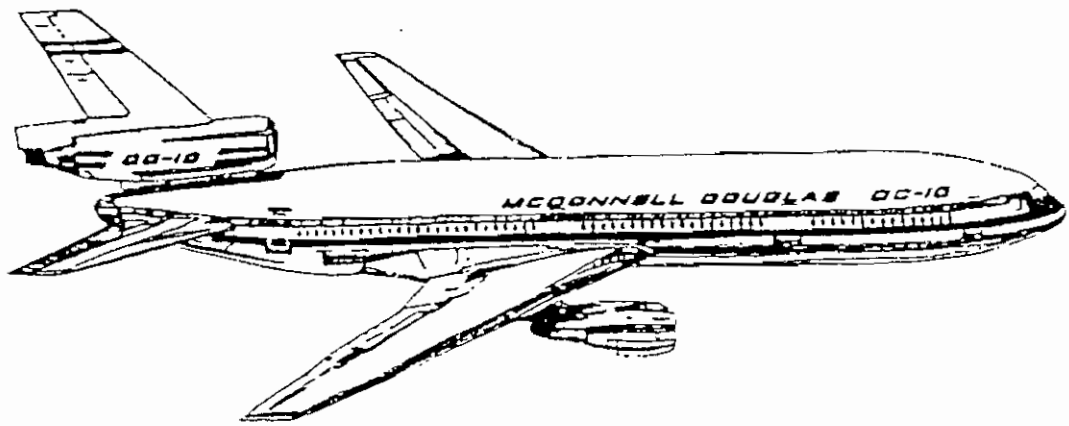
ESCUELA POLITECNICA NACIONAL

FACULTAD DE INGENIERIA ELECTRICA

TESIS PREVIA A LA OBTENCION DEL TITULO DE INGENIERO EN LA

ESPECIALIZACION DE ELECTRONICA Y CONTROL

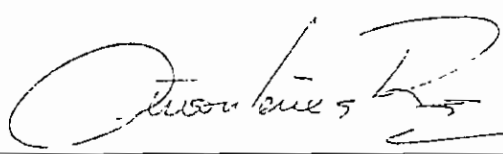
"DESARROLLO DE UN SISTEMA EXPERTO PARA DIAGNOSTICO DE FALLAS"



ALEX REVELO LOPEZ

QUITO, NOVIEMBRE DE 1993

CERTIFICO QUE EL PRESENTE TRABAJO HA SIDO REALIZADO EN SU TOTALIDAD POR EL SR. ALEX REVELO LOPEZ.



ING. CARLOS MONTENEGRO

DIRECTOR DE TESIS

En este trabajo se utilizó el conocimiento experimental disponible provisto por el fabricante del avión McDONNELL DOUGLAS DC-10, para diseñar un sistema experto de diagnóstico de fallas en el sistema eléctrico de potencia. Este último consta de 8 subsistemas de diagnóstico, cada uno de los cuales posee su propio proceso que utiliza en forma integrada esquemas de lógica, reglas inversas, hechos y eventualmente, marcos. Además, se han agregado 2 subsistemas de administración de información.

Para la implementación se trabajo con el shell INTELLIGENCE COMPILER, que integra varios paradigmas de representación de conocimiento y permite diseñar interfases amigables para el usuario. Así, en el proceso de diagnóstico se hizo uso de menús, ventanas de diálogos e hipertexto.

Los subsistemas de administración de información están constituidos por una base de datos inteligente y un conjunto de archivos de texto. La base de datos puede ser manejada independientemente por el usuario; contiene información histórica de las bitácoras de vuelo a partir del año 1990 y cuenta con variadas posibilidades de clasificación automática. El otro componente provee una vía de retro-alimentación para el mantenimiento y reactualización del sistema experto, a través del ingreso de información por parte de los usuarios.

Durante la ejecución de cada uno de estos 10 subsistemas se puede acceder a un sistema de ayudas en-línea.

El objetivo es que este sistema sirva de apoyo para los expertos en el mantenimiento del avión y en el departamento de control de calidad de la empresa.

1.5.1.2 INTERFASE PARA EL USUARIO.....53

1.5.1.3 LENGUAJE DE PROGRAMACION.....56

1.5.2 ATRIBUTOS Y APLICACIONES.....56

CAPITULO II. DESCRIPCION DEL SISTEMA A DIAGNOSTICAR

2.1 DESCRIPCION GENERAL.....64

2.2 UNIDAD DEL GENERADOR Y TRANSMISION
DE VELOCIDAD CONSTANTE.....65

2.3 GENERACION AC.....67

2.3.1 GENERACION, PROTECCION Y CONTROL DE AC.....71

2.3.1.1 GENERADOR AC DE MOTOR.....71

2.3.1.2 UNIDAD DE POTENCIA AUXILIAR (APU).....73

2.3.1.3 RELEVADOR DE GENERADOR, DE ENLACE
DE BUS AC, Y DE POTENCIA AUXILIAR.....74

2.3.1.4 UNIDAD DE CONTROL DE GENERADOR (GCU)...75

- POTENCIA DE CONTROL DC PARA GCUs.....76

2.3.1.5 UNIDAD DE CONTROL DE BUS (BCU).....81

- POTENCIA DE CONTROL DC DE BCU.....82

2.3.1.6 SWITCH DE CONTROL DEL GENERADOR.....85

2.3.1.7 SWITCH DE PARALELO DE GENERADOR.....85

2.3.1.8 SWITCHES DE ENLACE DE BUS AC.....86

2.3.1.9 SWITCH SELECTOR INDICADOR.....87

2.3.1.10 SWITCH DE POTENCIA EXTERNA PRINCIPAL...87

2.4 GENERACION DC.....87

2.5 POTENCIA EXTERNA.....89

2.6 DISTRIBUCION DE CARGA ELECTRICA.....90

2.7 LIMITACIONES DE POTENCIA ELECTRICA.....91

2.8 GLOSARIO DE ABREVIATURAS.....94

CAPITULO III. DESARROLLO DEL SISTEMA EXPERTO DE DIAGNOSTICO

3.1 INTRODUCCION.....101

3.2 ANALISIS DE FACTIBILIDAD.....106

3.3 DISEÑO CONCEPTUAL.....111

3.4 ADQUISICION DE CONOCIMIENTO.....119

3.5 REPRESENTACION DE CONOCIMIENTO.....121

3.6 PRUEBA Y VALIDACION.....127

3.7 MANTENIMIENTO.....134

CAPITULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 SISTEMA EXPERTO.....138

4.2 HERRAMIENTA DE DESARROLLO.....144

4.3 CRITERIOS DE DESARROLLO.....151

4.4 RECOMENDACIONES GENERALES.....154

APENDICES:

A. BIBLIOGRAFIA.....164

B. SHELL "INTELLIGENCE COMPILER".- GENERALIDADES...169

C. GUIA DEL USUARIO.....177

D. HOJAS DE DATOS Y PROGRAMAS.....194

- INDICE DE FIGURAS.....229

conocimiento a la que tienen acceso, para razonar acerca de problemas específicos [15].

La principal diferencia entre los sistemas expertos y los programas convencionales consiste no sólo en la entrega de la experiencia, pues muchos programas convencionales pueden desempeñar tareas "expertas"; sino también en la mayor separación posible entre los datos y las construcciones del programa. Esta separación permite una mayor flexibilidad en la generación, mantenimiento y entendimiento del programa. Así se hace posible una aplicación "estática" (eso es, no procedural) del conocimiento del dominio.

Por lo tanto, los sistemas basados en conocimiento difieren de los programas convencionales en la manera cómo estan organizados, cómo incorporan el conocimiento, cómo lo ejecutan y en la impresión que crean a través de sus interacciones [18].

Algunas aplicaciones corrientes de la ingeniería de conocimiento son:

- Diagnósticos médicos,
- Reparación de equipo,
- Configuración de computadoras,
- Interpretación de datos químicos, y elucidación de estructuras,
- Entendimiento de lenguaje e imágenes,
- Toma de decisiones financieras,
- Interpretación de señales,
- Exploración de minerales,
- Inteligencia militar y planificación,
- Consultoría de sistemas de computador, y diseño VLSI.

1.2 SISTEMAS INTELIGENTES BASADOS EN CONOCIMIENTO

La representación explícita del conocimiento y la aplicación de técnicas naturales de razonamiento ha sido materia de mucha investigación en Inteligencia Artificial. Una de las áreas más promisorias es la de los Sistemas Inteligentes Basados en Conocimiento, de la cual los Sistemas Expertos son un subgrupo. Un Sistema Inteligente Basado en Conocimiento usa el conocimiento (que se ha extraído de un experto humano) de una manera "inteligente" para resolver un problema, y organiza el conocimiento en tres niveles: datos, base de conocimiento, y control. Las computadoras organizadas de este modo, se llaman sistemas basados en conocimiento [26].

Estos sistemas cuentan con técnicas eficaces para representar el conocimiento (reglas de producción, planteamientos abductivos, y combinaciones de los dos, entre otros).

En el planteamiento con reglas de producción el conocimiento (especialmente el conocimiento "superficial") se organiza en: 1) una base de datos, hechos y relaciones, 2) un conjunto de reglas de producción (reglas...IF...THEN), y 3) una estructura de control o mecanismo de inferencia (es el componente que resuelve el problema).

En el planteamiento abductivo (la abducción es una forma muy común del razonamiento humano), a partir de un conjunto embrollado de observaciones se construye una hipótesis, la que se busca comprobar o refutar buscando o deduciendo más información (como síntomas adicionales p.e). Después, según sea necesario se repite el proceso. Esto es lo que se conoce

como "hipotetizar y probar". A semejanza del planteamiento con reglas de producción, este sistema consiste de una base de conocimiento, un motor de inferencia, y una interfase al usuario, pero difiere en la manera en que trabaja el motor de inferencia [1].

Los sistemas expertos pueden apoyar a diferentes clases de usuarios tales como diseñadores, operadores y personal de mantenimiento. En general, estos sistemas serán off-line (para diseñadores y personal de mantenimiento), y on-line (para operadores) [26].

El incentivo de aplicar la IA a la ingeniería proviene no sólo de la complejidad de la ingeniería moderna, sino también del costo tradicional, las limitaciones de tiempo y la disponibilidad limitada de la experiencia humana. La IA parece ofrecer una oportunidad para (1) capturar y retener experiencia que se ha acumulado a través de muchos años de ingeniería, (2) ampliar la experiencia que se necesita para difundir nuevas tecnologías y aplicaciones de diseño, y (3) ofrecer sistemas que razonen inteligentemente en tiempo real [10].

1.3 INGENIERIA DE CONOCIMIENTO

La ingeniería de conocimiento es el proceso de construcción de sistemas expertos, en los cuales el conocimiento se representa de una manera especifica. Se han identificado las siguientes categorías de aplicaciones de la ingeniería de conocimiento: interpretación, predicción, diagnóstico, diseño, planificación, supervisión, depuración, reparación, instrucción y control.

En términos generales, los sistemas expertos trabajan mejor en dominios de aplicación de conocimiento estrechos y específicos.

A su vez, una clasificación jerárquica desde una perspectiva de factores humanos, divide a los sistemas expertos de acuerdo con su propósito:

- desempeñar una tarea
- asistir una tarea, y
- enseñar una tarea

La primera categoría trata con sistemas expertos autónomos como los de robots autónomos, o sistemas de automatización. La segunda y tercera categorías se relacionan con sistemas de consulta experta [19].

El proceso de construción de un sistema experto consiste de dos actividades principales que usualmente se traslapan: 1) adquisición de conocimiento y, 2) implantación del sistema. Algunos términos que se usan para describir el proceso de construcción de un sistema experto aún no están muy bien definidos; estos términos incluyen: extracción de conocimiento, adquisición de conocimiento, implantación del sistema, inducción de máquina, y aun el mismo término de ingeniería de conocimiento. En la figura 1.1 se distingue el proceso de adquisición de conocimiento, del proceso de implantación del sistema. También se ha separado la extracción (manual o automática) de la inducción de máquina, puesto que éstas técnicas de adquisición son marcadamente distintas y han seguido diferentes trayectorias de desarrollo.

Una diferenciación importante del conocimiento tiene que ver con el conocimiento tecnológico científico ("conocimiento de funcionamiento"), y el conocimiento experimental ("conocimiento de utilización"). El primero es utilizado por diseñadores y personal de mantenimiento (sistemas off-line), mientras que el último es usado por los operadores (sistemas on-line) [20], [28].

Además los procesos de extracción y conceptualización también son complejos; y los ensayos realizados demuestran que la adquisición de conocimiento consume mucho tiempo. En algunos casos el proceso entero es probable que dure años más bien que meses mientras la tecnología aún desarrolla herramientas más eficaces para hacer esto [19].

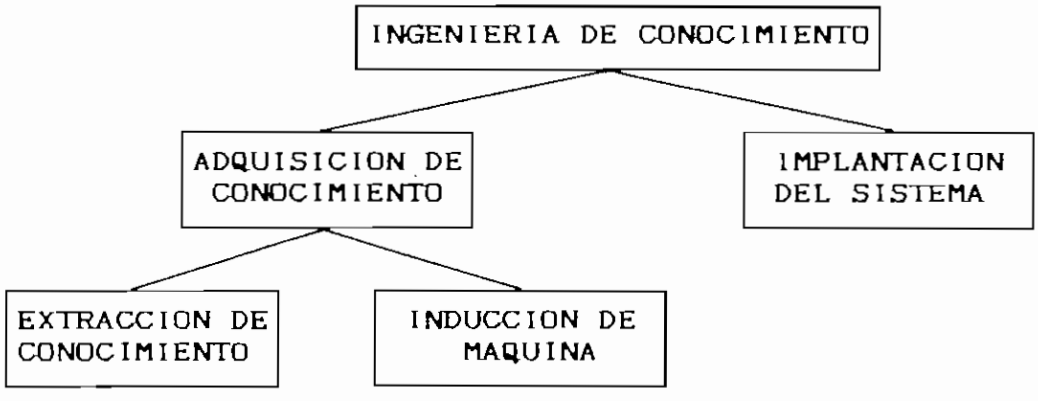


Figura 1.1. Relación entre los términos de la ingeniería de conocimiento.

Aun cuando un sistema experto de apoyo para predecir las consecuencias de algunas fallas técnicas por ejemplo, necesita normalmente sólo de ingeniería de conocimiento; sin embargo, el soporte procedural, los módulos de diagnóstico y control heurístico necesitan además conocimiento operacional; y al integrarse con el sistema de control y

La experiencia de un experto a menudo se la puede describir en piezas, cada una relativa al contexto de alguna situación pequeña y a las acciones apropiadas para tal situación. Los sistemas expertos proveen un mecanismo de inferencia para relacionar estas piezas dinámicamente al tiempo de su ejecución [18], [10].

En detalle, un experto no sólo posee un depósito de conocimiento conceptual (o declarativo) de un dominio, sino que también puede aplicarlo rápidamente en una tarea dada, por ello, tres cosas son esenciales para un experto: 1) tener conocimiento de conceptos, de su taxonomía e interrelaciones, y si es necesario poder resolver problemas nuevos razonando en base a los principios de su dominio, 2) tener una abundancia de respuestas disponibles para una situación específica, o respuestas parciales que acorten el proceso de resolución del problema, y 3) tener la habilidad de reconocer oportunidades para usar estos atajos e invocar la asociación "situación-acción" apropiada [10].

Además, la extracción del conocimiento de un experto, no consiste en transferir un modelo mental del cerebro del experto a la mente del diseñador del sistema, sino la formalización de un dominio en primer término. Idealmente, se requieren los modelos de estructuras conceptuales de la solución del problema, como un prerrequisito para el proceso de transferencia de conocimiento. Sin embargo, los planteamientos de la ciencia cognoscitiva todavía no han producido suficiente información para hacer posible un entendimiento completo de las estructuras de conocimiento, y de las estrategias de los expertos para la resolución de

problemas, de suerte que las aproximaciones actuales son incompletas y a menudo ad-hoc.

La situación se complica más por el hecho de que los expertos con frecuencia tienen fallas de memoria, o inconsistencias. Además, los expertos exhiben tendencias cognoscitivas difíciles de manejar: exceso de confianza, simplificación, y una débil preferencia por lo abstracto, así la evidencia es conflictiva y relativa [31]. Estas consideraciones son importantes a la hora de probar y validar sistemas expertos. Hasta hasta donde sea posible, las tendencias cognoscitivas deben filtrarse durante el proceso de extracción; y para reducir las posibilidades de que se den, a los expertos se les debe conscientizar de las tendencias de juicio comúnmente halladas. Además el proceso de extración debe incluir sondeos para promover la consideración de alternativas; y cuando los expertos manejan problemas de muestra en el proceso de extracción, debe tenerse en mente que la manera cómo se les presentan los problemas afectará el grado hasta el cual mostrarán tendencias cognoscitivas las reglas derivadas de aquí.

Tomando en cuenta estos puntos, se han sugerido las siguientes etapas para el proceso de ingeniería de conocimiento:

1. Extracción del conocimiento,
2. Filtración de tendencias cognoscitivas,
3. Representación del conocimiento y selección del esquema de control,
4. Desarrollo e integración del software,
5. Evaluación y validación del sistema, y

6. Prototipo avanzado del sistema experto.

Las etapas tres y cuatro, idealmente deberían llevarse a cabo sólo después que las dos primeras etapas se hayan completado. En realidad esto no es posible, y la experiencia sugiere que se requieren varias iteraciones a través de las primeras cinco etapas, antes que se contemple la etapa seis [19].

Por otro lado, existen dos opiniones competitivas acerca de la tarea de adquisición de conocimiento. Estas pueden describirse como de abajo-hacia arriba, y de arriba-hacia abajo. Quienes proponen el procedimiento de abajo-hacia arriba aspiran a estimar datos y conceptos del experto y después refinarlos iterativamente, en armonía con la siguiente descripción de la adquisición de conocimiento: "Es el proceso de extraer -como de una mina- aquellos rubíes de conocimiento fuera de las mentes de los expertos, uno por uno". Además se ha dicho que la construcción de sistemas expertos "es inherentemente experimental" [18], y que por lo tanto está caracterizada por un prototipeamiento rápido, que es esencialmente un proceso de abajo-hacia arriba. La suposición fundamental de este planteamiento es que un sistema experto está basado en una gran extensión de conocimiento específico de un dominio, y que hay pocos principios generales y fundamentales de la organización de este conocimiento en la mente de un experto. Sin embargo, la existencia de principios fundamentales y relaciones causales, puede ser una indicación de que el conocimiento experto es más independiente del dominio de lo que han asumido algunos investigadores [19].

Por eso, los que apoyan el procedimiento de arriba-hacia abajo, sostienen que hay un paso crucial que está ausente en el planteamiento de prototipeamiento, entre la identificación de las características pertinentes del dominio y la selección de métodos de solución: el de la interpretación de los datos dentro de alguna estructura coherente, un modelo, esquema, o forma canónica. Y se proponen cinco niveles de análisis de conocimiento: identificación, conceptualización, epistemológico, lógico e implantacional. Se han desarrollado estas ideas en una metodología de adquisición de conocimiento llamada KADS (1985) [19].

1.3.2 TECNICAS PARA ADQUISICION DE CONOCIMIENTO

Una de las primeras fases del desarrollo de cualquier sistema de computador, es la especificación del problema que se va a tratar. En el desarrollo de sistemas expertos, esta fase no es tan fácil de definir como para un sistema convencional. La especificación del problema suele ser bastante vaga. Se logran especificaciones adicionales a través de un período de adquisición de conocimiento. Después se lleva a cabo el prototipeamiento por uno o más ingenieros de conocimiento.

En algunos casos, se ha utilizado la naturaleza estructurada del conocimiento del dominio, en prototipos de sistemas para diagnóstico de fallas usados en sistemas consultivos on-line.

En general, las técnicas usadas para la adquisición de conocimiento pueden dividirse en dos categorías: extracción, e inducción de máquina.

Se recomienda también que en el proceso de adquisición de conocimiento el ingeniero de conocimiento use un lenguaje de términos (primitivas) a nivel de tareas para capturar conceptos importantes del dominio, antes que imponer sus propias primitivas; en segundo lugar, que se usen primitivas declarativas explícitas antes que descripciones procedurales, ya que la mayoría de los expertos entienden mejor representaciones declarativas; y en tercer lugar, no forzar a los expertos a generalizar ni a especificar información que no la tienen disponible [31].

Además, es importante tomar en cuenta que los expertos cambian sus estrategias de solución dependiendo del grado de indefinición del problema. En problemas bien definidos, una aproximación experta (que refleja los procesos físicos que ocurren en la realidad), difiere dramáticamente de la de un novato. Sin embargo, en muchos casos el experto es capaz de identificar factores de incertidumbre que impiden el uso de estrategias de solución precisas. En tales circunstancias adoptará modelos simples del proceso y un tanto imprecisos, pero también reconoce que los modelos simples no son precisos, y que el uso de un modelo preciso no es práctico. Las técnicas de adquisición de conocimiento deben ser capaces de habérselas con esta variación en la estrategia de los expertos [3], [19].

1.3.2.1 EXTRACCIÓN DE CONOCIMIENTO

En primer término, se tienen las técnicas de entrevista, que usualmente envuelven la recolección de información de los expertos del dominio, sea de manera implícita o explícita. Al principio se usaban los reportes escritos por los

expertos, pero esta técnica ahora ya no tiene aceptación, puesto que tales reportes tienden a tener un alto grado de prejuicios y pensamientos reflexivos. Las técnicas actuales incluyen entrevistas (tanto estructuradas como no estructuradas), cuestionarios y técnicas observacionales, tales como análisis de protocolos y recorridos a pie.

Se ha señalado que las metodologías de extracción de conocimiento "tienen más en común con la orientación del campo de trabajo de la antropología y la sociología cualitativa, que con la orientación experimental de muchas de las ciencias cognoscitivas" [19]. Por ello se sugiere qùe los ingenieros de conocimiento también usen la gran cantidad de literatura y experiencia, así como la larga tradición de las ciencias sociales en este campo de trabajo. Algunos escollos de la extracción de conocimiento se describen en base de esta experiencia de las ciencias sociales, en particular, se explican algunos problemas de las entrevistas como: la obtención de datos versus considerar al experto como persona, el temor del silencio, y fallas al escuchar, dificultad en formular preguntas y entrevistar sin un registro, así como problemas conceptuales tales como la metodología para realizar una entrevista no problemática, o culpar al experto de una entrevista fallida.

No obstante, para el desarrollo de sistemas expertos a gran escala, hay un significativo beneficio potencial que se gana al proveer a los expertos de un prototipo que criticar, sin involucrarlos en un proceso intensivo de adquisición de conocimiento [3], [19].

Otro planteamiento para la adquisición de conocimiento es el aprendizaje por interacción, en el cual el experto trata directamente con un programa de computador que le ayuda a clarificar sus ideas y a capturar su propio conocimiento.

Finalmente, otro planteamiento tiene que ver con el aprendizaje por inducción, en el cual un programa de computador 'destila' el conocimiento por medio de examinar datos y ejemplos. Su principal problema es la identificación de los atributos más apropiados sobre los cuales aplicar la inducción [3], [19].

1.3.3 METODOS DE REPRESENTACION DE CONOCIMIENTO

Como se indica en la figura 1.2, existen tres aspectos de la representación de conocimiento que son fundamentales en cualesquier herramienta para construcción de sistemas expertos: descripciones objeto (conocimiento declarativo tales como hechos), certidumbres, y acciones.

Un método de representar objetos es por medio de marcos con o sin herencia. (La herencia permite que las bases de conocimiento se organicen como agrupaciones jerárquicas de marcos, que heredan información de marcos de niveles superiores a ellos. Así, un mecanismo de herencia provee una forma de inferencia). Los marcos son estructuras tabulares de datos para organizar representaciones de situaciones u objetos para prototipos. Un marco tiene ranuras, que se llenan con información de objetos y relaciones apropiadas a la situación. Una versión de programación que se refiere a esto es la programación orientada a objetos, la cual utiliza objetos que tienen facilidades para transmitir mensajes entre sí; agregados a estos objetos hay procesos que pueden

activarse por el recibimiento de mensajes. El conocimiento declarativo también puede representarse por pares de parámetro-valor, por el uso de notación lógica y, en alguna medida por reglas.

Las acciones cambian una situación y/o modifican la base de datos pertinente. Las acciones se representan comúnmente por reglas. Para un fácil mantenimiento y un rápido acceso, estas reglas se agrupan en módulos (usualmente como subpartes del problema). Las acciones pueden representarse también en términos de **ejemplos**, que muestran las conclusiones o decisiones a las que se ha llegado. Los ejemplos son una forma de representación que facilitan la adquisición de conocimiento y la utilización de los sistemas inductivos. Además son mucho más fáciles de obtener de los expertos que las reglas, y a menudo son una expresión natural del conocimiento del dominio. Las acciones pueden expresarse también en notación lógica, que es la forma de representación de las reglas. Finalmente, las acciones pueden expresarse como procesos extraídos por:

- mensajes (en programación orientada a objetos) o
- cambios en una base de datos global que se conocen como **demons**.("Demons" son procesos que supervisan una situación y responden ejecutando una acción cuando se presentan sus condiciones de activación).

Además de la representación de objetos y acciones, se debe considerar hasta qué grado se conoce que el conocimiento o la información es correcta. Por eso la mayoría de las herramientas para construir sistemas expertos proveen técnicas para representar esta certidumbre. El planteamiento

más común es incorporar "factores de certeza", que son una derivación de aquellos utilizados en el sistema experto MYCIN. Se utilizan también la lógica difusa y la probabilidad.

Un modo alternativo de manejar incertidumbres o hipótesis tentativas, es considerar múltiples ambientes en los cuales son verdaderos o falsos diferentes detalles.

Otra consideración es si un modelo profundo (el cual es un modelo estructural o causal) del sistema se puede construir o no sin esfuerzo, con la herramienta para construir sistemas expertos que se considera, como ayuda para el razonamiento basado en modelos (el mismo modelo fundamental puede emplearse con frecuencia para otros usos, tales como la conservación del conocimiento y el entrenamiento) [14].

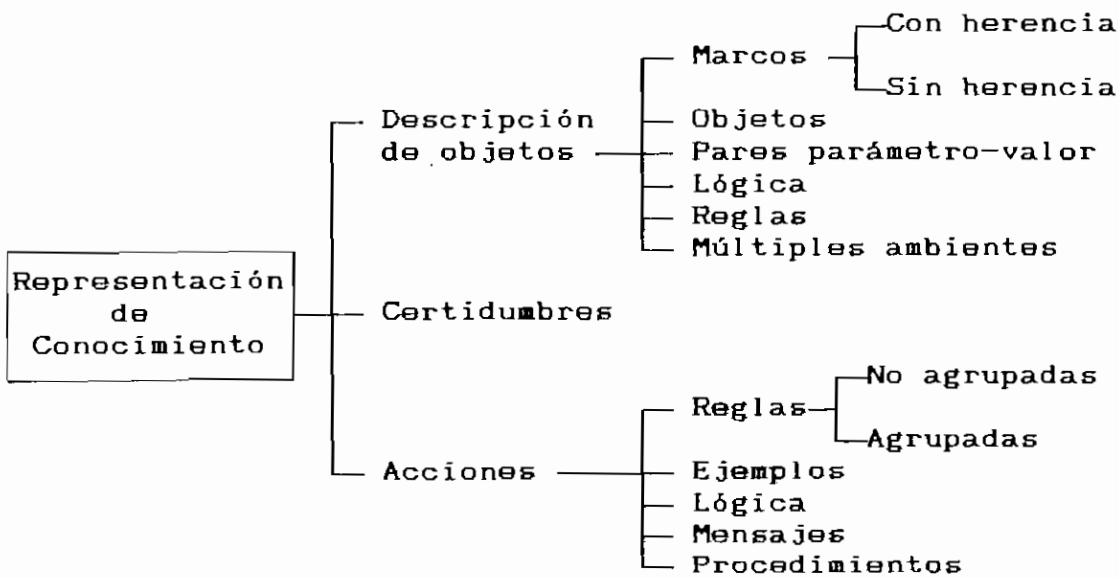


Figura 1.2. Diferentes métodos de representación de conocimiento.

