, el estándar nos aseguraría que estos paquetes trabajarían en los nuevos equipos que se fueran desarrollando y que se podrían incorporar a los anteriores sin mayores problemas.

b) Ya que los paquetes de aplicación que utilizan el estándar pueden operar con cualquier clase de equipo gráfico, entonces abarcan una mayor audiencia que los paquetes que dependan de ciertos equipos, ésto es particularmente significativo para los creadores de software comercial.

c) Los estándares gráficos en conjunto con los lenguajes de programación estandarizados facilitarían enormemente la comunicación y el intercambio de información entre sistemas gráficos, pudiendo, así, realizarse en forma directa.

d) El estándar facilita el problema de entrenamiento para los programadores de aplicaciones gráficas. Aprendiendo el estándar, un programador podría desarrollar sus programas en cualquier sistema que tuviera implementado el estándar.

Todos estos puntos le dan una gran significancia a los estándares gráficos, sin embargo los estándares presentan, también ciertos riesgos; la mayoría de los cuales son consecuencia de el hecho de que un 'estándar ideal' no existe:

a) Incorrectas implementaciones o pobres definiciones de un estándar pueden ocasionar problemas en los usuarios que no conocen el estándar o sus problemas potenciales.

b) Los estándares pueden detener la innovación; cuando el mercado se ve altamente dominado por "productos estándar", existe poca oportunidad de abandonarlo y ofrecer nuevas funciones.

3.2 Estandarización en tres categorías funcionales.

Varios son los estándares gráficos que han sido desarrollados, o se encuentran en desarrollo. Todos ellos pueden ser llamados "estándares de interface", ya que tienden a la estandarización en la comunicación entre dos elementos cualesquiera, hardware/software, de un sistema gráfico.

Dentro de el amplio campo de las gráficas computacionales, la estandarización de sistemas se puede clasificar en tres categorías funcionales:

a) Interface con los programas de aplicación (API).

Los estándares API son típicamente implementados como un conjunto de procedimientos externos o subrutinas que el programador puede encadenar a su código de aplicación para obtener gráficas en su dispositivo de salida. Actualmente los estándares API están representados por 3 proyectos: PHIGS (Programmer's Hierarchical Interactive Graphics Standard), GKS (Graphical Kernel System) y GKS-3D.

b) Interface con dispositivos virtuales (VDI).

Los estándares VDI se refieren a aspectos internos de los sistemas gráficos, conciernen a los fabricantes de dispositivos periféricos, fabricantes de tarjetas controladoras gráficas, fabricantes de chips e implementadores de software independiente. Estos estándares requieren independencia de dispositivos sin sacrificar funcionalidad, representan la interface entre los niveles superiores de software, y los dispositivos gráficos. Los estándares VDI, actualmente están representados por el proyecto CGI (Computer Graphics Interface).

c) Interface en el intercambio de información gráfica.

La estandarización a este nivel se refiere a el almacenamiento e intercambio de información entre sistemas CAD/CAM/CAE. En este paso la transferencia se realiza con base en un archivo externo en formato neutral. Algunos proyectos de estandarización en este nivel son: IGES (Initial Graphics Exchange Specification), SET (Standard D'Exchange et De Transfert), PDES (Product Data Exchange Specification), STEP (International Standar for the Exchange of Product Definitions) y CGM (Computer Graphics Metafile).

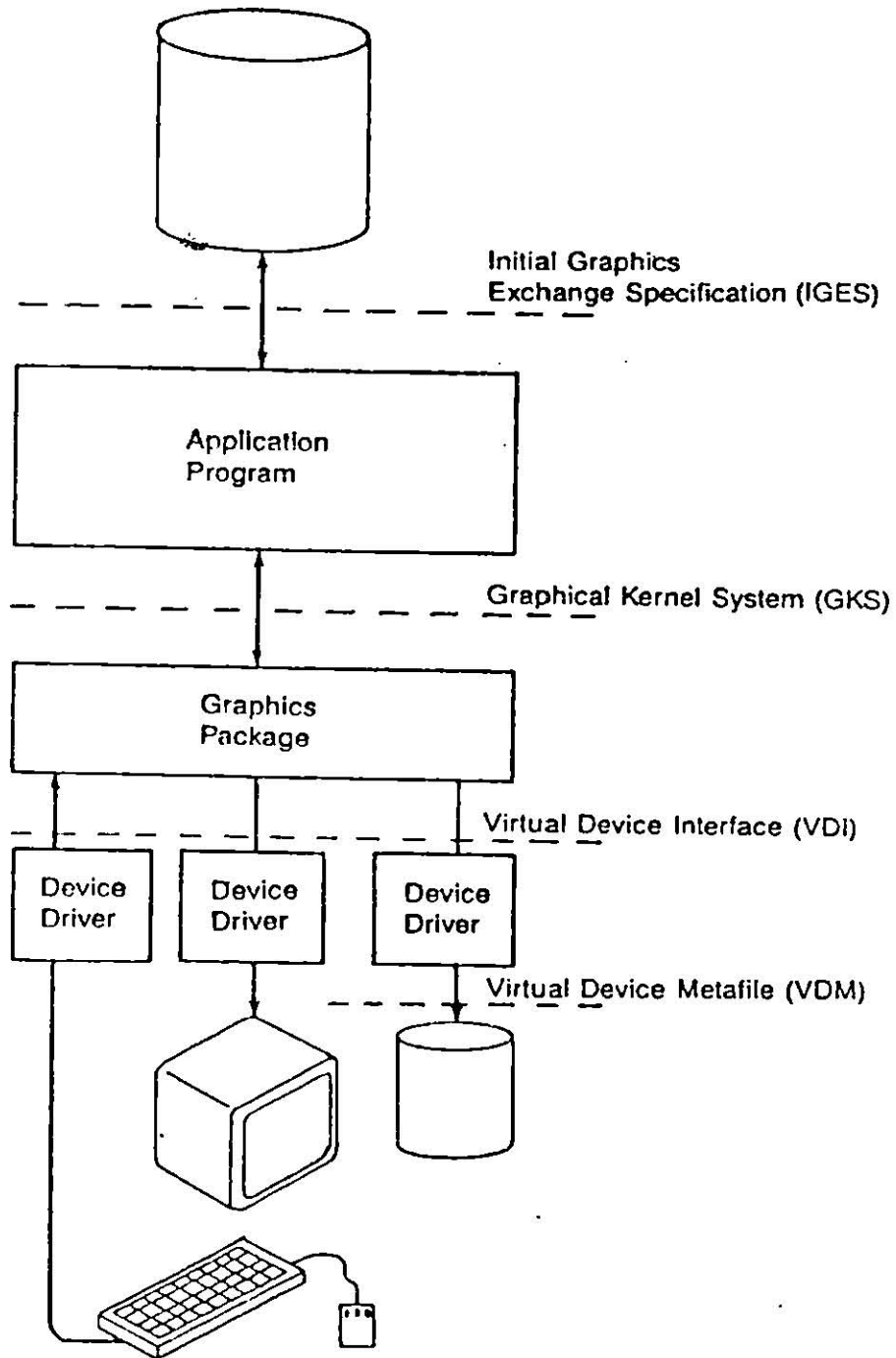


Fig. 3.1 Modelo de desarrollo de aplicaciones en un sistema gráfico.

4. Interface con programas de aplicación (API).

4.1 GKS (Graphical Kernel System).

GKS es un conjunto de rutinas gráficas cuyo objetivo principal es la generación y manipulación de imágenes, de una forma tal que no dependan de el dispositivo gráfico, ni de la computadora en los cuales es generada. Dichas imágenes varían desde simples gráficas X-Y, hasta complicados dibujos de ingeniería, o diagramas de circuitos impresos, o información astronómica (imágenes de satélites), etc. Todas estas aplicaciones deben estar contempladas en el GKS de tal manera que puedan ser generadas.

GKS es un estándar internacional aprobado por la International Standards Organization (ISO), actualmente representa el desarrollo mas completo en estándares gráficos dentro de esta categoría. Su precursor es el sistema gráfico CORE, desarrollado en los Estados Unidos pero no fue aceptado internacionalmente; actualmente GKS cubre la mayoría de las facilidades que el CORE presentó.

En GKS, las imágenes son construidas en base a un cierto número de blocks básicos de construcción. Estos blocks básicos de construcción o *primitivos*, como mejor se les conoce, representan los tipos de elementos básicos que se utilizan en la generación de una imagen en pantalla. Los cuatro primitivos básicos en GKS son:

- a) polilinea: define segmentos de línea conectados en base a una secuencia de puntos.
- b) marcas: utilizadas en la identificación de puntos.
- c) achurados: utilizados en la identificación de áreas.
- d) textos: secuencia de caracteres.

GKS representa la interface con las librerías de subrutinas gráficas, las cuales manipulan las opciones de entrada/salida de información gráfica para una gran variedad de aplicaciones.

4.1.1 Language Bindings.

GKS se encuentra definido para trabajar en forma independiente de los lenguajes de programación; ya que GKS puede ser accesado desde FORTRAN, C, PASCAL y ADA.

Esta independencia de lenguaje se logra realizar a través de los *language bindings*. Las cerca de 200 subrutinas GKS tienen un sólo nombre en su núcleo central, así mismo, todos sabemos que la manera de acceso a subrutinas es diferente para cada lenguaje, algunos tienen restricción en cuanto a longitudes de nombre, algunos en cuanto a caracteres especiales, etc., es así que los *language bindings* representan el mapeo necesario entre el nombre utilizado al llamar una rutina desde el programa de aplicación, y su nombre real en el núcleo GKS.

Entonces, finalmente, tenemos un sub-estándar a nivel de *language bindings*, que de alguna manera ayuda a que los programadores gráficos no tengan problemas mayores al cambiar de lenguaje de programación. Existe una tabla de *language bindings* por cada lenguaje de programación, entonces, cada sistema que cuente con el estándar GKS deberá también contar con las tablas de *language bindings* requeridas para ese sistema en particular.

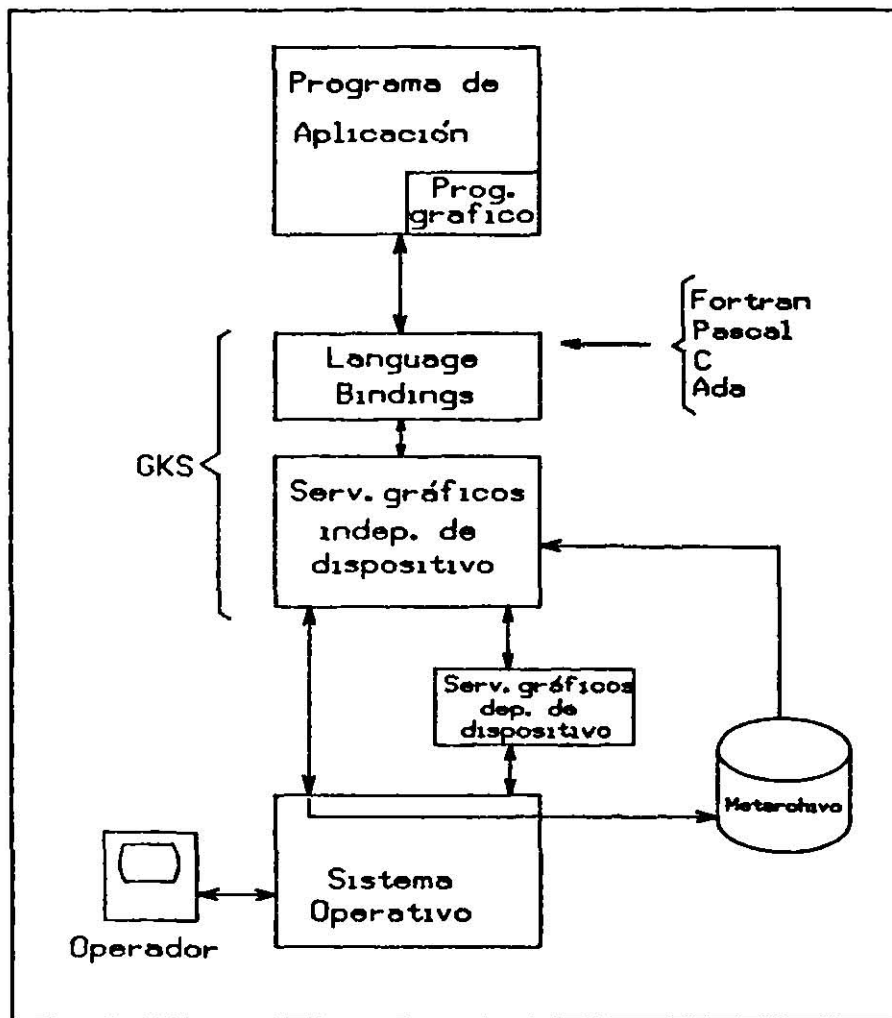


Fig. 4.1 Graphical Kernel System (GKS).

4.2 PHIGS (Programmer's Hierarchical Interactive Graphics Standard).

El estándar PHIGS, actualmente en desarrollo, y apoyado por ANSI/ISO, será el sucesor del GKS. PHIGS es útil en aplicaciones que manipulen complejas imágenes en 2D o en 3D en un ambiente altamente interactivo y dinámico. Todo esto se realiza utilizando una estructura jerárquica de datos y capacidades de edición flexibles.

El acrónimo PHIGS se puede interpretar como sigue: las palabras *programmer's*, *graphics* y *system*, indican que PHIGS es una interface de aplicación en la programación de un sistema gráfico, dicha interface se realiza entre los propios programas de aplicación y los dispositivos gráficos del sistema. La palabra *hierarchical* indica la forma de organización interna de la información gráfica, dicha estructura facilita la transferencia de información entre los modelos de aplicación y el sistema gráfico. Finalmente, la palabra *interactive* indica que PHIGS provee las facilidades para el trabajo interactivo y la modificación dinámica.

Un objetivo particularmente importante en PHIGS es el colaborar con el programador de aplicaciones gráficas en el desarrollo de sistemas que manejen y desplieguen rápidamente la información de los modelos o archivos gráficos. Prácticamente todas las aplicaciones gráficas pueden ser consideradas "manipuladores de modelos". CAD, CAE, CIM, control de procesos y simulación, son sólo algunas aplicaciones que presentan necesidades de un sistema gráfico como lo es PHIGS. Todas estas aplicaciones cuentan con grandes bases de datos organizadas de diferentes maneras, todas desean la interacción con la información del modelo de una manera rápida, y todas necesitan realizar las modificaciones al modelo de una forma interactiva.

4.2.1 Organización jerárquica del modelo.

La característica principal de PHIGS es la habilidad para manejar eficientemente la información gráfica de los modelos, tanto en el despliegue como en su actualización. La organización y contenido del modelo son los aspectos claves para cumplir con el objetivo.

La organización de la información en PHIGS es jerárquica con múltiples niveles. La organización jerárquica ya es común en muchas aplicaciones por lo que su incorporación a PHIGS no es muy complicada. Por otro lado, esta estructura de datos en PHIGS presenta muy pocas restricciones en cuanto a su organización, lo que brinda flexibilidad a los programadores.

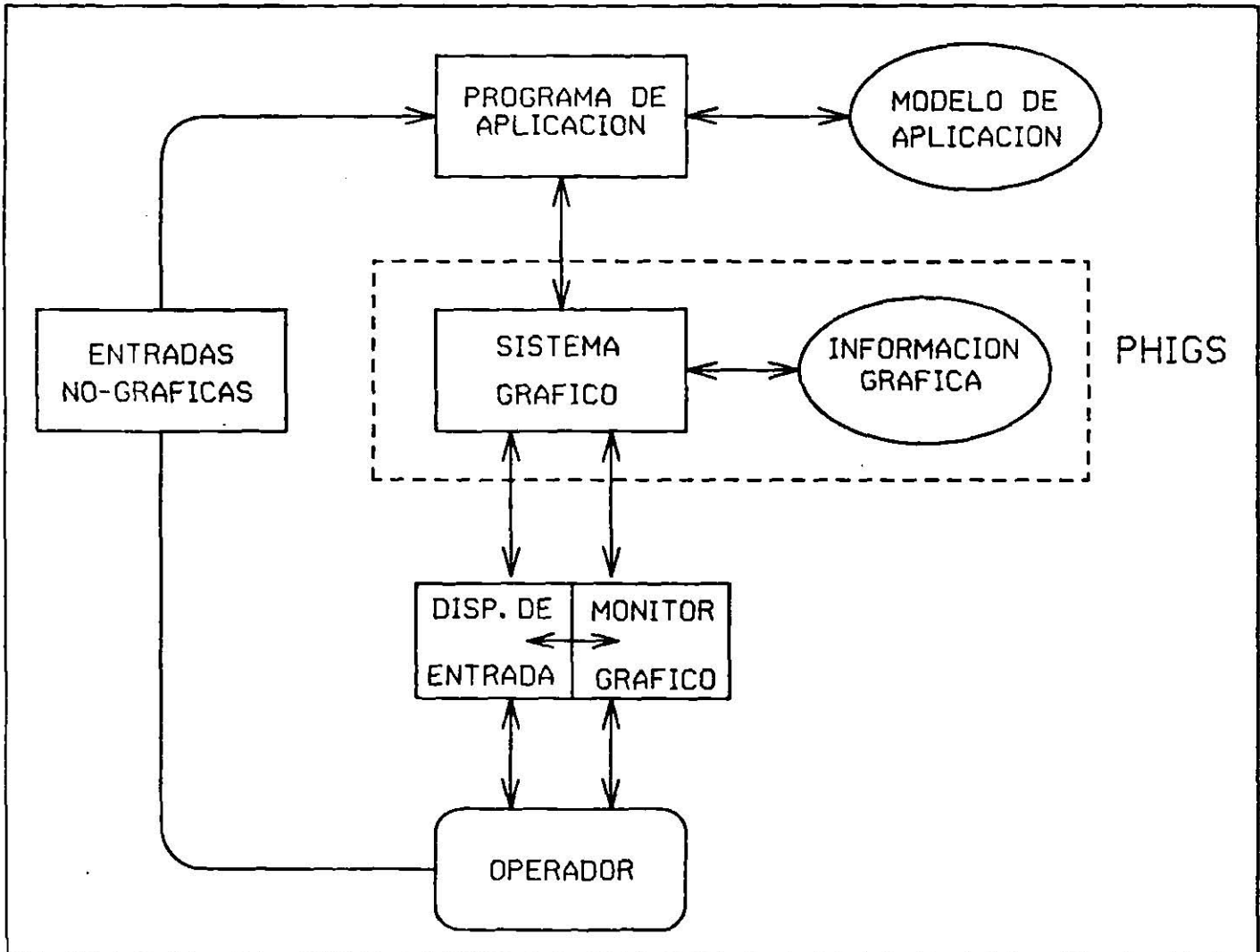


Fig. 4.2 PHIGS dentro de un sistema típico de aplicación gráfica.