1.3.4 MOTOR DE INFERENCIA

La figura 1.3 muestra las principales alternativas para llevar a cabo el proceso de inferencia. El planteamiento más usual es la clasificación, la cual es apropiada para

situaciones en las que hay un número fijo de posibles soluciones. De este conjunto, se evalúan conclusiones hipotéticas para ver si la evidencia las apoya. Esta evaluación se hace generalmente por encadenamiento hacia atrás a través de reglas si-entonces (esto es, antecedente-consecuente), empezando con las reglas que tienen las conclusiones hipotéticas como sus consecuentes. Se investigan después las reglas que tienen como su consecuente las condiciones que apoyan los antecedentes (condiciones de entrada) en la regla de la conclusión hipotética. Este proceso continúa recursivamente hasta que la hipótesis queda completamente probada, o hasta que se alcanza una respuesta negativa, o se agotan las posibilidades. Si ocurren cualquiera de los dos últimos resultados, se pueden intentar hipótesis adicionales hasta que se alcanza alguna conclusión o termina el proceso. En principio, el planteamiento de encadenamiento hacia atrás fue popularizado por el sistema experto MYCIN. Su correspondiente paquete de software para construir sistemas expertos, llamado EMYCIN, es el prototipo de virtualmente todos los paquetes comerciales (shells) disponibles hoy; puestos en marcha por hipótesis (esto es, activados por metas).

El encadenamiento hacia adelante comienza con datos que deben ser ingresados, o con el estado de la situación actual en una base global de datos. Los datos o la situación actual entonces se emparejan con las condiciones del antecedente, en cada una de las reglas pertinentes para determinar la aplicabilidad de la regla a la situación actual (la cual se representa en la base global de datos por un conjunto de

atributos y sus valores asociados). Entonces se selecciona una de las reglas que armoniza con estas condiciones (por ejemplo, por el uso de **meta-reglas**, las cuales ayudan a determinar el orden en el que se tratan las reglas; o por prioridades), y se usa el consecuente de la regla para añadir información a la base de datos o para actuar sobre algún proceder que cambia la situación global. El encadenamiento hacia adelante prosigue recursivamente (de una manera similar al encadenamiento hacia atrás), concluyendo cuando se alcanza el resultado o conclusión deseada, o cuando todas las reglas pertinentes se han agotado. La combinación de encadenamiento hacia adelante y hacia atrás ha demostrado ser muy útil en muchos casos.

El **razonamiento hacia adelante** (una forma más general de encadenamiento hacia adelante) puede hacerse con reglas manejadas por datos, o con procedimientos manejados por datos (demons).

El **razonamiento hipotético** se refiere a los planteamientos de solución en los que es necesario hacer suposiciones para habilitar el procedimiento de búsqueda y seguir adelante. Sin embargo, al final de la trayectoria de búsqueda puede suceder que ciertas suposiciones no sean válidas y deban retractarse. Este **razonamiento no-monotónico** (esto es, el razonamiento en el cual los hechos o conclusiones se retractan a la luz de nueva información) puede manejarse de varias maneras. Un planteamiento que reduce la dificultad de la computación, es llevar consigo múltiples soluciones (estas soluciones representan hipótesis diferentes) paralelamente, y descartar las inapropiadas a medida que se

reúne la evidencia que las contradice. A este planteamiento se le conoce como puntos de vista, contextos y ambientes, en herramientas diferentes. Otro planteamiento es seguir la pista de las suposiciones que apoyan la trayectoria de búsqueda actual, y buscar hacia atrás hasta el punto en el cual esta trayectoria se invalida. A este último planteamiento se le conoce como búsqueda no cronológica hacia atrás. Una aptitud relacionada es el mantenimiento de la verdad, que remueve creencias derivadas cuando sus condiciones ya no son válidas.

La programación orientada a objetos es un planteamiento en el cual la información acerca de un objeto y sus procedimientos se agrupan juntos dentro de una estructura de datos, tal como un marco. Estos procedimientos se ponen en marcha por mensajes que se envían al objeto desde un controlador central, u otro objeto. Este planteamiento es particularmente útil para simulaciones que involucran un grupo de objetos distintos, y para procesamiento de señales en tiempo real [6].

El planteamiento de inferencia de pizarrón está asociado con un grupo de sistemas expertos cooperativos que se comunican compartiendo información sobre una estructura común de datos, a la que se hace referencia con el nombre de "pizarrón". Para facilitar el desarrollo de la solución en el pizarrón se utiliza un sistema de agenda.

En el caso de las herramientas para la construcción de sistemas expertos, por lógica se refiere comúnmente a un planteamiento basado en la demostración de teoremas que implica unificación. La "unificación" se refiere a las

sustituciones de variables, ejecutadas de tal modo que se hace armonizar idénticamente a dos items. Las implantaciones de lógica común son versiones de un lenguaje de programación lógica (Prolog, por ejemplo), que utiliza un planteamiento de búsqueda exhaustivo a primera profundidad.

Una posibilidad importante de inferencia encontrada en algunas herramientas, es la habilidad de generar reglas o árboles de decisión inductivamente a partir de ejemplos. Los expertos humanos, a menudo son capaces de articular mejor su experiencia en forma de ejemplos antes que en forma de reglas. Por lo cual las técnicas de aprendizaje inductivo (que suelen tener una expresividad limitada) son con frecuencia métodos ideales de adquisición de conocimiento para prototipeamiento rápido, cuando los ejemplos pueden expresarse sencillamente en la forma de una conclusión asociada con una agrupación simple de atributos. Los constructores humanos del sistema experto resultante pueden después refinarlo iterativamente al criticar y modificar los resultados producidos inductivamente.

La inferencia inductiva usualmente comienza con uno de los parámetros de entrada y busca un árbol que contenga el mínimo número de decisiones que se necesitan para llegar a una conlusión. Este árbol de profundidad mínima se halla por un proceso cíclico a través de todos los parámetros como posibles nodos iniciales, y usando un planteamiento teórico de información para seleccionar el orden de los parámetros que se usarán para los nodos restantes, y para determinar qué parámetros son superfluos. Un "planteamiento teórico de información" es aquel que escoge la solución que requiere la

mínima cantidad de información para representarla. La profundidad del árbol es generalmente superficial (menos de cinco decisiones), así que grandes cantidades de ejemplos resultan en árboles amplios y superficiales.

Algunas herramientas incorporan demons que supervisan valores locales, y ejecutan procedimientos cuando se dan las condiciones de activación de estos demons. Estas herramientas son particularmente apropiadas para aplicaciones de supervisión.

Algunas herramientas ofrecen una selección de entre varios procedimientos de inferencia o de búsqueda. En sistemas construidos con tales herramientas, el diseñador posee medios para seleccionar la estrategia de inferencia dependiendo del estado del sistema. A tal control se le conoce como **meta-control**. Una forma de meta-control es el uso de **bloques de control**, que son procedimientos genéricos que le dicen al sistema los próximos pasos a seguir en una situación dada, de modo que se reduzca la búsqueda, y habilitando un gran número de reglas que se acomodan sin que el espacio de búsqueda se vuelva combinatoriamente explosivo.

Como la certidumbre de los datos, reglas y procedimientos es generalmente menor que 100 por ciento, la mayoría de los sistemas incorporan facilidades para el manejo de esta certidumbre. Por lo tanto, poseen alternativas para combinar reglas inciertas e información, determinando así un valor de certeza para el resultado.

A menudo se requiere el **emparejamiento de muestras** para mecanizar las técnicas de inferencia, particularmente para

emparejar los antecedentes de las reglas con el estado actual del sistema. La sofisticación del emparejamiento de muestras, afecta las capacidades del sistema. La variedad de este planteamiento va desde el emparejamiento de cadenas idénticas de caracteres, hasta variables, literales, y caracteres de sustitución (*, ?), e inclusive pueden incluir emparejamiento parcial y/o aproximado, que puede servir como razonamiento analógico.

Otras capacidades varían de herramienta a herramienta. Algunos motores de inferencia ofrecen cálculos matemáticos rápidos y sofisticados. Una de las capacidades más valoradas la suministran los motores de inferencia que pueden manejar bases de conocimiento modularizadas, o subproblemas de solución modularizada, mediante accesar y enlazar estos módulos según se necesiten [14].

Una consideración final respecto a las diversas posibilidades de inferencia es el grado de integración de todas ellas. Lo deseable es una integración completa, en especial para tratar problemas complejos.

1.4 SISTEMAS EXPERTOS PARA DIAGNOSTICO DE FALLAS

El diagnóstico de fallas es uno de los más grandes dominios de aplicación de los sistemas expertos de hoy, a pesar de que es todavía uno de los mejores ejemplos de tareas en las cuales el ser humano es mucho más efectivo que una computadora. En vista de ello, la meta de mejorar y asistir al desempeño humano en este campo, aún es la mejor manera de lograr un desempeño superior de un sistema global de diagnóstico [35].

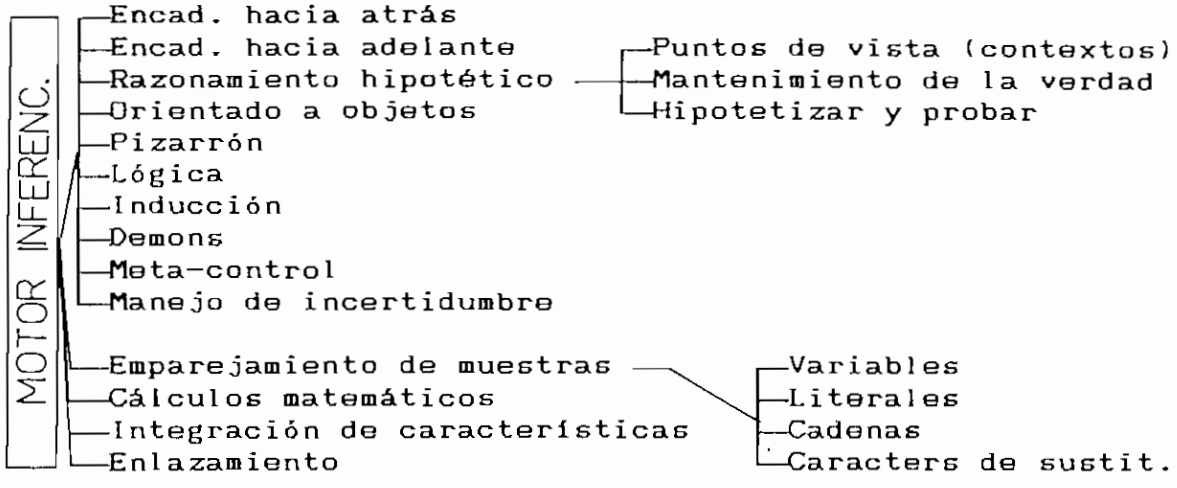


Figura 1.3. Posibilidades del motor de inferencia.

Aunque la mayoría de las aplicaciones anteriores se basaban en reglas; hoy en día se cuentan con técnicas poderosas que son capaces de razonar en base de modelos profundos, e implicarlos en un amplio alcance: aprendizaje a partir de la experiencia, información probabilística, y aprendizaje a partir de ejemplos [17].

Un humano que es experto en el diagnóstico y reparación de un aparato particular, eléctrico o mecánico, tiene muchas clases de conocimiento acerca del problema.

Como cualesquier otro ser humano, tiene cierto conocimiento de sentido común, el cual incluye las propiedades físicas del mundo, así como procedimientos básicos y mejores planteamientos para cada tipo de problema. Un experto también posee lo que podría llamarse "conocimiento fundamental del dominio", llamado en la literatura conocimiento "profundo", "causal", "funcional", o "físico". Este es el conocimiento que el experto adquiere en las etapas tempranas del desarrollo de su experiencia -es decir, lo que un novato en el área debería saber-. Esto incluiría

el conocimiento básico concerniente a cómo y porqué funciona el aparato, y los procedimientos para cazar fallas; proveyendo así la información de causa y efecto que se necesita cuando se diagnostica un malfuncionamiento. Un tercer tipo de conocimiento, conocido como "superficial", "compilado", "empírico", o "experimental", es lo que distingue a un experto de un novato; ya que provee "atajos" a través del conocimiento, permitiéndole al experto resolver más rápido los problemas, y con mayor eficiencia y precisión que un novato. Por supuesto, no existe realmente una línea clara entre cada uno de estos tipos de conocimiento. Estos son simplemente puntos discretos entre el conocimiento continuo de sentido común y el experimental. Esta continuidad de conocimiento a menudo cierra el círculo- por ejemplo, una experiencia puede pasar a convertirse también en sentido común. Un experto humano pone en juego todos los tipos de conocimiento antedichos, de una manera altamente integrada, cuando diagnostica un problema [11], [17],

La tecnología corriente de sistemas expertos, ha dependido mucho del conocimiento superficial. Sin embargo, las investigaciones más recientes, especialmente con respecto a diagnóstico y reparación, ha enfatizado el uso del conocimiento profundo [35], [32].

1.4.1 ESTRATEGIAS DE DIAGNOSTICO

Existen varias dimensiones en estrategias de diagnóstico e información. La idea clave es que la organización del conocimiento básico conduce a estrategias que enfocan diferentes niveles de representación de conocimiento, haciendo posible desarrollar diferentes sistemas de diagnóstico. El

conocimiento básico que se necesita para el diagnóstico, es el conjunto de malfuncionamientos y su relación con sus respectivas observaciones. Un solucionador de problemas puede tener todo esto directamente, o puede razonar a partir de algún otro conocimiento (a menudo llamado "causal", o modelos "profundos") para "compilar" el conocimiento de diagnóstico. A los sistemas que se les ha dado explícitamente este conocimiento, se les llama "sistemas de conocimiento compilado".

Típicamente un problema de diagnóstico empieza con la observación de algún comportamiento (valores de prueba, señales y síntomas, etc) que se reconoce como una desviación de lo que se desea, o se espera que sea. En esta etapa, el solucionador de problemas genera algunas hipótesis (posiblemente clasificadas) de las causas de falla. Esta tarea inicial de generación de hipótesis puede ser más o menos complicada, y más o menos controlada dependiendo del dominio y del conocimiento que el solucionador de problemas tenga. En esta etapa, un pequeño número de las mejores hipótesis se consideran como las diferenciales o grupo candidato. En un sistema compilado, el conocimiento de qué pruebas adicionales se pueden usar para confirmar o rechazar cada una de estas hipótesis, puede estar disponible explícitamente. Si este conocimiento no está disponible directamente pero se conoce la estructura del dispositivo, entonces se podrá asumir que su estructura ha cambiado en correspondencia a cada una de las hipótesis (en la lista diferencial) y razonar respecto a qué proceder seguir [25]. Esta estrategia sirve para introducir cada posible falla, y

observar su impacto en el sistema total. Después se empareja el estado actual del sistema con una biblioteca de fallas. Esto es lo que se conoce comúnmente como el planteamiento de modelos defectuosos. Esta información puede usarse para discriminar entre diferentes hipótesis de la diferencial.

Puesto que los expertos humanos hacen este razonamiento "directo" cualitativamente, en Inteligencia Artificial ha habido un sustancial interés en un cuerpo de técnicas llamadas simulación cualitativa. Al abandonar la precisión de la información numérica, la física cualitativa adquiere la habilidad de alcanzar conclusiones incluso con muy poca información; sólo que las predicciones pueden ser ambiguas: donde la física tradicional da una predicción exacta, la física cualitativa restringe las posibilidades a un conjunto pequeño. Sin embargo, para muchos propósitos conocer esto es suficiente, y cuando no lo es, la estructura de trabajo provista por el análisis cualitativo puede guiar la aplicación de conocimiento más preciso.

La totalidad de las estrategias disponibles y sus relaciones se ilustran en la figura 1.4. Este diagrama representa cuatro niveles básicos de representación de conocimiento. Estos son: estructural, conductual, funcional, y emparejamiento de muestras. De cada representación de conocimiento se puede derivar el próximo nivel superior de representación, en un sistema de diagnóstico [27].

Dada una representación de la conducta de los componentes de un dispositivo o sistema, y una representación de su estructura (p.e. la interconexión de los componentes), el razonamiento causal tiene la habilidad de generar

descripciones de la conducta del sistema como un todo. La simulación cualitativa y la consolidación han sido métodos para hacer esto. En general, esta técnica es útil para producir información del comportamiento por rangos de valores de los componentes.

Con frecuencia, puede ser necesario abstraer el comportamiento a un nivel superior a aquel en el que se especificó el componente. Por ejemplo, en un circuito electrónico, el comportamiento de los componentes tales como un transistor y un resistor, puede estar en términos de voltajes y corrientes; mientras que un dispositivo que los contiene puede describirse como un amplificador, u oscilador [11].

A partir del conocimiento estructural y a través del proceso de razonamiento cualitativo, razonamiento teleológico, y compilación, se puede obtener el modelo de un sistema experto basado en reglas.

En el otro extremo de la escala de conocimiento, se puede ingresar un sistema experto de conocimiento compilado en el nivel de emparejamiento de muestras compilado. Esto es típico de la mayoría de los sistemas expertos de hoy basados en reglas.

Se puede también entrar en ese nivel con una descripción basada en rasgos distintivos y salir con un sistema experto basado en reglas, o basado en casos y experiencia. Si se usa un modelo cualitativo, y se entra en el nivel de conocimiento conductual, se puede obtener un modelo cualitativo profundo del sistema basado sólo en la conducta.

En el nivel funcional se puede entrar con una simulación funcional del sistema, y usar un compilador de reglas para

generar un sistema basado en reglas; o permanecer en el nivel funcional de representación y tener lo que se conoce como un sistema de diagnóstico basado en modelos profundos de funcionamiento.

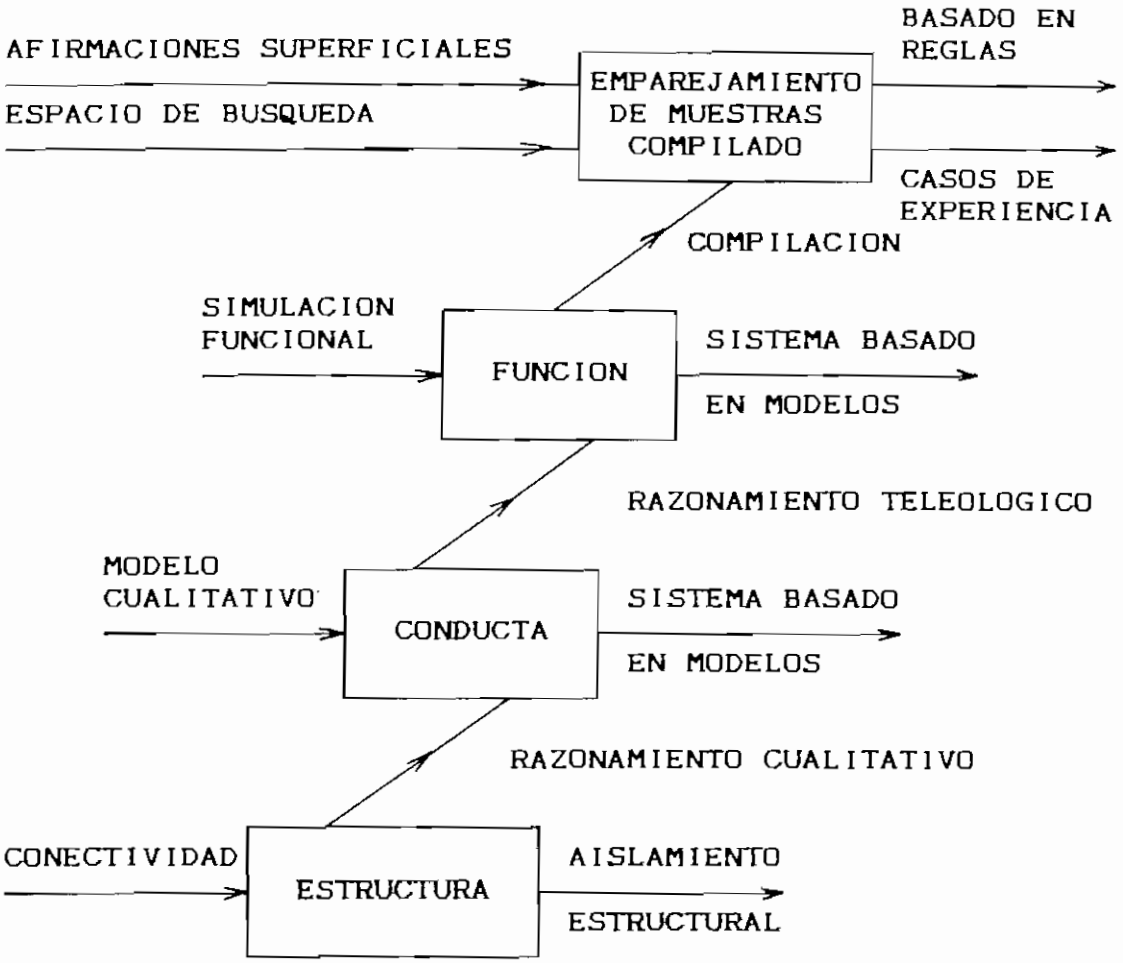


Figura 1.4. Niveles de razonamiento de diagnóstico.

Así, se pueden obtener cuatro tipos diferentes de sistemas de diagnóstico basados en cada uno de los niveles: Sistemas que usan conocimiento estructural primario, conocimiento conductual, conocimiento funcional, o emparejamiento de muestras compilado.

La meta real del proceso de diagnóstico, es generar un diagnóstico que pueda "explicar" todas las observaciones, especialmente fallas múltiples (dependiendo de las clases de

conocimiento disponibles, este caso puede ser muy complejo, o de complejidad controlable).

Debido al hecho de que ningún modelo "profundo" del dispositivo bajo diagnóstico se usa en los sistemas de conocimiento compilado, con frecuencia se les llama sistemas de conocimiento "superficial" [27].

Un paradigma más reciente en razonamiento superficial, es lo que se llama un sistema de diagnóstico basado en casos. En tal sistema, la información de diagnóstico se almacena como un cierto número de casos pasados. Cada caso describe las condiciones bajo las cuales fue aplicable y los resultados o tratamientos que surgieron en esa ocasión. Se usa un planteamiento indexado o de emparejamiento de muestras para identificar qué caso histórico es el más semejante al de la situación actual. Para conseguir el mejor emparejamiento, se emplean con frecuencia las técnicas normales de reconocimiento de modelos. Este es un planteamiento muy efectivo en situaciones donde las relaciones causales son muy complicadas o no están disponibles, y se cuenta con muchos casos pasados. En tales situaciones la superficialidad es exactamente lo que hace posible construir un sistema de diagnóstico efectivo [27], [35], [25].

1.4.1.1 RAZONAMIENTO ESTRUCTURAL

Uno de los tipos más simples de conocimiento que se puede usar en un sistema de diagnóstico, es la información estructural o de conectividad. En muchos sistemas puede ser difícil entender, o describir el funcionamiento del sistema, pero puede ser simple describir su conectividad. Los diagnosticadores son muy buenos también, usando la

información estructural para guiar un diagnóstico, aun cuando no entiendan la función de una parte del dispositivo. Este razonamiento puede ser eficiente, simple y rápido. De modo que suele ser deseable usar la información estructural para aislar una región defectuosa, y entonces emplear razonamiento más complejo para identificar el problema exacto. La información estructural también ayuda a derivar un gran número de posibles fallas directamente de la descripción del sistema, a través de un mecanismo de inferencia. Así, el sistema genera las reglas en vez de escribir cientos de ellas. Este acabado se deriva del examen de todas las conexiones estructurales y trayectorias garantizando que nada se olvide. Una vez que se ha usado la estructura para aislar la falla a una región, se puede usar la información funcional para un razonamiento adicional que aísle el elemento que esté fallando [25], [27].

1.4.1.2 RAZONAMIENTO CUALITATIVO

La mayoría de los sistemas de diagnóstico basan su inferencia y trabajo en datos reales que entran o se miden desde el ambiente. En estos sistemas, el primer paso es traducir los datos numéricos en valores cualitativos, que puedan ser analizados mejor por el resto del sistema del experto. Esta traducción de datos numéricos a cualitativos provee un medio importante de reducción de la información. Así, antes que preocuparse de un amplio rango de valores numéricos, ahora se puede enfocar y razonar en los estados cualitativos interesantes que el sistema puede exhibir. Un modelo cualitativo se puede usar para construir un sistema de diagnóstico completo [27].

1.4.1.3 MODELOS PROFUNDOS

Ejecutar un modelo de diagnóstico bajo varias suposiciones de falla, parece una buena manera de generar información acerca de qué observaciones esperar para cada falla. Sin embargo, para aclarar lo que significa poseer y ejecutar un modelo, conviene distinguir dos clases de modelos: 1) aquellos donde se usa numérica o cualitativamente una descripción matemática fundamental, y 2) aquellos donde se usa una representación de los resultados causales, por medio de los cuales el agente entiende la función del dispositivo. Este último tipo de conocimiento hace posible ir más allá de un simple funcionamiento de entrada/salida del sistema, y también posibilita el entendimiento de cada uno de los pasos fundamentales, y de la secuencia causal que se sigue.

La clave para el uso de modelos cualitativos cuenta con una representación apropiada de conocimiento, y un algoritmo para la manipulación de esa representación de conocimiento.

Sin embargo, los modelos ya sean conductuales, funcionales, o causales, están entre los mecanismos centrales para organizar sistemas de diagnóstico más poderosos. Estos modelos son la clave para los futuros sistemas de diagnóstico. Primariamente, se usan como un mecanismo de simulación directa. Esto es, dadas una serie de entradas en un nivel el modelo produce una descripción de las salidas al próximo nivel superior. La introducción de modelos defectuosos (modelos que representan al sistema en un estado defectuoso), son cruciales para verificar y confirmar las hipótesis de fallas.

Usando cadenas de conocimiento ya sea estructural o causal, es posible identificar qué componentes pueden estar fallando en el sistema.

Uno de los beneficios primarios de la información estructural, es identificar rápidamente las series de componentes que pueden estar fallando. Luego se usa la información funcional, para hipotetizar los modelos que pueden o no haber producido las fallas.

Se han hecho trabajos que enfocan cómo usar el conocimiento de funcionamiento para eliminar posibles candidatos, esto es, declarar la inocencia de una parte del sistema. Estos trabajos ilustran dos áreas importantes. Una es un ejemplo de cómo la función y estructura trabajan juntas para identificar y eliminar posibles candidatos. En segundo lugar, abordan una de las subáreas más importantes en la solución del problema de diagnóstico: Cómo se determina si la entrada o el sensor están mal, más bien que el sistema. Al razonar de nuevo acerca de cómo las posibles fallas no podrían haber ocurrido, es capaz de eliminar muchas posibilidades.

El uso primario de modelos es a través de un conjunto de ecuaciones cualitativas u otras que proveen una simulación de lo que hace cada componente. Un factor importante que restringe el progreso y desarrollo de complejos sistemas de diagnóstico, es la aptitud de razonar respecto de las ecuaciones actuales o de la teoría formal que representa a ese modelo, más bien que desde un modelo dibujado a mano [25], [27], [35].

1.4.1.4 USO DE LA EXPERIENCIA

Cuando se trata de construir un sistema práctico de diagnóstico, surgen varias limitaciones del razonamiento basado en modelos. Una es el hecho de que no todas las tareas que se necesitan para un sistema práctico, se cumplen fácilmente con el entendimiento de hoy del razonamiento basado en modelos. No es que sean teóricamente imposibles en un sistema de conocimiento profundo; es sólo que no es práctico incorporarlas, sea debido a falta de tecnología o a falta de recursos. En segundo lugar, a menudo se descubren fallas durante el funcionamiento real del sistema. Y es importante hacer correcciones para estas fallas muy rápidamente, a nivel del modelo profundo para lograr que el sistema continúe funcionando. Así, salen a la luz un conjunto de tareas para lograr el acoplamiento apropiado de los modelos profundos del sistema con el razonamiento superficial. Asumiendo que se ha derivado un sistema de diagnóstico de emparejamiento de muestras compilado, a través de alguna cantidad de razonamiento a partir del comportamiento estructural, o conocimiento funcional, se tendría un clásico sistema de diagnóstico basado en modelos. Sin embargo, en este punto se puede suplementar el sistema con aserciones superficiales adicionales, y desarrollar una base aun más grande de reglas que consista no sólo del conocimiento que se ha derivado de los niveles inferiores de la representación de conocimiento, sino también de razonamiento adicional, puramente superficial [27].

1.4.1.5 APRENDIZAJE

La incorporación de la experiencia como un suplemento al razonamiento profundo de un sistema, es el primer paso hacia un aprendizaje y auto-mejoramiento del sistema. Por ejemplo, un planteamiento de aprendizaje dirigido por fallas, se da cuando ocurre una falla, y el sistema de diagnóstico usa el método de aprendizaje de trozos compactos, para identificar qué información debería incorporarse en nuevas reglas desde los modelos del dispositivo. El sistema experto examina tres posibles razones de falla para confirmar una falla tomada como hipótesis. Puede haber habido una predicción inconsistente, una violación de restricciones, o una entrada inusual. Para cada una de estas situaciones existe una estrategia, para la corrección más óptima de la heurística al concentrarse en cómo distinguir una falla de otra. El sistema puede aprender un desempeño comparable a, y en algunos casos, mejor que solamente con el uso del conocimiento experto. En éstos trabajos, se investiga cómo la experiencia puede contribuir al refinamiento y modificación del proceso de razonamiento; además de los casos previos que suministran ejemplos para decisiones posteriores. Sin embargo, tan pronto como se desee añadir reglas superficiales a un sistema de modelo profundo para suplementar sus capacidades, también se necesitará un mecanismo de razonamiento para inferir analogías en los casos existentes llenando así vacíos, o refinando una selección de casos con los que ya se cuenta.

Se pensaría que el aprendizaje tiene un papel que cumplir en cada fase del sistema. Sin embargo, hasta la fecha, el

aprendizaje se ha enfocado únicamente para el refinamiento de reglas, o en la incorporación de conocimiento experimental [27].

1.4.1.6 DIAGNOSTICO DE LA CONDUCTA DEL SOLUCIONADOR DE PROBLEMAS.

Con el rápido incremento de la complejidad de los sistemas en desarrollo, se vuelve muy difícil entender y mantener los sistemas que se construyen. Al mismo tiempo, existe una tendencia hacia sistemas más autónomos. Para estos complejos sistemas futuros, será necesario trabajar de una manera autónoma para razonar respecto a su propio proceder en la solución de problemas. En este tipo de diagnóstico el dominio del problema no es un dispositivo físico, sino un solucionador de problemas. En algunos trabajos se ha ilustrado cómo se aplican en esto los métodos causales basados en modelos. En muchos casos no existe ningún modelo o patrón absoluto de la función correcta del sistema. En tales casos se introduce una técnica llamada de "razonamiento comparativo". Este planteamiento le permite al sistema de diagnóstico seleccionar su propio "criterio de comportamiento correcto"; a partir de los objetos dentro de su propio espacio de solución del problema [27].

En los primeros sistemas expertos, el conocimiento se representaba en la forma de reglas de producción. El motor de inferencia era un mecanismo muy simple. Sencillamente encontraba el grupo de reglas que podrían concluir la hipótesis actual, e intentaba una a la vez. Para satisfacer una

regla se requerían otras reglas. Esto condujo a la recursión y al encadenamiento hacia atrás.

En cambio ahora se reconoce ampliamente en sistemas de resolución de problemas que "el conocimiento es poder"; y en su diseño ha habido un interés creciente en avanzar hacia estructuras de conocimiento múltiple más complejas. Los trabajos sobre modelos causales muestran que se están haciendo progresos significativos en esto [30], [35].

1.4.2 UN SISTEMA EXPERTO DE DIAGNOSTICO PARA DOMINIOS ELECTRICOS.

Puesto que los sistemas expertos basados en conocimiento superficial poseen limitadas aptitudes de explicación, y se degradan rápidamente cuando se enfrentan con problemas que apenas caen fuera de su dominio de experiencia. La investigación de sistemas expertos especialmente en diagnóstico y reparación, enfatiza el uso del conocimiento profundo a través de modelos físicos o causales. Estos modelos pueden contener un entendimiento muy profundo de cómo funciona el dispositivo. Por lo tanto, se trata de combinar este entendimiento con la hábil experiencia de un experto para resolver problemas. Sin el mismo conocimiento que está disponible a un experto humano, no se puede esperar que un sistema experto se comporte de una manera precisa y robusta.

Un sistema experto, llamado el Modelo de Diagnóstico Integrado (MDI), intenta aplicarse a algunas de estas cuestiones en el área limitada del diagnóstico y reparación de dispositivos eléctricos. El MDI difiere de muchos sistemas expertos de varias maneras: Contiene explícitamente

dos tipos distintos de conocimiento, experimental y fundamental. Cada uno de ellos se estructura y representa separadamente con su propia base de conocimiento y motor de inferencia. De modo que cada uno es un sistema experto en sí mismo. Estos dos sistemas expertos se han integrado en un sencillo sistema experto por vía de un ejecutor, que puede alternar sobre cualquiera de los dos en un problema de diagnóstico (ver figura 1.5). Esta base de conocimiento también provee un medio de "traducir" el conocimiento entre las bases de conocimiento experimental y fundamental. El resultado es un sistema experto que puede resolver problemas, aun cuando no exista conocimiento experimental para manejar la situación, resultando en una degradación más suave de sus aptitudes en la periferia de su conocimiento. Además suministra diferentes niveles de explicación. El modelo construído se usa para una simulación cualitativa del funcionamiento correcto e incorrecto del dispositivo [11].

Al llevar a cabo un diagnóstico, los humanos primero examinan la información inicial y las observaciones de un malfuncionamiento. Basándose en este análisis, deducen hipótesis, las cuales constituyen un marco de referencia para adquirir más datos a partir de más observaciones, pruebas, etc. Cuando se confirma una hipótesis con la suficiente certidumbre, se intenta rectificar el problema. Si la reparación falla, el experto reinicia el proceso de diagnóstico [11].

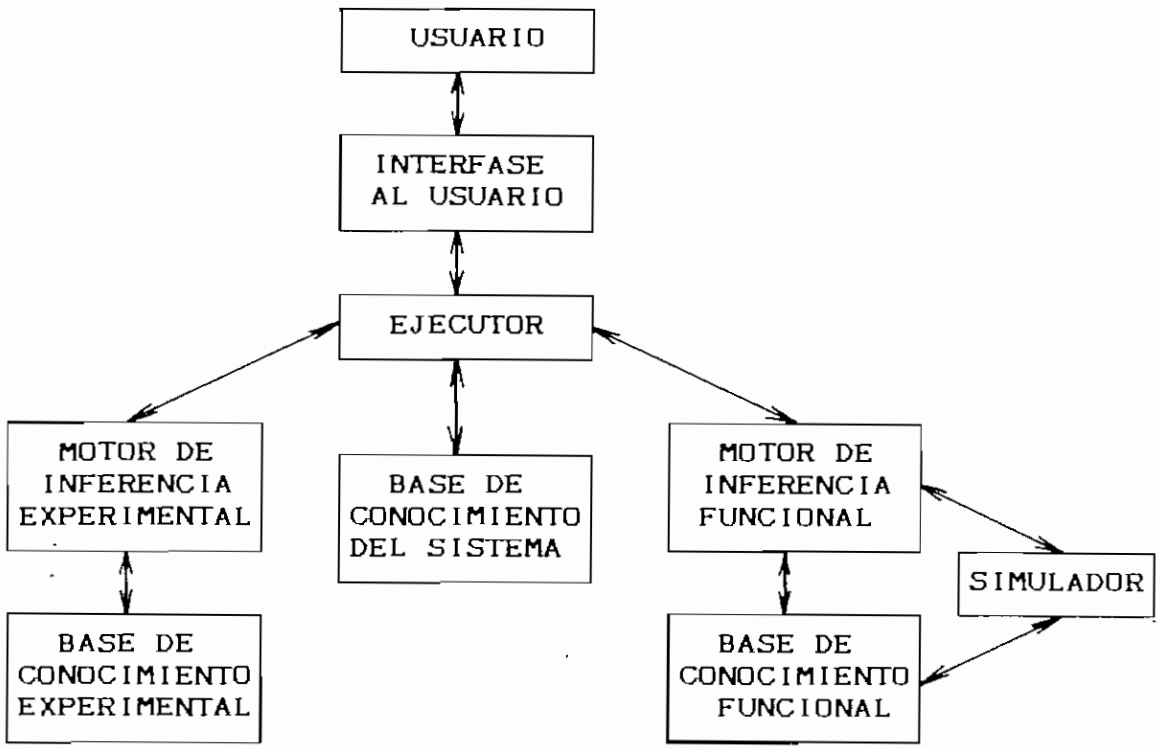


Fig. 1.5. Vista global del modelo de diagnóstico integrado.

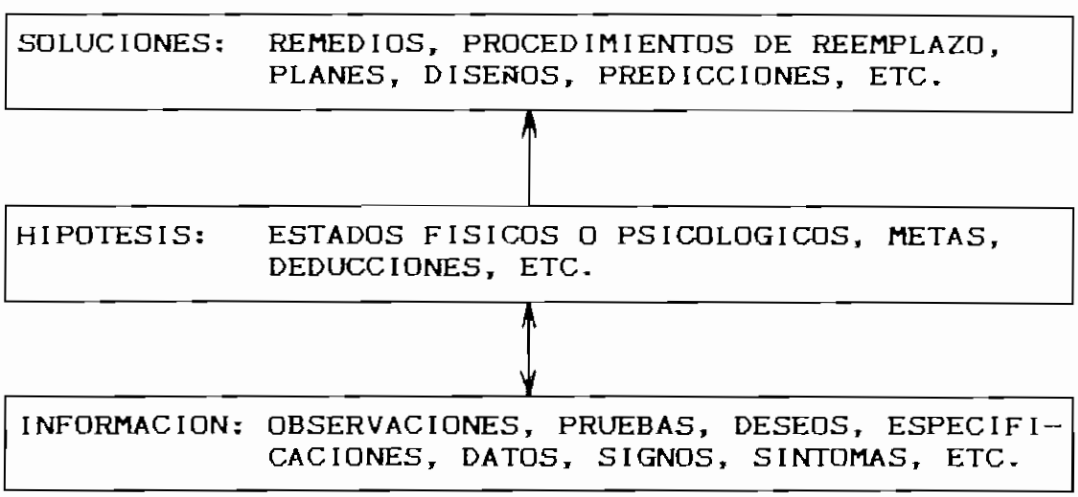


Figura 1.6. Estructura general de la base de conocimiento experimental.

Este proceso se sigue sin importar qué tipo de conocimiento esté guiando al humano en cualquier momento: experimental, fundamental, sentido común, o algún otro tipo.

El MDI usa redes semánticas para estructurar la base de conocimiento en tres niveles específicos de conocimiento del problema (Ver figura 1.6). Un nivel -el nivel de información- contiene conocimiento concerniente a observaciones, pruebas, etc. Estas se asocian con el conocimiento en el nivel de hipótesis, el cual provee sugerencias de la posible condición física del dispositivo. Estas hipótesis se asocian también con datos en el nivel de soluciones, donde se puede sugerir una reparación. Esa técnica suministra una estructura de conocimiento para el sistema experto, y una estructura para llevar a cabo un diagnóstico.

Además, durante el proceso de diagnóstico un experto humano tiende a moverse hacia atrás y hacia adelante entre los varios tipos de conocimiento que posee. El MDI trata de integrar los dos tipos de conocimiento que están disponibles: La información adquirida por una de las fuentes de conocimiento se hace disponible sin esfuerzo a la otra por vía del ejecutor. Además, una cierta cantidad de conocimiento se traslapa en las dos bases de conocimiento, pues no hay una línea clara entre ellas. Sin embargo, es probable que sin importar cómo se clasifique el conocimiento, algo del experto fundamental se aplicaría mejor antes que algo del experto experimental y viceversa. Así, El MDI empieza el diagnóstico dando al experto experimental (que es más eficiente en encontrar la solución al problema, si el problema es conocido) el control del proceso de diagnóstico, y si este falla en encontrar una solución entonces cambia el control al experto fundamental.

Mientras el experimental se desenvuelve, el fundamental "vigila" y toma la información del problema adquirida por el un experto como entrada para su capacidad de simulación. Esto se hace a través de la base de conocimiento del sistema; incluso el experto experimental adquiere la nueva información obtenida por la simulación a través de la misma base de conocimiento. Sin embargo, una vez que el experto experimental ha salido de sus posibilidades, ya no se lo usa de nuevo porque no tiene nada más que contribuir. El experto fundamental toma lo que se aprendió y procede a partir de aquí.

El MDI se lo hizo con la meta de aplicar sus conceptos a los dispositivos analógicos del mundo real: El controlador electrónico para el motor CFM56 construido por la General Electric para su uso en aviones KC-135, incluyendo algunos de los 737, 747, 767 [11].

1.5 HERRAMIENTAS COMERCIALES PARA CONSTRUCCION DE SISTEMAS EXPERTOS

El desarrollo de nuevos sistemas expertos se facilita cada vez más por las mejores herramientas para construirlos que se hacen disponibles. Se ha reportado que estas herramientas hacen posible desarrollar un sistema experto en un orden de magnitud de tiempo menor que lo que tomaría con el uso tradicional de lenguajes de programación simbólica, tales como Lisp [6], [14].

1.5.1 ESTRUCTURA

El núcleo de un sistema experto consiste de una base de conocimiento con su correspondiente motor de inferencia, que opera sobre la base de conocimiento para desarrollar una

solución o respuesta deseada. Además, se requiere una interfase final al usuario, o una interfase para un arreglo de sensores y actuadores para la comunicación con su ambiente de trabajo. (Un "ambiente" es un sistema o situación operado por, o en contacto con, el sistema experto). Además, para facilitar el desarrollo de un sistema experto, una herramienta (ESBT) debe también incluir una interfase para el diseñador del sistema.

El carácter y calidad de estas interfases son dos de las principales diferenciaciones entre ESBT's comerciales y ESBT's desarrolladas en universidades y usadas en investigación. Son también importantes en la estructura de las ESBTs:

- las interfases a otros softwares y bases de datos, y
- las computadoras en las cuales las ESBTs correrán durante el desarrollo del sistema experto y en su destino final donde el usuario.

La figura 1.7, resume la estructura de una ESBT.

La representación de conocimiento y el motor de inferencia se han descrito ya en los apartados 1.3.3 y 1.3.4 respectivamente [14].

1.5.1.1 INTERFASE PARA EL DISEÑADOR

Varias herramientas ofrecen diferentes niveles de aptitudes que el constructor del sistema experto puede usar para diseñar el sistema. Las herramientas más simples son shells dentro de las cuales se inserta el conocimiento en una modalidad estructurada y específica. Las herramientas más sofisticadas son generalmente más difíciles de aprender, pero en cambio le permiten al diseñador una elección más

amplia de representaciones de la base de conocimiento, estrategias de inferencia, y la forma de la interfase final al usuario. Se proveen varios niveles de asistencia en depuración. La figura 1.8, da una indicación de las posibles opciones (dependientes de la herramienta) que están disponibles para cada aspecto de la interfase al diseñador [14].

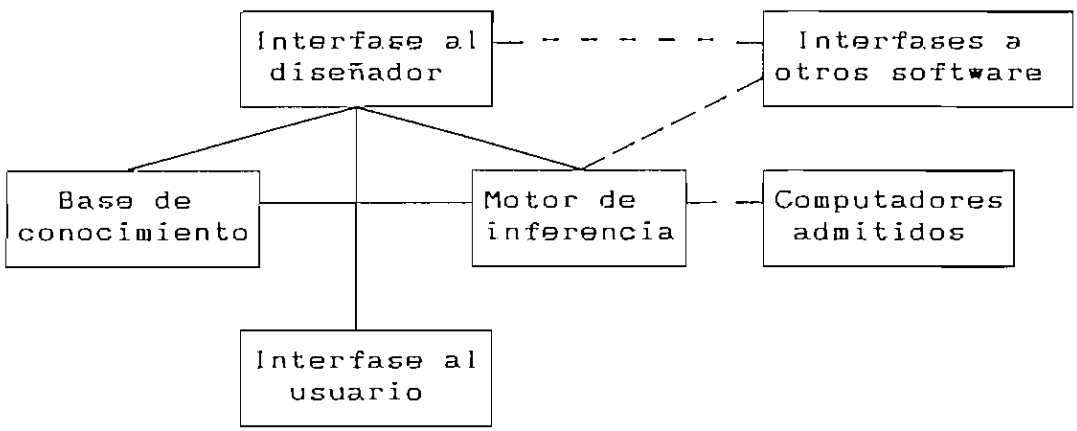


Figura 1.7. La estructura de una herramienta para construir sistemas expertos (ESBT's.-Las líneas sólidas representan relaciones básicas, y las líneas de puntos representan aspectos relacionados.)

1.5.1.2 INTERFASE PARA EL USUARIO

Una vez que se ha construido el sistema experto, su utilidad depende en gran parte de la interfase final al usuario. La figura 1.9, da una indicación de varias facilidades para el usuario que se hallan en las ESBT's. Puesto que la mayoría de los sistemas expertos son realmente ayudantes inteligentes, la interfase final al usuario se diseña con frecuencia para que permita un diálogo interactivo.

Este diálogo y/o la entrada inicial se presentan al usuario muchas veces como arreglos estructurados de entrada de datos, que contienen selecciones de menús.

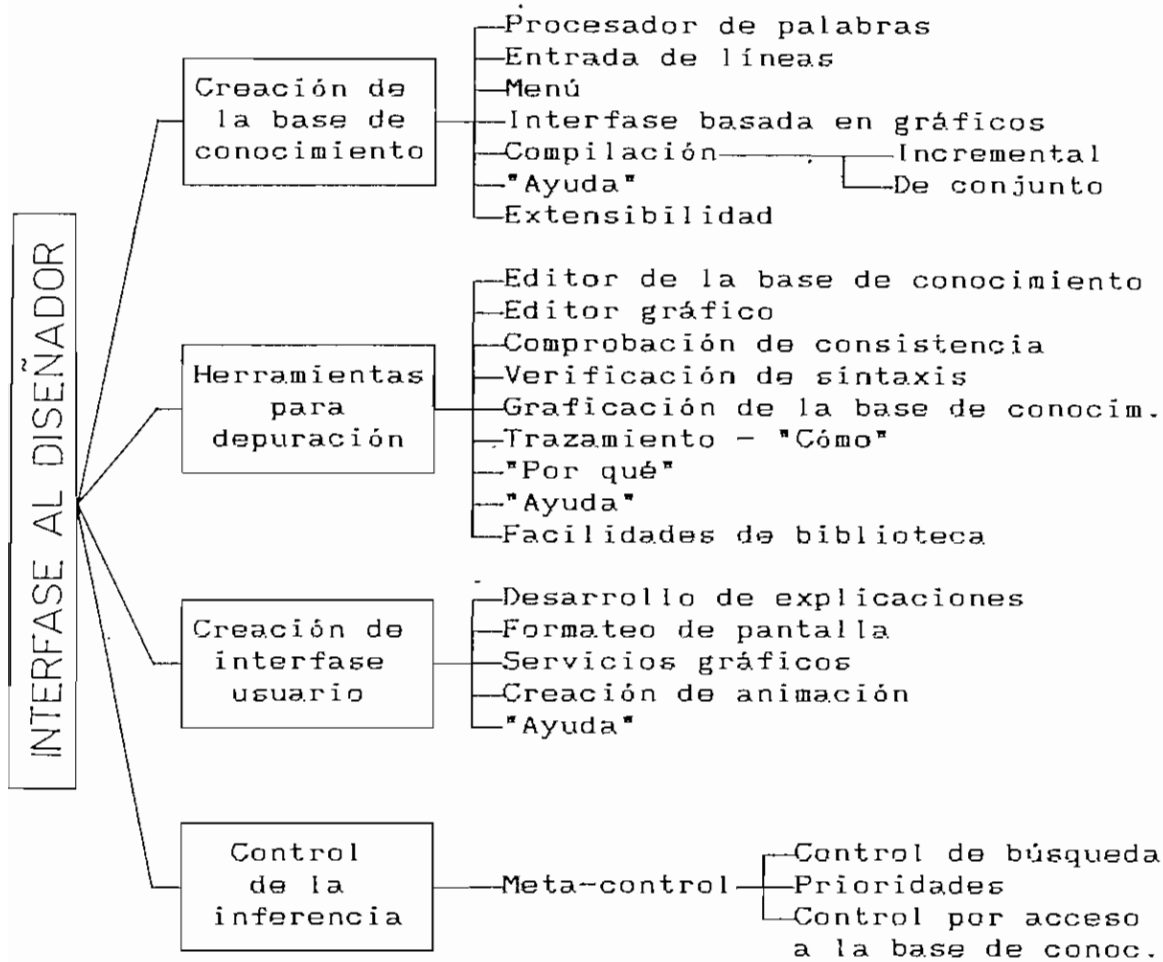


Figura 1.8. Posibilidades para la interfase al diseñador.

En algunos casos, para incrementar la flexibilidad del sistema, los sistemas aceptarán respuestas múltiples e inciertas del usuario, y todavía llegarán a conclusiones (aunque se reduzca la certidumbre de las conclusiones resultantes). En sistemas sofisticados, los gráficos se usan a menudo para mostrar la línea de razonamiento cuando el sistema responde a las preguntas "cómo" del usuario. En sistemas más simples, se presentará una lista de las reglas que apoyan las conclusiones del sistema. Las ESBTs también responden a la pregunta del usuario "¿por qué necesita usted esta información?", al citar la regla para la cual se requiere dicha información. La habilidad del sistema para

responder a las preguntas del usuario "Por qué" y "Cómo" es importante, ya que incrementa la confianza del usuario final en la aptitud del sistema para tomar decisiones.

Otras capacidades que a menudo se encuentran en ESBTs, son facilidades que le permiten al usuario seleccionar valores alternativos de parámetros y observar el efecto en las consecuencias (estas facilidades apoyan la pregunta "qué pasaría si..."), y que le permiten llevar a cabo una línea de cuestionamiento, de modo que el sistema no necesita dedicarse a áreas que el usuario ve que son inaplicables o innecesarias; y además la capacidad de guardar ejemplos para uso o consideración futura.

Las herramientas muy sofisticadas incluyen además gráficos interactivos y facilidades de simulación [14].

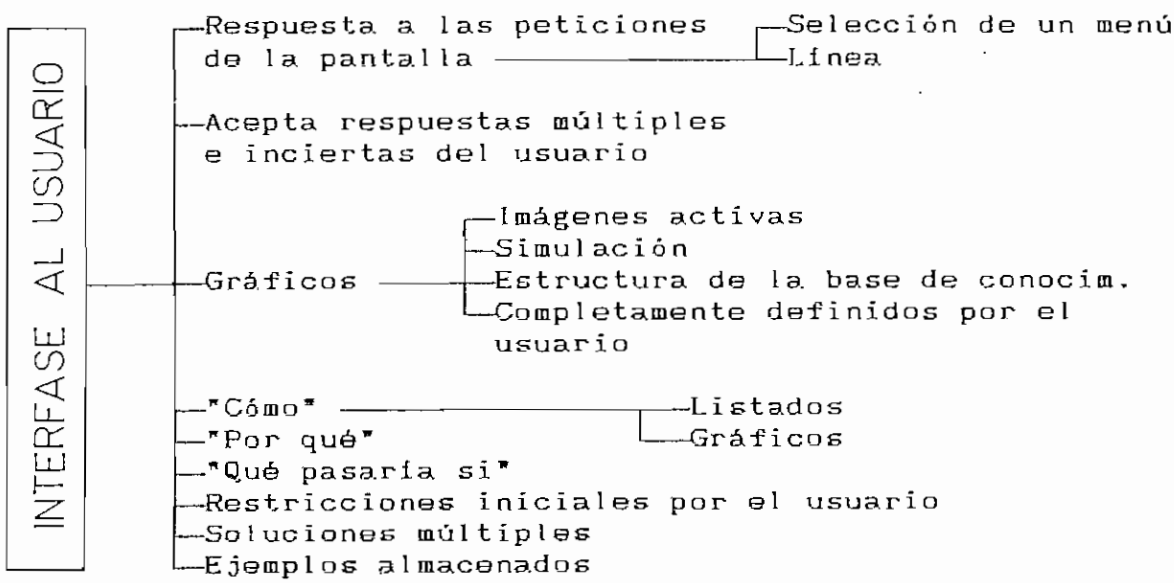


Figura 1.9. Posibilidades para la interfase final al usuario.

1.5.1.3 LENGUAJE DE PROGRAMACIÓN

La figura 1.10 esquematiza este aspecto de una ESBT. En general, las herramientas más sofisticadas se han escrito en Lisp. Sin embargo, aun estas herramientas están siendo reescritas en lenguajes tales como C, para aumentar la velocidad, reducir las demandas de memoria, y promover la disponibilidad de una variedad más grande de computadores, por su portabilidad.

Además las herramientas suelen tener enganches de lenguaje, para acceder a otros programas, o enganches de bases de datos, a fin de alcanzar otra información. En algunos casos, el sistema experto generado por la herramienta es completamente encajable en otros sistemas, lo que le hace más autónomo [14].

1.5.2 ATRIBUTOS Y APLICACIONES

De primera consideración son las aplicaciones funcionales que se pueden construir fácilmente con determinadas ESBT's. A continuación se hace una revisión de las mayores aplicaciones:

CLASIFICACION. Es por mucho, la función que más se aborda con los sistemas expertos, y se refiere a la selección de una respuesta desde un conjunto fijo de alternativas sobre la base de la información ingresada.

Algunas subcategorías de clasificación son:

* Interpretación de mediciones. Esto se refiere a la selección de hipótesis hecha en base de datos medidos e información de consecuencia natural.

* Diagnóstico. En diagnóstico, el sistema no solamente interpreta datos para determinar la dificultad sino también

busca datos adicionales cuando se requieren para apoyar su línea de razonamiento.

* Depuración, tratamiento, o reparación. Estas funciones se refieren a la toma de acciones o recomendación de medidas para corregir una situación desfavorable que se ha diagnosticado.

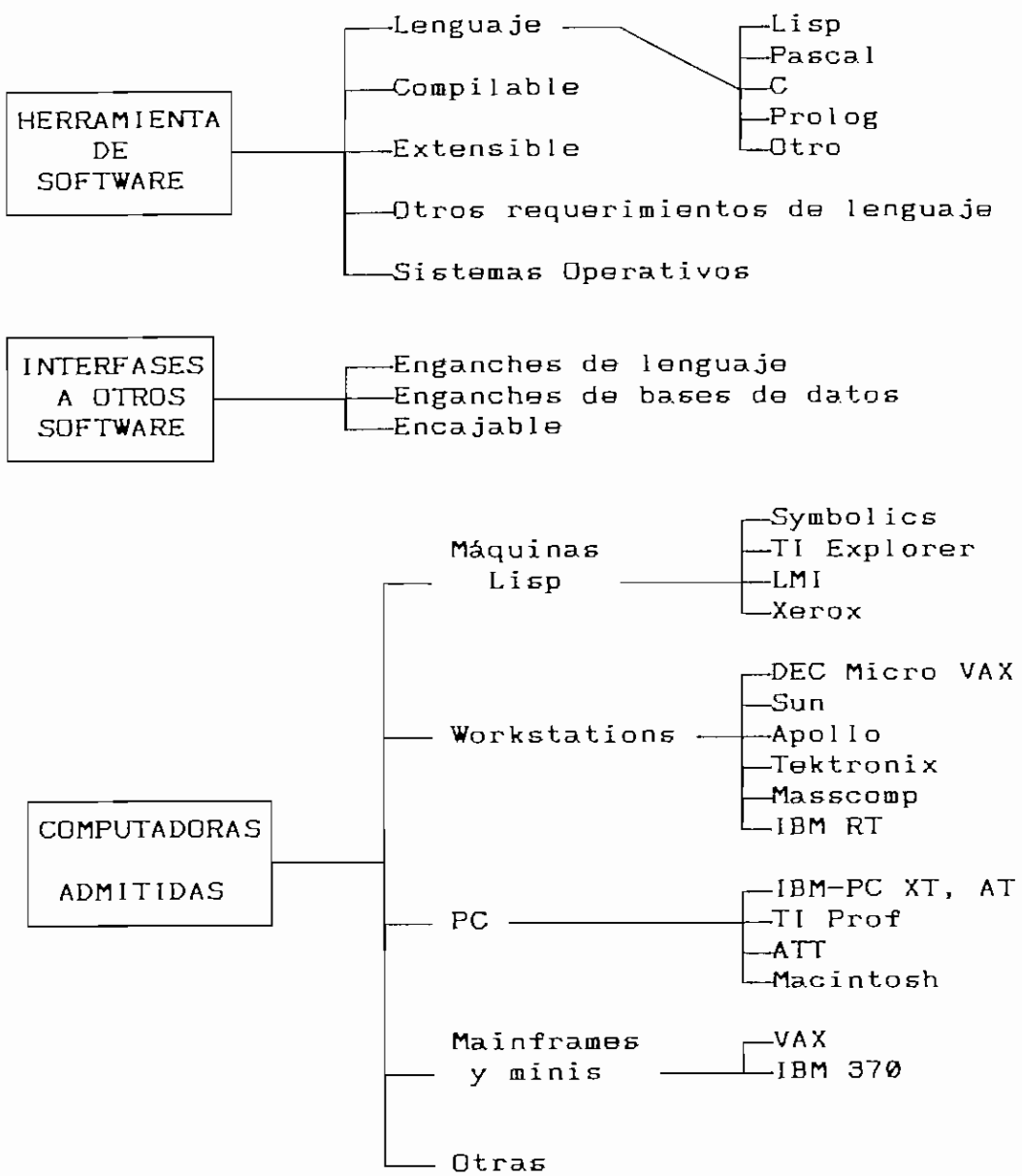


Figura 1.10. Aspectos de software y hardware de herramientas.