Los blocks de construcción en PHIGS son conocidos como *estructuras*, los cuales a su vez contienen *elementos* que definen los atributos de salida o las referencias de la estructura.

Como cualquier otro sistema gráfico, PHIGS utiliza atributos para definir la apariencia de un primitivo en el dispositivo de salida. Los atributos que utiliza PHIGS son, también los mismos que utiliza cualquier otro sistema gráfico, pero la diferencia es que en PHIGS éstos se encuentran ligados en una estructura jerárquica, que es referenciada al momento del despliegue de los elementos.

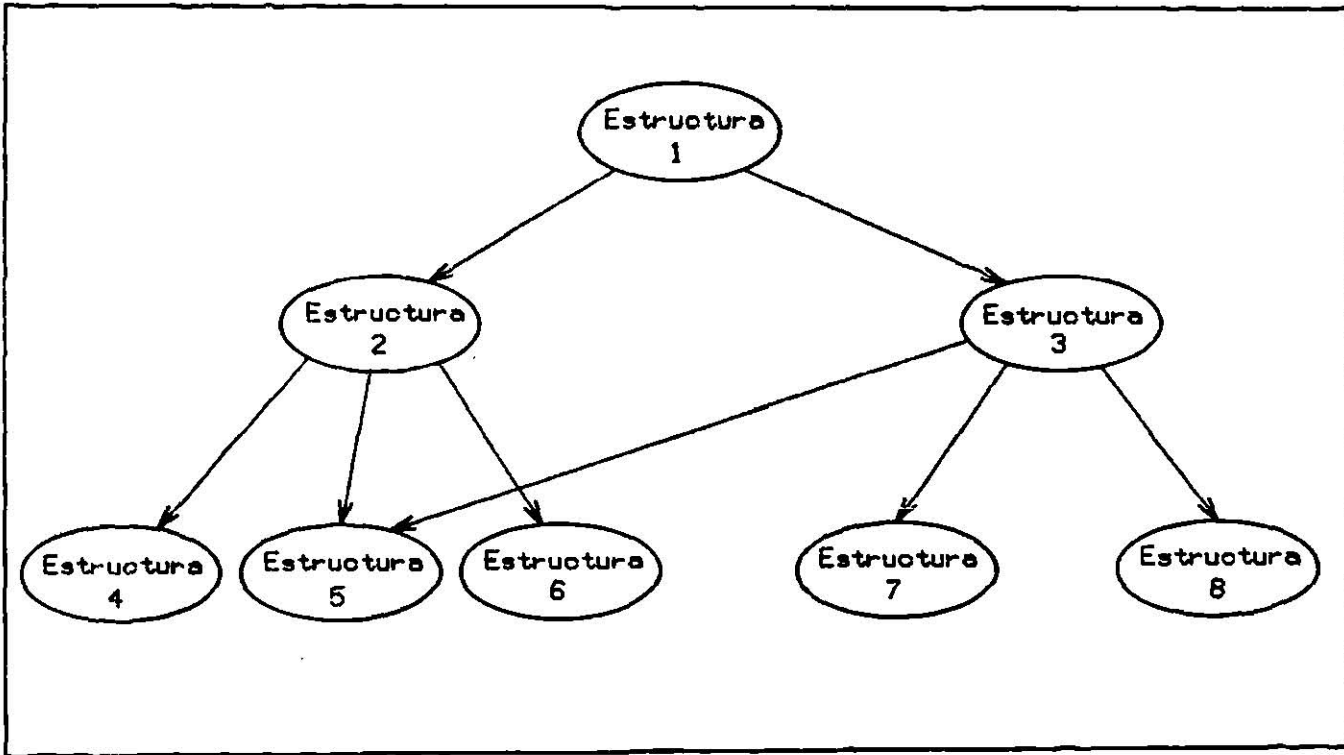


Fig. 4.3 Organización de datos jerárquica en PHIGS.

En la figura 4.3 la estructura jerárquica tiene su raíz en la estructura 1, esta estructura raíz realiza llamadas a las estructuras 2 y 3, quienes a su vez llaman a las estructuras del nivel inferior. En general, dentro de la estructura jerárquica, las llamadas a las estructuras de niveles inferiores se realizan desde las estructuras de los niveles superiores.

4.3 Relación GKS - PHIGS.

Al momento en que PHIGS surge como un segundo estándar gráfico dentro de el nivel de interface con los programas de aplicación (justamente después de GKS y su extensión en 3D), muchos se preguntarán la necesidad de contar con múltiples estándares dentro del mismo nivel de la arquitectura de un sistema gráfico. Algunas de las mayores diferencias entre PHIGS y GKS se pueden resumir enseguida:

- GKS trabaja en base a una organización de datos horizontal. Historicamente, los segmentos del GKS han sido utilizados para representar información gráfica en lugar de poder contar con, tan sólo, información de referencia acerca del modelo gráfico.

- Una vez que un segmento GKS es creado, su contenido no puede ser modificado, sólo se le podrán modificar ciertos atributos que afecten el segmento en su totalidad. Por otro lado PHIGS permite modificar cualquier parte de una estructura en cualquier momento.

- GKS encadena los primitivos con sus respectivos atributos al momento de su creación; para poder modificar éstos atributos, el segmento debe ser destruido y creado de nuevo con los nuevos. En PHIGS los primitivos de salida se encadenan con sus atributos correspondientes al momento del despliegue, facilitando la modificación de múltiples datos.

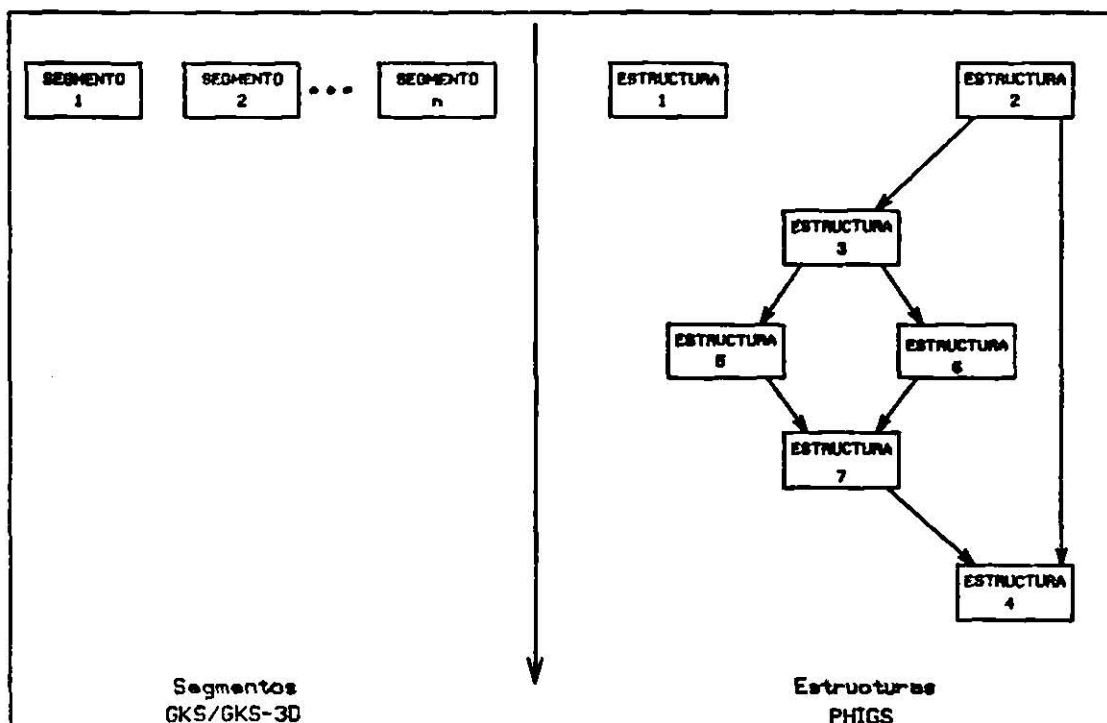


Fig. 4.4 Comparación de estructuras de datos en GKS y en PHIGS.

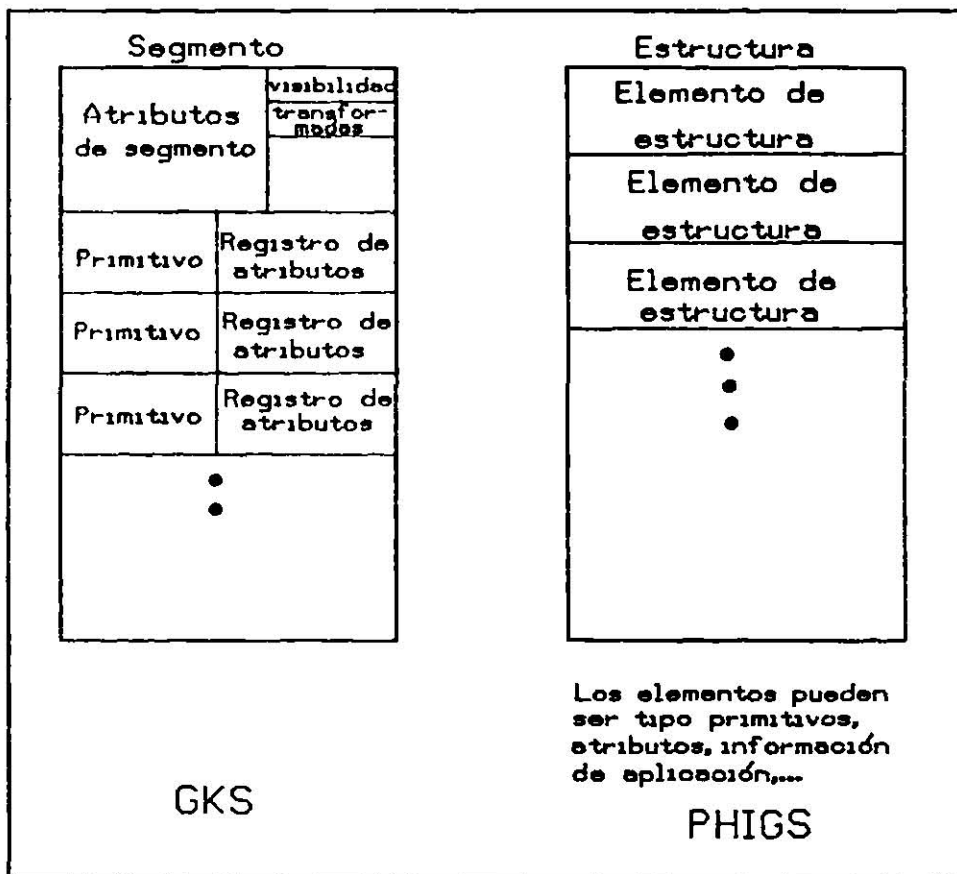


Fig. 4.5 Comparación entre el contenido de un segmento en GKS y una estructura en PHIGS.

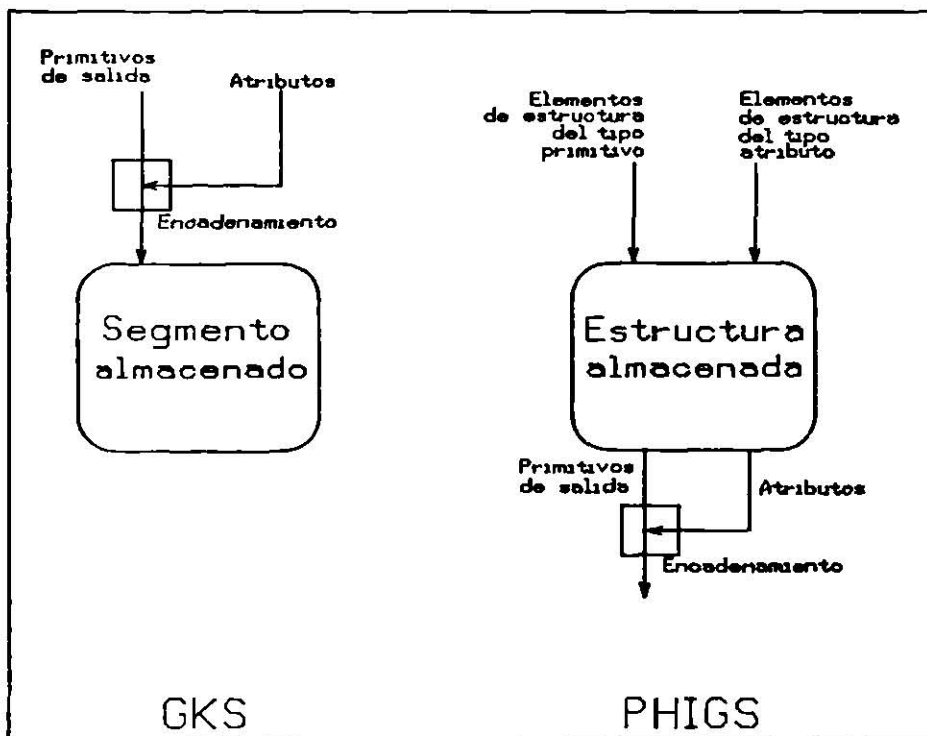


Fig. 4.6 Comparación en el encadenamiento de atributos en GKS y en PHIGS.

- GKS cuenta con transformaciones del modelo limitadas, que son conocidas como transformaciones de normalización; éstas son aplicadas una sólo vez a cada segmento. PHIGS, sin embargo, permite múltiples transformaciones que son aplicadas dentro de el espacio del modelo, el espacio en el cual los modelos son creados y manipulados. Es así que GKS cuenta con un espacio de composición al nivel de la imagen, después de que los primitivos han sido normalizados; estrategia que es inconsistente con su función de sistema manipulador de imágenes. PHIGS es un sistema manipulador de modelos.

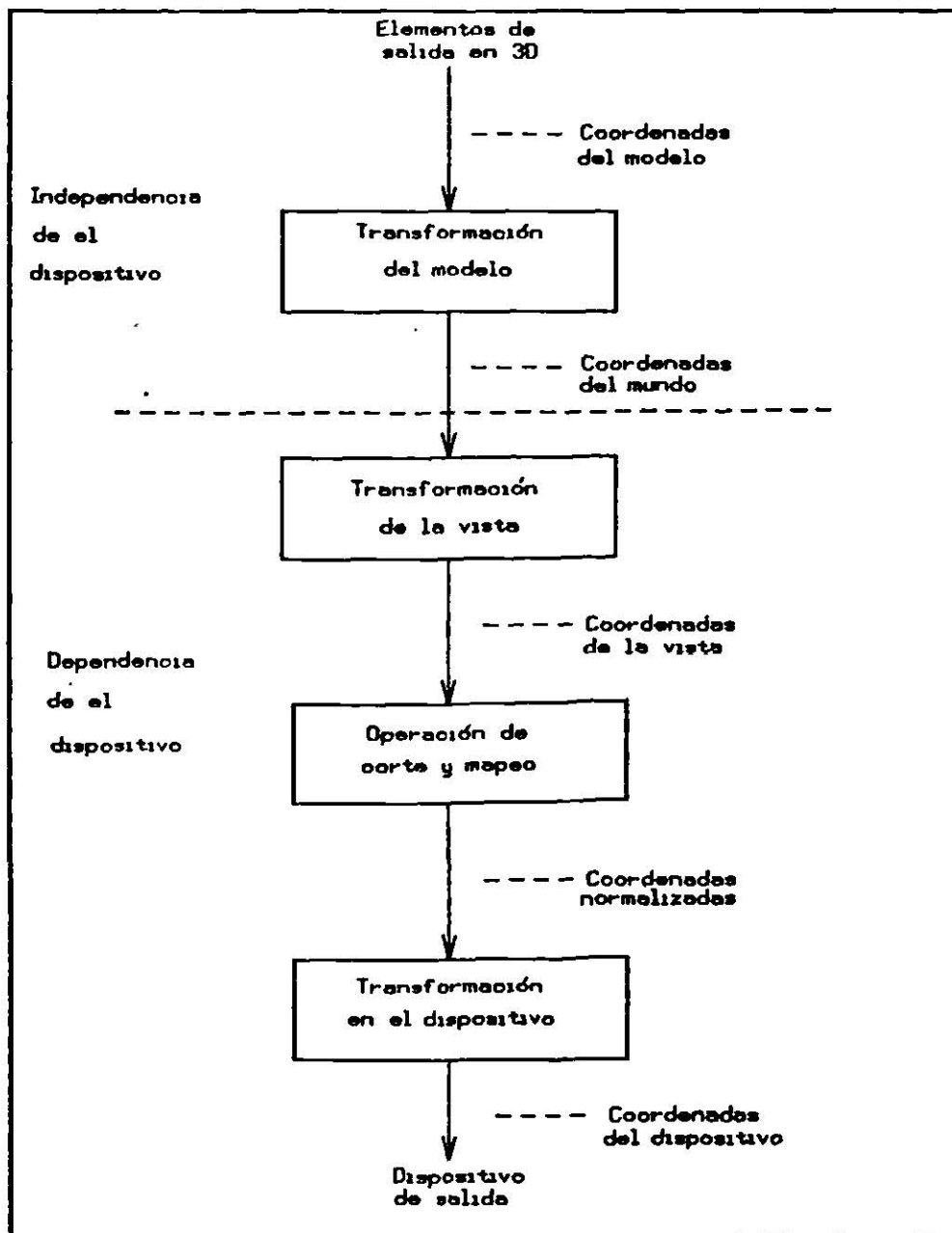


Fig. 4.7 Secuencia de transformaciones en los sistemas de coordenadas en PHIGS.