* Uso de consejero. Un sistema experto en calidad de fachada para un programa de computador o para una sección de maquinaria, puede ser muy útil a un usuario inexperto. Tales sistemas dependen tanto de las metas del usuario como de la situación en curso para sugerir qué hacer a continuación. Así, el dispositivo se desenvuelve según cambia el estado del ambiente. El uso de consejeros puede ser útil para guiar a los usuarios a través de procedimientos en otros dominios (por ejemplo, reparación de automóviles y pilotaje de aviones).

Se puede considerar que la clasificación y otras aplicaciones funcionales son de dos tipos: razonamiento superficial y razonamiento profundo.

En razonamiento superficial, no se emplea ningún modelo del sistema; el planteamiento asumido es, escribir una colección de reglas, cada regla aseverando que una cierta situación garantiza una cierta respuesta o conclusión. (Estas relaciones situación-respuesta se escriben usualmente como reglas heurísticas, recogidas y almacenadas a partir de la experiencia). En razonamiento profundo, el sistema compone modelos causales o estructurales del dominio de interés para ayudarse a llegar a la conclusión. Así, los sistemas que emplean modelos profundos son potencialmente más competentes y pueden degradarse más hábilmente que aquellos que se fían del razonamiento superficial.

DISEÑO Y SÍNTESIS. Se refiere a la configuración de un sistema sobre la base de un conjunto de posibilidades alternativas. El sistema experto incorpora restricciones que

el sistema debe satisfacer y guía los pasos que el sistema debe dar para satisfacer los objetivos del usuario.

ASISTENTE INTELIGENTE. Aquí el énfasis está en tener un sistema que dependiendo de las necesidades del usuario pueda aconsejar, suministrar información, o desempeñar varias subtareas.

PREDICCIÓN. Se refiere al pronosticamiento de lo que sucederá en el futuro sobre la base de la información actual. Este pronóstico puede depender sólo de la experiencia o puede implicar el uso de modelos y fórmulas. Los sistemas más dinámicos pueden usar la simulación para apoyar el pronóstico.

PROGRAMACIÓN DE ACTIVIDADES. Se refiere al ordenamiento en el tiempo de un conjunto dado de tareas, de modo que puedan hacerse con los recursos disponibles y sin interferir con otras.

PLANIFICACIÓN. Es la selección de una serie de acciones a partir de un grupo complejo de alternativas para satisfacer las metas del usuario. Es más complejo que la programación de actividades puesto que las tareas ya no son dadas sino escogidas. En muchos casos, las limitaciones de tiempo y recursos no permiten que todas las metas se satisfagan. En estos casos se busca el resultado más deseable.

SUPERVISAMIENTO. Se refiere a la observación del progreso predicho o destinado de una situación corriente, y a dar la alerta al usuario o al sistema si hay una desviación de lo esperado o usual. Aplicaciones típicas de esto son los vuelos espaciales, procesos industriales, condiciones de pacientes, y acciones enemigas.

CONTROL. El control es una combinación del supervisamiento de un sistema y de la toma de acciones apropiadas en respuesta a la vigilancia del alcance de metas. En muchos casos tales como la operación de vehículos o máquinas, el retardo tolerable de la respuesta puede ser tan pequeño como de milisegundos. En tal caso se puede aludir al sistema como un sistema de tiempo real ("tiempo real" se define como "respondiendo dentro de un retardo permisible de tiempo", a fin de que el sistema controlado permanezca dentro de sus límites de operación) [20].

RECOPILACIÓN DE INFORMACIÓN. Un sistema que desempeña esta función puede recibir información y retornar una nueva síntesis u organización. Una aplicación puede ser la determinación inductiva de un árbol de decisión a partir de ejemplos. Otras pueden ser la evaluación de situaciones militares o del mercado de valores sobre la base de la entrada de datos e información de consecuencia natural.

DESCUBRIMIENTO. Es similar a la recopilación de información, excepto que el énfasis está en hallar nuevas relaciones, orden, o conceptos. Esta es todavía un área de investigación. Los ejemplos incluyen hallar nuevos conceptos matemáticos y leyes elementales de la física.

OTRAS. Hay otras funciones tales como aprendizaje que son directamente sub-clasificables bajo las que se han enumerado hasta aquí. En muchos casos, estas funciones (y algunas de las que ya se han mencionado) pueden ser ingeniosamente descompuestas en funciones discutidas previamente. Así por ejemplo, el diseño y algunas otras funciones pueden

separarse a menudo en subtareas que pueden resolverse por clasificación [14], [30].

• VERY ○ SOMEWHAT LITTLE	DIAGNOSIS AND CLASSIFICATION	DATA ANALYSIS AND INTERPRETATION	DESIGN AND SYNTHESIS	PREDICTION AND SIMULATION	MONITORING	USE ADVISOR	INTELLIGENT ASSIST.	PLANNING AND SCHEDULING	CONTROL
INFERENCE APPROACH									
BC	•	○	○		○	○	○	○	○
FC & FORWARD REASONING	○	•	•	•	•	•	○	•	•
BC, FC, & FORWARD REASON.	•	•	•	○	○	○	•	•	○
HYPOTHETICAL REASONING	•	•	•	○	○	○	○	•	
OBJECT-ORIENTED	○	○	○	•	○	○	○	○	•
BLACKBOARD	•	•	•	○	○	○	○	•	
INDUCTION	•	○	○			○	○		○
OBJECT DESCRIPTION									
FRAMES	•	○	•	○	○	○	○	•	○
FRAMES W/ INHERITANCE	•	○	•	○	○	○	○	•	○
OBJECTS	○	○	○	•	○	○	○	○	•
PARAMETER VALUES PAIRS	○	○	○		○	○			○
LOGIC	○	○	○	○	○	○	○	○	○
RULES	○	○	○	○	○	○	○	○	○
CERTAINTIES	•	•	○	○	○	○	○	○	○
ACTIONS									
RULES	•	•	•	•	•	•	•	•	•
EXAMPLES	•	○	○	○		○	○		○
LOGIC	•	•	•	•	○	○	○	○	○
MESSAGES	○	○	•	•	○	○	○	•	•
PROCEDURES	○	○	•	•	•	○	○	•	•

Tabla I. En esta tabla, se hace un intento de relacionar la importancia de varios atributos que se han hallado en diferentes ESBTs con relación a sus aptitudes para la construcción de sistemas expertos que desempeñen diferentes funciones.

Finalmente, la figura 1.11, sumariza algunos de los aspectos que determinan la "utilidad global de una herramienta particular". Además de los costos y aplicabilidad funcional, se debe eligir la herramienta de acuerdo al tamaño del sistema que se va a construir. Entre los factores que más contribuyen al grado de satisfacción, tanto del diseñador como del usuario final, están la naturaleza de la

interacción o visualización (por ejemplo, si tiene lugar por medio de un menú o gráficos), cuán fácilmente se recupera de errores, la ayuda en-línea que provee, la estética percibida, y la transparencia del sistema. También es muy importante cuán fácilmente se aprende el sistema.

En muchos casos será necesario intentar construir las mismas aplicaciones con cada herramienta para poder diferenciarlas mejor entre si [14], [29].

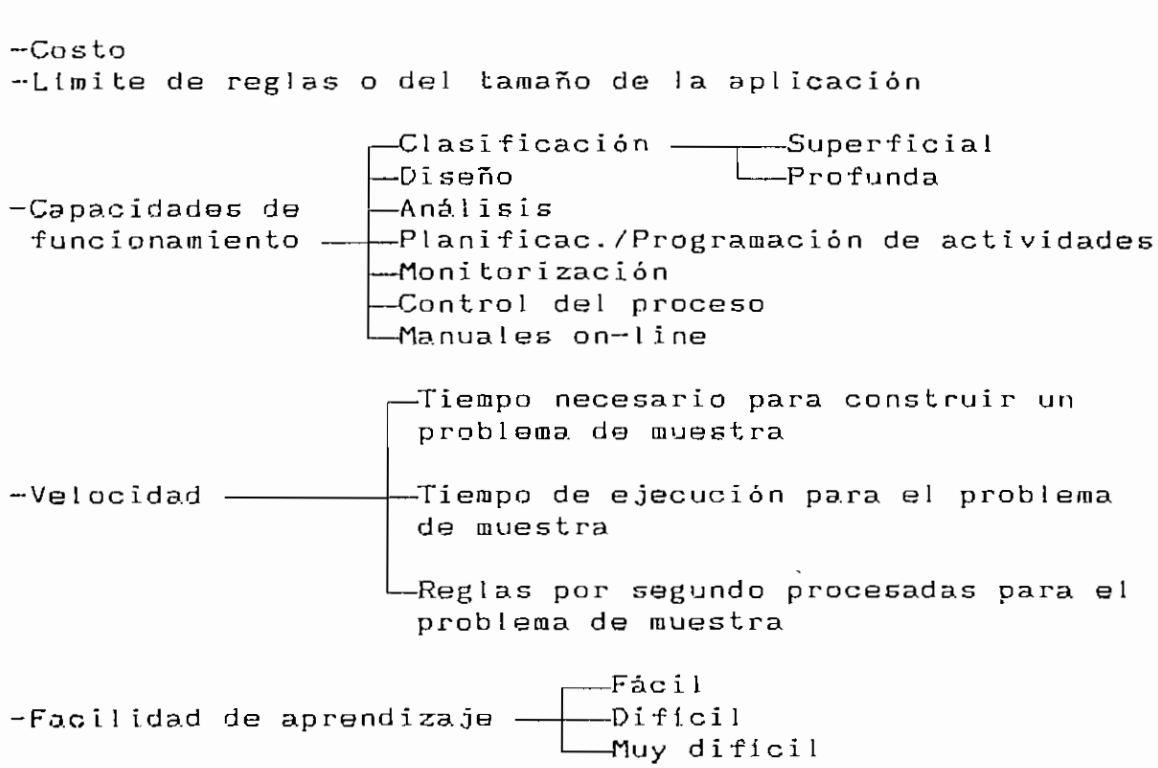


Figura 1.11. Consideraciones para evaluar la utilidad global de una herramienta.

DESCRIPCION DEL SISTEMA A DIAGNOSTICAR

2.1 DESCRIPCION GENERAL

El sistema de potencia eléctrica del aeroplano consiste de un sistema de generación de corriente alterna y otro de corriente continua, con sus correspondientes sistemas de distribución (figuras 2.1 y 2.2) [7].

Tres generadores AC de 90 KVA cada uno, sin escobillas, refrigerados por aire, y accionados por el motor proveen de corriente alterna trifásica, de 400 Hertz, 115/120 voltios, para la operación normal del avión. La velocidad de cada generador de motor se mantiene constante por una transmisión de velocidad constante. Un cuarto generador, idéntico a los tres generadores AC de los motores, está instalado en la unidad de potencia auxiliar (APU) para servir como una fuente de potencia auxiliar en tierra, y para suplir potencia suplementaria durante el vuelo.

Además se suministra potencia de emergencia por medio de un sistema de bateria/ inversor estático, o por un sistema de potencia de generador/ transformador-rectificador movido por aire (ADG).

El suministro de corriente continua se lo obtiene de cuatro transformadores-rectificadores no regulados de 75 amperios, y por una fuente de baterías de níquel-cadmio.

Existen dos receptáculos externos de potencia localizados en el panel de potencia de tierra, en la sección central delantera inferior del fuselaje. Uno es para conectar una

fuente de potencia externa al sistema principal de distribución de potencia eléctrica, y otro es para conectar dicha fuente al sistema de potencia eléctrica de las cocinas.

Los clavijeros del centro de potencia eléctrica y equipo, localizados en el compartimiento central adjunto, proveen sitios para distribución de la potencia eléctrica.

2.2 UNIDAD DEL GENERADOR Y TRANSMISION DE VELOCIDAD CONSTANTE

La velocidad de cada generador se mantiene constante en 8,000 rpm durante todas las variaciones de velocidad del motor, por medio de una transmisión coaxial asociada de velocidad constante (CSD), equipada con engranajes diferenciales. Esta transmisión consiste de un sistema hidráulico con un depósito integral de aceite, una unidad de potencia, el gobernador, y una desconexión eléctrica. Además está localizada en la parte delantera de la caja de engranajes adjunta, la misma que cuenta con una manija de interconexión que conecta la transmisión CSD al generador AC que está instalado en la parte posterior de dicha caja. Un sistema de sujeción-desprendimiento rápido (QAD) asegura la transmisión a la caja de engranajes. La transmisión se puede remover por separado, como una unidad individual, sin remover el generador AC. El acceso es a través de la entrada del capuchón del ventilador derecho.

La frecuencia del generador se mantiene normalmente dentro de límites estrechos, en 400 Hertz, por el gobernador de velocidad de la transmisión CSD, que corrige su velocidad de salida. Aunque la velocidad del eje de entrada a la

transmisión puede variar, la velocidad del eje salida permanece constante, y la frecuencia de salida del generador se controla en 400 (± 4) Hertz.

El CSD es capaz ya sea de sumar o restar de la velocidad que recibe de la caja de engranajes del motor para mantener esta frecuencia en el generador.

La rotación de salida de la transmisión CSD puede detenerse bajo procedimientos específicos, al presionar momentáneamente el switch CSD PRESS TO DISC correspondiente, localizado en el panel del ingeniero de vuelo (figura 2.4). Después de ser desacoplado el CSD, únicamente se lo puede volver a acoplar mientras el aeroplano está en tierra, y con el motor respectivo apagado y detenido. El acoplamiento se lleva a cabo al tirar manualmente del mango de reimplantación del CSD, hasta que la espiga del solenoide dé un chasquido dentro de su posición.

Cada uno de los tres sistemas de generador consisten esencialmente de una transmisión CSD con una desconexión eléctrica, un intercambiador de calor del CSD, un controlador de carga y controles e indicaciones asociadas.

La potencia de entrada para cada generador AC se transmite desde el motor, y se controla a través de los efectos combinados del sistema de engranajes diferenciales, dos unidades hidráulicas variables, y dos unidades hidráulicas fijas.

El circuito de indicación de temperatura del CSD tiene dos escalas: OUTLET y RISE. La primera va desde 20 °C hasta 180 °C, mientras que la segunda va de 0 °C a 40 °C. En condiciones normales, el indicador muestra la temperatura de

salida (OUTLET). Presionando el pulsante CSD TEMP (PUSH FOR RISE), ubicado en el panel eléctrico del técnico de vuelo (figuras 2.3 y 2.4), el indicador suministra información relativa al salto de temperatura.

Cuando la presión del aceite del sistema hidráulico del CSD desciende debajo de 120 PSI, se ilumina en el panel del técnico de vuelo la indicación ámbar CSD OIL PRESS LO.

2.3 GENERACION AC

El sistema de generación AC funciona de tal manera que cada generador se conecta a través de un relevador de generador (GR) a su bus respectivo. A su vez, cada bus de generador se conecta al bus de enlace AC, a través de su relevador de enlace de bus respectivo (BTR) (figuras 2.2 y 2.3). Los generadores se pueden operar como sistemas separados sin conexión en paralelo, cada uno suministrando potencia a sus correspondientes buses de generador y con un generador conectado al bus de enlace AC. O con todos los generadores en paralelo. O aislados, sin ningún generador en el bus de enlace AC.

Sin embargo, se prefiere la operación en paralelo, durante la cual todos los generadores comparten la totalidad de las cargas eléctricas. La conexión automática en paralelo para los generadores de motor principales, se efectúa al presionar momentáneamente el switch PARALLEL GENS, localizado en el panel del ingeniero de vuelo (figura 2.4); siempre y cuando los tres switches AC BUS TIE estén en la posición NORM y la salida de cada generador sea de la calidad apropiada. Para lograr el paralelismo de un

generador puede ser necesario una ligera estrangulación de avance/retardo.

Cualquier operación que abre un relevador de enlace de bus automáticamente desarmará el paralelismo. Además, no se logrará poner en paralelo los generadores si se está suministrando potencia externa al bus de enlace AC, si cualquier circuito de protección indica que uno o más BTR's han saltado, si cualquier switch para aislar el enlace del bus AC está en la posición ISOL, si cualquier relevador de potencia auxiliar (APR) está cerrado, o si se ha seleccionado el modo de tierra dual (aun si no se logra la configuración de tierra dual del sistema de generación).

Si ocurre la pérdida de un generador, su bus respectivo se abastecerá automáticamente desde el bus de enlace AC, se trate de operación paralela o no.

El orden preferencial para suministrar potencia al bus de enlace AC es 2-1-3; y el primer generador operativo en ese orden siempre empieza con el generador No.2 que se conecta al bus de enlace AC.

Cuando los switches de enlace de bus AC se colocan en la posición ISOL, los generadores y el circuito preferencial son inoperativos y los relevadores de enlace de bus no se pueden cerrar automáticamente. Mas bien un relevador de enlace de bus se cerrará únicamente cuando el switch correspondiente de enlace de bus AC esté en la posición NORM, en armonía con la operación normal del sistema.

En el sistema eléctrico del aeroplano se encuentran dos buses de emergencia AC y dos DC. Sin embargo, el control de emergencia y la operación del aeroplano se aseguran

solamente con un bus AC y un bus DC energizados, si ambos buses están asociados con el mismo canal, 1 o 3 (ver figura 2.2). Normalmente, estos buses de emergencia se asocian con los canales 1 y 3 de generador, y suministran potencia a los sistemas eléctricos que se consideran necesarios para volar y efectuar un aterrizaje satisfactorio. En el caso de que ninguna potencia eléctrica esté disponible desde los generadores principales o desde el generador de la APU; al situar el switch EMER PWR localizado en el panel de arriba en el compartimiento de vuelo en la posición ON, se conectará el sistema de baterías al bus DC izquierdo de emergencia y al inversor estático que suministrará 115 VAC al bus AC izquierdo de emergencia.

Por otro lado, la selección del modo de tierra dual es posible ya sea con los generadores AC operando con paralelismo o sin él. Cuando el selector de modo LAND, localizado en el panel de control de guía de vuelo se acciona momentáneamente, y la palanca que emplea el segundo piloto automático situada en el panel de control de inclinación y guía de vuelo se coloca en la posición CMD, se enciende la luz DUAL LAND SELECTED localizada en el panel del ingeniero de vuelo. Cuando el bus 1 y el bus 3 AC están siendo abastecidos por cualesquier dos generadores separados, excepto una combinación del generador 2 y la APU, se enciende la luz DUAL LAND PWR ON localizada en el panel del ingeniero de vuelo y se inicia el modo de operación de tierra dual. Durante el modo de tierra dual, si se pierde ya sea la fuente de potencia AC 1 o 3 (200 milisegundos o más) por cualesquier razón, los generadores permanecerán

aislados, con un "NO GO" para Tierra Dual (Tierra Simple únicamente).

Asimismo, el generador impulsado por aire (ADG) provee una fuente de potencia de emergencia en vuelo si hay una pérdida de larga duración del sistema eléctrico principal, suministrando 115 VAC ya sea a una bomba hidráulica auxiliar o al bus AC derecho de emergencia. El switch selector de carga ADG de dos posiciones, localizado en el panel de arriba del compartimiento de vuelo, se pone en la posición normal HYD cuando se despliega el ADG y su potencia de salida se aplica a una bomba hidráulica auxiliar. La potencia de salida del ADG también se la puede aplicar a los buses AC y DC derechos de emergencia (a través del transformador-rectificador 3) colocando el switch selector de carga en la posición R EMER ELEC.

Durante la operación del sistema de generación AC, existen dispositivos automáticos de protección que aíslan las fallas del sistema. Así, el sistema eléctrico puede controlarse y supervisarse a través de sus controles e indicaciones asociados localizados en el compartimiento de vuelo.

Con todo, las baterías del aeroplano se usan para suplir potencia al bus DC izquierdo de emergencia y al inversor estático, el cual, a su vez suministrará potencia al bus AC izquierdo de emergencia (figura 2.2).

Finalmente, la potencia eléctrica a los buses de las cocinas se la puede suministrar a través de los buses AC de los generadores del aeroplano, a partir de cualquier generador o generadores de motor principales, o también desde el generador de la APU, o durante el mantenimiento en tierra

por el uso combinado de potencia exterior principal y potencia exterior de cocina.

2.3.1 GENERACION, PROTECCION Y CONTROL DE AC

En esta sección se describirán con mayor detalle los componentes del sistema de generación, protección y control de AC (figuras 2.2 y 2.3).

2.3.1.1 GENERADOR AC DE MOTOR

El generador de AC consiste de un generador de magneto permanente (PMG), un generador excitador, y el generador AC principal contenido en un alojamiento simple de dos piezas. El campo principal del generador, la armadura de excitación, los rectificadores rotativos, y los magnetos permanentes rotativos están montados sobre el eje del generador. La armadura del estator del generador, el campo de excitación, y la armadura del estator del PMG están montadas en el alojamiento del generador.

El PMG es un generador trifásico, en estrella sin tierra. El rotor del PMG es un montaje rotativo de magneto permanente de ocho polos, y el devanado del PMG está en el estator del mismo. La salida del PMG es no regulada y varía según lo hagan la carga o la velocidad. La salida del PMG es la potencia fuente para la excitación del generador, y una de las fuentes para el control y operación de la unidad de control del generador (GCU).

El campo excitador provee de potencia al campo rotativo del generador AC, y tiene dos devanados en paralelo, con un termistor en serie con uno de ellos para suministrar una resistencia casi constante de campo a través del rango de temperatura de operación. La armadura de excitación

trifásica está en estrella sin tierra y conectada a seis diodos de potencia de silicio, ubicados sobre el ventilador del generador, los cuales forman un puente rectificador trifásico de onda completa. La potencia DC se conecta desde el puente rectificador al campo rotativo del generador AC, el cual induce potencia trifásica en el devanado de salida del generador principal.

Los devanados trifásicos del estator del generador AC están conectados en estrella, con el extremo de cada fase conectado a un terminal del bloque terminal de seis lugares, situado externamente sobre el generador. Los terminales T1 y T4 corresponden a la fase A, T2 y T5 a la fase B, T3 y T6 a la fase C. Los terminales de neutro T4, T5 y T6 pasan a través de transformadores individuales de corriente ubicados dentro del generador, y después son conectados al bloque terminal. Estos transformadores son el lado neutral del lazo de protección de corriente diferencial. Un enchufe simple situado sobre el marco del generador se conecta al PMG, al campo excitador, a los transformadores de corriente, y a la parte correspondiente del indicador de falla. Los terminales T4, T5, y T6 y el marco del generador se conectan externamente por una barra de bus.

Los cojinetes principales del generador tienen un depósito separado de aceite para su lubricación; además existe un tapón exterior con una varilla de inmersión integral y un tapón de drenaje para cada depósito. Si algún cojinete principal llegara a fallar, en el generador se cuentan con dos cojinetes auxiliares de apoyo que permiten cinco horas de operación. Además en cada uno de estos cojinetes se tiene

un detector que recoge una señal cuando falla su cojinete principal asociado, haciendo que se encienda una indicación luminosa.

2.3.1.2 UNIDAD DE POTENCIA AUXILIAR (APU)

El generador APU es idéntico a los generadores AC de motor, con la excepción de que está instalado en la APU y es accionado por un tren de engranajes adjunto. La refrigeración de aire para el generador APU la suministra su ventilador a través de un sistema de canales.

A su vez, el generador de la unidad de potencia auxiliar (APU) provee una fuente de potencia eléctrica auxiliar. Además, cuando se hace disponible la potencia eléctrica auxiliar y los switches de selección de bus de generador APU están en la posición ON, el generador APU se quitará del bus automáticamente y después, el generador de motor se conectará al bus. En el caso que ocurra el paro de un motor, su bus de generador se transferirá automáticamente al generador APU al detenerse su generador respectivo.

Por eso es que el generador de la APU se lo puede conectar a uno, dos, o a los tres buses de generador, o al bus AC del servicio de tierra únicamente.

Además existen circuitos entrelazados que impiden el paralelismo desde el generador de la APU con los generadores principales o con la potencia exterior; o el paralelismo desde los generadores principales con el generador de la APU o con la potencia exterior.

2.3.1.3 RELEVADOR DE GENERADOR, RELEVADOR DE ENLACE DE BUS AC, Y RELEVADOR DE POTENCIA AUXILIAR

Los tres relevadores de generador (GR), los tres relevadores de enlace de bus AC (BTR), y los tres relevadores de potencia auxiliar (APR) son unidades idénticas que están localizadas en el centro de potencia eléctrica en el compartimiento central adjunto (figuras 2.2 y 2.3). Cada motor puede ser conectado a, o removido de su bus de generador respectivo, o del bus de enlace AC por un relevador de generador asociado y un relevador de enlace de bus AC respectivamente. El generador APU puede ser conectado a cualquiera, o a todos tres, de los buses de generador por un relevador de potencia auxiliar asociado.

Cada relevador de potencia es operado eléctricamente, magnéticamente asegurado, y clasificado como de 115/200 VAC, 400 Hz, con una corriente continua de 350 Amperios. Además de un switch de bobina y seis switches auxiliares que tienen contactos normalmente abiertos y normalmente cerrados conectados al enchufe del relevador, este posee tres grupos de contactos principales de potencia, uno por cada fase. La potencia AC de entrada se conecta a los terminales T1, T2, y T3; a su vez la potencia de salida se conecta a los terminales L1, L2, y L3.

El switch de transferencia de bobina del relevador se conecta en serie con las bobinas de cierre y de salto para coordinar estas acciones respectivas, y para remover la potencia de estas bobinas después de cada operación.

La potencia DC de operación normal de la bobina está entre 18 y 28 voltios. Dos diodos zener cátodo a cátodo se

conectan en cada bobina, para limitar el efecto de la inductancia (pico de voltaje) sobre el switch de transferencia, cuando este interrumpe el circuito de la bobina.

2.3.1.4 UNIDAD DE CONTROL DEL GENERADOR (GCU)

Para cada uno de los tres canales del sistema principal de potencia se cuenta con una unidad de control del generador (GCU), así como para el canal aislado del sistema de potencia auxiliar. Las cuatro unidades son idénticas y están localizadas en el clavijero de habilitación, en el compartimiento central adjunto (figura 2.1).

Cada GCU combina las funciones de regulación de voltaje del generador y, control y protección del sistema. El circuito regulador de voltaje es completamente estático y emplea el sensamiento del voltaje promedio con un elevado control inicial de fase. Los circuitos de sensamiento, de retardo de tiempo, lógicos, y las etapas de salida son también circuitos estáticos. A través de un control V/ADJ se ajusta el voltaje de salida del generador asociado. Girando el control en el sentido de las manecillas del reloj se incrementa el voltaje, a la inversa decrece el voltaje.

Situados en el panel frontal de la GCU están cinco indicadores FLRU, el switch de inicialización de FLRU, el switch y la indicación luminosa de comprobación, cuatro fusibles de indicaciones luminosas con dos de repuesto, y un tornillo de ajuste de voltaje. Un receptáculo de inserción dual sobre el panel posterior de la GCU conecta el cableado eléctrico del aeroplano a la GCU.

El equipo de prueba intraconstruido de la GCU (BITE) cumple dos funciones. Primero, se comprueban ciertos circuitos de protección pasiva por medio de indicaciones luminosas. Segundo, se analizan las fallas para determinar la unidad reemplazable (más probable) de la línea que ha fallado (FLRU).

POTENCIA DE CONTROL DC PARA GCU's PRINCIPALES

La potencia de control DC para cada canal GCU principal se subdivide en: Potencia para el regulador de voltaje y, potencia para control y operación DC.

En cuanto a la potencia para el regulador de voltaje, su fuente de potencia DC es el PMG únicamente. El PMG se conecta al primario del transformador de la GCU que tiene dos devanados secundarios. Uno de estos se conecta al rectificador y filtro trifásico de onda completa que suministra potencia DC al regulador de voltaje, cuando los contactos del relevador de campo GCR están cerrados.

Respecto a la potencia para control y operación DC, se suministra de fuentes diferentes. Una fuente es el devanado secundario del transformador del PMG de la GCU. Este voltaje AC pasa por un rectificador trifásico de onda completa hacia los seis alimentadores de potencia DC (DC1 hasta DC6). Las líneas de potencia DC de la GCU están aisladas por diodos y protegidas por fusibles, de suerte que la pérdida de un alimentador debido a una falla no causará la pérdida de otros alimentadores. La segunda fuente de potencia DC es la red de interconexión de potencia de control de la GCU "Confluencia de potencia DC". Esta red se abastece de las tres GCU's principales, de la GCU de la APU y de los buses

de potencia de control DC de la BCU. Una tercera fuente de potencia (GCU APU únicamente) se la obtiene a partir del bus de baterías del aeroplano. Se suministra voltaje DC a la GCU desde el bus de baterías a través de un interruptor (CB). Antes de la conexión de la fuente del bus de batería a los seis alimentadores de potencia DC en la GCU se encuentran un filtro y un diodo. Para las GCU's principales, uno de los seis alimentadores de potencia GCU se usa también para suplir potencia de red a la GCU y, otro alimentador de potencia suple potencia de la GCU a la red. La GCU APU y la BCU suplen potencia únicamente a la red y no reciben potencia de ella. Cada entrada y salida está aislada de la red por un diodo y un interruptor (CB) de modo que la pérdida de una alimentación, debido a una falla, no causará la pérdida de las otras alimentaciones.

Esta red asegura que una fuente de 28 VDC esté disponible a todo tiempo para control de la GCU y funciones de protección, sin importar de qué canal se esté suministrando potencia DC (incluyendo la APU y canales de potencia externa).

Para desactivar la potencia GCU principal con propósitos de mantenimiento, es necesario asegurarse que su canal asociado esté desenergizado, que el generador asociado no esté girando, y que el CB de la red "IN" de la GCU asociada esté abierto. Estas condiciones se deberían cumplir cuando se remueven o se instalan GCU's principales.

Para desactivar la potencia GCU de la APU con propósitos de mantenimiento, es necesario asegurarse que el generador APU

no esté girando y que el CB del bus de batería a la GCU de la APU esté abierto.

En la red de potencia de control de la BCU GCU se incluye un panel de protección de aislamiento del sistema de potencia (PSIP), localizado en el compartimiento central adjunto. Usando diodos, zeners, y fusibles, el panel PSIP asegura el voltaje AC a tierra hasta que la fuente se abra (por el fusible correspondiente en el panel). Esto protege a la GCU APU y a la BCU del daño que se les pudiera causar por fallas AC o DC (ingreso al bus de batería con voltaje AC) o por una entrada AC a la BCU por vía de las espigas entrelazadas de control de potencia externa. Los fusibles se ubican externamente sobre el panel PSIP para facilidad de reemplazo.

El relevador de control del generador (GCR) es retenido magnéticamente y posee contactos que controlan la aplicación y remoción de la alimentación de potencia del PMG, a través del regulador de voltaje al campo excitador del generador. La potencia de operación del GCR es la potencia de control GCU de 28 VDC.

El cierre manual del GCR se logra colocando el switch de control del generador correspondiente, localizado en el panel superior delantero del ingeniero de vuelo, en la posición RESET momentáneamente. La bobina del GCR se cerrará a condición de que no exista una señal de salto causada por una falla. El GCR entonces se cerrará y permanecerá así a través de todas las condiciones de operación normal del sistema.

El salto manual del GCR se logra con la operación manual de la manija de control de fuego para los generadores principales, y por la operación del switch de control de fuego para el generador APU. El salto automático se logra cuando la bobina de salto del GCR se energiza con las señales de retardo de tiempo de bajo voltaje, de sobrevoltaje, de retardo de tiempo de protección diferencial, y de circuitos de protección del alimentador paralelo abiertos.

Las mecanismos de paro internos previenen el funcionamiento cíclico del GCR. Si una falla hace saltar el relevador, este se cierra cuando el switch de control del generador se coloca en la posición RESET momentáneamente. Si la falla no se elimina, o si ocurre otra falla mientras el switch está en cualquier posición el GCR saltará y permanecerá así.

La función de potencia disponible (PR) recibe señales desde el GCR y circuitos de secuencia de fase, de bajo voltaje, y de baja velocidad durante los procedimientos normales de arranque del sistema. La PR impide el cierre del relevador de generador hasta que las condiciones de operación del generador sean satisfactorias, y el interruptor de control del generador se coloque en la posición ON. Una vez activada la PR, el salto del relevador de generador se controla por las salidas de los circuitos de protección. La PR se desenergiza cuando el GCR salta, o existe una condición de baja velocidad, o el interruptor de control del generador está colocado en la posición OFF.

El relevador del generador (GR) se controla por la operación automática de la GCU. Se tiene la operación automática del

GR durante el arranque del motor, y durante el retardo de baja velocidad normal debido a las funciones de protección. El GR se cierra automáticamente si no ocurre ninguna falla y las salidas de voltaje y frecuencia del generador han alcanzado los límites especificados. El salto automático del GR ocurre cuando el GCR ha saltado, o existe una condición de baja velocidad.

Se provee también un control automático del relevador de enlace de bus (BTR), con anulación manual. Los dispositivos estáticos se usan para suministrar señales de cierre, mientras que para señales de salto se usa un relevador.

Cualquier salto del sistema de protección que también resulta en el salto del GR (diferente de las fallas de bus), vuelve a cerrar automáticamente el circuito de paro temporal del enlace de bus (BTLO) con una señal desde el GR.

Dos dispositivos estáticos de paro (LO) impiden que el sistema se comporte cíclicamente durante las condiciones de falla. Un LO previene el ciclaje del GCR y GR después de que ocurre una señal de salto. La inicialización se lleva a cabo poniendo el interruptor de control del generador en RESET, y después en la posición ON. Las señales de falla subsiguientes producen un salto por cada operación de inicialización.

El voltaje de salida del sistema AC se regula controlando la cantidad de potencia entregada desde el regulador al campo excitador del generador. El regulador de voltaje (VR) sensa el voltaje del sistema, compara el voltaje con un valor de referencia y usa la señal resultante para controlar la excitación del generador. El regulador controla la cantidad

de potencia entregada al campo excitador del generador, mediante variar el tiempo de ON y de OFF de la salida del amplificador de potencia modulada por ancho de pulso, de frecuencia constante.

Un circuito supervisa la frecuencia de una sola fase del generador de voltaje de magneto permanente (PMG). Cuando ocurre una sobrefrecuencia (OF), el relevador de enlace de bus asociado (BTR) salta (pero no se enclava) después de un retardo de tiempo. Los canales buenos volverán a la frecuencia normal. En el canal que ha fallado, la señal de OF hará que salte el relevador de control de generador (GCR) y por consiguiente, el relevador de generador (GR) y desenergizará el canal después de un retardo de tiempo. Cuando el GR del canal falloso salta, su BTR volverá a cerrarse automáticamente, reconectando el bus de carga del canal falloso al bus de enlace.

2.3.1.5 UNIDAD DE CONTROL DE BUS (BCU)

Está localizada en el clavijero de equipo en el compartimiento central adjunto. La BCU suministra protección al bus de enlace AC, protección y control de la potencia externa principal y de la potencia externa de cocina, control de paralelismo automático, lógica preferencial, y la configuración de tierra dual del sistema eléctrico.

Todos los relevadores de la BCU están herméticamente sellados y eléctricamente retenidos. Los componentes en miniatura están localizados en las placas de circuitos impresos que se encajan en una placa maestra. La BCU posee un equipo de prueba intraconstruido (BITE) que comprueba

ciertos circuitos de protección pasiva a través de sus indicaciones luminosas.

El circuito de autoparalelismo (AP) compara las frecuencias del generador y del bus de enlace con miras a su paralelismo. Cuando la desviación de frecuencia está dentro de 0 a 4 Hertz, el circuito de AP originará una señal de paralelismo de modo que la diferencia de ángulos de fase en el instante del paralelismo esté dentro de 90 grados. La señal de AP actúa cerrando cualquier BTR abierto. El generador no puede ser puesto en paralelo si se está suministrando potencia exterior al bus de enlace AC, si cualquier APR está cerrado, si cualquier switch de aislamiento de enlace de bus AC está en la posición ISOL, si se presenta un salto de BTR, o si al sistema se lo ha seleccionado para la configuración de tierra dual.

Situada en el panel frontal de la BCU está una indicación luminosa de comprobación (CC), un switch y tres fusibles de indicaciones luminosas con dos de repuesto. Un receptáculo de inserción dual en el fondo del panel de la BCU conecta el cableado eléctrico del aeroplano a la BCU.

POTENCIA DE CONTROL DC DE LA BCU

Potencia de control DC para la BCU.- La lógica de tierra dual utiliza la potencia del bus directo de bateria que se suministra a la BCU solamente cuando se inicia el modo de tierra dual. Esta potencia se regula dentro de la BCU y se aisla de toda otra potencia de la BCU. El bus de enlace DP (TDP) usa la potencia GCU del canal apropiado para hacer saltar y parar los BTR's. Los circuitos de derivación de carga a la cocina utilizan la potencia GCU asociada para

ordenar un salto, o dirigir la potencia BCU para energizar los relevadores de derivación de carga BCU, y la potencia del bus DC asociado para energizar la bobina de salto del relevador de control de cocina. Los circuitos de control APR 2 y APR 3 se energizan desde la potencia de control GCU de la APU. La potencia de control y operación BCU restante se suministra desde dos fuentes diferentes. Una fuente es el bus de baterías a través de un CB externo y un diodo interno. Una segunda fuente se deriva de la potencia AC trifásica, suministrada a la BCU a partir de la potencia externa, rectificada por un rectificador trifásico de onda completa. Estas dos fuentes suministran cuatro alimentaciones de potencia DC (DC1 hasta DC4) que se aíslan por diodos y se protegen con fusibles, de modo que la pérdida de una alimentación debido a una falla no producirá la pérdida de las otras. Una alimentación DC de potencia suministra una entrada de potencia DC a la red de interconexión de potencia de control GCU.

Para desenergizar la potencia BCU con propósitos de mantenimiento es necesario asegurarse que la fuente principal de potencia externa esté apagada, que todos los motores del aeroplano estén detenidos, y que la APU tampoco esté funcionando. El switch de batería del aeroplano debe estar en la posición OFF. Se debe cumplir con estas condiciones cuando se remueva o se instale una unidad BCU.

Cuando la potencia externa se conecta al aeroplano, la BCU suministra control automático del relevador de potencia externa (EPR). El EPR se cierra automáticamente cuando el

switch principal de potencia externa se coloca en la posición EXT PWR, con tal que la potencia exterior sea de la calidad apropiada y que el bus de enlace AC esté inactivo. Cualquier señal de falla desde los circuitos de protección, hará saltar automáticamente el EPR. El EPR puede hacerse saltar manualmente también poniendo el interruptor principal de potencia externa en la posición OFF.

Cuando la potencia exterior de cocina se conecta al avión, la BCU suministra la derivación de carga automática de cocina en sobrecorriente, y controla el relevador esclavo de potencia externa de cocina, que permite a las cocinas transferirse desde los buses de generador a la potencia externa de cocina, a través de los relevadores de transferencia de cocina.

El relevador esclavo de potencia de cocina se cierra cuando el switch de potencia externa de cocina se coloca en la posición GLY EXT PWR, si la potencia externa de cocina es de la calidad apropiada y se energizan uno o más buses eléctricos principales, ya sea por la potencia eléctrica externa principal o por cualquier generador del avión. Cualquier señal de falla desde los circuitos de protección hará saltar automáticamente el relevador esclavo de potencia de cocina.

El panel de protección de aislamiento del sistema de potencia (PSIP), protege a la BCU del daño que se le podría causar al poner voltaje AC en la entrada de batería, o al poner una entrada AC en la punta de enlace del control de potencia externa.

El sistema eléctrico de potencia se controla primariamente desde la estación del ingeniero de vuelo. A menos que se establezca otra cosa, los controles descritos en los siguientes parágrafos están localizados en el panel delantero superior del ingeniero de vuelo.

2.3.1.6 SWITCH DE CONTROL DEL GENERADOR

Tres switches de control, uno por cada generador de motor se los puede colocar en las posiciones de ON, OFF o RESET. Cuando un switch está en la posición ON la operación del generador es automática. En la posición OFF no se puede conectar el generador a su bus. La posición RESET, de contacto momentáneo, se usa para cerrar el relevador (de campo) de control de generador, después que se ha hecho saltar el relevador por la operación automática de los circuitos de protección, o por la operación manual de la manija de control de fuego.

Particularmente, debe tenerse en cuenta lo siguiente: "El generador se inicializaría normalmente una sola vez por cada falla ocurrida. Pero si la falla salta al generador después de inicializarlo, la falla debe ser localizada y corregida antes de intentar colocar el generador en su bus de nuevo" [7].

2.3.1.7 SWITCH DE PARALELO DE GENERADOR

Para iniciar la operación de autoparalelismo de los generadores se usa un switch de botón pulsante, de contacto momentáneo. Cuando se presiona el switch PARALLEL GENS, se pone en marcha un sistema de autoparalelismo. Tan pronto como se satisfacen las condiciones para poner en paralelo un generador, el sistema de autoparalelismo cierra el relevador

de enlace de bus correspondiente, conectando el bus del generador al bus de enlace AC. No hay un orden definido en el cual los generadores se conecten en paralelo, y puede transcurrir un tiempo considerable entre la operación del switch PARALLEL GENS y la conclusión del paralelismo. Se puede usar los cambios en la velocidad del motor (movimientos de estrangulación) para conectar en paralelo su generador correspondiente, con tal que la frecuencia del generador esté dentro de los límites normales. Cualesquier operación que abra un relevador de enlace de bus desarma el sistema de autoparalelismo. Por lo que el sistema debe volverse a poner en marcha si de nuevo se desea la operación paralela.

2.3.1.8 SWITCHES DE ENLACE DE BUS AC

Se pueden fijar tres switches de enlace de bus AC, uno por cada generador de motor, en las posiciones NORM e ISOL. Normalmente, los switches están en la posición NORM y los generadores se los puede operar en paralelo o no. Colocando un switch en la posición ISOL se abre el relevador de enlace de bus AC. Cuando todos los tres switches están en la posición ISOL, los generadores no pueden operar con paralelismo o sin él, se tiene el circuito preferencial inoperativo, los relevadores de enlace de bus no pueden cerrarse automáticamente, y el sistema puede operar en el modo aislado únicamente. Un relevador de enlace de bus se lo puede cerrar, según la operación normal del sistema, cuando el switch de enlace de bus asociado está colocado en la posición NORM. Ya se mencionó antes en el apartado 2.3 las razones que imposibilitan el paralelismo de los generadores.

2.3.1.9 SWITCH SELECTOR INDICADOR

El switch selector indicador es un switch giratorio de ocho posiciones, sin posición OFF (figura 2.4).

La potencia externa de cocina, la potencia externa principal, el generador APU, o el voltaje y frecuencia del generador accionado por aire (fase A) se muestran en los indicadores AC VOLT y FREQ cuando el switch está en la posición respectiva de GLY EXT PWR, EXT PWR, APU, o ADG. Con el switch en la posición AC GEN/DC BUS 1, 2, o 3, el voltaje y frecuencia AC (fase A) del generador seleccionado se muestra en los indicadores FREQ y AC VOLT, y el voltaje del bus DC asociado se muestra en el indicador de voltaje DC. El voltaje y amperaje DC de la batería se muestran en el indicador DCV AMP; y los del bus AC de emergencia izquierdo se muestran en los indicadores de AC VOLT y FREQ, cuando el switch está en la posición BAT & L EMER AC.

2.3.1.10 SWITCH DE POTENCIA EXTERNA PRINCIPAL

El switch de potencia externa principal se retiene magnéticamente en la posición EXT PWR. La posición EXT PWR coloca la potencia externa en el bus de enlace AC siempre y cuando sea aceptable. La posición OFF remueve la potencia del bus de enlace AC. El switch se vuelve atrás a la posición OFF si se desenergiza la potencia externa, o si ocurre una falla en el suministro de potencia externa.

2.4 GENERACION DC

La potencia para el sistema de generación DC se la obtiene de cuatro unidades no reguladas transformador-rectificador de 75 amperios, las cuales convierten AC a DC para las cargas de 28 voltios DC que se requieren. La entrada es de

potencia trifásica, 400 Hertz, 115/200 VAC y la salida es de potencia de 28 VDC (figura 2.2).

Los transformadores rectificadores normalmente operan aislados, con la palanca de switches de enlace DC asegurada en la posición OPEN. No obstante, se puede lograr que funcionen en común colocando los switches de enlace DC en la posición CLOSE.

Los transformadores-rectificadores (T/R) No. 1 y 2A reciben potencia desde el bus de generador AC 1 y 2 respectivamente. El T/R No. 2B recibe potencia desde el bus de generador AC 2 a través del bus AC del servicio de tierra. El T/R No. 3 recibe potencia desde el bus AC del generador 3, a través del bus AC de emergencia derecho.

Durante la operación normal del sistema, los T/R No. 1, 2A, y 3 abastecen de potencia a los buses DC 1, 2, y 3 respectivamente. Además los T/R No. 1 y 3 suplen potencia a los respectivos buses DC izquierdo y derecho de emergencia. Los T/R No. 2A y 2B dan potencia al bus de batería. El T/R No. 2B también suministra potencia al bus DC de servicio de tierra. Los buses DC pueden ser abastecidos de potencia desde cualquiera de los T/R a través del interruptor de circuito de control remoto (RCCB) de enlace transversal, y los relevadores de enlace DC, los cuales se controlan por la operación manual de los switches de enlace DC.

El sistema a baterías de 28 voltios obtiene potencia de una fuente de baterías de níquel-cadmio. La batería suministra potencia a cargas DC específicas y al inversor estático, el cual asume cargas AC específicas. La batería se carga a través de un cargador de batería de corriente limitada,

cuando la potencia AC está disponible en el bus AC del servicio de tierra y el switch de la batería está en ON.

Se requiere de potencia de control DC, tanto para la GCU como para la BCU; su obtención y utilización se trata en detalle en los apartados respectivos de los ítems 2.3.1.4 y 2.3.1.5.

2.5 POTENCIA EXTERNA

Hay dos receptáculos de potencia externa, uno para el sistema de distribución de potencia eléctrica principal, y otro para el sistema de distribución de potencia a las cocinas (figura 2.2).

El bus de enlace AC se energiza por una fuente de potencia externa suministrada a través del receptáculo de potencia externa principal y del relevador de potencia externa, siempre y cuando se coloque el switch de potencia externa que se encuentra en el panel delantero superior del ingeniero de vuelo, en la posición EXT PWR. Los buses AC y DC del servicio de tierra, se energizan por la fuente de potencia externa suministrada a través del receptáculo de potencia externa principal y del relevador del servicio de tierra, poniendo en ON el switch del servicio de tierra en el panel de potencia de tierra, o en el panel del servicio de tierra de la cabina.

Los buses de las cocinas se energizan por una fuente de potencia externa, a través del receptáculo de potencia externa de cocina y de los relevadores de transferencia respectivos, con tal de que el switch de potencia externa de cocina esté en la posición GLY EXT PWR, y los switches del bus de cocina estén en la posición ON, y que además los

buses principales estén energizados por la potencia externa principal o por cualquier generador. Los switches están localizados en el panel delantero superior del ingeniero de vuelo.

2.6 DISTRIBUCION DE CARGA ELECTRICA

La distribución de carga eléctrica a partir de las fuentes de potencia a las varias cargas del aeroplano, se lleva a cabo por los relevadores de generador, de enlace de bus, de potencia auxiliar, de potencia externa, de transferencia de cocina; y buses de generador, buses AC y DC, bus de batería, bus directo de batería, y buses de emergencia AC y DC. Los componentes del sistema eléctrico de potencia se localizan en el centro de potencia eléctrica, y clavijeros asociados al equipo en el compartimiento central adjunto. Además en clavijeros de equipo y panel de control DC en el compartimiento de aviónica, y sobre el panel del equipo eléctrico de cocina (figuras 2.1, 2.2 y 2.4).

Los controles e indicaciones que se usan para controlar y supervisar el sistema eléctrico de potencia se ubican en paneles, en el compartimiento de vuelo, adelante y a la izquierda en la cabina principal en la cocina inferior, en el compartimiento central adjunto y en el panel de potencia de tierra ubicados en la sección central delantera inferior del fuselaje. En el caso de la potencia externa o de la APU el switch POWER del panel de reaprovisionamiento no está disponible para operaciones de carga de combustible. El switch POWER suplirá potencia de batería automáticamente a los circuitos de control de combustible y al inversor

estático, para una medición estándar de combustible únicamente.

2.7 LIMITACIONES DE POTENCIA ELECTRICA

Los valores mostrados en la columna normal, son rangos de operación normal. Las indicaciones fuera del rango normal indican posibles anormalidades del sistema.

Los límites mostrados en la columna permisible no se deberían exceder, y no se recomienda una operación prolongada en estos límites.

Cualquier condición que produzca cualquier indicación en la columna permisible se debería corregirla a la primera oportunidad.

Se puede dañar el equipo eléctrico del aeroplano si los sistemas eléctricos AC han sido sometidos a voltajes más altos que 130 voltios fase-neutro, o 224 voltios fase-fase. Cualquier sobrevoltaje AC aplicado a los transformadores-rectificadores resultará en un sobrevoltaje DC aplicado al equipo DC. Todos los equipos eléctricos, de navegación, de radio y radar sujetos a sobrevoltajes dañinos deberían ser examinados y comprobados funcionalmente, de acuerdo a las prácticas de mantenimiento aplicables [7].

	Normal Min o Max	Permisible Min Max
Indicador de voltaje AC bus 1, 2, o 3	112 a 118 Volts	108 122 Volts
Indicador de voltaje AC APU	112 a 118 Volts	108 122 Volts
Indicador de voltaje AC (Modo auxiliar)	112 a 118 Volts	108 122 Volts
Indicador de voltaje AC pot. ext. principal	112 a 118 Volts	108 122 Volts

	Normal Min o Max	Permisible Min	Max
Indicador de voltaje AC pot. ext. cocina	112 a 118 Volts	108	122 Volts
Indicador de voltaje AC bus AC de emer. izq.	112 a 118 Volts	108	122 Volts
Indicador de voltaje AC bus AC de emer. izq. config. de emergencia	108 a 122 Volts	108	122 Volts
Indicador de frec. AC bus 1, 2, o 3	396 a 404 Hz (Puede variar ± 2 Hz)	380	420 Hz
Indicador de frec. AC APU	396 a 404 Hz (Puede variar ± 2 Hz)	380	420 Hz
Indicador de frec. AC APU (Modo auxiliar)	396 a 404 Hz (Puede variar ± 2 Hz)	380	420 Hz
Indicador de frec. AC pot. ext. principal	396 a 404 Hz	380	420 Hz
Indicador de frec. AC pot. ext. de cocina	396 a 404 Hz	380	420 Hz
Indicador de frec. AC bus AC izq. de emer.	396 a 404 Hz	380	420 Hz
Indicador de frec. AC bus AC izq. de emer. config. de emergencia	396 a 404 Hz	380	420 Hz
Indicador de carga AC GEN 1, 2, y 3 en no-paralelo	sobre 0 bajo 1.0	0	1.0
Indicador de carga AC	La desviación entre dos medidas cualesquiera no debe exceder ± 0.15 con una carga total de .6 a .9 NOTA: Si el sistema no reúne este requisito, cargue a 1.3 del total y reúna los requisitos de la siguiente línea.		

	Normal Min o Max	Permisible Min	Max
Indicador de carga AC GEN 1, 2, y 3 en paralelo	Cada indicador de carga debe mostrar dentro del ± Ø.1 de carga promedio con carga mínima de 1.3 de la carga total	Ø	1.Ø
Indicador de carga AC	sobre Ø bajo 1.Ø	Ø	1.Ø
Derivación de carga AC sobrecorriente de cocina	1.5 en no menos que 1Ø segundos, y no sobre 5Ø segundos.		
Indicador de voltaje DC bus 1, 2, y 3	24 a 29 Volts	24	29 Volts
Indicador de voltaje DC batería sin funcionamiento del cargador	26 a 29 Volts	22	3Ø Volts
Indicador de voltaje DC Batería con el cargador funcionando	29 a 34 Volts	28	38 Volts
Indicador de carga DC T/R 1, 2A, 2B, y 3 en no-paralelo	sobre Ø bajo 1.Ø	Ø	1.Ø
Indicador de carga DC T/R 1, 2A, 2B, y 3 en paralelo	La diferencia entre las lecturas alta y baja no exceda ± Ø.2	Ø	1.Ø
Indicador de amperaje DC cargador pulsante de batería (la longitud del pulso de carga depende del estado de carga de la batería)	35 a 65 Amperios	35	65 Amps
Indicador de voltaje DC comprobación de potencia de emergencia	26 a 29 Volts	22	3Ø Volts
Indicador de amp DC comprobación de pot. de emergencia	37 a 65 Amps	25	75 Amps
Indicador de voltaje DC potencia de emergencia configuración de emer.	22 a 29 Volts	22	3Ø Volts

	Normal Min o Max	Permisible Min Max
Indicador de amp DC potencia de emergencia configuración de emer.	hasta 100 Amps	hasta 100 Amps
Indicador de voltaje AC generador accionado por aire	112 a 118 Volts	108 122 Volts
Indicador de frec AC generador accionado por aire	370 a 430 Hz	370 430 Hz

NOTA: Inicialmente, los indicadores de frecuencia se calibran desde 380 hasta 420 Hz. Es normal que el indicador se clave en 420 Hz durante la operación del ADG.

Temperatura de salida del aceite de la transmisión CSD	hasta 146 °C	hasta 163 °C
Subida de la temperatura del aceite de la transmisión CSD	hasta 25 °C	hasta 40 °C

NOTA: Los rangos de sobrecarga del generador para operación durante corto tiempo son:

(1) de 1.5 por 5 minutos
(2) sobre 1.5 por 5 segundos (el rango de sobrecarga del generador es de 2.0 por 5 segundos)

NOTA: Los rangos de sobrecarga del transformador rectificador para operación de corto tiempo son:

(1) de 1.5 por 5 minutos
(2) sobre 1.5 por 5 segundos (el rango de sobrecarga del transformador rectificador es 2.0 por 5 segundos).