5. Interface con dispositivos virtuales (VDI).

5.1 CGI (Computer Graphics Interface).

CGI es un estándar actualmente en desarrollo que está siendo soportado por las organizaciones ANSI e ISO. CGI representa un dispositivo virtual el cual puede ser formalizado mediante una tarjeta de circuitos en un dispositivo gráfico, o como un equipo independiente conectado al mismo.

En 1977 el sistema CORE presentó un modelo de sistema gráfico con dispositivos-independientes (DI), lo cual era posible mediante manejadores de dispositivo (drivers). Aun así, ni el sistema CORE de 1977, ni su revisión de 1979 pudieron identificar exactamente cuáles características de dependencia podían separarse bajo una interface dependiente/independiente del dispositivo (DI/DD). Sin embargo estos esfuerzos marcan el precedente hacia un estándar CGI.

Posteriormente el nombre de Interface con Dispositivos Virtuales (VDI) reemplazó a DI/DD, indicando que un estándar podía alcanzarse mediante la combinación hardware, firmware, y software. Ahora, mediante la resolución presentada por ISO con el número TC97/SC21/WG2 en junio de 1984, se modificó el nombre del proyecto, para quedar finalmente como Computer Graphics Interface.

La realización de CGI debe estar orientada hacia un medio ambiente donde residen otros controladores de interface. En nuestro contexto, un sistema significa una colección de interfaces, una de las cuales es CGI. Cuando se trabaja con un sistema, el usuario ejecuta una tarea y observa los resultados; si este sistema aparenta ser sólo una pieza de hardware, como una PC o una estación de trabajo con capacidades de procesamiento locales, entonces el usuario no se percata de que su tarea está siendo descompuesta y distribuida dentro de el sistema, es decir, CGI es parte integral y elemento apropiado de el sistema. Entonces CGI puede ser implementado como un dispositivo disponible para los sistemas gráficos (similar a los servidores de archivos), que es soportado por el sistema operativo; o de otra manera implementado como un recurso integral de los sistemas gráficos que manejen los proveedores. Esta última opción sería válida si los sistemas trabajaran con una interface de alto nivel similar para todos, ejemplo GKS.

Generalmente hablando, un dispositivo es un elemento de hardware el cual es controlado por alguno otro mas poderoso. Dichos dispositivos generalmente realizan actividades de procesamiento gráfico muy específicas. La mayoría de los sistemas gráficos cuentan con un dispositivo de interface en el cual las actividades de procesamiento y dibujo son divididas (ej. procesador de archivos gráficos). En CGI el término dispositivo es un término un poco mas restrictivo aplicado a el hardware con capacidades limitadas a la agrupación, almacenamiento, y ejecución de entradas/salidas de información.

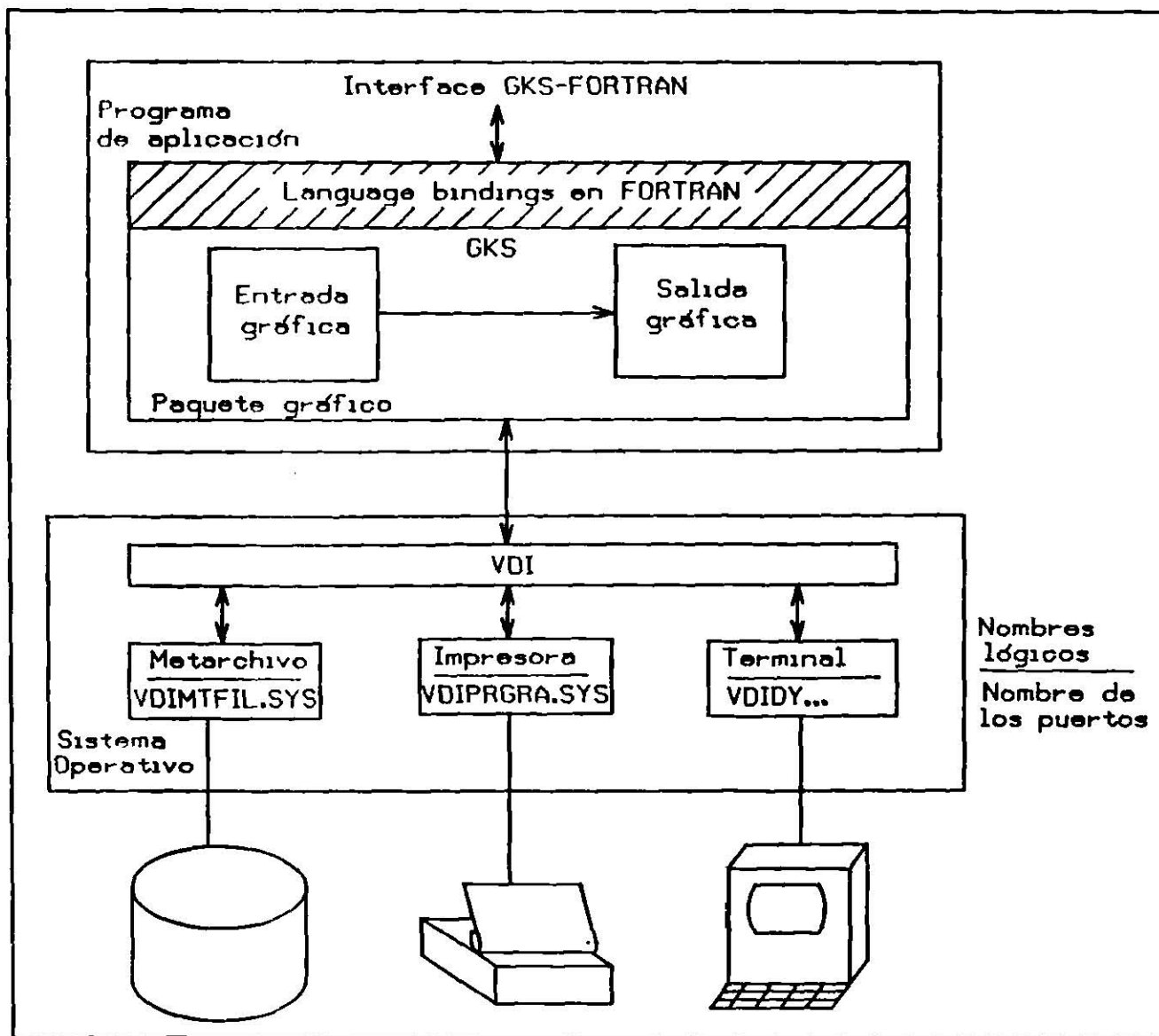


Fig. 5.1 Relación entre un estándar API (ej. GKS) y un estándar VDI.

6. Interface en el intercambio de información entre sistemas.

Muchas compañías medianas y grandes cuentan ya, entre sus diferentes divisiones con sistemas de cómputo heterogeneos que deben compartir información. La comunicación entre ellos se hace necesaria en el trabajo cotidiano. Los sistemas computacionales, pequeños, medianos o grandes, con aplicaciones CAD/CAM/CAE no se encuentran exentos de esta situación.

Como se comentó antes, varias organizaciones iniciaron el desarrollo de sistemas gráficos simultáneamente, el avance se fue presentando en paralelo por varios caminos diferentes, no era posible definir algunos lineamientos generales que estandarizaran su desarrollo ya que se estaba compitiendo en el desarrollo de nueva tecnología.

Tiempo después de que los sistemas gráficos iniciaron su comercialización, la necesidad de comunicación y transferencia de información entre equipos diferentes se hizo patente. Tarde o temprano cada usuario de sistemas CAD/CAM/CAE necesita intercambiar información entre sistemas con software no compatible. Esto se puede deber, entre otras cosas, a lo siguiente:

- Debido a que la compañía maneja paquetes de software gráficos de diferentes proveedores.
- Debido a que el departamento de manufactura cuenta con un sistema gráfico de programación, para las máquinas de control numérico, diferente al que utiliza el departamento de diseño de esa misma compañía
- Debido a que los clientes de una compañía desean recibir en su sistema CAD/CAE, la información de diseño que resulte de un cierto proyecto, y los sistemas de las dos compañías son diferentes.
- Debido a que los miembros de un departamento de ingeniería necesitan acceso a la información gráfica de un diseño en particular para trabajar con ella con algunos paquetes de análisis que ellos mismos desarrollaron.

Anteriormente, cuando se requería transportar un diseño de un sistema a otro, era necesario volver a realizar ese diseño en el sistema destino, lo que originaba altos costos en tiempo y dinero.

En el inicio de los años 70's la solución que se le dio al problema, fue la de desarrollar traductores directos uno-a-uno entre sistemas.

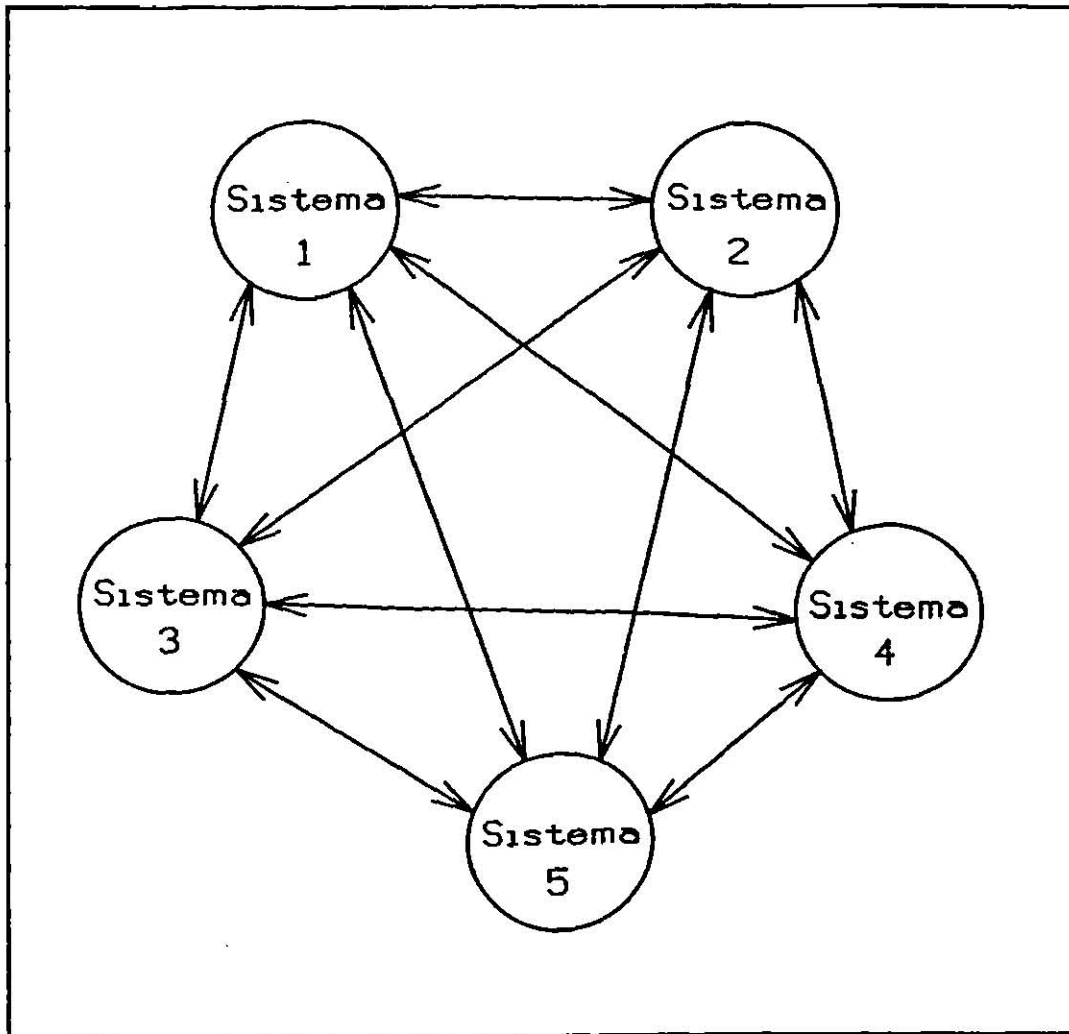


Fig. 5.1 Contando con traductores directos entre sistemas, se requieren 12 traductores para comunicar 4 sistemas en ambos sentidos, agregar un quinto sistema obliga a contar con 8 traductores directos adicionales.

Esto vino a solucionar el problema a medias ya que era necesario contar con un traductor directo para cada sistema con el cual se deseaba intercambiar información.

Esta nueva situación de proliferación de traductores directos obligó a la creación de el concepto de un archivo con formato neutral. Contar con un formato neutral alivia la situación anterior reduciendo el número de traductores.

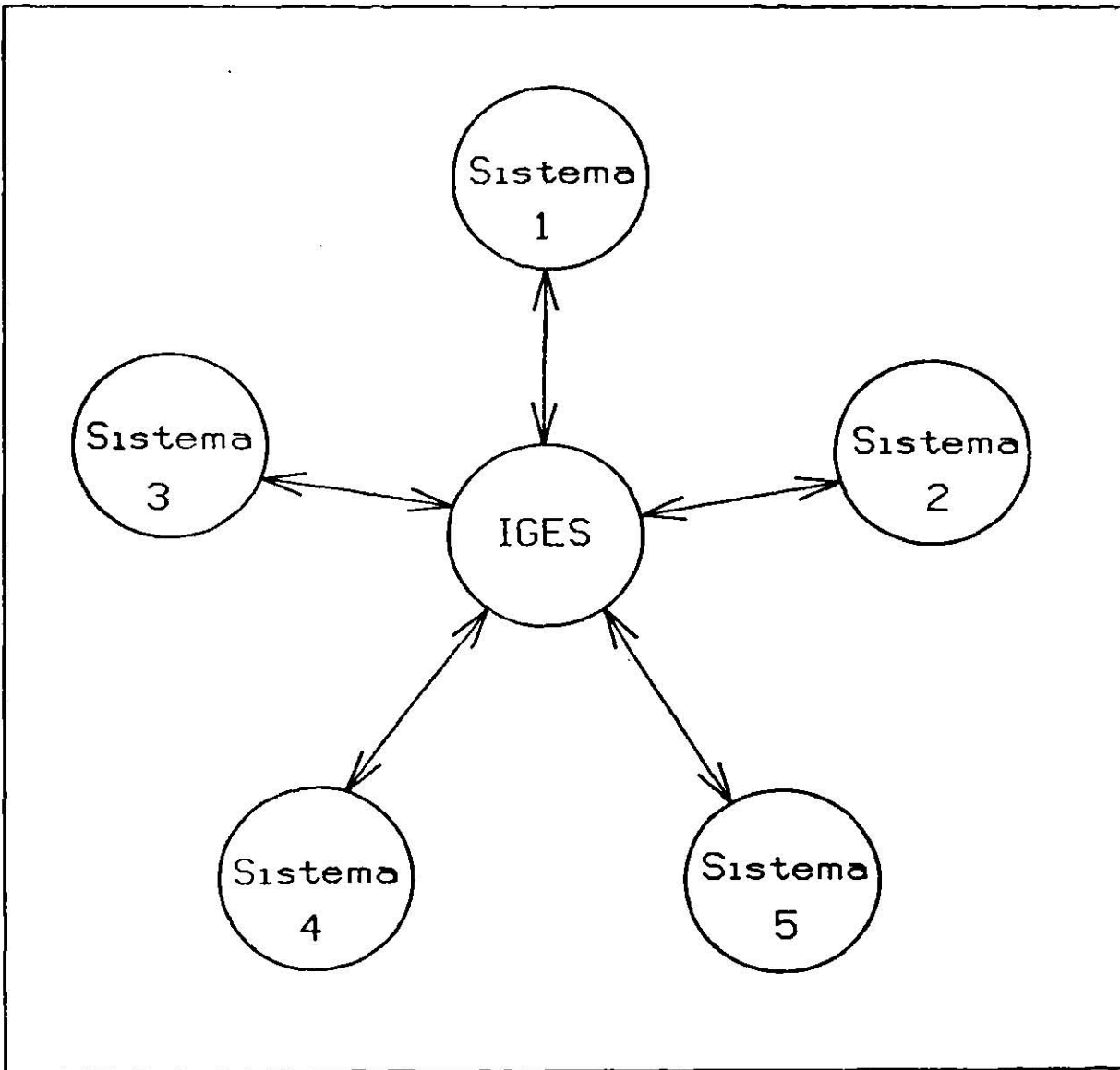


Fig. 6.2 El intercambio de información utilizando un archivo neutral se realiza en dos pasos y requiere dos traductores. La comunicación entre 4 sistemas utiliza 8 traductores, un quinto sistema obliga a contar, solamente, con 2 traductores adicionales.

Otro de los beneficios del archivo con formato neutral es que asegura que los archivos "historia" guardados en este formato por largo tiempo, pueden ser, en cualquier momento, activados de nuevo utilizando un sólo traductor, y trabajarlos, aun, en algún sistema de dibujo mucho mas moderno que en el cual fueron creados originalmente.

6.1 Formatos neutrales.

Principalmente, seis formatos neutrales se encuentran disponibles para el intercambio de información entre sistemas:

- a) IGES (Initial Graphics Exchange Specification) Implementado y proyectado.
- b) XBF2
- c) SDF
- d) SET (Standard D'Exchange et De Transfert)
- e) SIF
- f) ANSIS

No existe un consenso nacional o internacional que determine la elección de alguno de ellos como el mejor formato para el intercambio de información. De acuerdo a la evolución de éstos, serán los usuarios quienes determinen cuál de ellos tiene un mejor funcionamiento y cuáles parámetros son críticos en su definición.

- IGES implementado (IGES versión 2.0) incluye geometría en alambrión, superficies y anotaciones, todo ésto forma un subconjunto de la definición completa del formato IGES.
- IGES proyectado incluye todo lo tratado en las versiones 2.0 y 3.0 y una parte experimental de sólidos.
- XBF2 fue desarrollado por Computer Aided Manufacturing International Inc. (CAM-I), está basado en IGES y tiene capacidad de trabajo con sólidos.
- SDF es también similar al IGES, sin embargo maneja exclusivamente geometría, fue desarrollado por Vought Corp.
- SET es un formato desarrollado por la compañía Aerospatiale de Francia, se encuentra, ahora, antes del formato de la International Standards Organization (ISO) en competencia con IGES como formato internacional. La definición de los elementos en SET es similar al IGES pero tiene una estructura diferente, mucho mas compacta.
- SIF fue desarrollado por Intergraph Corp. como su formato estándar para el intercambio de información.
- ANSIS, que es una sección de el estándar estadounidense Y14.26M (IGES), fue desarrollado junto con éste pero de una forma diferente. Este formato no puede trabajar anotaciones.