2.8 GLOSARIO DE ABREVIATURAS

Las siguientes abreviaturas se encuentran tanto en el texto como en las ilustraciones relativas al sistema de potencia eléctrica [7]:

AC	Corriente Alterna
ADG	Generador Accionado por Aire
AGSTR	Relevador de Enlace AC del Servicio de Tierra
APR	Relevador de Potencia Auxiliar
BAT	Batería
BCU	Unidad de Control de Bus

BF/BLO	Falla de Bus/ Suspensión de Bus
BITE	Equipo de Prueba Intraconstruido
BTLO	Suspensión Temporal de Enlace de Bus
BTR	Relevador de Enlace de Bus
CBR	Relevador del Cargador de Batería
CC	Comprobación de Confiabilidad
CCT	Transformador de Corriente de Control
CSD	Transmisión de Velocidad Constante
CT	Transformador de Corriente
DC	Corriente Directa
DP	Protección de Corriente Diferencial
DT1R	Relevador 1 de Enlace DC
DT3R	Relevador 3 de Enlace DC
DXT	Enlace de Cruce DC
EPC	Centro de Potencia Eléctrica
FLRU	Unidad Reemplazable de la Línea que ha Fallado
GCR	Relevador de Control de Generador
GCU	Unidad de Control de Generador
GEN	Generador
GR	Relevador de Generador
GSR	Relevador del Servicio de Tierra
GTR	Relevador de Transferencia de Cocina
ICU	Unidad de Control de Indicador
IND	Indicador
LC	Controlador de Carga
LECR	Relevador de Control Izquierdo de Emergencia
LSR	Relevador de Derivación de Carga
OC	Sobrecorriente
OE	Sobre-excitación
OP	Fase Abierta
OV	Sobrevoltaje
PMG	Generador de Magneto Permanente
QAD	Sujeción-Separación Rápida
RCCB	Interruptores de Circuito de Control Remoto
REIR	Relevador de Aislamiento de Emergencia Derecho
RETR	Relevador de Transferencia de Emergencia Derecho
TR	Transformador Rectificador
UE	Baja Excitación
UF	Baja Frecuencia

UV Bajo Voltaje

VR Regulador de Voltaje

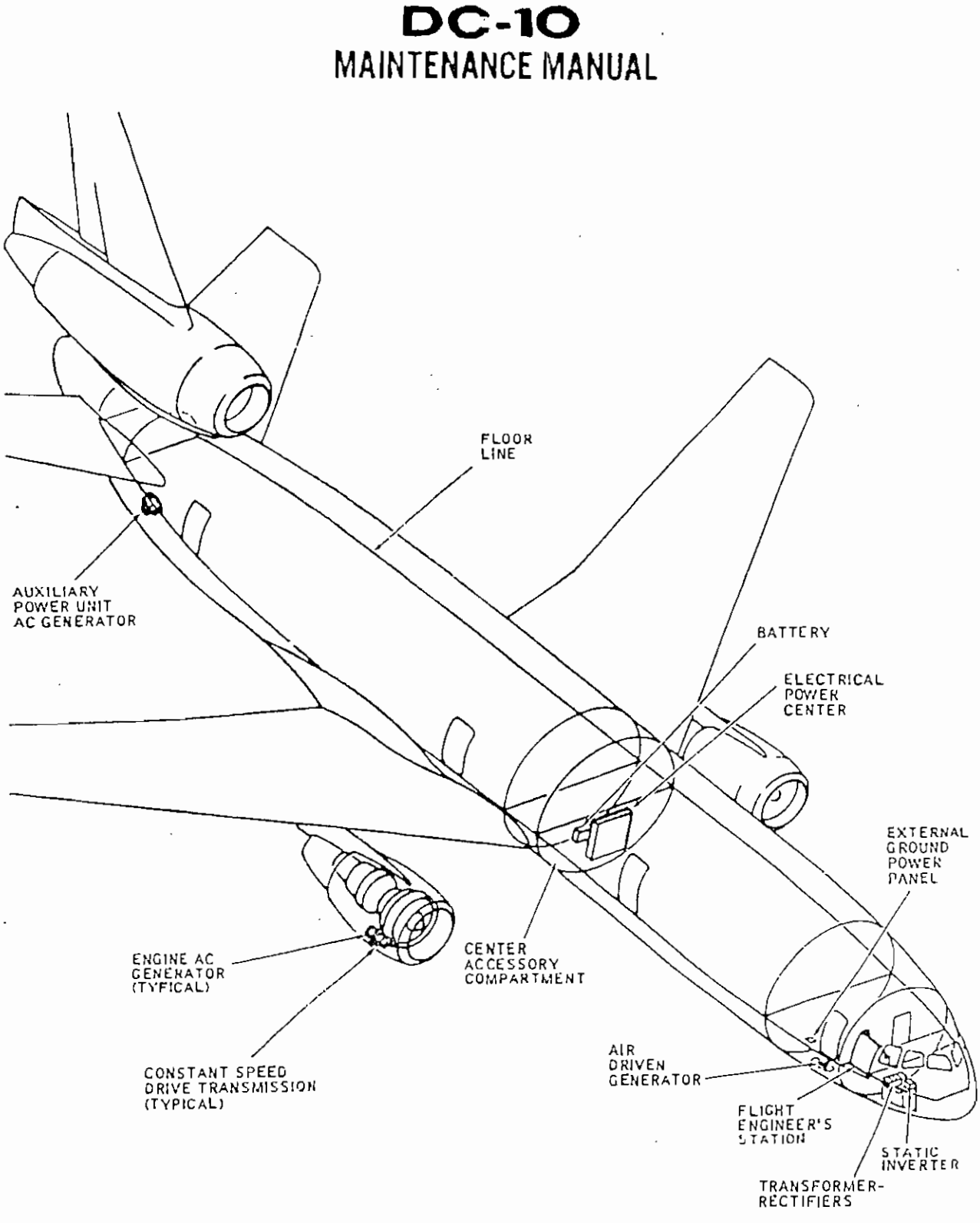


Figura 2.1. Componentes del Sistema Eléctrico de Potencia.

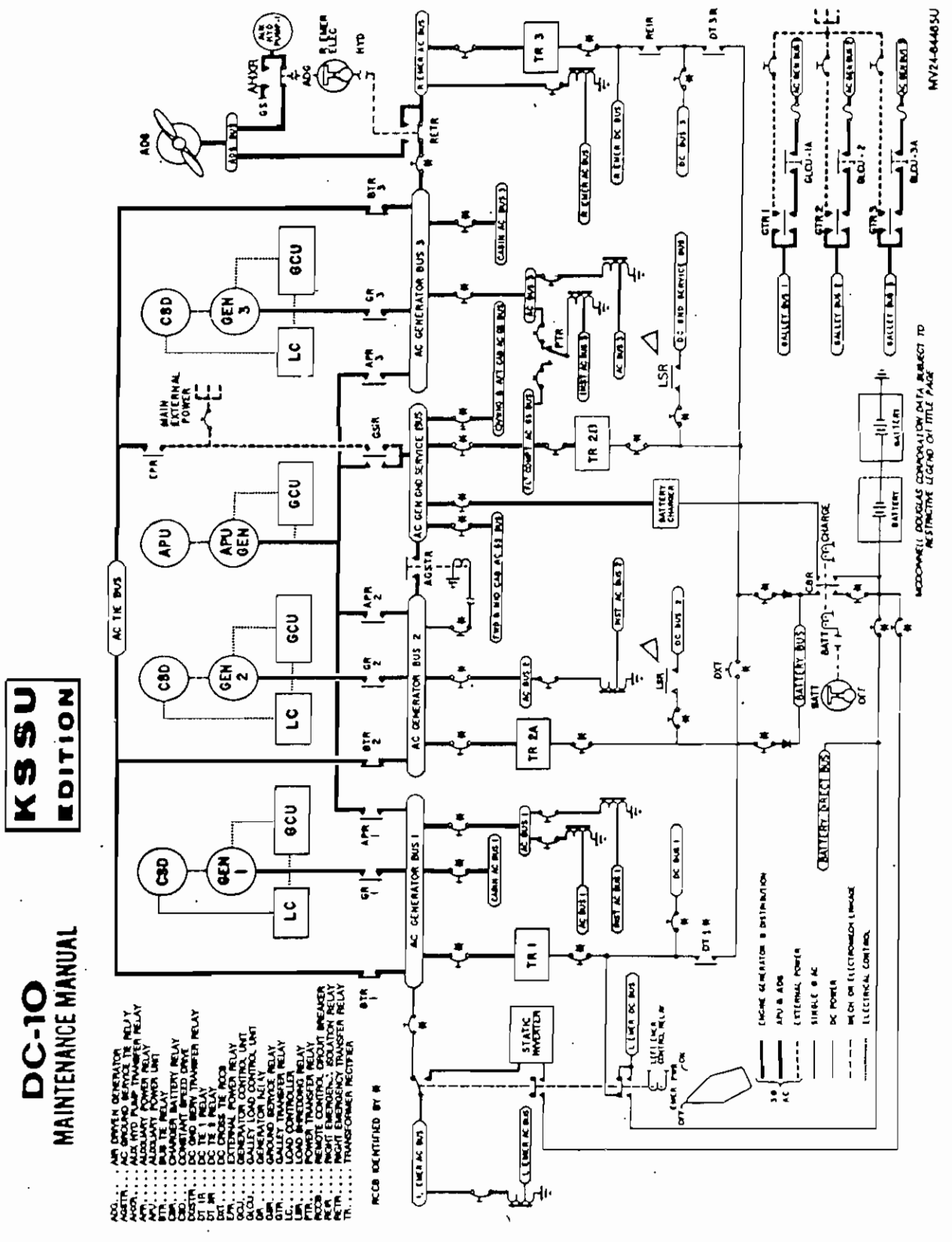


Figura 2.2. Esquema del Sistema Eléctrico de Potencia.

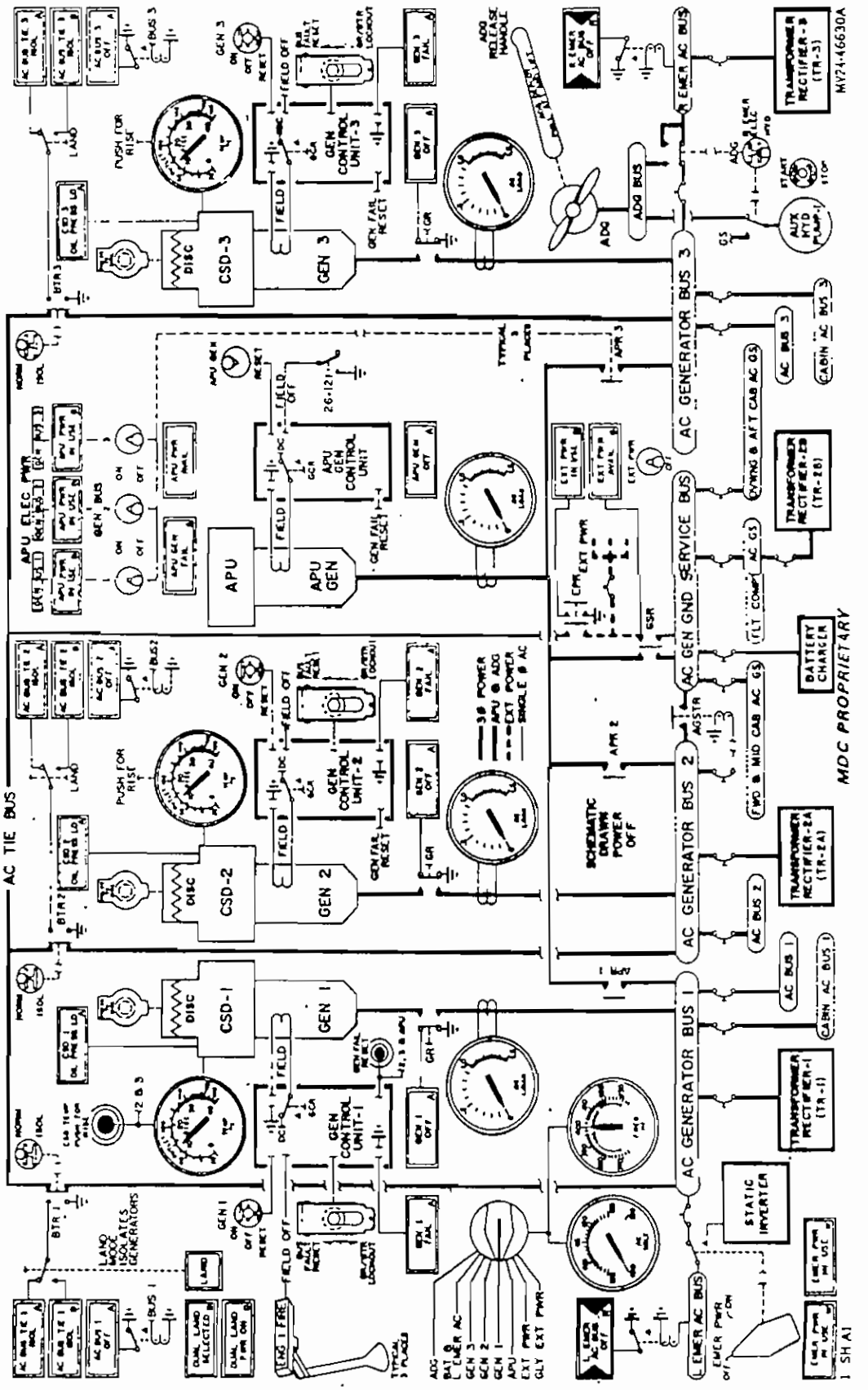


Figura 2.3. Esquema del Sistema de Potencia AC.

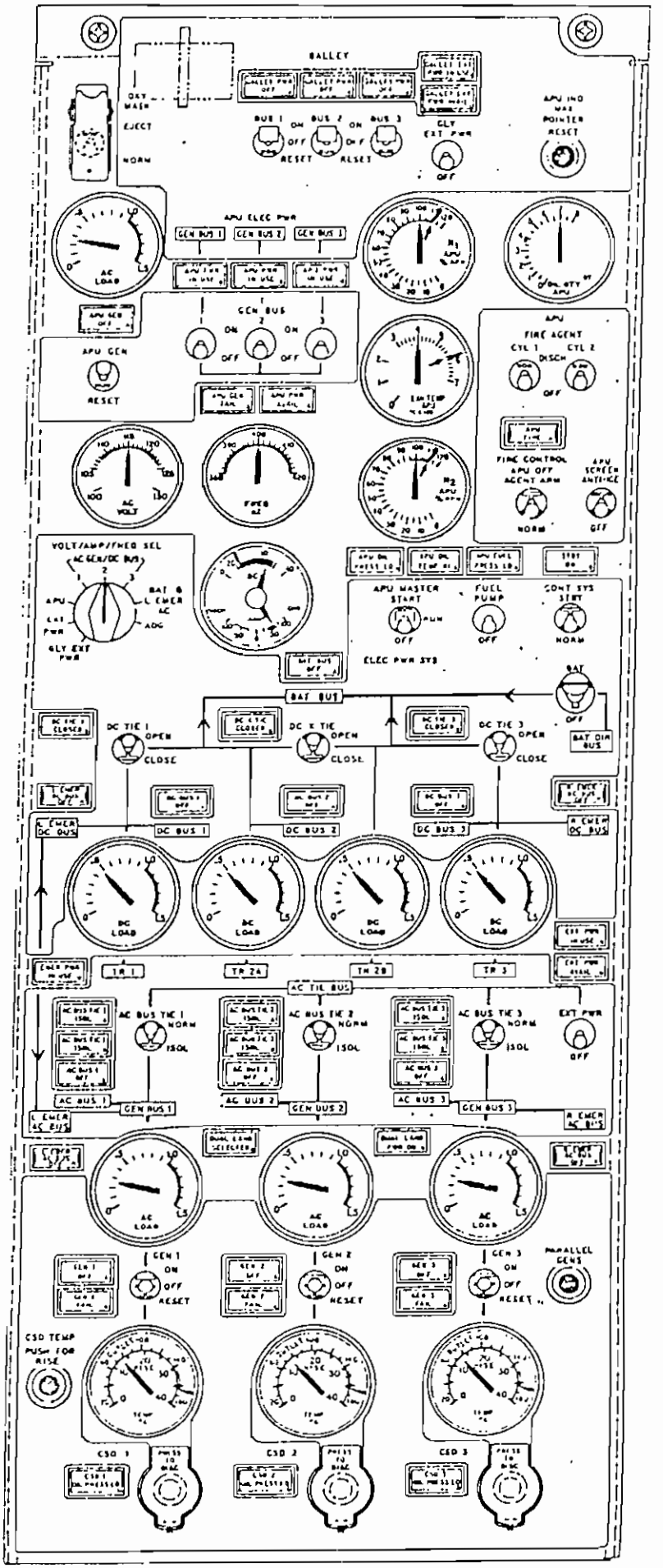


Figura 2.4. Panel Eléctrico del Ingeniero de Vuelo.

DESARROLLO DEL SISTEMA EXPERTO DE DIAGNOSTICO

3.1 INTRODUCCION

Generalmente se empieza a construir un sistema experto con un conjunto de preguntas en la mente [30]:

- ¿Por qué se está construyendo este sistema experto?
- ¿Se justificará el esfuerzo de construir un sistema experto?
- ¿Cuáles serán los réditos de la inversión?
- ¿Cómo se lo va a construir?
- ¿Quién va a ayudar?
- ¿Qué herramientas están disponibles?
- ¿Puede construirse el sistema del modo en que se lo preveé?
- ¿Qué puede salir mal?
- ¿Cómo se puede dar mantenimiento al sistema después que haya sido construido?

Así, la intención de construir un sistema experto para diagnóstico de fallas en un avión, tuvo varios antecedentes: Se hizo una búsqueda extensiva y minuciosa de información sobre el tema y se encontró por ejemplo, el diseño de un sistema de control de vuelo para corregir fallas en tiempo real, propuesto para la automatización de procedimientos de emergencia en aviones (estos procedimientos requieren de una respuesta rápida y de complejas decisiones de la tripulación de vuelo) [16]. Esta idea serviría después para el modelo

final. Además se encontraron otros prototipos relacionados [10], [25].

Desde el comienzo se tuvo la intención de hacer algo que sirviera como una aplicación práctica de la tecnología de sistemas expertos, en una empresa nacional, investigando las posibilidades de las herramientas comerciales disponibles para este fin. Pues se podía contar con uno de estos shells: INTELLIGENCE COMPILER [36].

Este sistema experto sería una valiosa ayuda alternativa -no una sustitución- del manual de mantenimiento del avión para los inspectores de mantenimiento; puesto que también pondría a su alcance información histórica pertinente. Funcionaría en el departamento de mantenimiento y control de calidad de la empresa propietaria de la aeronave -según lo acordado-.

Este avión posee 50 sistemas enumerados en la clasificación ATA 100 del fabricante MacDONNEL DOUGLAS DC-10 [8]. Y puesto que los sistemas expertos de éxito tienen un enfoque limitado de la tarea, y para que el presente trabajo constituyera un proyecto realizable, se decidió que el prototipo se circunscribiría a un solo sistema, el # 24 de POTENCIA ELECTRICA, que consta básicamente de 8 subsistemas. El ATA 24 es vital para todos los restantes, y uno de los más extensos.

Ningún apoyo financiero estaría disponible. Por eso, mas bien el interés por investigar y ver en funcionamiento el sistema resultante, se consideró como una motivación importante para hacer este esfuerzo. Pues a cambio se contaba con la ayuda técnica necesaria de expertos.

A pesar de tratarse de un sistema complejo y de no disponer de herramientas para adquisición de conocimiento, sin embargo se poseía un conocimiento organizado del dominio (árboles de fallas del manual de mantenimiento del avión, y bitácoras de vuelo desde el año 1990).

Con los futuros usuarios de este sistema experto: El personal de control de calidad y los expertos se trabajó durante varios meses, y su ayuda fue esencial para adentrarse en la concepción de lo que resultaría ser el sistema experto resultante.

Conjuntamente con este análisis introductorio se examinó las características del shell INTELLIGENCE COMPILER respecto a sus facilidades para el diseño de pantallas, creación de ventanas y opciones de menú, etc. y se corroboró su idoneidad para las exigencias de este proyecto.

Todo esto es importante para poder construir y ensayar lo más pronto posible los primeros prototipos, ya que sólo así se pueden esclarecer algunos aspectos esenciales tanto del diseño como de la herramienta en cuestión. Como por ejemplo, averiguar si el sistema puede ser hecho del modo en que se lo conceptúa. Una manera de lograr una mejor perspectiva a esta cuestión es pensar en todos los posibles modos de falla y construir un "árbol de falla" hasta donde sea posible.

Inicialmente se pensó que el sistema se lo podía construir del modo en que se lo preveía (aunque más tarde hubo que hacer algunos ajustes en esto, según se lo indicará más adelante).

Ya desde un punto de vista profesional un proyecto de desarrollo de sistema experto puede fallar por diversas razones [30]:

- Subestimar el conocimiento que se requiere y la dificultad en adquirirlo, o si el experto halla difícil expresar cómo está estructurado el conocimiento.
- Durante el proyecto, los miembros clave del equipo de desarrollo se van sin dejar suficiente documentación de su actividad. El sistema nunca se recuperará por completo del caos interno resultante.
- El proyecto tiene insuficiente apoyo financiero.
- El motor de inferencia es demasiado lento cuando el sistema se pone en marcha, produciendo un desempeño pobre e inaceptable en tiempo-real.
- Los usuarios no pueden usar el sistema debido a que no entienden cómo operarlo, ya sea por un pobre diseño de la interfase o por la ausencia de instrucciones apropiadas.
- No hay nadie disponible para reactualizar y dar mantenimiento al sistema a medida que cambia el conocimiento del dominio. Consecuentemente, el sistema no lleva el paso con el conocimiento del dominio.

Por todo esto, al principio del proceso de desarrollo se debería hacer un intento serio de pensar completamente en las etapas de construcción de un sistema experto, y anticipar cualesquier defectos o escollos que se puedan encontrar (un defecto suficientemente serio puede conducir a

que se suspenda el proyecto, mientras que en otros casos puede producir una modificación en las estrategias).

Quizá, lo que podría salir mal en este proyecto sería que el desarrollo del sistema tomara más tiempo que el previsto (más tarde se dio efectivamente esta situación, debido a la familiarización con la complejidad del sistema por un lado, y por otro debido a las dificultades financieras inherentes al proyecto).
Finalmente, después de la construcción del sistema, se pensó en proveer facilidades para su mantenimiento y reactualización (esto se describirá más adelante).

Como en cualquier caso de desarrollo de software, construir un sistema experto requiere de alguna disciplina en los métodos y procesos que se usan. Imponer esta disciplina puede ser difícil como resultado de detalles sociales que interfieren con la construcción de sistemas expertos, y que son diferentes de los detalles involucrados en el desarrollo de software convencional.
Una característica final que puede ser pasada por alto es saber cuándo detenerse. Un sistema experto siempre puede ser refinado, re-escrito en un nuevo shell, y así sucesivamente. Decidir cuándo se detiene el desarrollo y el sistema llega a ser operacional puede ser difícil. Detenerse demasiado pronto puede resultar en un sistema inmaduro, inadecuado para los usuarios. Detenerse demasiado tarde no resulta económico por lo menos; y después de cierto punto, el porcentaje de réditos al esfuerzo se vuelve desfavorable.

Con estos antecendentes se prosiguió adelante con el ciclo formal de desarrollo de un sistema experto [30], que se esquematiza en la siguiente figura 3.1:

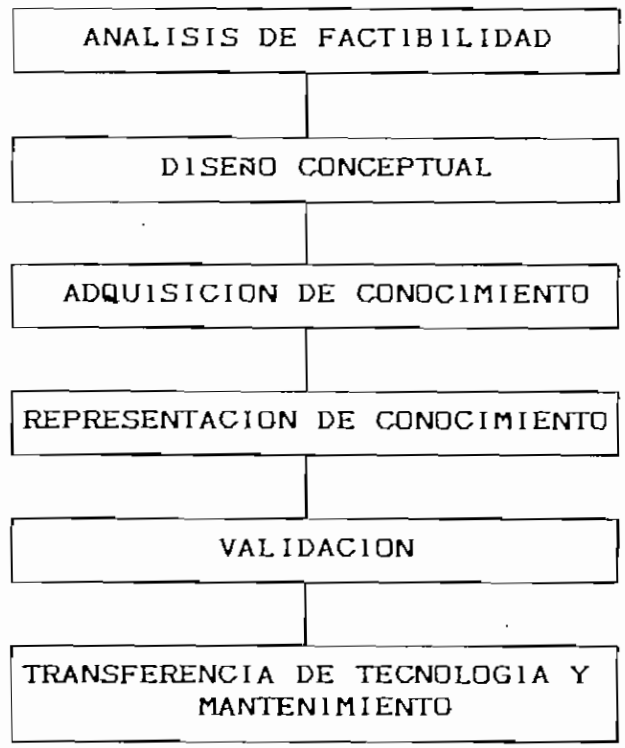


Figura 3.1. El ciclo de desarrollo de un sistema experto.

3.1.1 ANALISIS DE FACTIBILIDAD

Una manera de evitar el fracaso en la construcción de un sistema experto, es seleccionar una tarea manejable. La tarea no debería ser demasiado difícil (al presente, las tareas como los juegos de ajedrez, e innovadores diseños de ingeniería caen dentro de esta categoría), ni deberían depender de grandes cantidades de sentido común (puesto que los sistemas expertos dependen del conocimiento que enfocan). También ayuda mucho si hay un buen modelo de cómo se ejecuta la tarea. En el desarrollo de un sistema experto

médico por ejemplo, es de gran utilidad tener una buena idea 'a priori' de qué tipos de enfermedad existen y cómo pueden clasificarse dentro de una jerarquía.

Cuando el desarrollo del sistema experto es una empresa comercial, surgen factores adicionales que se asocian con el desarrollo de cualquier producto; y como cualquier producto nuevo, un sistema experto debería ser al menos, tan bueno como las alternativas que suplementa o reemplaza. Por ejemplo, construir un sistema experto como una alternativa al método simplex de programación lineal no sería sabio en casos donde se requiere de esta técnica, puesto que los productos existentes ya hacen muy bien este trabajo.

Se pueden resumir los requerimientos para análisis de factibilidad en sistemas expertos, con las siguientes preguntas:

- ¿Carece la tarea de una solución adecuada ya existente de software?

  No tiene caso duplicar algo que ya funciona bien.

- ¿Tiene la tarea un enfoque limitado?

  Es importante darse cuenta que las mejores tareas de sistemas expertos son mas bien de naturaleza limitada y tienen un enfoque limitado. Tratar de desarrollar un sistema experto que intente resolver todos los problemas de la humanidad de una vez resultará en fracaso.

- ¿Son los expertos demostrablemente mejores que los aficionados?

  Por lo general es difícil intentar una tarea que ha resultado igualmente ardua para expertos y novatos.

Esto daría a entender que la tarea es inherentemente difícil de tratar.

- ¿Cambiarán las reglas del juego de la noche a la mañana?

  Es importante saber si es probable que se mantengan estables los criterios para llevar a cabo la tarea. De otro modo será difícil capturar el conocimiento.

- ¿Está un experto humano disponible y encargado?

  Es esencial tener un experto que esté dedicado al proyecto y que suministre el conocimiento pacientemente.

- ¿Está la destreza enseñada o documentada?

  El material de enseñanza o la documentación interna y los preceptos para ejecutar la tarea del sistema experto sirven muchas veces como una invalorable fuente de información.

- ¿Cuáles serán los criterios para el éxito o el fracaso?

  Es importante saber qué constituirá el éxito para el sistema experto. ¿Tiene que ser el sistema experto 100% correcto todo el tiempo, o será aceptable un porcentaje razonable de éxito o de corrección parcial?

  En algunos casos, aun si el sistema tiene éxito el 50 % del tiempo todavía será de gran ayuda. Por ejemplo, en el caso de un sistema que diagnostica exitosamente fallas de computador el 50 % del tiempo, y dice "No sé" en lo restante. Este sistema todavía puede ser un éxito si actúa como un filtrador de problemas que maneja casos de rutina, y que pasa los problemas más difíciles al experto humano.

El dominio de operación de este sistema experto es en el campo de la ingeniería, y la tarea que va a ejecutar es de diagnóstico de fallas en el sistema eléctrico de potencia (ATA 24) de un avión DC-10. La tarea es de apoyo a los expertos, como inspectores de mantenimiento del avión.

Se confió en la factibilidad del proyecto por la existencia de un buen modelo de cómo se ejecuta esta tarea -es decir- conocimiento, información teórica e histórica, diagramas, etc.

A pesar de que no se trataba de un proyecto comercial, se consideró que este sistema sí podría equipararse a sus alternativas. Se comprobó que, aunque para aviones más modernos como los "Airbus" existe ya un sistema computadorizado a bordo que diagnóstica las fallas durante el vuelo, y las registra para su corrección cuando el avión aterriza y pasa al "modo de mantenimiento"; no existe tal cosa para el DC-10, siendo necesaria la intervención del ingeniero de vuelo.

En vista de que se da una gran prioridad a la capacitación constante de los expertos en cursos especializados de entrenamiento fuera del país, es difícil encontrar aficionados ejerciendo en este campo. Y no hay lugar para ellos por la gran responsabilidad que reviste el mantenimiento de un avión comercial de pasajeros, ni tampoco para ensayos ni errores de inexpertos. No obstante sí se cuenta con un equipo de varios expertos, que poseen un excelente registro de confiabilidad, y con reportes continuos provenientes tanto del fabricante como de

asociaciones de aerolíneas, que recogen la experiencia de flotas enteras de aviones DC-10.

Los criterios para desempeñar la tarea estuvieron bien establecidos desde el principio por la naturaleza misma del conocimiento estructurado existente; de modo que tampoco se esperaron cambios súbitos en estos criterios.

La factibilidad del proyecto se vio reforzada por la asignación -de parte de la gerencia de mantenimiento- de un experto que estuvo particularmente dedicado a este proyecto.

Una fuente esencial de ayuda, debido a la complejidad del sistema, fue la documentación de la destreza para cazar fallas en el manual de mantenimiento del avión, con el detalle de los síntomas de falla, árboles de falla y acciones finales para cada uno de los subsistemas y para la mayoría de casos posibles dentro del ATA 24 (estimado por los expertos alrededor de un 90 %).

Finalmente, los detalles que determinan el éxito de este sistema experto son:

- La exactitud con la que refleje la experiencia de las cartas de caza de fallas, desde los primeros síntomas hasta el reemplazo o sugerencia final.
- El sistema de ayudas incorporado que suplementa a todos los procedimientos secuenciales o acciones de mantenimiento en cualquier instante.
- La interfase amistosa al usuario con un sistema práctico de menúes, y ventanas de diálogos fácilmente comprensibles y manejables.
- La información histórica (bitácoras de vuelo), convenientemente procesada y fácilmente accesible en

una base de datos relacional. Las estadísticas son muy importantes para el control de calidad y la evaluación de la confiabilidad del entero sistema de apoyo a la aeronave, es decir, los recursos humanos, técnicos, administrativos, etc.

- A pesar de que en primera instancia este es un sistema de apoyo en tierra, sin embargo la velocidad de ejecución, no deberá ser desanimadoramente lenta (de varios segundos).
- Un sistema para el ingreso de información pertinente de parte de los usuarios; con miras a la reactualización y mantenimiento del sistema experto.

Se consideró que estas metas eran realistas y que se podían alcanzar con este proyecto.

3.1.2 DISEÑO CONCEPTUAL

Una vez que la tarea está bien formulada y es claro que el desarrollo del sistema experto es apropiado, se puede empezar la tarea del diseño conceptual enfocada en una porción restringida pero manejable del dominio del conocimiento. La conceptualización es una exploración por el experto, posiblemente con la asistencia de otros, dentro de los conceptos y relaciones principales que componen el conocimiento del experto para resolver problemas. También se exploran las restricciones generales para la resolución de problemas.

Un entendimiento aproximado de cómo funcionará el sistema experto, obtenido durante el proceso de diseño conceptual, puede ayudar a los expertos a enmarcar su conocimiento de una manera apropiada. Así, el experto desempeña un papel más

importante en la ingeniería de conocimiento. La interacción entre un ingeniero de conocimiento y un experto, puede ser una sociedad y una búsqueda conjunta de conocimiento.

Los temas que se deben abordar en el diseño conceptual se alistan a continuación:

- ¿Cuáles son los mayores objetos, conceptos, o entidades, en el dominio de conocimiento?
- ¿Cómo deberían describirse los objetos, conceptos y entidades?
- ¿Cuáles son las relaciones importantes que existen entre los objetos?
- ¿Cómo están restringidas estas relaciones en el dominio del conocimiento?
- ¿Cuáles son las características generales del problema que se tratará?
- ¿Cómo obtendrá el sistema experto información acerca del problema en curso?
- ¿Cómo interactuará el sistema con el usuario?

Una ventaja final del diseño conceptual es que forza a los expertos a estructurar su conocimiento y lo saca al aire libre. El resultado es una conceptualización de los componentes y una organización de la experiencia. Este proceso de explicación del conocimiento continuará en la fase de adquisición de conocimiento.

En esta fase de diseño conceptual, los árboles de falla con los que se pudo contar y su relación con los controles e indicadores del panel eléctrico en la estación del ingeniero de vuelo fueron de gran ayuda. En vista de que en ellos se

presentan también los conceptos y relaciones principales, y las restricciones generales para el sistema. De aquí se obtuvo una idea general de cómo funcionaría el sistema una vez concluído.

Fundamentalmente, se vió que el conocimiento con el que se contaba se trataba del llamado conocimiento superficial, compilado, o experimental; el cual provee atajos a través del conocimiento del dominio, permitiéndole al experto resolver los problemas más rápida, eficiente y precisamente. Además, los expertos cuentan con información un poco más profunda en el manual detallado de diagramas de todos los sistemas del avión (a este manual se hace referencia en el proceso de razonamiento con conocimiento superficial, por ejemplo, cuando se ha llegado al último paso del árbol de falla sin resolver el problema).

El sistema ATA 24, se subdivide en 8 subsistemas, los cuales en sí cuentan con más subsistemas, unos dentro de otros. Sin embargo, a partir de cierto nivel de profundidad, ni siquiera los expertos en el mantenimiento del avión tienen acceso a ellos, sino que tienen que remitirlos al fabricante especializado de tales subsistemas. Como por ejemplo: La Unidad de Control del Generador (GCU), el Controlador de Carga (LC), los Generadores, la Transmisión de Velocidad Constante (CSD), etc. Así sucede que, cuando no remiten los síntomas de falla después de un cuidadoso procedimiento secuencial, se reemplazan sistemas enteros, como los supracitados. Por lo tanto, los conceptos básicos que se manejan en este caso cuentan con muchos dispositivos, tanto por el enorme grado de complejidad como por el ahorro de

tiempo, dinero y sobre todo la confiabilidad (en réditos de seguridad para el avión y los usuarios) que representa hacerlo así. Esta es una simplificación importante, que hace factible la realización de este prototipo experimental de sistema experto.

Básicamente entonces, se manejan los siguientes conceptos fundamentales: 1) síntomas de falla, basados en una diversidad de switches, breakers (incluyendo los de control remoto), indicadores analógicos y luminosos, 2) árboles de falla y 3) reemplazos o acciones finales concernientes a dispositivos enteros (ayudados por conocimiento profundo). Por supuesto, también existe el conocimiento de sentido común, bajo ciertos lineamientos y sugerencias.

Estos objetos y conceptos se describen en base a ventanas de diálogos manejadas por reglas inversas. Los diálogos a su vez componen los árboles de falla que se codifican en una base de hechos.

Las relacion básica e importante entre todos estos objetos es su interdependencia como partes de un todo global que identifica, aísla y corrige la falla. Con una reorganización del conocimiento se podría examinar el ATA 24 bajo el paradigma de marcos, con atributos de temperaturas, voltajes, corrientes, frecuencias, indicaciones luminosas, analógicas, etc. Puesto que eso y nada más es lo que muchos objetos pueden tener en común, pues se los trata como 'cajas negras' que generan síntomas de falla en sí mismas, o en otros sistemas inter-relacionados o dependientes. Así, se ve que las relaciones entre objetos en el presente esquema de

diseño, siguen el mismo patrón que la naturaleza del conocimiento existente en el manual de mantenimiento: Experimental, compilado o superficial [7], [30].

Los casos de diagnóstico de fallas se enfocan por un lado, en tiempo real por la tripulación de vuelo, para su posterior corrección en tierra. Y por otro lado, durante el mantenimiento que los expertos llevan a cabo en tierra. Ambos enfoques se complementan entre sí para un tratamiento más confiable del problema. Generalmente, ambos enfoques se centran en torno al conocimiento superficial.

El sistema experto obtiene -del usuario- la información para suministrar guía en un problema dado, ya que él es quien guía al sistema en primer término (en lugar de que el sistema lo haga con él, es decir se define el nivel de intrusividad). Con esta información, el sistema buscará en su base de conocimiento las respuestas que el usuario puede considerar, exhibiendo para esto ventanas de diálogos que involucran la posibilidad de hacer selecciones.

En el panel eléctrico del ingeniero de vuelo están arreglados y nominados los diversos controles e indicadores en sitios exclusivos, para cada uno de los 8 susbsistemas del ATA 24 (figura 2.4). Esto facilita la observación y el diagnóstico de fallas. Otro detalle tiene que ver con los acrónimos de sistemas específicos que se usan extensamente en la codificación del conocimiento GCU, LC, CSD, GEN, CB, RCCB, etc.

Respecto a la interacción del sistema con el usuario se ha adoptado 2 niveles de intrusividad [35]. Básicamente, existen 5 maneras en que el humano y la computadora pueden

interactuar en términos de intrusividad: Dos niveles extremos (los más intrusivos) son el nivel en el que el humano dirige (es decir le asigna tareas al computador. Por ejemplo, el computador responderá a las peticiones del operador para desempeñar una subtarea o responder una pregunta), y el nivel en el que la computadora dirige (esta cuestiona al humano por información o para desempeñar algunas tareas, y el humano no tiene otra elección que seguir estas peticiones).

En el medio, los niveles en los que el humano sugiere (puede imponer restricciones al procesamiento de la computadora), y la computadora sugiere (puede suministrar alguna información o advertencia al humano, el cual es libre de atenderla o no dependiendo de su evaluación de la situación), le permiten a un solucionador de problemas -el humano o el computador- ser moderadamente intrusivos entre sí. Finalmente hay un nivel independiente en el cual ninguno de los dos influye en el otro.

Los sistemas expertos típicos usan solamente los dos niveles extremos de interacción. Algunos sistemas usan solamente uno, otros los dos.

En el presente caso se tiene primariamente el nivel extremo dirigido por el humano y uno de los niveles intermedios, en el que la computadora sugiere y el humano decide (en parte porque no se trabaja en tiempo real). Con respecto a esto es oportuno señalar que el diagnóstico de fallas aún sigue siendo uno de los mejores ejemplos de tareas en las que el humano es mucho más efectivo que la computadora. Por lo tanto, mejorar o asistir el desempeño del humano en el

diagnóstico de fallas sigue siendo la mejor manera de mejorar el rendimiento en este campo. Con este valioso antecedente se escogieron los 2 niveles de intrusividad mencionados.

Así, la interacción del sistema con el usuario sería a través de pantallas, menús, diálogos, y ayudas on-line.

Otra técnica muy importante utilizada aquí, tiene que ver con las bases de datos relacionales [30], [36].

Una base de datos relacional es un conjunto de registros, cada uno de los cuales se ajusta a un formato común. Es decir que cada registro contiene el mismo número, tipo, y orden de campos. Cada campo representa un detalle de información particular, tal como el nombre de un empleado, su número de licencia de conductor, salario, etc.

Los marcos también son conjuntos de registros que se ajustan a convenciones de formatos. Sin embargo, los marcos difieren de las bases de datos relacionales en que los archivos de los marcos pueden contener diferentes tipos de registros en el mismo archivo.

Las bases de datos relacionales son útiles para almacenar grandes números de registros (conjuntos de información). Cada detalle de información de un registro es independiente y está relacionado con su similar de otro registro por un índice común.

Las bases de hechos se asemejan a las bases de datos relacionales en que cada registro es independiente de cualquier otro, es decir no hay herencia entre registros. Por eso es que las bases de hechos pueden verse como bases

de datos relacionales que contienen registros solamente con un detalle de información.

Así, se encuentra una relación y una justificación entre todas las técnicas y procedimientos conceptualizados para este sistema de diagnóstico: reglas, lógica, bases de hechos, ventanas de diálogos, base de datos relacional y marcos.

Con respecto al sistema para ingreso de información, se lo haría en el formato usual de memorándums, e implicaría el manejo de archivos con las facilidades de programación lógica del shell INTELLIGENCE COMPILER, para cada uno de los 8 subsistemas.

Además cada uno de estos se trataría como un sistema experto independiente en sí mismo -es decir- tendría su propia base de conocimiento, motor de inferencia e interfase al usuario. No están integrados para interacturar entre sí debido a la cantidad de información y a sus requerimientos de memoria. Para hacer esto -que sería deseable- se necesitaría una tarea mucho más grande, compleja y exigente, que involucraría más tiempo y recursos.

A continuación se alistan los componentes independientes del sistema total:

- Sistema de Transmisión de Velocidad Constante (CSD)
- Sistema de Generación AC
- Sistema de Generación DC
- Sistema Izquierdo de Potencia de Emergencia
- Sistema Derecho de Potencia de Emergencia
- Sistema de Potencia Externa
- Sistema de Potencia de las Cocinas

- Sistema de Distribución de Carga Eléctrica
- Base de Datos Relacional
- Ingreso de Información (Memorándums)

Con referencia a la figura 1.7; para cada uno de los componentes del sistema de diagnóstico, se resume su estructura así:

- Interfase al usuario: Menús y ventanas de diálogos e hipertexto, con ayudas on-line.
- Base de conocimiento: Hechos, lógica, reglas, y marcos (limitados).
- Motor de inferencia: Encadenamiento hacia atrás, lógica, y marcos integrados entre sí. Emparejamiento de muestras de variables, literales y cadenas de caracteres.

3.1.3 ADQUISICION DE CONOCIMIENTO

Generalmente se acepta que esta tarea es la más difícil en la construcción de sistemas expertos. La utilidad global de un sistema experto depende de la calidad de su conocimiento, y esta a su vez depende del éxito en adquirirlo.

Diferentes escuelas de pensamiento proponen varias estrategias o metodologías para la adquisición y representación de conocimiento.

El conocimiento puede ser sorprendentemente difícil de reconocer cuando se lo tiene que extraer desde un fondo ruidoso de sugerencias, y aun de acontecimientos

relacionados. La adquisición de conocimiento puede asemejarse a un proceso de destilamiento, donde se debe aislar los hechos y reglas esenciales, de información que contiene muchas impurezas. En vez de estar almacenado en varios manuales, el conocimiento en muchos dominios está disponible solamente a los expertos, y puede no haber sido escrito nunca en una forma estructurada. En tales casos, la tarea de construir un sistema experto debe empezar por adquirir o capturar el conocimiento experto.

En el presente caso, la tarea de adquisición de conocimiento sí consumió una buena parte del tiempo. Sin embargo se facilitó mucho por la documentación existente para el diagnóstico de fallas en los manuales del avión, tanto de conocimiento profundo como superficial, y en la información histórica (bitácoras de vuelo).

Toda esta información se hizo fácilmente accesible. No obstante se requería saber leerla, interpretarla y seleccionarla.

Quizás desarrollando este proyecto entre dos personas se podría haber organizado una labor más extensa de ingeniería de conocimiento. Con todo, se dio un uso cabal al conocimiento del manual de mantenimiento y a las bitácoras de vuelo, dentro del alcance de trabajo de esta tesis de grado (que provee también un punto de partida para futuros trabajos).

Al consultar a los expertos de mantenimiento en cuanto al porcentaje de validez que asignan al conocimiento existente en los manuales (para el diagnóstico de fallas), ellos asignaron valores que flucutaban desde un 75 a un 90% de

acierto y credibilidad. Por lo tanto, el sistema experto basado en esta fuente de información, complementado con el sistema de ayudas que contiene referencias e información de conocimiento más profundo, y con las bitácoras de vuelo; no estaría desautorizado, sino que sería útil, según la opinión de los expertos. Así, este sistema sería un asistente de consulta experta.

3.1.4 REPRESENTACION DE CONOCIMIENTO

Según se ha mostrado en el Capítulo I, el conocimiento puede expresarse en términos de estructuras de conocimiento tales como hechos, reglas y marcos. Una vez que se adquiere el conocimiento, se usa un sistema de representación para codificarlo (dentro de la base de conocimiento) en una forma utilizable por el sistema experto.

Un rasgo de la representación práctica del conocimiento es la necesidad de subdividirlo en contextos a medida que la cantidad de conocimiento se incrementa. Saber cómo organizar y dividir el conocimiento durante su proceso de representación puede ser un desafío en dominios donde el conocimiento es complejo. Y sin embargo allí es donde se presentan diferentes tipos de problemas, que se tratan generalmente en diferentes contextos de conocimiento.

Las estructuras que se usaron para representar el conocimiento fueron reglas de producción, exclusivamente de encadenamiento hacia atrás -mucho más flexibles que las reglas hacia adelante-, junto con una base de hechos que contiene codificados los árboles de falla completos (es

decir, las secuencias de procedimientos con todas sus ramificaciones, alternativas, observaciones, etc.) mediante diálogos guiadores e informativos que involucran selecciones para el usuario.

También se ha hecho uso de marcos en los sistemas más grandes (generación AC, DC, distribución, y base de datos relacional), pero escasamente. Donde la eficiencia de la programación simbólica así lo requería. Por ejemplo, para un gran ahorro de memoria en la codificación y en el tiempo de ejecución (se obtuvieron buenos resultados al definir ciertas variables globales en las ranuras de los marcos). Ya se mencionó antes la posibilidad de re-estructurar el conocimiento en base a características comunes a los sistemas: Voltajes, corrientes, temperaturas, frecuencias, indicaciones luminosas, etc. Dando así mayor expansión y posibilidades a la programación orientada a objetos.

La organización del conocimiento mantiene las mismas subdivisiones de contexto que en los manuales y en el panel eléctrico del ingeniero de vuelo.

Un rasgo interesante de la representación de conocimiento vinculada con reglas, es el uso de diálogos con posibilidad de ofrecer instrucciones y selecciones al usuario [36], además de un sistema de ayudas en línea. Esta facilidad de la interfase al usuario se ha usado extensamente en la base de conocimiento.

Otra excelente característica para interacción con el usuario es la técnica de hipertexto [36], que puede vincularse directamente con reglas (y por ende con el conocimiento), o con ventanas de diálogos (para la

interfase). En el presente caso se ha relacionado el hipertexto tanto con reglas inversas como con diálogos, con excelentes resultados, por su gran utilidad y simplificación de diseño. En particular, en la etapa inicial de presentación (lo más detallada y completa posible) de los síntomas de falla, previa a los procedimientos secuenciales de los árboles de falla. Por eso en respuesta a una selección dentro de un menú, aparecerá hipertexto en la mayoría de los casos. Así se superaron los problemas de abreviación excesiva o de fragmentación indeseable en la descripción detallada de los síntomas de falla, en detrimento de su presentación y comprensión (esto ocurrió con la lógica directa de predicado, menús, y diálogos con selecciones que se los ensayó y rechazó). Así se pudo incluir además los códigos de falla del manual de mantenimiento; lo cual constituye una referencia útil para el usuario.

La siguiente etapa de la inferencia, el diagnóstico de fallas en sí, se manejó de otra manera.

En primer lugar, se encontró el modo de manejar una variedad de información -sea texto explicativo, opciones, teclas de función, etc.- en cada paso de la inferencia diseñando ventanas de diálogos que contengan todas estas posibilidades (según se necesiten). Así se estructuró el proceso en pasos secuenciales completos.

El diseño del modelo de ventana de diálogo que se adoptó es el siguiente: Cada diálogo contiene un título en la barra de vídeo reverso que indica el nivel correspondiente del árbol de fallas. A continuación viene el texto explicativo (ítems

de texto) del procedimiento que debe llevarse a cabo. Después se encuentra un ítem de selecciones, dentro del cual el usuario puede elegir una opción situándose con las teclas de flechas hacia arriba o hacia abajo y presionando F10.

De inmediato aparece la siguiente ventana de diálogo que contiene el próximo paso del procedimiento secuencial de diagnóstico, de acuerdo a la selección previa hecha por el usuario, y así se continúa hasta que se alcanza el último paso de la inferencia -el reemplazo, o sugerencia final-.

La última categoría de ítems que se utilizan, son precisamente las teclas de funciones (F3 a F12); de las que se utiliza sólo F10 por simplificación de diseño. En la misma línea se encuentra a la derecha la tecla de escape (ESC) para retroceder hasta cualquier nodo anterior del árbol de fallas. Al retroceder se cierran los diálogos. Y se abren a medida que se avanza en el diagnóstico.

Cada diálogo puede contener las selecciones que se deseen, en un sólo ítem o en varios. En el presente caso, se utilizan 2 opciones en la mayoría de diálogos, en vista de que no se requiere más.

Todos los árboles de falla se han estructurado en base a este modelo de diálogos. Cada diálogo y cada ítem reciben un nombre al momento de crearlos. Estos nombres, junto con todas las opciones que contengan el (los) ítem(s) de selecciones, son la información que interviene activamente en la inferencia y en la representación de los árboles de falla en la base de hechos.

Esta lógica de razonamiento utilizando reglas inversas (que se puede ver en las hojas de programas), toma en cuenta la

importancia fundamental de cerrar los diálogos a medida que se retrocede con la tecla ESC (de lo contrario, pronto se agotaría la memoria RAM).

En resumen, las reglas, la lógica de predicado y la lógica booleana son esencialmente la estructura que representa y maneja el conocimiento a través del motor de inferencia. Así, los árboles de falla manejan diálogos y selecciones codificados en lógica de predicado como hechos. En especial, se ha buscado que la lógica de predicado (también definida por el diseñador) sea concisa, rápida y fácil de codificar.

En el caso de la base de datos relacional, se ha buscado estructurar una representación que contenga toda la información técnica contenida en la página de una bitácora de vuelo. Es decir: fecha, número de página, número de serie (con relación a los dispositivos removidos y reemplazados), ítems de página y extensiones de ítem, descripción del problema, y acción correctiva. Adicionalmente, se ha incluído una opción para agregar información formalizada: La categoría del sistema al cual corresponde el problema -pues así se podrá obtener una presentación de todos los problemas que correspondan a cualquiera de los 8 subsistemas-.

La interfase al usuario de la base de datos es similar a la de los sistemas de diagnóstico (excepto por el hipertexto, que no hace falta aquí). El ménú contiene 4 opciones: Mirar, agregar, modificar y suprimir registros. En su programación se ha utilizado íntegramente reglas inversas.

Para la definición del esquema bajo el cual se almacenarán los registros, se ha preferido el estilo más conciso de la

lógica de predicados relacionales, más bien que la representación al estilo de marcos relacionales (los cuales tienen una mejor aplicación en bases que contienen más campos en sus registros). Se han definido varios índices (6), para todos aquellos campos que se pueden utilizar como referencia en la búsqueda de registros: Fecha, número de bitácora, ítem, números de serie -removidos y reemplazados-, y categoría del sistema al que corresponde el problema.

El último ejecutable, correspondiente al ingreso de información nueva por el usuario, también utiliza en su programación reglas inversas más los predicados que sirven para la manipulación de archivos. Su interfase al usuario en presentación de menú contiene 8 opciones, para cada uno de los subsistemas del ATA 24, que además cuentan con un archivo editable con extensión .MNT, en el que se almacena la información acumulativamente en forma de memorándums.
Cuando se escoge en el menú un subsistema, se abre su archivo correspondiente, se obtiene el memo del usuario y se lo agrega al final del archivo, después se cierra el archivo. Mientras esto sucede, se manejan otros dos archivos comunes a cualquier subsistema (MEMO y RESPONSE), exclusivamente con propósitos de exhibición al usuario de su memo ingresado.
Tanto la base de datos relacional, como el sistema de ingreso de memos, tienen un sistema de ayuda en línea accesible a todo tiempo con F1 (lo mismo que cada uno de los sistemas de diagnóstico).

3.1.5 PRUEBA Y VALIDACION

Para los sistemas expertos, una gran parte de la validación tiene que ver con el conocimiento contenido en la base de conocimiento (que puede ser muy difícil de validar). La evaluación y validación de sistemas expertos requiere un amplio rango de criterios y métodos.

Durante la fase de validación se verifican los requerimientos para el sistema experto y los criterios de éxito o fracaso. Los métodos de validación deberían aplicarse a la validez de las conclusiones, a las razones para llegar a esas conclusiones, y a los métodos básicos de razonamiento en sí. La inferencia misma es un método para establecer argumentos válidos, y puede usarse para determinar si hay reglas inconsistentes en la base de conocimiento. Las explicaciones del shell se pueden usar para verificar si la utilización del conocimiento tiene sentido. Las historias del caso también se pueden usar para validar el conocimiento, al evaluar la salida del sistema experto, en contraste con los resultados efectivos de casos previos.

Una manera de averiguar si un sistema experto desarrollado funciona en realidad y cuán bien lo hace sería consultar a un experto, pero contar solamente con la opinión del experto, no suministra respuestas científicas a las preguntas que se refieren a la calidad de la base de conocimiento.

Intuitivamente, el objetivo de hacer el esfuerzo de validación es asegurar que el sistema experto se comporte "correctamente". Aunque la validación incrementa la

confianza en el comportamiento futuro del sistema, este aún pudiera cometer errores en situaciones inesperadas -claro- los expertos humanos también cometen errores y se los tolera, excepto cuando llevan a una catástrofe. En un sistema no trivial de software es muy difícil alcanzar una certeza total en su correción. Con todo las preguntas clave para tolerar los errores en un sistema experto son: "¿Cuál es el precio que se debe pagar por cada error?" y "¿Con cuánta frecuencia se cometen los errores?".

Mas bien que aspirar a verificar el comportamiento de un sistema experto, se debería tratar de medir su desempeño.

La validación puede hacerse ya sea juzgando el desempeño final del sistema experto como una caja negra, o enfocando la calidad de la experiencia misma, es decir, mirando dentro de la caja negra.

Aparte de medir el desempeño de un sistema experto resultante, el análisis lógico y empírico de la experiencia dentro de la base de conocimiento puede indicar inconsistencias y errores. En algunos casos este planteamiento de validación puede ser preferible, porque se puede usar para probar secciones de la base de conocimiento, aun antes de construirse un prototipo completo del sistema experto.

Sin embargo en muchos casos el comportamiento de entrada-salida determinará el éxito o fracaso de un sistema experto, y no se puede pasar por alto la validación de la caja negra. La validación de la caja negra es también, a menudo, más objetiva que otros métodos, puesto que resulta menos afectada por los sentimientos, prejuicios, etc.

La validación de sistemas expertos es un problema crucial para aplicaciones en las cuales se tiene que "acertar la primera vez". En tales ocasiones, sería inapropiado depurar el funcionamiento del sistema experto, debido a que no se pueden tolerar las fallas, como en sistemas que tratan con la defensa civil, reacción a terremotos, caza de fallas en una planta nuclear, y ayudas en la cabina del piloto. Por ello se ha visto la necesidad de métodos de experimentación que puedan construir escenarios para probar el desempeño del sistema experto, y la validez del conocimiento experto incorporado, antes de su operación.

Algunas de las técnicas para verificar y validar modelos de simulación, son útiles también en probar y simular la operación de un sistema experto. Estas incluyen, el uso de información histórica para probar el sistema experto, evaluación de la validez de la fachada del sistema por expertos, y el uso extensivo del sistema experto dentro de un entorno simulado.

Verificando la completez y consistencia de la base de conocimiento se pueden analizar conflictos, redundancia, suposiciones, reglas ausentes, reglas circulares, conclusiones inalcanzables, y metas con callejones sin salida.

Estos tipos de planteamiento de validación son una manera poderosa de comprobar la estructura lógica de los hechos, y las reglas en la base de conocimiento; pero no enfocan directamente el problema de si las reglas individuales mismas son correctas. Se debería recordar que la validez lógica no es lo mismo que la verdad, puesto que un argumento

lógicamente válido puede estar basado en suposiciones equivocadas.

Se puede evaluar también la validez global de un sistema experto, es decir, si su apariencia y desempeño son aceptables para aquellos que lo usarán. Al efectuar este juicio de aceptabilidad, se debe tener el cuidado de explicar factores subjetivos que se relacionan más con la apariencia que con el desempeño del sistema. Algunos criterios de validez global incluyen lo siguiente:

Validez de la Fachada. ¿Se ve y se siente bien el sistema experto en términos de su base de conocimiento, estructura de inferencia, y salida?

Naturalidad. ¿Qué tan bueno es el sistema en dar al usuario la sensación de tratar con un experto más bien que con una máquina?

Alcance. ¿Cuán bien cubre el sistema experto la extensión de problemas importantes en su dominio de conocimiento?

Robustez. ¿Cuán sensitivo es el sistema experto a cambios menores en la descripción del problema?

Desempeño. ¿Cuán bien se ajusta el desempeño del sistema a la realidad? Por ejemplo, ¿da al consultor de inversiones consejo acerca de inversiones lucrativas?

Precisión. ¿Cuán precisas son las predicciones hechas por el sistema? Por ejemplo, ¿suministra el sistema de diagnóstico de fallas una descripción detallada, o simplemente una idea de la localización general de la falla?

Confiabilidad. ¿Qué proporción de los consejos o predicciones del sistema pueden clasificarse como equivocadas?

El proceso de validación de este sistema experto se facilitó por la existencia de conocimiento del dominio.
La valoración del comportamiento de entrada-salida se facilitó por las cartas de "caza de fallas". Este tipo de validación se la hizo manualmente, comprobando todas las conclusiones posibles de los síntomas y árboles de falla existentes, con todas sus alternativas registradas para el ATA 24. En esto fue de ayuda la numeración secuencial de los procedimientos en cada uno de los diálogos. Así se verificó al sistema experto como una caja negra exitosamente.

La validación de la inferencia (mirando dentro de la caja negra) se llevó a cabo practicando trazados paso paso de los procesos de razonamiento establecidos para el diagnóstico. De este modo se comprobó la estructura lógica de las reglas, los hechos, y el uso de la memoria al tiempo de la ejecución del sistema. La validez de las razones para llegar a ciertas conclusiones vino establecida desde el conocimiento del dominio, y se verificó su seguimiento con el motor de inferencia y las facilidades para depuración del shell.