La tabla 1 muestra solamente si la información puede ser definida, no se incluye referencia alguna sobre los problemas de proceso. Cada formato tiene sus puntos fuertes y sus puntos débiles. IGES proyectado, XBF2 y ANSIS por ejemplo, tienen capacidades similares en la definición de la geometría de un archivo de diseño, sin embargo no tienen iguales características en el manejo de la información correspondiente a sus bases de datos.

	Formatos						
	IGES implementado	IGES proyectado	XBF2	SDF	SET	SIF	ANSI 5
Geometría 3D	●	●	●	●	●	●	●
Sólidos	○	●	●	●	●	●	●
Relación BD / Geometría	●	●	●	●	●	●	○
Base de Datos (BD)	●	●	●	●	●	●	○
Información de Manufactura	○	●	●	●	●	●	○

●	Necesidad cubierta
◐	Necesidad parcialmente cubierta
○	Necesidad no cubierta

Tabla 1. Comparación de características entre formatos neutrales.

6.2 Alternativas actuales.

El éxito o fracaso de los estándares en formatos depende, en gran parte, de quienes son las organizaciones que los desarrollan y los soportan. La mayoría de los usuarios de sistemas gráficos se inclinan por formatos aceptados internacionalmente, de ahí se puede observar que la competencia final, actualmente, se centra entre los formatos IGES y SET.

IGES es, en estos momentos, el formato mas popular en Estados Unidos; SET, por otro lado, está adquiriendo gran popularidad entre los países de Europa.

Los otros 4 formatos mencionados fueron desarrollados por compañías particulares para resolver problemas de comunicación internos.

6.3 IGES (Initial Graphics Exchange Specification).

IGES ha estado ganando aceptación mundial como el formato más popular para la transferencia de información entre sistemas gráficos. IGES fue desarrollado en 1979 bajo la dirección de National Bureau of Standards (proyecto ASC Y14, Engineering Drawings and Related Documentation Practices), y fue aprobado como su estándar en enero de 1980 con el objetivo principal de facilitar la transferencia de información entre sistemas gráficos. El 21 de septiembre de 1981 la American National Standards Institute (ANSI) aprobó IGES como parte de su estándar Y14.26M, "Digital Representation for Communication of Product Definition Data".

Hasta el momento se han publicado 3 versiones del formato, y se espera la publicación de una cuarta versión para fines de 1987.

IGES

<i>versión</i>	<i>fecha</i>
<i>1.0</i>	<i>1981</i>
<i>2.0</i>	<i>1982</i>
<i>3.0</i>	<i>1986</i>
<i>4.0</i>	<i>1987</i>

Tabla 2. Publicaciones del formato IGES.

El comité IGES, que inicialmente estuvo formado por 3 organizaciones voluntarias, ahora alcanza a tener 100 compañías y 600 miembros.

IGES fue diseñado originalmente para corregir el problema ocasionado por los traductores directos, ya que IGES trabaja con base en un archivo neutral. Un beneficio directo de los archivos neutrales es que éstos pueden ser almacenados durante largos períodos de tiempo, el comité IGES tiene un acuerdo de compatibilidad que significa que archivos IGES creados hoy, podrán ser accesado por los nuevos sistemas CAD/CAM que se vayan desarrollando.

IGES, por sí mismo, es tan solo un documento descriptivo de la información que debiera escribirse en un archivo de datos, los programadores deben crear los procesadores para traducir la información de sus archivos, a formato IGES y viceversa.

El proceso de traducción de datos através de IGES se compone de dos elementos: el preprocesamiento, y el postprocesamiento.

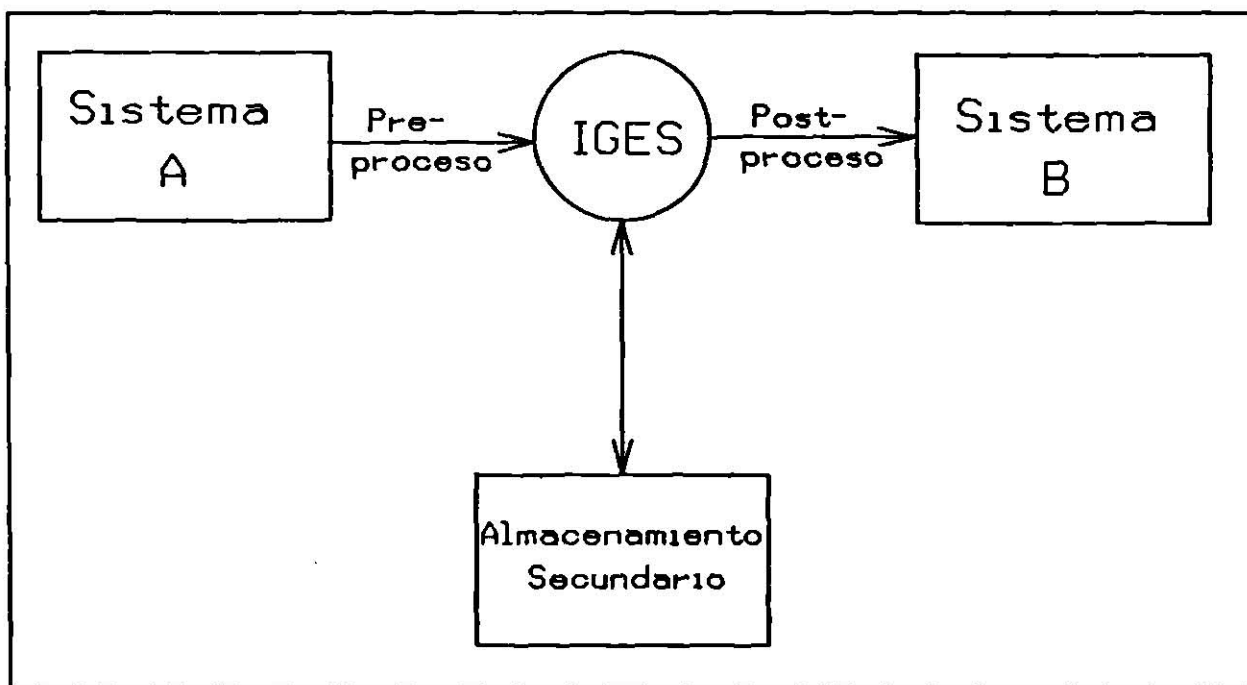


Fig. 6.4 Comunicación vía formato IGES.

6.3.1 Formato de archivo IGES.

Como en la mayoría de los sistemas CAD/CAM, IGES está basado en entidades. Las entidades varían desde objetos geométricos simples, como puntos, líneas y círculos, hasta algunos mas sofisticados como dimensiones, celdas y superficies esculpidas.

Las entidades en IGES se encuentran divididas en 3 categorías: geometría, anotaciones y estructura. Las entidades de geometría como líneas, círculos y superficies definen objetos. Las entidades de anotación incluyen las dimensiones y textos, y en general todos los elementos que auxilian la visualización y documentación de un objeto. Las entidades de estructura se forman de las diversas maneras en que los sistemas CAD/CAM combinan algunas otras entidades para facilitar su uso. Un ejemplo de una entidad de estructura viene a ser una subfigura (también conocida como block, celda o ditto en diferentes sistemas CAD/CAM), las cuales son almacenadas en bibliotecas para ser llamadas desde los dibujos que la necesiten; por ejemplo, una sólo subfigura puede estar formada de todos los elementos que se necesitan en el trazo de una llanta, ahora, al realizar el dibujo de un coche, sólo se hará el llamado a la subfigura "llanta" en cuatro ocasiones, sin necesidad de dibujarla cuatro veces.

En su inicio, el formato IGES estuvo sólomente definido en un código ASCII, accesible por el usuario, y utilizando registros de 80 caracteres. Buscando reducir el tamaño de los archivos (los archivos IGES típicamente son de 5 a 10 veces mas grandes que el archivo de diseño que representan), se definió un formato ASCII reducido y un formato binario, sin embargo la mayoría de los procesadores IGES actuales, aun soportan sólomente el formato ASCII original.

Los archivos son divididos en 5 secciones: sección inicio, sección global, sección directorio, sección parámetros, y sección término.

La sección inicio está compuesta sólomente por el texto utilizado en la propia documentación del archivo IGES.

La sección global está compuesta por 24 parámetros de naturaleza global, como son: nombre del archivo, autor, fecha de creación, unidades de medición, etc

La sección directorio (DE) contiene la información que define los parámetros de cada una de las entidades en el archivo, parámetros como tipo de entidad, color, tipo de línea, espesor, nivel y una matriz de transformación para posicionar la entidad en archivos de 3 dimensiones.

La sección parámetro contiene información específica para cada entidad, por ejemplo, para líneas, contiene las coordenadas x,y,z de sus puntos extremos, para superficies, contiene sus valores x,y,z y los vectores de definición de la superficie.

Se requieren dos líneas dentro de la sección directorio por cada entidad, y una o más líneas en la sección parámetro por cada entidad. Ya que los archivos IGES típicamente contienen miles de entidades, estas secciones constituyen el grueso de el archivo.

La sección término está formada por una sólo línea al final del archivo que contiene el número de registros (o líneas) en cada sección.

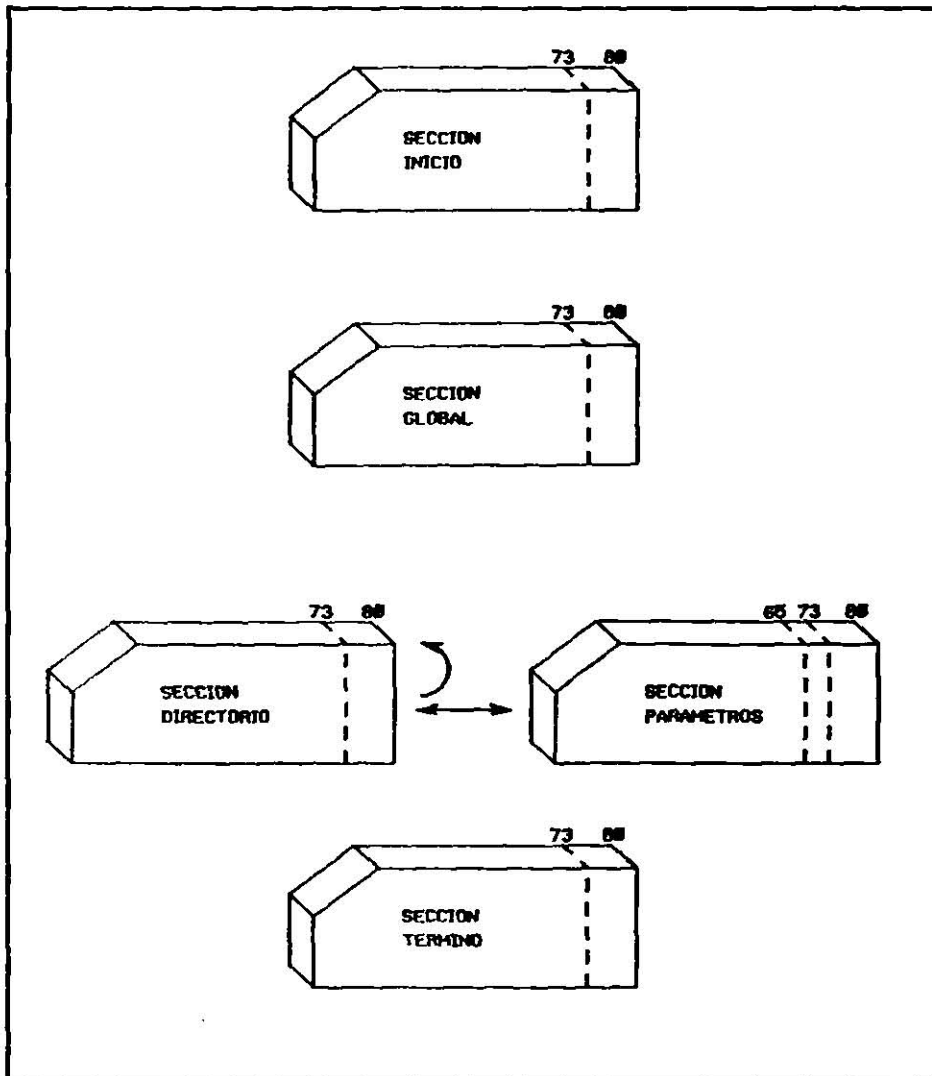


Fig. 6.5 Los archivos IGES se dividen en cinco secciones: inicio, global, directoria, parámetro y término.

El 'Documento IGES' describe el formato para cerca de 50 diferentes entidades, muchas de las cuales presentan varias opciones alternas, lo que resulta en cerca de 150 posibles formatos. Todas las entidades tienen la misma información definida en la sección directorio, sin embargo la información en la sección parámetro puede variar desde simples valores x,y,z que definen un punto, hasta cientos de valores que definen una superficie esculpida. Algunas entidades contienen texto, el cual es utilizado en dimensiones o notas en un dibujo, así mismo, algunas entidades pueden hacer referencia a otras formando así estructuras.

Las entidades en la versión 1.0 de IGES fueron básicamente entidades de alambrión y algunas superficies, ésto es, básicamente las encontradas en los sistemas CAD/CAM del final de los años 70's. IGES ha evolucionado, y la versión 3.0 incluye, no sólomente otras superficies adicionales sino además algunas entidades para aplicaciones específicas como modelación por elemento finito (FEM), diagramas eléctricos, diseño de tarjetas de circuitos impresos, diseño de plantas y diseño arquitectónico. La versión 4.0, la cuál se espera esté lista a fines de 1987, también soportará modeladores de sólidos. En general IGES tiende a formar un superconjunto de toda la información que manejen los sistemas CAD/CAM del momento.

6.3.2 Situación actual de IGES.

IGES es soportado por mas de 30 distribuidores de sistemas gráficos, es, generalmente, el primer método utilizado por éstos al desarrollar software para el intercambio de información entre sistemas.

Lo que un usuario espera al realizar un intercambio de información, es tomar un archivo gráfico, traducirlo a formato IGES, leer ese archivo desde otro sistema, y tener, en éste, el 100% de la información original.

La práctica muestra que algunos procesadores traducen el 100% de la información, sin embargo, algunos otros fallan y traducen muy poca o ninguna información.

La calidad de los procesadores IGES varía grandemente. Algunos son capaces de manejar superficies de revolución o superficies esculpidas, mientras que otros manejan exclusivamente líneas, puntos y arcos. Algunos procesadores son flexibles, y tienen opciones para leer archivos generados por diferentes sistemas, mientras que otros sólo pueden leer archivos generados por sus propios sistemas.

La mayoría de los procesadores IGES manejan, como requisito mínimo, líneas, puntos y arcos, ya que éstas forman las entidades básica para cualquier sistema de dibujo por computadora, el soporte de las otras entidades varía grandemente de procesador a procesador; la única manera de estar seguro de las características de un procesador IGES es el realizar pruebas exhaustivas con información real.

Un método efectivo para evaluar una transferencia de información vía IGES es el preparar 2 o 3 ejemplos prácticos reales en donde se incluyan diferentes tipos de elementos gráficos. Es necesario comparar los diseños en los 2 sistemas (origen y destino), así mismo manipular la información, editar textos, realizar mediciones, verificar unidades, realizar escalamientos y rotaciones de la información gráfica, encontrar intersecciones entre elementos y superficies, etc.

La mayoría de los problemas con los procesadores IGES son originados por la forma en que éstos son implementados, cada proveedor de sistemas CAD/CAM desarrolla sus propios procesadores IGES, algunos lo hacen mejor que otros, después de todo la organización IGES es una organización voluntaria y los programadores de cada una de éstas incluyen tanto de las especificaciones del formato como mejor les parezca. Debido a la gran variedad de información definida en IGES, no existe algún procesador que soporte todas ellas, más aún, la mayoría soporta menos de la mitad. El problema se complica por las diferencias de interpretación que se tienen de las mismas especificaciones del formato IGES.

Existe también otro problema, la certificación de los procesadores IGES. El comité IGES pertenece a la National Bureau of Standards de los Estados Unidos, la cual prohíbe realizar acciones de certificación de productos, más aun, en vista de posibles problemas legales, ninguna organización de estándares se atreve a afirmar que el producto de cierta compañía cumple o no con un estándar particular. Sin embargo, para cubrir este problema, el comité IGES se encuentra trabajando en conjunto con la Society of Automotive Engineers (SAE) en el desarrollo de un programa de validación de procesadores. Bajo este programa, SAE certificará que un procesador IGES particular trabaje tan bien como su creador lo asegura. Este programa ya se debe encontrar listo, se tenía programado iniciar su operación durante el verano de 1987. La información utilizada en la prueba de los procesadores IGES será del dominio público y estará disponible para cualquier usuario o programador de procesadores interesado en realizar alguna prueba personalmente. Sin embargo, reiteramos; el grado de sofisticación de un procesador IGES depende totalmente de sus creadores.

6.4 SET (Standard D'Exchange et De Transfert).

El formato SET trabaja en base a un archivo neutral que contiene, esencialmente, el mismo tipo de información que el formato IGES. SET fue desarrollado por la compañía aérea francesa Aerospatiale, los trabajos fueron iniciados en 1983. SET ha sido utilizado, inicialmente en la transferencia de información relativa al diseño de aeroplanos, en particular, lo relativo al diseño de el nuevo jet Airbus 320 (proyecto A320), en donde participaron 4 compañías aéreas europeas. (Aerospatiale-Francia, British Aerospace-Inglaterra, CASA-España, Messerschmidt Balkow Blohm-MBB-Alemania Occidental).



Fig. 6.6 Proyecto A320 de Aerospatiale.

Cada una de estas compañías cuenta con su propio sistema gráfico diferentes entre sí, por ejemplo, British Aerospace utiliza Anvil 400, MBB utiliza CADAM, CASA utiliza Computervision CADD5 4X, y Aerospatiale utiliza CADD5 4X, CDM 300, CADD5 3 y su propio software en donde se incluyen SIGMA (sistema para el diseño y manipulación de formas complejas), RIFE (sistema de diseño de circuitos impresos) y SAO (lenguaje gráfico que comunica y define las especificaciones en el sistema de diseño).

La principal ventaja de SET es el formato del archivo, el cual es mucho más compacto que su equivalente en IGES. SET utiliza el concepto de blocks y subblocks (equivalente a las entidades en IGES). Un block de dimensionamiento en SET contiene la información de la sección directorio en IGES, además de 5 subblocks que contienen el texto y las líneas de cota.

Antes de dar inicio a la definición del formato SET, el personal de Aerospatiale verificó el funcionamiento de IGES, sin embargo, la versión de IGES disponible en 1983 no trabajó lo bien que se quisiera, los traductores (preprocesadores y postprocesadores) eran pobres y se limitaban a traducir solamente puntos, líneas, arcos y textos, es decir, manejaban exclusivamente lo relativo a los dibujos de detalle; por lo tanto se concluyó que la inclusión de IGES en el proyecto A320 era inadecuada, no era conveniente tomar ese formato como el estándar de intercambio de información.

Entonces se definieron 2 objetivos principales que debería cumplir SET: servir como el formato para el intercambio de información gráfica y no gráfica entre los sistemas CAD/CAM/CAE y, por otro lado, utilizar un formato compacto, ya que existen miles de dibujos que deben ser almacenados durante años y deben de poder ser traducidos sin riesgo alguno en los modernos sistemas gráficos que se vayan desarrollando.

En marzo de 1984 es publicada por AFNOR (Association Francaise de Normalisation), la organización francesa de estándares, la primera especificación del formato SET conocida como la versión 1.1. Esta versión cubría lo relativo al intercambio de superficies, alambres 3D, puntos, líneas, arcos, splines y, en general, todo lo necesario en un dibujo de detalle.

Las primeras pruebas con SET 1.1 mostraron un ahorro en el tamaño de el archivo neutral de entre 7 y 10 veces con relación al formato IGES, el formato SET es mucho más compacto, así mismo, el tiempo en que SET realiza la traducción es al menos 3 veces menor que el utilizado en IGES.

Las pruebas de comparación de formatos realizadas entre octubre de 1984 y enero de 1985 fueron detenidas cuando se comprobó el óptimo funcionamiento de SET sobre IGES. En febrero de 1985 el comité coordinador del proyecto A320 decidió poner a SET en operación, nombrándose un grupo supervisor de la implementación.

El proyecto A320 es la primera gran aplicación de SET, el segundo es el diseño de Hermes, el transbordador espacial europeo programado para su lanzamiento en la década de los 90's.

SET es un proyecto de largo alcance, los ingenieros de Aerospatiale están trabajando en incluir, en versiones futuras, lo relativo al intercambio de superficies complejas (o superficies esculpidas), modelado de sólidos modelado por elemento finito y control numérico.

Hoy en día, traductores al formato SET, han sido desarrollados, o se encuentran en desarrollo, por parte de las compañías proveedoras de sistemas gráficos. Se estima que toma, aproximadamente, un año-hombre de trabajo el desarrollo y prueba de un traductor a SET para cada sistema gráfico CAD/CAM/CAE. Las especificaciones se encuentran definidas y abiertas para que cada proveedor de sistemas CAD/CAM/CAE, los tome, desarrolle sus traductores específicos, los distribuya, y los soporte.

Actualmente, en los Estados Unidos, no existe algún proveedor de sistemas gráficos que soporte SET, sin embargo, SET, como un estándar francés ha sido propuesto como estándar europeo internacional.

Queda por verse, si en un futuro SET se convierte en una alternativa en el duro mercado estadounidense.

6.5 CGM (Computer Graphics Metafile).

ANSI señala un conjunto de elementos básicos que integran la información manejada por los sistemas gráficos computacionales, este conjunto es conocido como Computer Graphics Metafile (CGM). El diseño de este estándar se encuentra basado en el trabajo de varios grupos, principalmente se encuentra influenciado por el trabajo de el Comité de Planeación de Estándares Gráficos de el Grupo Especial de Interés en Gráficas Computacionales de la Asociación de Maquinaria Computacional (ACM-SIGGRAPH GPSC). Este trabajo conocido como sistema CORE, contiene una propuesta para un metarchivo que fue difundida en 1979.

El CGM fue originalmente desarrollado por el comité técnico X3H3 en marzo de 1981, posteriormente, en el período 1982-1984, fue refinado en cooperación con el Grupo de Trabajo 2 de el Subcomité en Lenguajes de Programación de el Comité Técnico en Procesamiento de Información de la Organización Internacional de Estándares (ISO TC97/SC21/WG2).

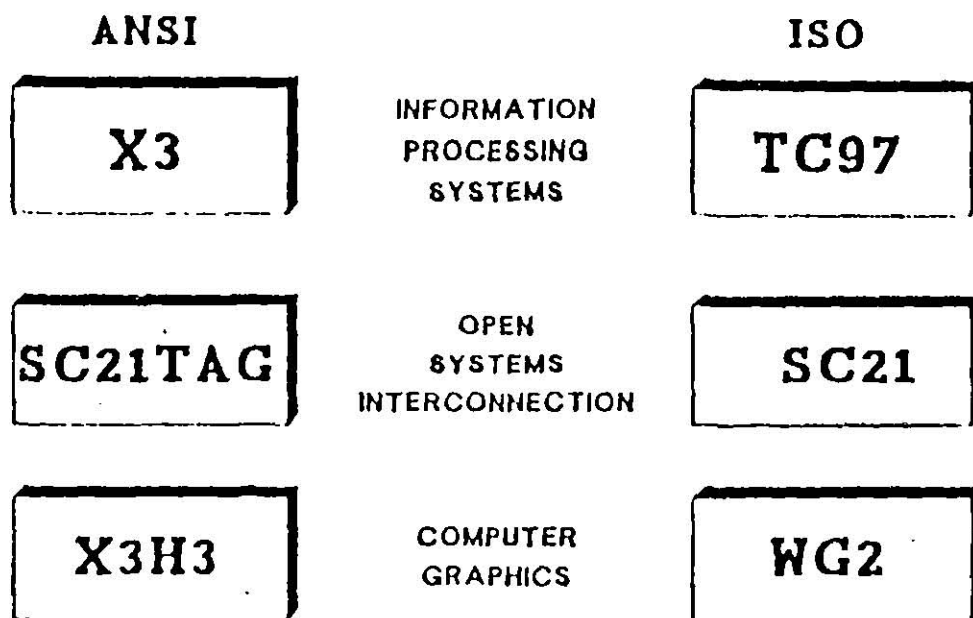


Fig. 6.7 Relación entre los grupos de trabajo de ANSI e ISO.

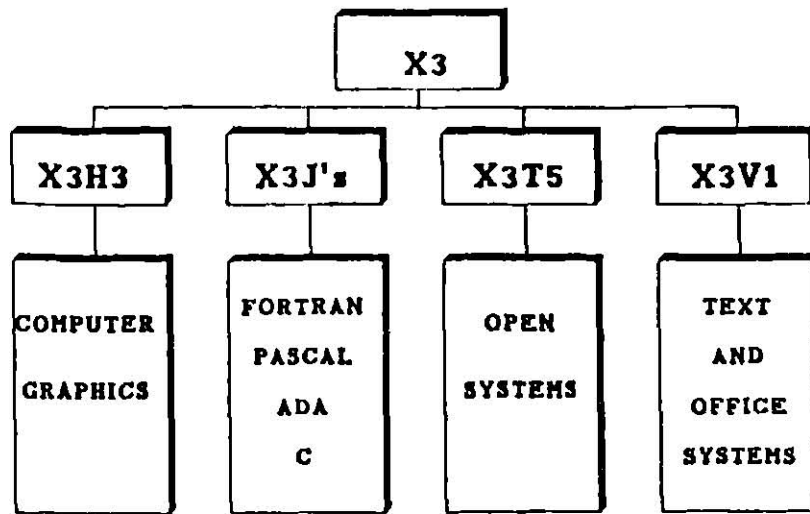


Fig. 6.8 Comités técnicos relacionadas con el área de sistemas de información en ANSI.

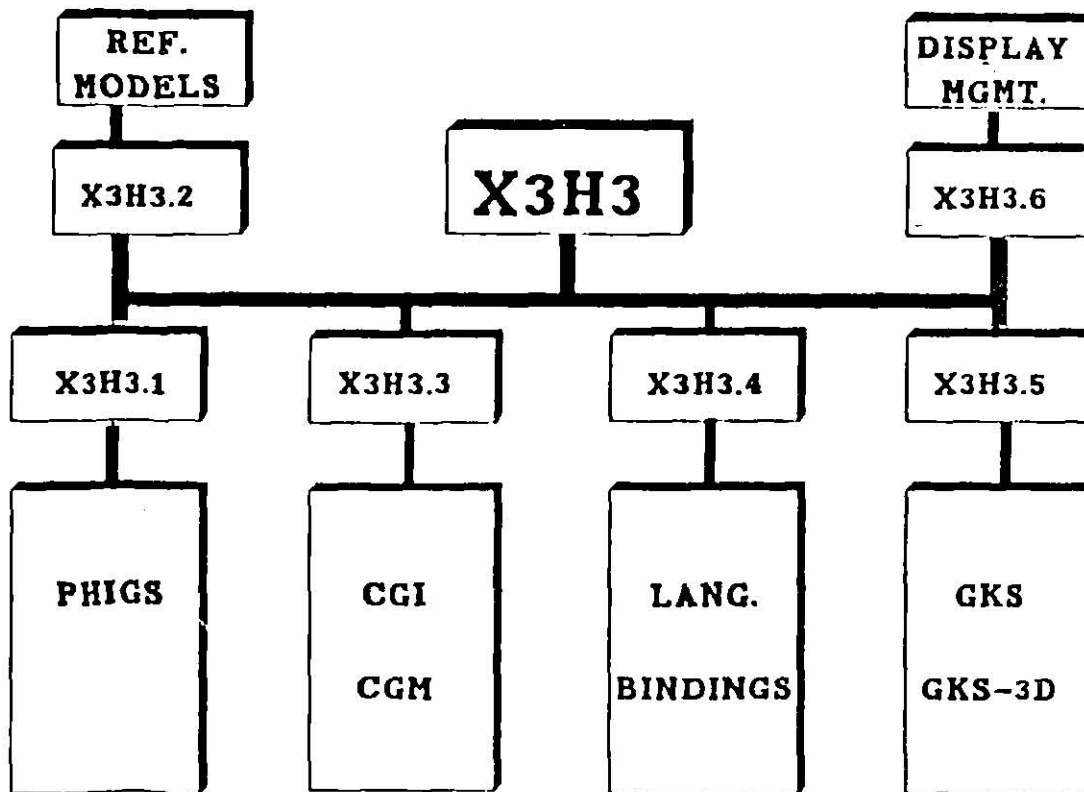


Fig. 6.9 Comités técnicos relacionadas con el área de computación gráfica en ANSI.

El nombre de este estándar fue modificado de Virtual Device Metafile (VDM) a Computer Graphics Metafile como resultado de la resolución transmitida por ISO TC97/SC21/WG2 en junio de 1984. CGM pretende ser una interface en el nivel de información gráfica, así la industria gráfica computacional contará con una definición estándar de un archivo para la captura, transferencia y almacenamiento de imágenes.

El formato de el metaarchivo consiste de un conjunto de elementos que pueden utilizarse para describir imágenes de una forma compatible entre diferentes sistemas de diferentes arquitecturas, y entre diversos dispositivos que difieren en capacidades y diseño. Así bien, CGM está estrechamente relacionado con las capacidades funcionales de el estándar CGI, sus elementos son definidos consistentemente con el proyecto CGI.

Este estándar promueve el intercambio de información, lo cual facilita el compartir trabajo entre diferentes instalaciones y reduce el tiempo de proceso al regenerar información gráfica. La transferencia de información se puede realizar vía medios magnéticos o vía líneas directas de comunicación.

El conjunto de elementos del CGM utilizan una terminología estandarizada, lo cual permite a las comunidades industriales y académicas el desarrollo de programas de instrucción o capacitación, basados en las técnicas y metodologías de programación que estos elementos utilizan.

Algunos beneficios adicionales derivados de la utilización de el estándar CGM son:

- las salidas gráficas generadas durante una sesión de trabajo pueden ser archivadas para chequeos posteriores.
- la generación de interfaces de un metarchivo con diferentes tipos de plotter se facilita, e igualmente se puede estandarizar.
- la generación de imágenes que formarán una secuencia de animación gráfica, no necesariamente debe ser en el mismo orden en que serán posteriormente desplegadas.
- las mismas imágenes ya generadas pueden ser revisadas varias veces, sin necesidad de volver a realizar el cálculo.

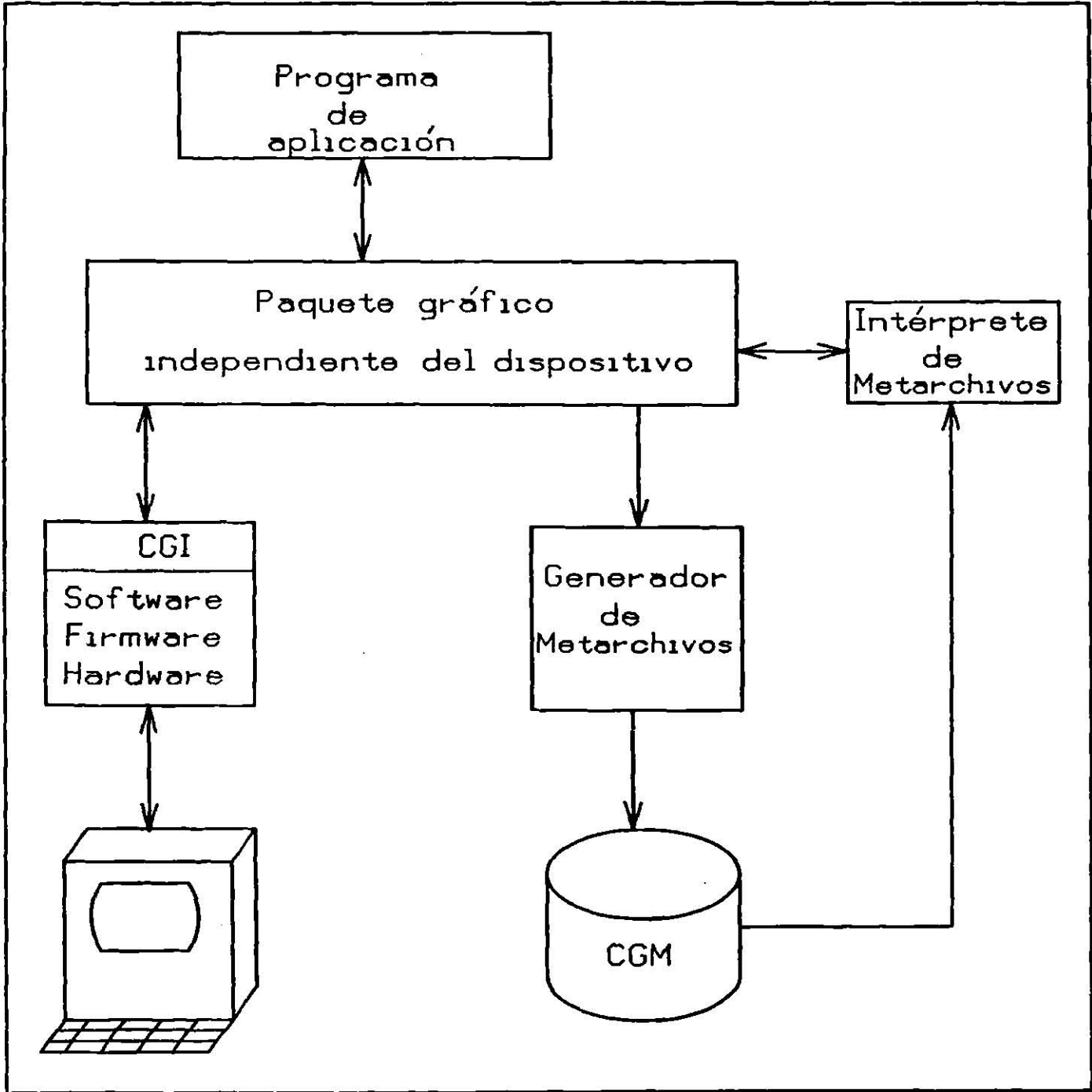


Fig. 6.10 Relación entre CGI y el medio ambiente de aplicación gráfica.

CGM es un estándar para el intercambio de archivos con información de imágenes gráficas exclusivamente, es decir, CGM está involucrado en la generación y transferencia de información gráfica independiente del dispositivo, para ser desplegada en una amplia variedad de equipos. CGM no contempla la transferencia de información de definición de productos, (geométrica o no geométrica) entre sistemas CAD/CAM; ése tipo de definiciones se encuentran contempladas en el formato estándar IGES. Específicamente CGM e IGES trabajan con diferente información, para diferentes propósitos y a diferentes niveles de detalle.

6.6 Formatos de base de datos externos.

La necesidad de transferencia de información entre sistemas CAD/CAM es grande, IGES no ha sido una respuesta perfecta al problema, SET por otro lado, apenas se está incorporando al mercado estadounidense, es por esto que algunas compañías particulares se han inquietado y desarrollado sus propios formatos "estándar".

Algunos ejemplos de éstos formatos, conocidos como formatos de base de datos externos, son el DXF (para AutoCAD), SIF (para Intergraph), Archivos Universales (para I-DEAS), Archivos Neutrales (para PATRAN), etc.

Los formatos de base de datos externos son creados por los proveedores de software CAD/CAM/CAE para brindarles a sus usuarios un camino de acceso a la base de datos de sus diseños, es decir, la utilidad de éstos formatos es para los usuarios quienes se encuentran desarrollando aplicaciones en donde es necesario hacer referencia a información gráfica de algún archivo.

El formato IGES también podría servir para éste objetivo, sin embargo, algunas compañías evitan su uso debido a lo extenso e impráctico del formato.

El formato de base de datos externo es, generalmente, muy similar a el formato interno utilizado por el proveedor, sin embargo, está diseñado para facilitar un acceso rápido a la información. Generalmente se representa en código ASCII, accesable desde cualquier terminal, aun y cuando, normalmente las bases de datos internas se encuentran escritas en código binario. (Ver apéndice B)

Más que todo, debido a la influencia del mercado, algunos proveedores llaman "estándares" a sus formatos de base de datos externos.

La desventaja de éstos formatos externos es que se encuentran totalmente definidos por una sólo compañía, la compañía que los diseña, y si el software de ésta compañía no soporta una entidad particular, por consecuencia, su formato externo tampoco la soportará.

Es por esto que, generalmente, éstos formatos no son utilizados para el intercambio (entrada o salida) de información entre sistemas gráficos; IGES comparativamente, define todo tipo de entidades que puede manejar cualquier software de sistemas CAD/CAM/CAE.

6.7 Alternativas futuras.

Tratando de entender la proyección futura de los estándares, el siguiente paso del IGES es PDES (Product Data Exchange Specification) el cuál está siendo desarrollado por la organización IGES. PDES aspira llegar más lejos que IGES definiendo modelos más conceptuales, utilizando un formato basado en la estructura de datos que utilizan los modeladores de sólidos, así mismo, PDES manejará información no-geométrica como, materiales a ser utilizados en la fabricación, costos, pesos, proveedores, etc. La representación de la información se basará en una arquitectura de 3 niveles: el nivel de aplicación, el nivel conceptual (o lógico) y el nivel físico.

PDES está dirigido para aplicaciones como mecánica, eléctrica, y AEC (Architectural and Engineering Construction). Se tiene planeado tener listo el primer ensayo de las especificaciones del formato para fines de 1987, sin embargo, los procesadores PDES estarán disponibles hasta principios de la década de los 90's.

Definitivamente es un formato orientado a futuro, para trabajar con sistemas CAD/CAE sofisticados que utilicen modeladores de sólidos y aún inteligencia artificial.

Por otro lado la International Standard Organization (ISO) se encuentra estudiando los formatos existentes actuales como el IGES, SET y VDAFS (estándar alemán para el intercambio de superficies) para utilizarlos en la definición de su propio formato STEP (International Standard for the Exchange of Product Definitions) el cual esperan tener listo también para el inicio de la década de los 90's.

7. Caso práctico.

Bajo la misma arquitectura de hardware mostrada en la sección 2.1, Vitro Tec se encuentra trabajando con dos paquetes de software de diseño mecánico de diferentes proveedores, el paquete I-DEAS de SDRC (Structural Dynamics Research Corporation) y el paquete MEDS (Mechanical Engineering Design System) de Intergraph Corporation.

El paquete I-DEAS es un paquete de diseño mecánico, cubre las fases de modelación geométrica, preprocesamiento, análisis y postprocesamiento. La creación de la geometría se realiza utilizando las técnicas de modelación de sólidos (CSG); las etapas de preproceso, análisis y postproceso utilizan la técnica de elementos finitos (FEM, FEA).

La siguiente figura muestra la estructura del paquete que se basa en una estructura totalmente modular.

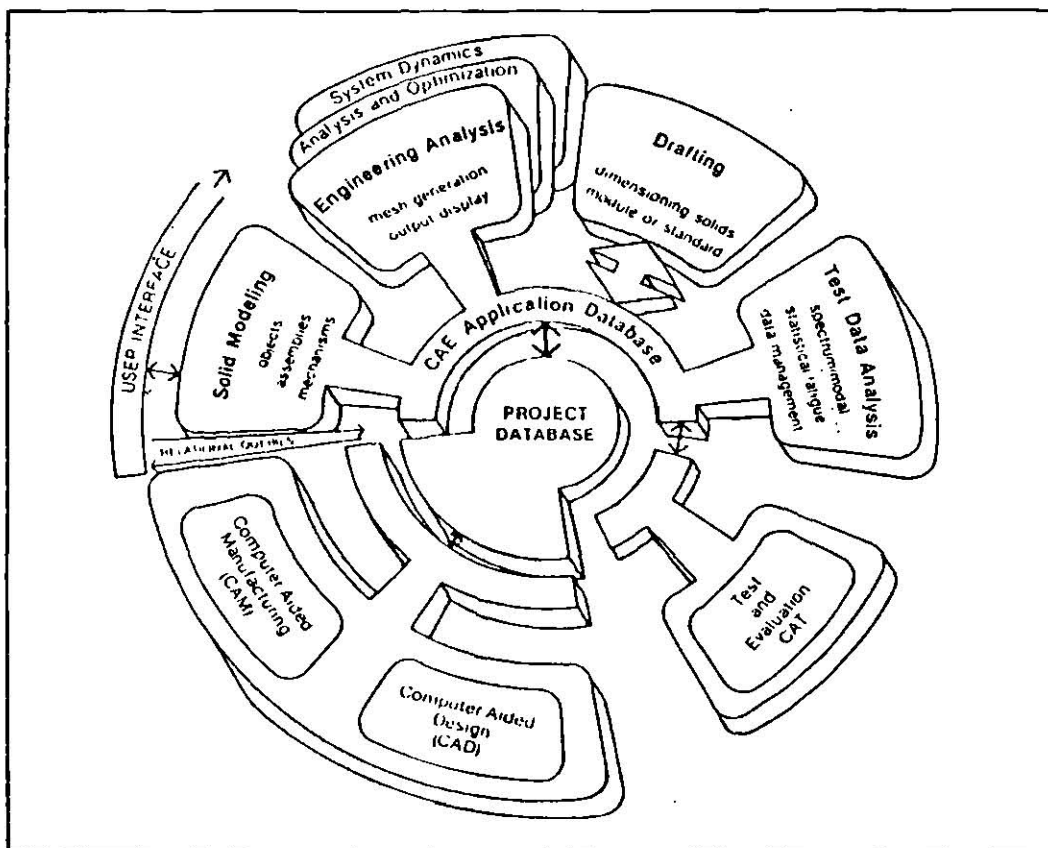


Fig. 7.1 Paquete de software I-DEAS.

El paquete MEDS, por otro lado, trabaja en base a representaciones de figuras en alambrión (2 o 3 dimensiones), y superficies, incluyendo superficies esculpidos. (surface modeling)

La siguiente figura muestra la ubicación del paquete MEDS dentro de la configuración de software de una computadora VAX.

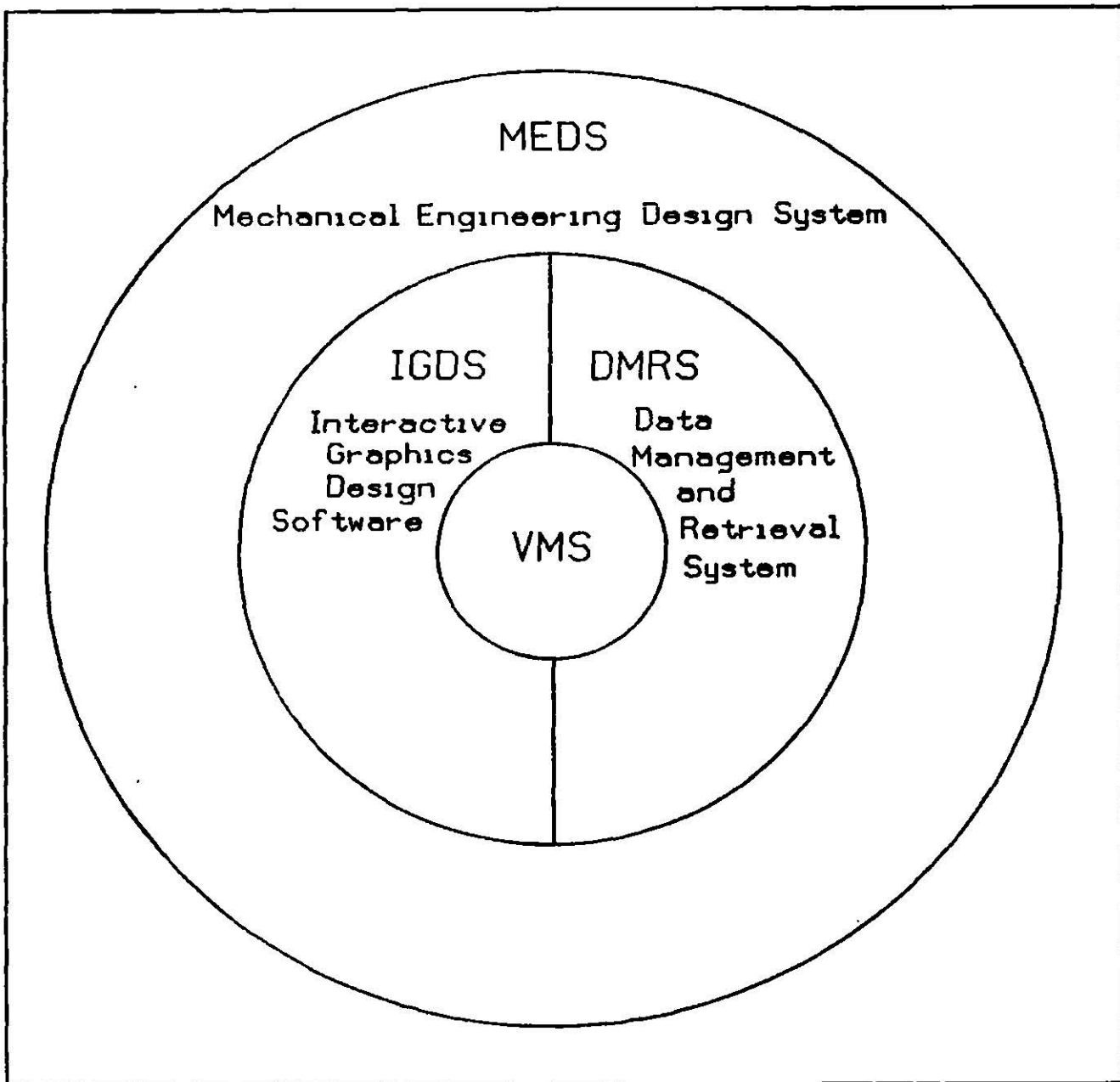


Fig. 7.2 Paquete de software de diseño mecánico en una computadora VAX.

El siguiente diagrama muestra los pasos necesarios en la conversión de un archivo de trabajo creado con Geomod para que pueda ser accesado por MEDS, y viceversa.

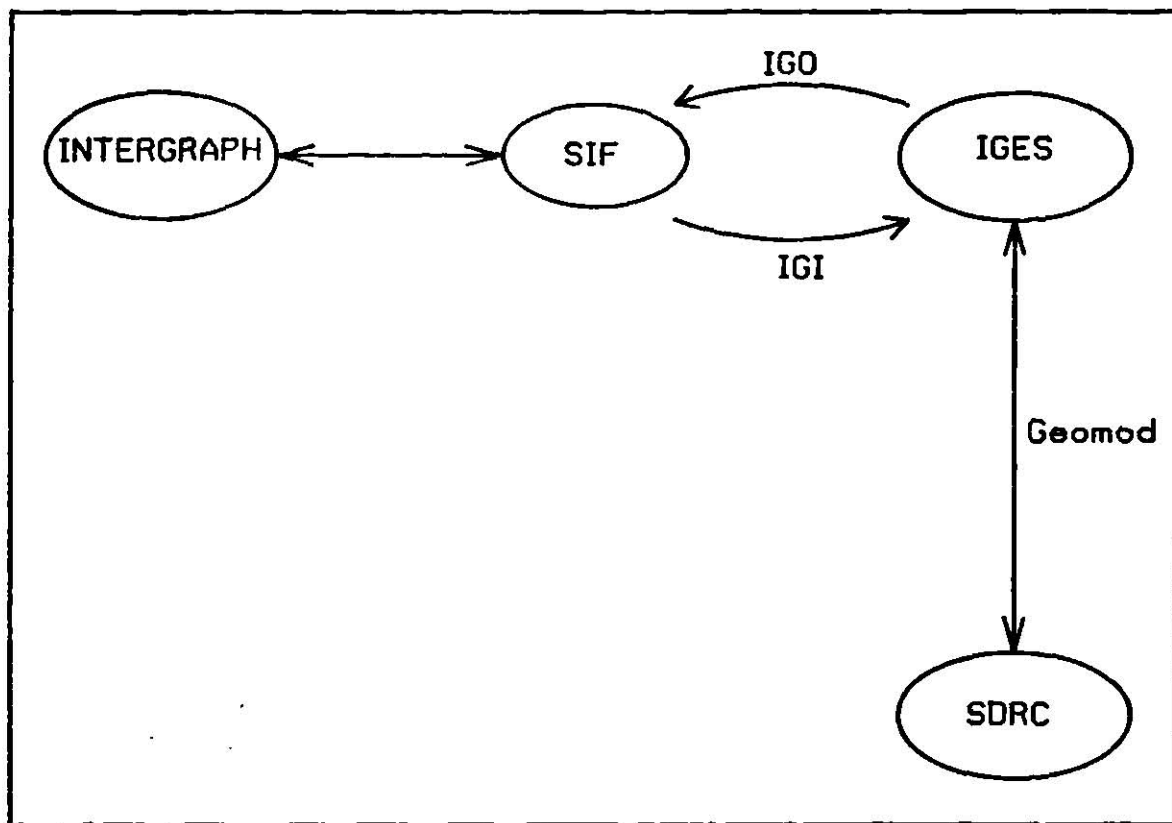


Fig. 7.4 Fluja de información en la transferencia MEDS - I-DEAS.

El nombre colocado sobre cada flecha indica el paquete de software utilizado para pasar de un formato a otro.

Como se puede ver, el cambio del formato SDRC al formato IGES se realiza en un sólo paso utilizando solamente el Geomod, es decir, el mismo software que se utiliza en la modelación.

Por otro lado, en la conversión de un archivo en formato IGES a formato Intergraph se necesitan 2 procesadores: el IGES y el ISIF. Ambos son productos Intergraph, pero se manejan aparte del paquete MEDS.

En el sistema Intergraph, el procesador IGES es totalmente dependiente del procesador ISIF (Intergraph Standard Interchange Format), es decir, el procesador IGES no puede trabajar sin ISIF.

Como se mencionó antes, el formato SIF es un formato de base de datos externo, ASCII, creado por Intergraph para brindar a sus usuarios un medio de acceso a las bases de datos de sus archivos gráficos, ahora bien, el formato SIF también representa un paso obligatorio en el camino de transferencia de información vía IGES.

El procesador ISIF se compone, a su vez, de 3 procesadores: GTA, ASI y TRI. Tanto el procesador ASI como el TRI realizan la conversión de formato de SIF a IGDS (formato propio de Intergraph), a su vez el procesador GTA realiza la traducción en el sentido inverso.

Tres son los archivos de trabajo que utiliza el procesador ISIF:

- archivo.ASC define las especificaciones de creación para el archivo SIF.
- archivo.BIN archivo de control requerido por los tres procesadores, contiene especificaciones y características requeridas por los procesadores.
- archivo.SIF archivo con información gráfica escrito en el formato neutral SIF.

El procesador IGES de Intergraph está, también, compuesto de dos procesadores: IGI e IGO. Los archivos IGES pueden ser accedidos, procesados y finalmente traducidos a formato SIF mediante el procesador IGES IGI. Así mismo archivos IGES pueden ser creados, partiendo de un archivo SIF, al utilizar el procesador IGES IGO.

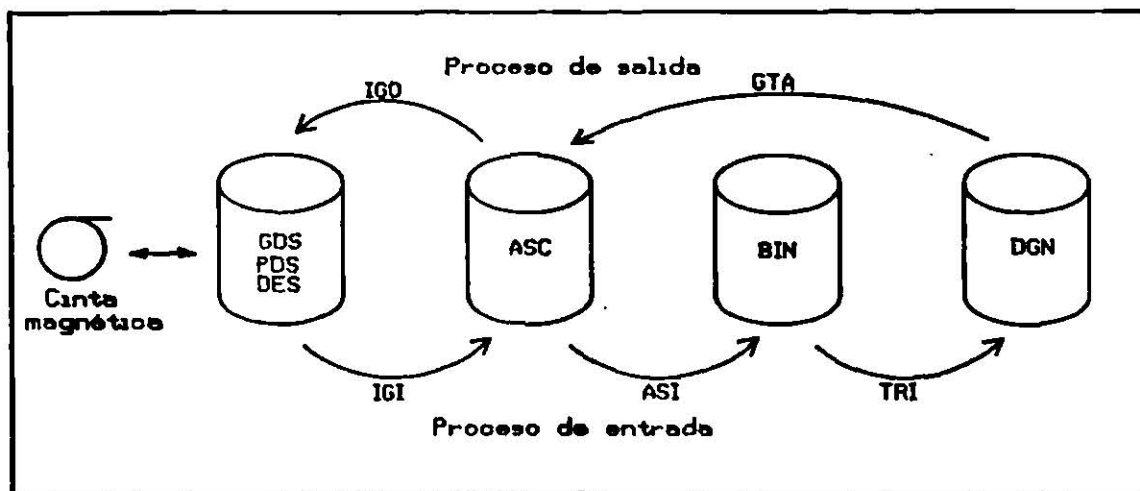


Fig. 7.5 Archivos de trabajo necesarios en la transferencia de información en un sistema Intergraph.

Existen tres archivos que son con los que trabaja el procesador IGES de Intergraph. Como se mencionó antes, en un archivo IGES la información se encuentra clasificada en cinco secciones, pues bien, la información de estas cinco secciones se encuentra dividida en los tres archivos de la siguiente manera:

- la sección global y la sección inicio se encuentran en un archivo con la extensión GDS.
- la sección directorio se encuentra en un archivo con extensión DES.
- la sección parámetro y la sección término se encuentran contenidas en un archivo con extensión PDS.

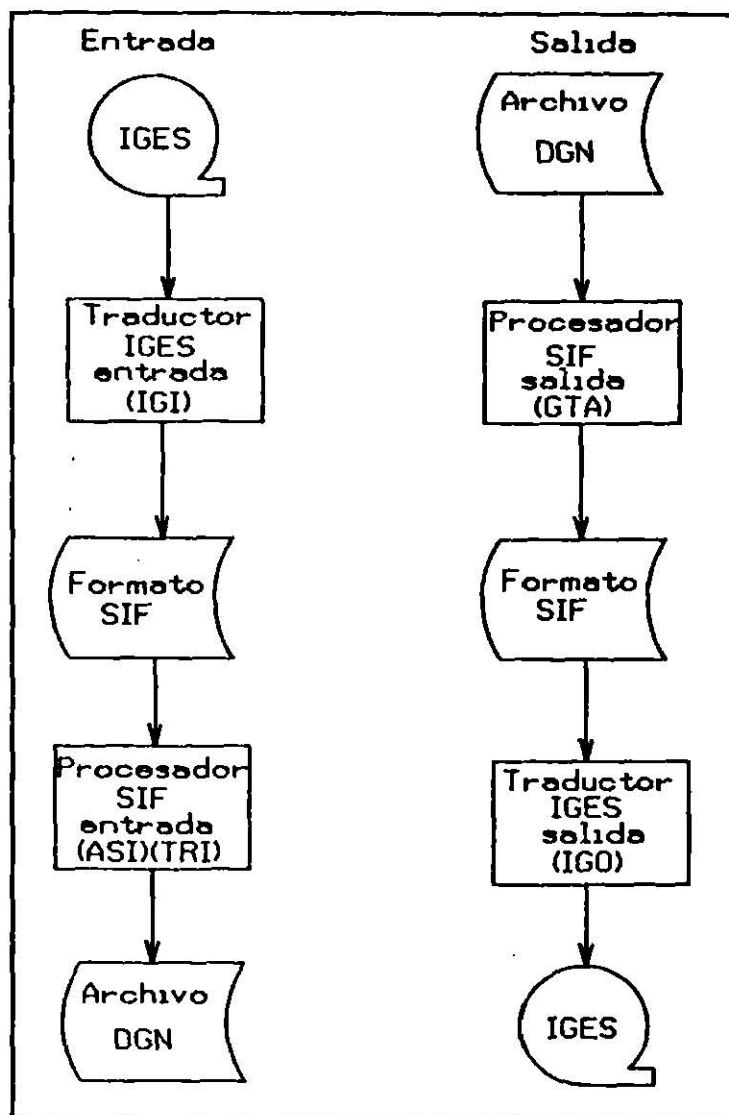


Fig. 7.6 Proceso de transferencia de información en un sistema Intergraph, utilizando sus procesadores IGES e ISIF.

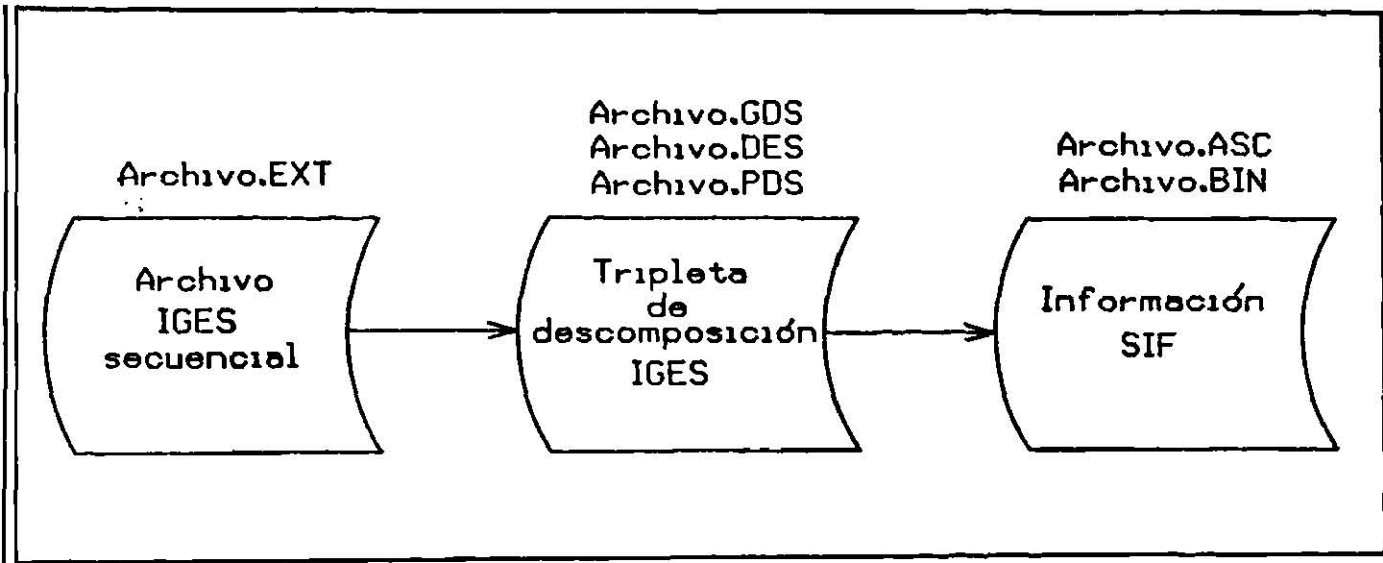


Fig. 7.7 Procesador IGI.

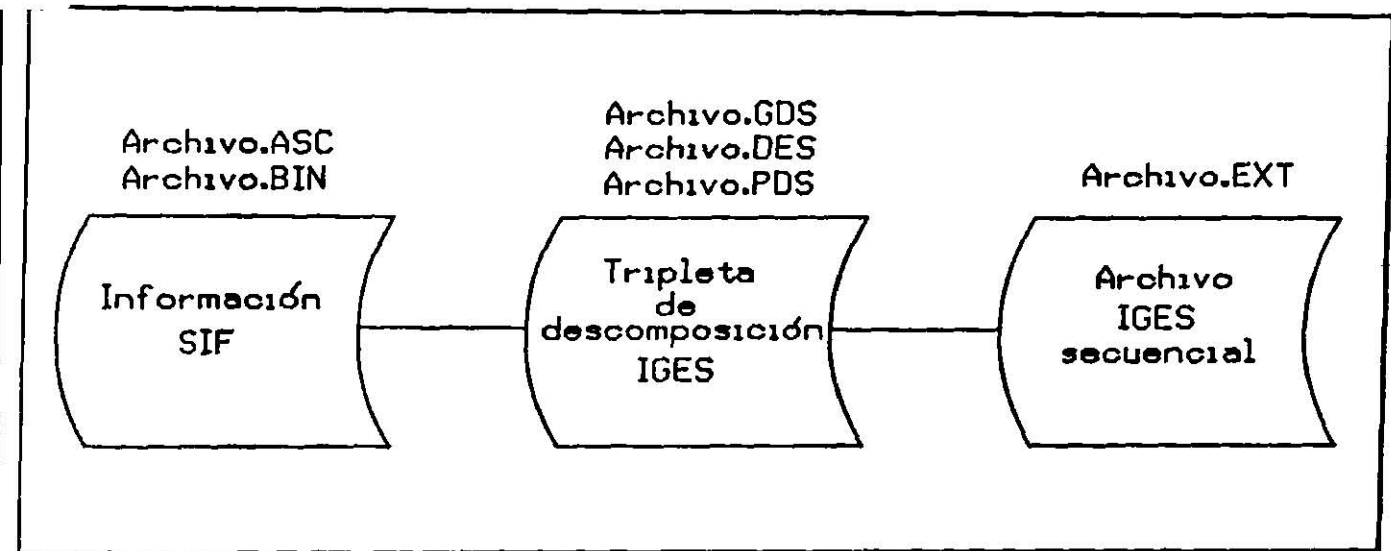


Fig. 7.8 Procesador IGI.

8. Conclusiones.

Definitivamente el desarrollo tanto de hardware como de software por diferentes caminos, manejado por las diferentes compañías interesadas, vino a resultar en una gran variedad de sistemas que en la mayoría de los casos no "hablan" el mismo lenguaje. Es evidente que la estandarización en sistemas gráficos es necesario y benéfica.

Entre todos los temas que se encuentran alrededor de los estándares gráficos y que, por lo tanto, se incorporan al desarrollo y utilización de los sistemas CAD/CAM, dos son los que resaltan por su relevancia: productividad y funcionalidad.

Es así que surgen las preguntas:

- ¿Efectivamente los estándares aumentarán la productividad?
- ¿La aplicación de estándares, degradará la funcionalidad hasta niveles inaceptables?

Definitivamente no es posible dar una respuesta global, los estándares deben analizarse por separado. Es evidente que, para el caso particular del IGES, que representa el estándar mayormente utilizado en la transferencia de información, tanto la productividad como la funcionalidad se ven seriamente afectados. Muchas inconformidades existen alrededor de IGES.

Se ha observado que la publicación de estándares es una tarea que toma buen tiempo en realizarse, de tal manera que cuando éste se concluye, los sistemas gráficos ya han evolucionado, el estándar se encuentra limitado en relación a los sistemas utilizados en ese momento. Esta situación es, hasta cierto punto, entendible y justificable, hay que estar consientes que los estándares son normas, se debe tener cuidado en no detener la innovación y también en no limitar la utilización de todas las capacidades que marcan la diferencia entre un sistema bueno y uno regular.

La estandarización total, tanto en hardware como en software es difícil de alcanzar, el "estándar perfecto" que nos permita liberarnos de los problemas actuales en cuanto a compatibilidades continúa siendo imaginario. Existen grandes ventajas reconocidas que motivan la continuidad de trabajo de los comités de estándares internacionales.

Debemos tener presente que los estándares representan un producto mas en el mercado, el surgimiento de "nuevos estándares" representa ventas para algunas compañías.

Antes de aceptar un estándar como elemento integral de un sistema se deben conocer a fondo sus características, su historia, su funcionalidad y cuáles son las compañías que lo soportan; así también, su aceptación debe ser consecuencia de un estudio técnico detallado de evaluación técnica.

Viendo la situación general en que se encuentra México, estamos obligados a meditar bien nuestras decisiones, y a fomentar la comunicación y la realización de reuniones técnicas con el objetivo de transmitir experiencias.

9. Bibliografía.

Libros.

- [1] CAD/CAM Computer Aided Design and Manufacturing
Mikell P. Groover, Emory W. Zimmers
Prentice-Hall, Inc.
1984
- [2] Device Independent Graphics
R. F. Sproull, W. R. Sutherland, M. K. Ullner
McGraw-Hill Book Company
1985
- [3] Computer Integrated Manufacturing
Paul GRónky
Prentice/Hall International, UK, Ltd.
1986
- [4] Computer Graphics. A Programming Approach
Steven Harrington
McGraw-Hill Book Company
1985
- [5] Principles of Interactive Computer Graphics
W. M. Newman, R. F. Sproull
McGraw-Hill Book Company
1984
- [6] The Smart Manager's Guide to Selecting and Purchasing
CAD/CAM Systems
L. Stephen Wolfe
CAD/CAM Publishing, Inc.
1984
- [7] An Analysis of CAD/CAM Applications
Richard Stover
Prentice-Hall, Inc.
1984

[8] Fundamentals of Interactive Computer Graphics

J. D. Foley, A. Van Dam
Addison-Wesley
1982

[9] Introduction to the Graphical Kernel System (GKS)

F. R. A. Hopgood, D. A. Duce, J. R. Gallop, D. C. Sutcliffe
Academic Press, Inc.
1983

Artículos.

[1] IGES. One answer to the problem of CAD database exchange.

Ralph J. Mayer
Byte Magazine
Junio 1987

[2] Defining Product Data for Integrated CAD/CAM

Charles S. Skinner
CAE Magazine
Agosto 1985

[3] CAE of a Laser Machining Center

Michael A. Krutilla
Proceedings of the 2nd. Biennial International Machine Tool
Technical Conference
1984

**[4] A survey of graphics standards and their role in information
interchange**

Peter R. Bono
Proceedings of the NCGA Computer Graphics '86

[5] Understanding the Computer Graphics Interface (CGI)

Peter R. Bono, David B. Arnold
Proceedings of the NCGA Computer Graphics '87

[6] The Computer Graphics Virtual Device Interface (VDI)

Thomas Powers, Andrea Frankel, David Arnold
IEEE Computer Graphics and Applications Magazine
Agosto 1986

- [7] Language Bindings for Computer Graphics Standards
Madeleine R. Sparks, Julian R. Gallop
IEEE Computer Graphics and Applications Magazine
Agosto 1986
- [8] SET Links European CAD
Arielle Emmett
Network, Official Publication of the Society for Computer-
Aided Engineering
Septiembre 1986
- [9] PHIGS: A Standard, Dynamic, Interactive Graphics Interface
David Shuey, David Bailey, Thomas P. Morrissey
IEEE Computer Graphics and Applications Magazine
Agosto 1986
- [10] The Computer Graphics Metafile
Loffon Henderson, Margaret Journey, Chris Osland
IEEE Computer Graphics and Applications Magazine
Agosto 1986
- [11] GKS-3D: A Three-Dimensional Extension to the Graphical
Kernel System
Richard F. Puk, John I. McConell
IEEE Computer Graphics and Applications Magazine
Agosto 1986
- [12] The Reference Model for Computer Graphics
George S. Carson, Eileen McGinnis
IEEE Computer Graphics and Applications Magazine
Agosto 1986
- [13] Basics of Hierarchical Graphics
Machine Design Magazine
Octubre 1985
- [14] Reporte comparativo IGES-ISIF vs. Geodraw
José I. Tijerina
Vitra Tec, Reporte interno
Abril 1986

Manuales.

- [1] **The CAD/CAM Handbook**
Carl Machover, Robert E. Blauth
Computervision Corporation
1980
- [2] **VAX Initial Graphics Exchange Specification (IGES)**
Document *DIXD3470
Intergraph Corporation
Septiembre 1984
- [3] **Initial Graphics Exchange Specification (IGES)**
Document *DIXD4120
Intergraph Corporation
Febrero 1985
- [4] **I-DEAS Application Guide**
Structural Dynamics Research Corporation
Julio 1987
- [5] **Metafile for the Storage and Transfer of
Picture Description Information**
American National Standard Institute
Document *ANSI X3.122-1986
1986
- [6] **Graphics Standard Handbook**
Edited by Edmund Van Deusen
ISBN 0-939078-01-5
1985

Apéndice A

Ejemplo archivo IGES

Apéndice A: Ejemplo archivo IGES.

SDRC I-DEAS 3.9: Object Modeling

10-DEC-87 09:35:31

DATABASE:
VIEW: No stored VIEW
Task: OBJECT
Object: No stored OBJECT

UNITS = SI
DISPLAY: No stored OPTION
Bin: 1-MAIN

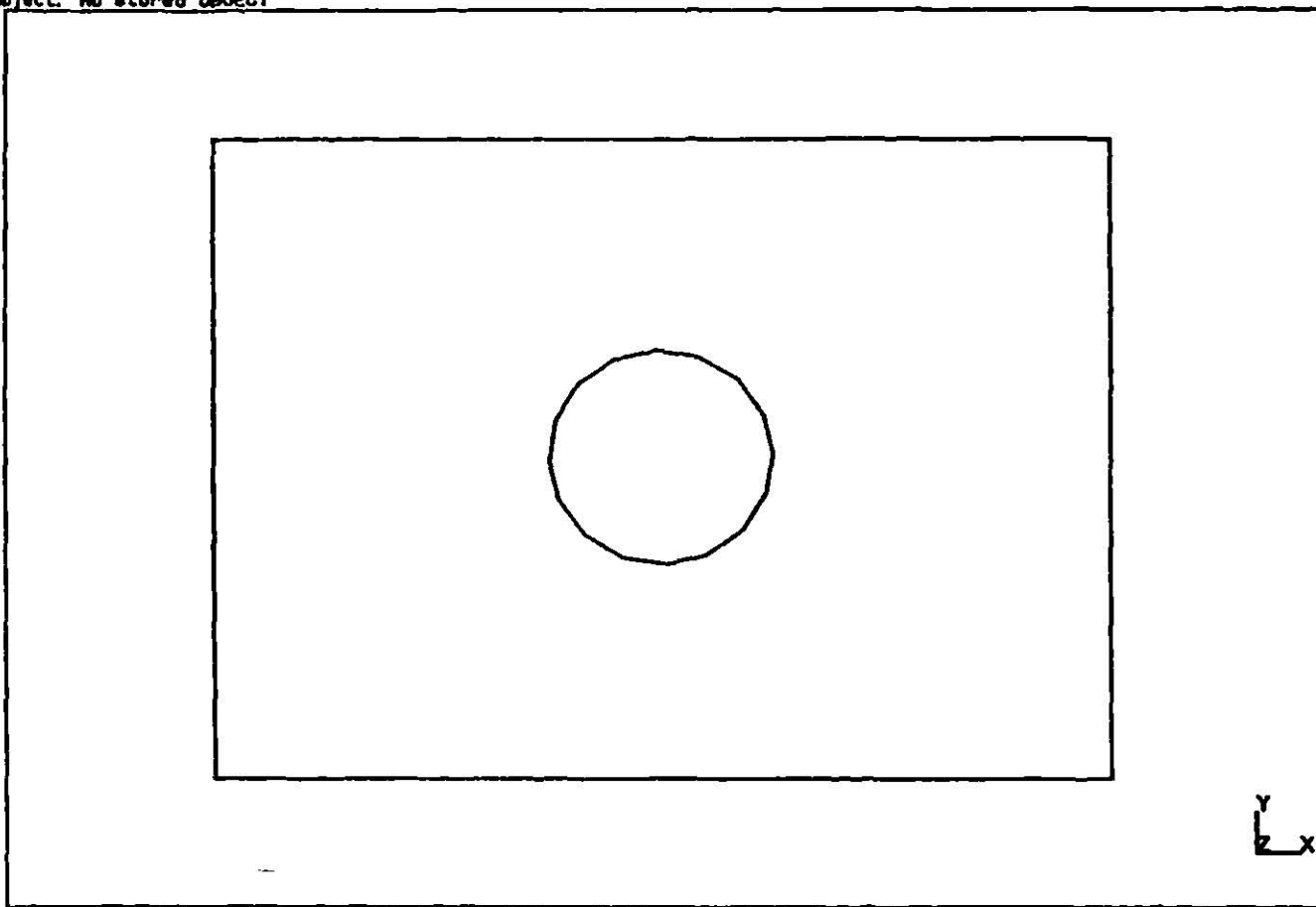


Figura geométrica representada
en los formatos Universal e IGES.

EJEMPLO ARCHIVO IGES

,,4HNONE,11HCUADRO.IGES, 36HSDRC GEOMOD;V3.0 ,VAX ,TYPE-IGES30,,32,8, S
 24,8,56,11HCUADRO.IGES,1.0,2,2HMM ,,,13H871223.092500,1.0E-02,9.0E+03,,G

110	1	1	1	0	0	0	0000000000	G
110	0	15	2	0	0	0	0000000000	D
110	3	1	1	0	0	0	0000000000	D
110	0	15	2	0	0	0	0000000000	N
110	5	1	1	0	0	0	0000000000	D
110	0	15	2	0	0	0	0000000000	D
110	7	1	1	0	0	0	0000000000	D
110	0	15	2	0	0	0	0000000000	D
124	9	1	1	0	0	0	0010000000	D
124	0	15	3	0	0	0	0000000000	D
100	12	1	1	0	0	9	0000000000	D
100	0	15	2	0	0	0	0000000000	D
110, 0.5000000E+04, 0.6000000E+04, 0.0000000E+00,								1P
0.5000000E+04, 0.3000000E+04, 0.0000000E+00, 0, 0;								1P
110, 0.5000000E+04, 0.3000000E+04, 0.0000000E+00,								3P
0.9000000E+04, 0.3000000E+04, 0.0000000E+00, 0, 0;								3P
110, 0.9000000E+04, 0.3000000E+04, 0.0000000E+00,								5P
0.9000000E+04, 0.6000000E+04, 0.0000000E+00, 0, 0;								5P
110, 0.9000000E+04, 0.6000000E+04, 0.0000000E+00,								7P
0.5000000E+04, 0.6000000E+04, 0.0000000E+00, 0, 0;								7P
124, -0.1000000E+01, 0.0000000E+00, 0.0000000E+00, 0.6982177E+04,								9P
0.0000000E+00, 0.1000000E+01, 0.0000000E+00, 0.499683E+04,								9P
0.0000000E+00, 0.0000000E+00, -0.1000000E+01, 0.0000000E+00, 0, 0;								9P
100, 0.0000000E+00, -0.1782275E+02, -0.4996826E+03, 0.0000000E+00,								11P
0.0000000E+00, 0.0000000E+00, 0.0000000E+00, 0, 0;								11P
1G	3D	12F	13					T

Apéndice B

Ejemplo formatos de base de datos externos

Apéndice B: Ejemplo archivo Universal Software I-DEAS de SDRC.

SDRC I-DEAS 3.9: Object Modeling

10-DEC-87 09:35:31

DATABASE:
VIEW: No stored VIEW
Task: OBJECT
Object: No stored OBJECT

UNITS = SI
DISPLAY: No stored OPTION
@inc 1-MAIN

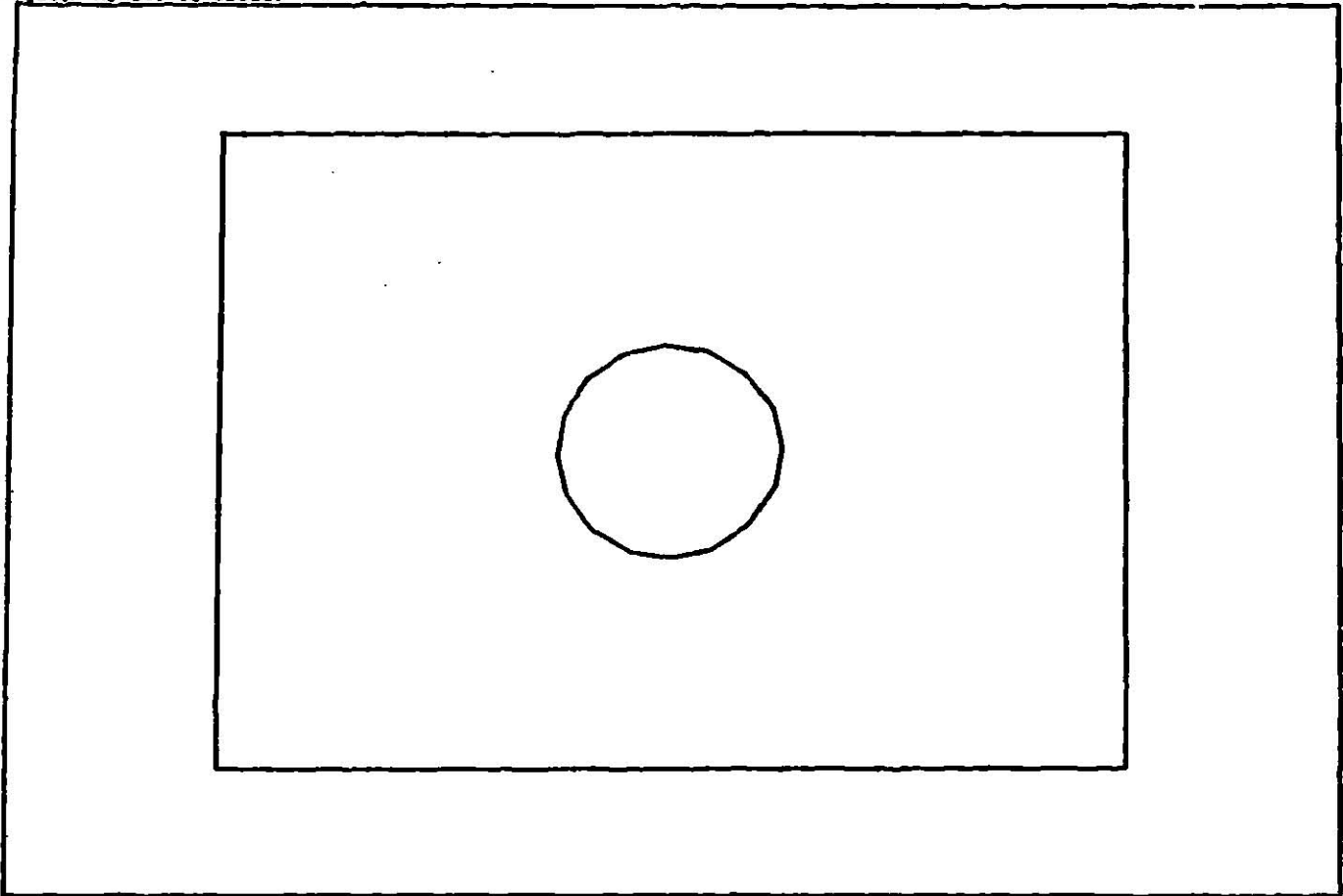


Figura geométrica representada
en los formatos Universal e IGES.

-1
151
NONE

SDRC I-DEAS 3.9: Monitor
23-DEC-87 08:22:3
Never Never
SDRC I-DEAS 3.9: Object Modelins
23-DEC-87 09:33:1

156
1Metric abs (SI)
1.00000E+00 1.00000E+00 1.00000E+00
-1
-1

449

1PROFILE1			1	1	0	2	
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	
5.00000000E+00	6.00000000E+00	0.00000000E+00	1.00000000E+00	5.00000000E+00			
3.00000000E+00	0.00000000E+00	1.00000000E+00	0.00000000E+00	0.00000000E+00			
1.00000000E+00	1.00000000E+00						
5.00000E+00	6.00000E+00	0.00000E+00					
5.00000E+00	3.00000E+00	0.00000E+00					
5.00000000E+00	3.00000000E+00	0.00000000E+00	1.00000000E+00	9.00000000E+00			
3.00000000E+00	0.00000000E+00	1.00000000E+00	0.00000000E+00	0.00000000E+00			
1.00000000E+00	1.00000000E+00						
5.00000E+00	3.00000E+00	0.00000E+00					
9.00000E+00	3.00000E+00	0.00000E+00					
9.00000000E+00	3.00000000E+00	0.00000000E+00	1.00000000E+00	9.00000000E+00			
6.00000000E+00	0.00000000E+00	1.00000000E+00	0.00000000E+00	0.00000000E+00			
1.00000000E+00	1.00000000E+00						
9.00000E+00	3.00000E+00	0.00000E+00					
9.00000E+00	6.00000E+00	0.00000E+00					
9.00000000E+00	6.00000000E+00	0.00000000E+00	1.00000000E+00	5.00000000E+00			
6.00000000E+00	0.00000000E+00	1.00000000E+00	0.00000000E+00	0.00000000E+00			
1.00000000E+00	1.00000000E+00						
9.00000E+00	6.00000E+00	0.00000E+00					
5.00000E+00	6.00000E+00	0.00000E+00					
6.98217726E+00	4.99968243E+00	0.00000000E+00	1.00000000E+00	5.29047394E+00			
3.54791188E+00	0.00000000E+00	7.07106769E-01	7.49968243E+00	4.51782274E+00			
0.00000000E+00	1.00000000E+00	5.31567907E+00	2.84125447E+00	0.00000000E+00			
7.07106769E-01	7.01782274E+00	4.00031757E+00	0.00000000E+00	1.00000000E+00			
4.60902166E+00	2.81604934E+00	0.00000000E+00	7.07106769E-01	6.50031757E+00			
4.48217726E+00	0.00000000E+00	1.00000000E+00	4.58381653E+00	3.52270675E+00			
0.00000000E+00	7.07106769E-01	6.98217726E+00	4.99968243E+00	0.00000000E+00			
1.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00	2.50000000E-01			
2.50000000E-01	5.00000000E-01	5.00000000E-01	7.50000000E-01	7.50000000E-01			
1.00000000E+00	1.00000000E+00	1.00000000E+00	0.00000000E+00	0.00000000E+00			
1.00000000E+00							
6.98218E+00	4.99968E+00	0.00000E+00					
7.49836E+00	4.45952E+00	0.00000E+00					
6.86545E+00	4.01845E+00	0.00000E+00					
6.98218E+00	4.99968E+00	0.00000E+00					

Archivo archivo DXF.
Software AutoCAD de Autodesk, Inc.

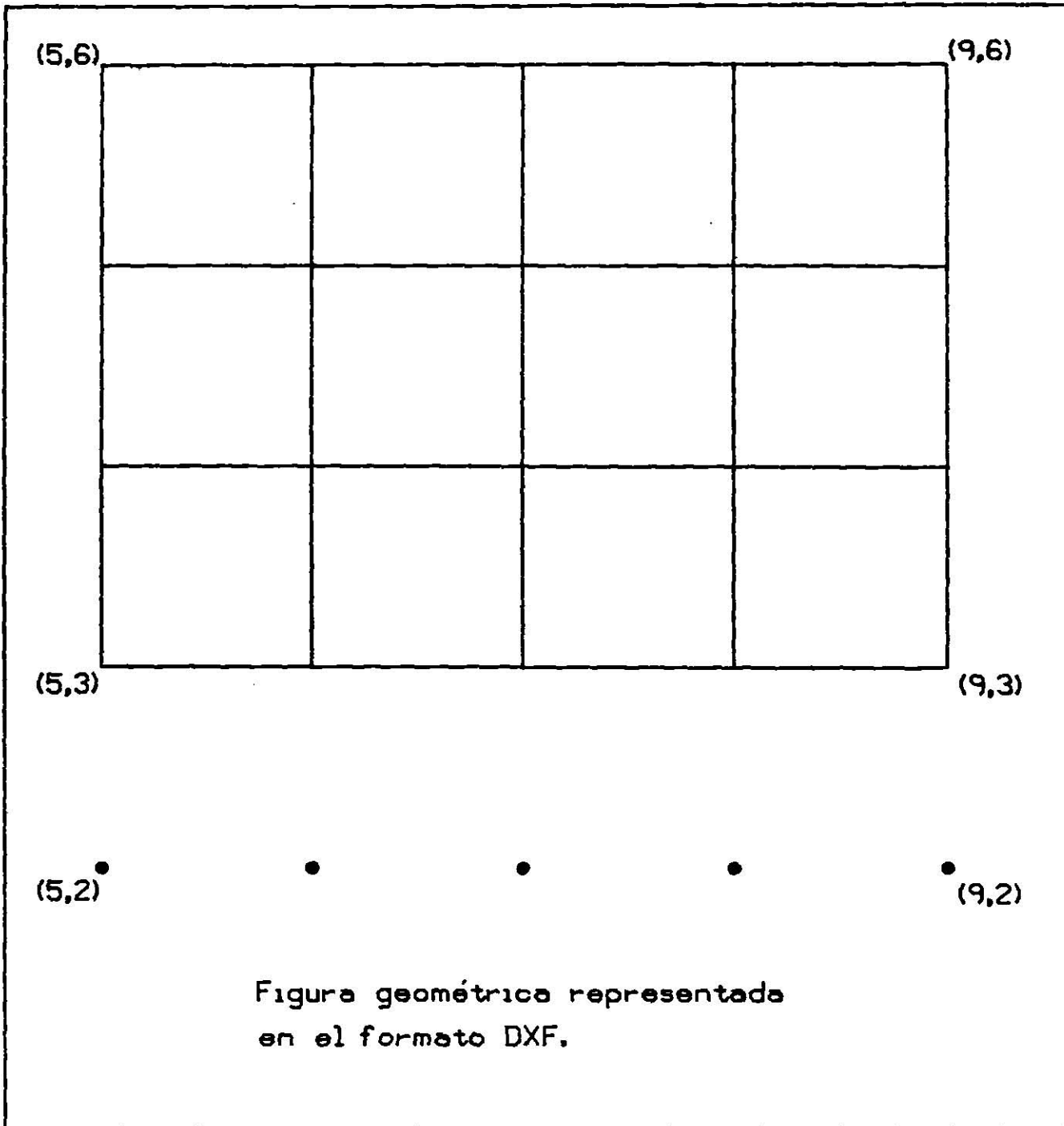


Figura geométrica representada
en el formato DXF.

0
SECTION
2
HEADER
9
\$ACADVER
1
AC2.10
9
\$INSBASE
10
0.0
20
0.0
9
\$EXTMIN
10
0.0
20
0.0
9
\$EXTMAX
10
9.0
20
6.0
9
\$LIMMIN
10
0.0
20
0.0
9
\$LIMMAX
10
12.0
20
9.0
9
\$VIEWCTR
10
7.3781
20
4.5
9
\$VIEWSIZE
40
9.0
9
\$SNAPMODE
70
1
9
\$SNAPUNIT
10
1.0
20
1.0
9
\$SNAPBASE
10

\$SNAPANG
50
0.0
9
\$SNAPSTYLE
70
0
9
\$SNAPISOPAIR
70
0
9
\$GRIDMODE
70
0
9
\$GRIDUNIT
10
0.0
20
0.0
9
\$ORTHOMODE
70
1
9
\$REGENMODE
70
1
9
\$FILLMODE
70
1
9
\$TEXTMODE
70
0
9
\$DRAGMODE
70
1
9
\$LTSCALE
40
1.0
9
\$OSMODE
70
0
9
\$ATTMODE
70
1
9
\$TEXTSIZE
40
0.2
9
\$TRACEWID
40
0.05
9

\$CLAYER
7
0
9
\$DIMSCALE
40
1.0
9
\$DIMASZ
40
0.18
9
\$DIMEXO
40
0.0625
9
\$DIMDLI
40
0.38
9
\$DIMEXE
40
0.18
9
\$DIMTP
40
0.0
9
\$DIMTM
40
0.0
9
\$DIMTXT
40
0.18
9
\$DIMCEN
40
0.09
9
\$DIMTSZ
40
0.0
9
\$DIMITOL
70
0
9
\$DIMLIM
70
0
9
\$DIMITIH
70
1
9
\$DIMITOH
70
1
9
\$DIMSE1
70

0
9
\$DIMITAD
70
0
9
\$LUNITS
70
2
9
\$LUPREC
70
4
9
\$AXISMODE
70
0
9
\$AXISUNIT
10
0.0
20
0.0
9
\$SKETCHINC
40
0.1
9
\$FILLETRAD
40
0.0
9
\$AUNITS
70
0
9
\$AUPREC
70
0
9
\$MENU
1
ACAD
9
\$ELEVATION
40
0.0
9
\$THICKNESS
40
0.0
9
\$VIEWDIR
10
0.0
20
0.0
30
1.0
9
\$LIMCHECK
70

10 1
9
\$CHAMFERA
40
0.0
9
\$CHAMFERB
40
0.0
0
ENDSEC
0
SECTION
2
TABLES
0
TABLE
2
LTYPE
70
1
0
LTYPE
2
CONTINUOUS
70
64
3
Solid line
72
65
73
0
40
0.0
0
ENDTAB
0
TABLE
2
LAYER
70
1
0
LAYER
2
0
70
64
62
7
6
CONTINUOUS
0
ENDTAB
0
TABLE
2
STYLE
70
1
0

40
0.0
41
1.0
50
0.0
71
0
42
0.2
3
TXT
0
ENDTAB
0
TABLE
2
VIEW
70
0
0
ENDTAB
0
ENDSEC
0
SECTION
2
BLOCKS
0
ENDSEC
0
SECTION
2
ENTITIES
0
LINE
8
0
10
5.0
20
6.0
11
5.0
21
3.0
0
LINE
8
0
10
5.0
20
3.0
11
9.0
21
3.0
0
LINE
8

20
3.0
11
9.0
21
6.0
0
LINE
8
0
10
9.0
20
6.0
11
5.0
21
6.0
0
LINE
8
0
10
5.0
20
5.0
11
9.0
21
5.0
0
LINE
8
0
10
5.0
20
4.0
11
9.0
21
4.0
0
LINE
8
0
10
6.0
20
6.0
11
6.0
21
3.0
0
LINE
8
0
10
7.0
20
3.0
11

LINE
8
0
10
8.0
20
6.0
11
8.0
21
3.0
0
POINT
8
0
10
5.0
20
2.0
0
POINT
8
0
10
6.0
20
2.0
0
POINT
8
0
10
7.0
20
2.0
0
POINT
8
0
10
8.0
20
2.0
0
POINT
8
0
10
9.0
20
2.0
0
ENDSEC
0
EOF