A continuación se presenta un repaso de los requerimientos establecidos para este sistema experto junto con sus criterios de éxito o fracaso:

- Con respecto a la exactitud de la experiencia estructurada en las cartas de caza de fallas, esta resultó satisfactoria pues los expertos atribuyeron hasta un 90 % de efectividad a estos procedimientos, que se verificaron con éxito. Así, este sistema experto de diagnóstico, con un sistema de ayudas en línea y una base de datos relacional (información histórica) cubre satisfactoriamente la extensión de problemas importantes en su dominio de conocimiento, con una buena precisión.
- El sistema de ayudas que funciona a todo tiempo suplementando al sistema se incorporó con éxito a cada uno de los 10 ejecutables que componen el sistema global. Es sencillo de manejar y cuenta con un arreglo jerárquico de opciones, teniéndose así un sistema de "árboles de ayuda".
- La interfase amistosa al usuario presenta menús y diálogos incluyendo hipertexto, lo cual influye favorablemente en la presentación, comprensión e interacción con el usuario. Se ha buscado estandarización y naturalidad en sus interacciones. Esto hace que el sistema experto se vea y se sienta bien con respecto a su base de conocimiento, estructura de inferencia, y salida. De esta manera de demostró la validez de su fachada.
- La información histórica se incorporó con mucho éxito en una base de datos relacional -con opciones para mirar, agregar, modificar y suprimir registros- completamente manejable por el usuario. La misma que

contiene registros con toda la información que se puede hallar en una página de bitácora. Y es capaz de proveer una clasificación automática de todos los incidentes de falla por subsistemas (entre otras cosas). Además puede ser ampliada y modificada hasta el grado que se desee por parte del usuario (más detalles en la guía del usuario).

- La comprobación del ejecutable SNL también dio los resultados que se esperaron, pues le permite al usuario escribir memorándums acerca de los síntomas de falla que no aparezcan alistados en el sistema de diagnóstico general. De esta manera se cuenta con una vía de retroalimentación desde los expertos hacia el ingeniero de conocimiento, con miras a la reactualización y mantenimiento del sistema experto de diagnóstico (más detalles en la guía del usuario).
- En el sistema, el usuario toma la iniciativa desde el principio, y para reducir las posibilidades de error debidos a variaciones en el ingreso de la información de entrada, el usuario no tiene que escribir nada, sólo elegir opciones (excepto en la base de datos relacional y en el ingreso de memos, en los cuales también se ha procurado reducir al mínimo lo errores de ingreso de información). Por eso, desde cualquier nodo de un árbol de falla la inferencia le permite al usuario retroceder para escoger otro camino, mostrándole en cada nivel anterior las selecciones que se hicieron entonces.
- Por el conocimiento involucrado, así como por las características de funcionamiento del sistema, su

desempeño se ajusta a la realidad para la que fue diseñado, como asistente en el control de calidad y diagnóstico de fallas.

- El tiempo de respuesta del sistema no representa un inconveniente puesto que es casi instantáneo.

3.1.6 MANTENIMIENTO

Una vez que el sistema experto se hace disponible a los usuarios, fuera del equipo de trabajo original, surge el problema de darle el mantenimiento apropiado. Es importante no olvidarse de los posibles problemas en el mantenimiento de un sistema experto cuando se enfoca por primera vez el problema de construirlo.

Por ejemplo, la interfase al usuario necesitará ser adaptada según el requerimiento de los usuarios. Durante la vida de un sistema experto, tenderán a ocurrir modificaciones significativas de esta interfase por dos razones:

1. Con el transcurso del tiempo, se volverá mucho más fácil modificar la interfase que modificar la base de conocimiento fundamental y el motor de inferencia del sistema experto.
2. La evaluación del usuario y su aceptación del sistema serán influenciados mucho por la calidad de la interfase. Algunos usuarios pudieran preferir inclusive un sistema experto menos competente pero con una mejor interfase. Por ejemplo esto puede ocurrir en medicina, donde los buenos modales al lado de la cama del paciente pueden valorarse más que otras

consideraciones, al determinar la aceptabilidad de un médico para un paciente.

La portabilidad del software del sistema experto es muy necesaria, y el sistema desarrollado en esta ocasión es completamente portátil. Los sistemas expertos para ayudar a un piloto de aviación por ejemplo, tendrán que ejecutarse en un hardware que se ajuste a las restricciones de un entorno donde el peso y el espacio son críticos.

Una característica de los primeros sistemas expertos era que podían correr solamente en cierto tipo de máquina muy costosa. La tendencia actual hacia la creación de shells de sistemas expertos en lenguajes convencionales, C en particular, está produciendo shells y sistemas expertos más portátiles (como en el presente caso).

Mientras que la base de conocimiento y el motor de inferencia serán relativamente portátiles; sin embargo la interfase al usuario probablemente continuará siendo un tanto dependiente de la máquina. En este sistema experto los atributos esenciales de la interfase al usuario, están presentes prácticamente en cualquier computador corriente de la actualidad, aunque pueden ser realzados en computadores más costosos. Shells que poseen atributos gráficos sofisticados requieren igualmente de computadores más costosos.

Son inevitables las nuevas versiones de un sistema experto operacional a medida que el sistema se reactualiza para hacer frente, ya sea al conocimiento equivocado, o ausente, o a un entorno cambiante que produce nuevo conocimiento.

Teniendo presente esta necesidad se ha provisto el sistema para ingreso de memos por parte de los usuarios.

Esto da énfasis al hecho de que un sistema experto requiere soporte técnico. Este es un buen argumento para evitar que solo en los pocos individuos que originalmente construyeron el sistema se deposite una confianza excesiva.

La consideración de este hecho se hace evidente, por ejemplo, en la base de datos relacional. La misma que se ha diseñado de tal manera que el usuario pueda desenvolverse por sí mismo y no dependa de otros en la práctica.

La sencillez de manejo del sistema de diagnóstico también se hizo con la misma mentalidad.

Con todo, este sistema está para recibir mantenimiento, especialmente al recoger observaciones valiosas de los expertos mediante el sistema de memos SNL, particularmente para los síntomas de falla no alistados en cada subsistema de diagnóstico, y para el margen de no-efectividad del 10 % al 25 % máximo en el caso de las cartas de caza de fallas.

CONCLUSIONES Y RECOMENDACIONES

Este capítulo se ha organizado en 4 secciones, en lo que respecta a conclusiones y recomendaciones: 1) del sistema experto, 2) de la herramienta de desarrollo INTELLIGENCE COMPILER, 3) de criterios de desarrollo, y 4) recomendaciones generales.

4.1 SISTEMA EXPERTO

1.- Si el papel del sistema experto que se propone desarrollar es de apoyo para la toma de decisiones, el ingeniero de conocimiento debería identificar las debilidades en el proceso de razonamiento experto, y proponerse completarlo.

En el presente caso, aparte de reflejar la exactitud en el proceso de diagnóstico de fallas, los expertos valoraban mucho la información histórica, la cual constituiría un auxiliar de apoyo valioso para ellos en la toma de decisiones. Por eso es que se ha puesto mucho énfasis en desarrollar una base de datos inteligente que le brinde autonomía de uso al usuario.

2.- En el proceso de diagnóstico se obtuvo mejores resultados dándole la iniciativa al humano más bien que al sistema de computador.

Se ensayó el nivel en el que la computadora dirige y el humano obedece sujetándose a las iniciativas asumidas por el computador, pero esto sólo condujo a problemas complejos consumidores de tiempo.

No se puede negar lo atrayente que suele ser pensar en un hábil sistema inteligente, capaz de hacerse cargo de mucha responsabilidad por sí solo, pero la realidad es que un sistema de computador puede ser de mayor utilidad si sirve como un sistema de apoyo al usuario y es dirigido por este. Más todavía cuando el diagnóstico de fallas sigue siendo una de las tareas en las que el humano es mucho más efectivo que una computadora.

Cuando se seleccionó el nivel de intrusividad adecuado, se mejoró mucho la eficiencia del sistema y su programación. Recordando lo mencionado en el capítulo anterior; se trabaja en dos niveles: 1) el humano dirige y la computadora obedece, y 2) la computadora sugiere y el humano decide. Con esta medida, la interfase amistosa para el usuario fue más sencilla de elaborar, en base a menús y diálogos estandarizados.

3.- En este sistema experto, se ha seguido el enfoque tradicional de abajo-hacia arriba en la adquisición de conocimiento (es decir que se extrae de los expertos conocimiento específico y se lo refina). Este es un enfoque experimental de prototipeamiento rápido. Y eso es lo que se buscaba desde el comienzo.

Este método aparte de sus ventajas tiene también sus desventajas. Se tiene la alternativa de usar el método de arriba-hacia abajo.

4.- Se hizo necesario volver a la fase de Diseño Conceptual en varias ocasiones, hasta encontrar la organización final

del conocimiento que más se ajustaba a las necesidades del proyecto, y a los recursos del software.

Por ejemplo, la descripción de los síntomas no se la pudo representar satisfactoriamente, ni con lógica de predicado, ni con menús, ni con diálogos que involucraban selecciones; por su extensión, contexto y presentación. Puesto que se fragmentaba o se abreviaba demasiado, afectando la eficiencia del sistema, y la comprensión del usuario.

Así, después de ensayar estas 3 alternativas, mas bien se optó por manejar esta etapa con una técnica sencilla y eficaz de interfase al usuario: Hipertexto, manejado con reglas inversas.

La ventaja de hacer esto, aparte de superar los inconvenientes ya mencionados, fue que se pudo añadir los códigos del manual de mantenimiento -como variables globales- en ranuras de marcos (la programación se hizo más eficiente y se logró un mayor ahorro de memoria), y además como una útil referencia visual de hipertexto (para el manual de mantenimiento del avión).

5.- En el caso de los árboles de falla -en cambio- se vio que se podía manejar mayor cantidad de información en cada paso de la inferencia incluyendo el conocimiento en ventanas de diálogos. Aquí se puede incluir textos editables o no por el usuario, selecciones -únicas o múltiples-, botones, y factores de certeza.

Esencialmente, se utilizó como componentes de los diálogos a textos explicativos, selecciones únicas, y botones. Cada selección es una ramificación del árbol de falla. Con este

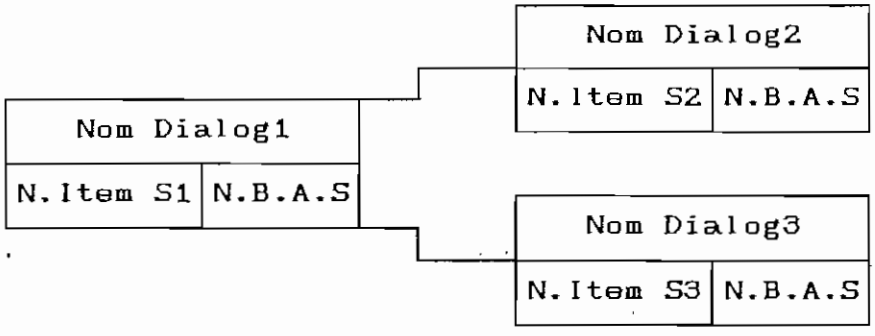
modelo, el proceso de inferencia se acortó de una manera clara y concisa.

Sin embargo cuando se utilizan muchos diálogos -como en este caso- hay que tener cuidado de adoptar el estilo de programación apropiado -en lo que concierne a cerrar los diálogos oportunamente- ya que de lo contrario, se consumirá demasiada memoria RAM e inesperadamente pronto se puede llegar a quedar fuera de memoria, en algún punto de la inferencia. En las hojas editables de programas se puede apreciar este esquema, en el que se conecta con un "or" lógico la cláusula que cierra el diálogo al presionar <ESC>, -o <ENTER> en esta misma opción- liberando así la memoria cuando se retrocede en el árbol de fallas.

Además se ahorrará tiempo en la codificación de los diálogos en la base de conocimiento, si se adopta una representación de referencia en el papel para cada diálogo y árbol de falla a medida que se los va creando. Por ejemplo, será necesario registrar el nombre del diálogo, y los nombres y contenido de los ítems -particularmente de aquellos que contienen información que interviene activamente en la inferencia, como los de selecciones y botones por ejemplo- (los ítems de texto explicativo se los trata pasivamente). Esta información también será fundamental para el uso del predicado BUTTON-CHOSEN, que se utiliza para manejar toda la información contenida en un diálogo.

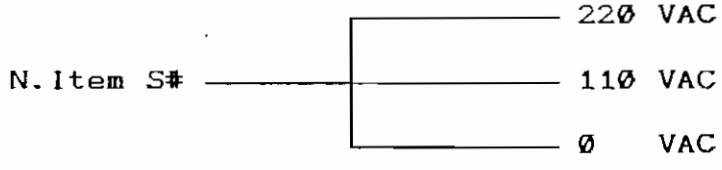
La siguiente representación sirvió mucho para este propósito, evitando invocaciones repetidas y engorrosas al editor de diálogos (IDG), desde el editor de programas (ICE)

o desde el ejecutor (ICX) para averiguar los datos mencionados:



Donde:
Nom Dialog#= Nombre del Diálogo respectivo
N.Item S# = Nombre del Item de Selecciones respectivo
N.B.A.S = Nombre del Botón de Aceptación de Selecciones respectivo

También se registrará el contenido del ítem de selecciones para la inferencia, por ejemplo:



Este es sólo un ejemplo -de un trozo de árbol- del método empleado aquí.

Un diálogo puede incluir uno o varios ítems de las 9 posibilidades que existen; asimismo uno o varios botones de aceptación de ítems. Por cada nombre de botón de aceptación se hace necesario agregar un predicado BUTTON-CHOSEN, con este nombre y el del diálogo. A su vez, con el nombre del ítem se obtendrá la información que suministre para la inferencia (una selección hecha por el usuario por ejemplo), con el mismo formato que se utiliza para el caso de los marcos, y en el antecedente de la regla inversa cuya conclusión es el predicado BUTTON-CHOSEN.

Una característica que hizo mucha falta; fue la de poder exhibir las ventanas de diálogos en una localización específica de la pantalla. El shell I/C no tiene, al menos en esta versión, provisión para hacer esto -aunque sí se los puede mover a voluntad después que aparecen-.

Se trató de compensar esta falta diseñando los diálogos de una anchura estándar (35 caracteres), para que cuando aparezcan -en el centro de la pantalla- sólo se note la variación en la altura del diálogo. Además con este formato se puedan reacomodar varias ventanas en la pantalla sin que se traslapen entre sí.

6.- Cuando se utilice el sistema de memos, una vez que se haya escrito una línea completa y continuado a la siguiente, no se podrá regresar a corregir la línea anterior. Cada línea contiene 80 caracteres y se recomienda prever esto, continuando a la línea siguiente con <Enter>, en vez de esperar a que pase automáticamente, al superar los 80 caracteres.

7.- Concerniente a la base de datos relacional, se pueden plantear las siguientes inquietudes:

- Obtener información por año, por mes, y por día separadamente. Este puede ser un buen objetivo al implementar una base de datos relacional, que contenga información acerca de registros de fallas. En el presente caso se tiene la fecha completa en un solo campo.
- Otra cosa útil para una referencia rápida podría ser un listado de todas las fechas y bitácoras que se encuentran en la base de datos, antes de ingresar a la misma.

- Además sería interesante añadir las horas de vuelo, con referencia a los calendarios de mantenimiento de la aeronave para los sistemas involucrados en la base de datos. Eventualmente también el número de vuelo, procedencia, destino, nombres de la tripulación, y responsables por el mantenimiento.
- Quizá para lograr manejar toda esta información eficientemente, se podrían definir varios esquemas en la base de datos y obtener solamente la información pertinente a través del álgebra de relaciones.

4.2 HERRAMIENTA DE DESARROLLO (SHELL INTELLIGENCE COMPILER)

1.- El shell INTELLIGENCE COMPILER [36] es una de las herramientas que proveen un conjunto de diversos paradigmas avanzados y poderosos que incluyen: Marcos con agregados de procedimientos, valores activos y pre-definidos, orientación a objetos con herencia y excepciones; reglas con encadenamiento hacia adelante y hacia atrás, ambas con cuantificación existencial y universal de variables; razonamiento inexacto y cálculo de certidumbres; razonamiento semi-exacto con alternación de inferencia exacta e inexacta; bases de datos relacionales; árboles-B; y estructuras definidas por el usuario.

Este es uno de los paquetes que ofrecen las ventajas de una integración uniforme de diversas estructuras de conocimiento. Por ejemplo, un marco puede invocar una regla inversa, que invoca a un grupo de reglas hacia adelante, que invocan a un grupo de reglas inexactas con factores de certeza, que invocan a una regla inversa ya sea de los agregados de procedimientos de un marco -para leer o

escribir- o que invoque hipertexto, o ventanas de diálogos, o una función en lenguaje PASCAL, C, etc.

El código de I/C es muy compacto y eficiente, y permite que los sistemas expertos se encajen en otras aplicaciones, o se enlacen con bases de datos externas; o que se utilicen hasta en computadoras portátiles.

2.- Hipertexto se puede manejar en cualquier momento, en combinación con cualquier paradigma para representación de conocimiento a través del predicado LBODY, o de alguno definido por el usuario. Hipertérminos diferentes pueden invocar hipertextos diferentes o a uno solo inclusive (a través de una regla inversa con el parámetro de enlace, común a todos los hipertérminos). Así, se presentan facilidades para la creación de hiper-pantallas para interfases.

3.- La jerarquía padre-hijo-hermano tiene varias aplicaciones dentro del shell I/C:

- En el diseño del sistema de ayudas para el usuario.
- En los diálogos, para obtener selecciones -únicas o múltiples- entre otras cosas. En general, para capturar la información de un ítem específico.
- En los marcos, a través de los términos de objetos, para accesar la información de las ranuras -lectura o escritura-.
- En los registros de las bases de datos relacionales, para obtener la información de algún campo en particular de un registro.

El formato en el caso del sistema de ayudas difiere de los otros tres.

4.- El uso del predicado NO-BACKTRACK, aparte de servir para la liberación de memoria RAM, es muy útil para controlar las recursiones y evitar los rebotes indeseables del control de la inferencia. Es utilizable solamente en reglas inversas. Esta es otra ventaja para preferir las reglas inversas a las directas, en este shell particularmente.

5.- Para hacer la transferencia de tecnología al usuario al término del desarrollo de la aplicación, debe incluirse invariablemente el archivo *.BAT (creado en I/C, o su expresión equivalente: ICX -a ... -f ARCHIVO.ICP, en el diseño de una pantalla de menú o en otro *.BAT), y el *.ICP. Eventualmente se incluirán los archivos *.DLG (cuando se han creado diálogos en el IDG), *.HYT (cuando se ha diseñado hipertexto), *.DAI (cuando se ha diseñado una base de datos relacional), y *.HLP (cuando se ha creado un sistema de ayudas on-line). De los archivos del shell, se incluirá solamente el ejecutor ICX.EXE.

Ninguno de estos archivos es editable, excepto el *.BAT, y el *.HLP (menos su codificación de diseño).

6.- Al utilizar el IDG para diseñar diálogos, conviene diseñar los ítems comunes a todos los diálogos una sola vez (botones de aceptación o cancelación, ítems de selecciones, etc.), y para el resto de diálogos usar la opción para copiar ítems; ya sea en el mismo diálogo, o en otro; en el mismo archivo o en otro; y dentro o fuera del directorio de trabajo. Esto ahorrará tiempo y esfuerzo.

7.- Como un recordatorio; en el editor se puede hacer uso de 3 alternativas para agregar información que se repite:

- Definiendo macros con Alt =, y ejecutándolos con Alt -.

- Usando la selección de búsqueda y reemplazo, en la opción 3 del menú (Search).
- Usando la opción 2 (Edit) del menú, con las selecciones de mover (Cut) o copiar (Copy).

8.- De los símbolos alfanuméricos, se ha notado que sólo los puntos se pueden utilizar como parte de los argumentos de los predicados (definidos en el shell o por el usuario).

9.- Si al ejecutar la aplicación en el modo "stand-alone" (desde el DOS) se observa que se daña la presentación del menú; entonces convendrá usar el predicado REPAINT-SCREEN para corregir este problema. Se ha notado que cuando se utiliza hipertexto en particular, es necesario hacer esto, ya sea asociando dicho predicado con los predicados APP-STARTUP o MENU-CHOSEN.

10.- Al diseñar el sistema de ayudas, será útil recordar que el archivo de ayudas *.HLP resultante habrá que copiarlo del directorio en que se lo creó al directorio del I/C para que se lo identifique y funcione (desde el ambiente de ejecución ICX), de lo contrario no se lo detectará. Sin embargo, al ejecutar la aplicación desde el DOS, basta que el mencionado archivo esté junto con los demás que componen la aplicación.

11.- Cuando la inferencia no puede averiguar el valor de verdad de alguna cláusula involucrada en el proceso de razonamiento, entonces pregunta por él. Esto causa una buena impresión al usuario. Para conseguir que el sistema pregunte lo que no conoce, se utiliza un switch, el predicado ASK-ON. Sólo que se ha encontrado un comportamiento aleatorio e indeseable cuando se utiliza este predicado con reglas directas. Específicamente, asociado con el predicado -INVOKE

con hipótesis- (usado para desencadenar el proceso de razonamiento con reglas directas). El problema se centra en la hipótesis, sea una constante, una variable o una expresión. Se sugiere averiguar mejor este incidente.

12.- A continuación se llama la atención sobre otros aspectos específicos del shell 1/C que sin ser decisivos sí afectan la utilización cabal de todos sus detalles:

- No se ha logrado instalar el paquete en el modo extendido en clones de IBM. Se ha pensado que puede deberse a recursos exclusivos existentes en el ROM BIOS de IBM.
- Una debilidad del editor ICE, es que no se puede ingresar directamente caracteres ASCII que no aparecen en el teclado. En español se requieren de la vocales con tildes, las letras ñ, Ñ, ü y los caracteres ¡, ¿, entre los más utilizados. Para ingresar estos caracteres se necesita usar otro editor.

  Después de escribir un archivo de programa en otro editor, será necesario marcar los predicados con la tecla F9 en el ambiente del ICE.

  Lo antedicho es válido para la escritura de programas, incluído hipertexto y los archivos de ayudas; mas no para el texto de los diálogos. Esta es la razón por la que se decidió escribir todo el texto de los diálogos en letras mayúsculas.
- En los archivos de diálogos *.DLG, se ha observado que no se reducen de tamaño cuando en ellos se suprime información. Para obtener un archivo que indique el tamaño que corresponda a la realidad (si se le ha suprimido información), lo que se ha hecho es crear

otro archivo y copiar en este los diálogos del archivo original uno por uno. Se sugiere investigar mejor esto.

- No se ha logrado que funcione un ítem en el IDG, que resultaría sumamente valioso para el ingreso de texto formateado en ventanas de diálogos. Este es el ítem # 8, llamado "Text Box". Esto representaría un gran ahorro de tiempo y esfuerzo, y un mejoramiento en la presentación del texto en los diálogos. En vez de este ítem, se tuvo que utilizar el # 1 (Messages), con el que se diseña el texto línea por línea (en el formato de anchura que se adoptó); en vez de hacerlo como un bloque de texto, de una sola vez.
- Sería bueno hacer una revisión de índices que incluyan 2 o más campos (multi-campo), para las bases de datos relacionales.
- Los predicados para crear ventanas en una posición específica: WINDOW, BOXED-WINDOW, y BOX junto con el predicado TEXT, funcionan bien en el ambiente del ICX; pero en el modo "stand-alone" se daña la presentación al usuario. Por eso, cuando se desee presentar una interfase usando ventanas, se tiene la alternativa de utilizar en su lugar el predicado POP-TEXT que cumple un objetivo similar.
- Se ha tenido dificultad en afirmar hechos con factores de certeza dentro del programa en el momento de su compilación. No hay problemas cuando son afirmados por el proceso de inferencia. Con hechos sin factores de certeza no se ha presentado esta dificultad.

- Al diseñar menús, si se desea que los ítems de sus opciones desplegados hacia abajo tengan más de 20 caracteres de longitud, hay que evitar usar la opción MENU ACTIONS perteneciente a la opción 2 (Menus) del IDG; de lo contrario, los ítems se truncarán en esta longitud. En este caso, es mejor utilizar el predicado MENU-CHOSEN y los predicados OPEN-DIALOG, RUN-DIALOG, y CLOSE-DIALOG. Así se podrá extender la longitud de los ítems a 40 caracteres.
- Se sugiere estudiar con la opción de ejecución de trazado paso a paso, los programas bajo el archivo EXAMPLES y si es posible, imprimirlos; ya que allí se encuentra algún detalle informativo que no se presenta en el texto del manual -por ejemplo, ciertos predicados y sus usos-.
- Si con un monitor VGA monocromático no se logra instalar exitosamente el paquete, o no se aprecia el subrayado de los predicados; conviene revisar la posición del jumper de selección MONO/COLOR en la tarjeta VGA.

  Si la dificultad es que no se logra instalar y correr el paquete, se puede intentar poniendo este jumper en COLOR; para luego de la instalación, volverlo otra vez a MONO, a fin de poder apreciar el subrayado de los predicados nuevamente.

  Por supuesto, con un monitor VGA color, no se tienen estos problemas. Y además, se podría ensayar la instalación en modo extendido.

4.3 CRITERIOS DE DESARROLLO

1.- La utilización de una herramienta eficaz para construir sistemas expertos es un factor decisivo para el éxito en el desarrollo de una aplicación, su uso definitivamente representa una gran ventaja, en vez de intentar construir un sistema experto a partir de un lenguaje como PROLOG o C.
Pues además de las facilidades que estas herramientas brindan para el diseño de interfases al usuario, y la edición, ejecución y depuración de la base de conocimiento con integración de varios paradigmas; se ha comprobado que estos shells pueden reducir el tiempo para desarrollar una aplicación hasta en un orden de magnitud menos que el que requeriría construirlo a partir de LISP, PROLOG, etc.

2.- Cuando se utiliza un shell para el desarrollo de sistemas expertos, se vio lo acertado de la sugerencia de iniciar familiarizándose con la interfase al usuario del shell cuando ya se está listo para empezar con la programación. Para el I/C esto quiere decir empezar creando una interfase con el IDG ("Generador Inteligente de Diálogos"). Esto es necesario al principio porque:

- Facilitará la clarificación de los objetivos y requerimentos de la aplicación. En el presente caso, facilitó la concepción del diseño del software.
- Uno de los criterios clave de validez global para la aceptación de un sistema experto tiene que ver con la calidad de la interfase al usuario.
- Además con el transcurso del tiempo será más fácil modificar la interfase al usuario que la base de conocimiento, o el motor de inferencia.

Al principio se trató de fijar la atención en la inferencia en sí y en el diseño de reglas. Después, cuando se adquirió la suficiente familiaridad con el IDG, se vio la necesidad de un cambio en la concepción del sistema. Esto también representará un ahorro de tiempo.

Ya en el IDG, será necesario pasar lo más pronto posible del modo de aprendiz al modo experto de funcionamiento.

3.- Antes de promover el desarrollo de cualquier sistema experto, se recalca la necesidad de tomarse tiempo para reflexionar en todas las etapas de diseño involucradas (figura 3.1). Si no se hace así, puede haber un apresuramiento en el que se pasen por alto factores esenciales que deben considerarse al principio -antes de invertir tiempo y esfuerzo innecesariamente-.

4.- En el cumplimiento de las dos primeras etapas de diseño: Análisis de Factibilidad y Diseño Conceptual, la ayuda de los expertos y usuarios fue sumamente útil. La guía y sugerencias iniciales, los diagramas presentados, y la comprensión de la manera cómo trabajan, fueron una ayuda primordial en la introdución y comprensión del dominio.

La conclusión es que desde el principio se debe trabajar con los expertos, así como con los futuros usuarios del sistema.

5.- El problema de utilizar el método de abajo-hacia arriba para la adquisición de conocimiento, estriba en que si se desea que el sistema experto sea capaz de manejar casos nuevos, que involucren aspectos complejos no representados en el conocimiento adquirido inicialmente, se requiere una mayor re-estructuración del conocimiento mas bien que un sintonizamiento fino del mismo (insistiendo en continuar

usando el método de abajo-hacia arriba). Pues se corre el riesgo de "fallar en apreciar las demandas de la tarea" [19], y requerir de más tiempo y recursos que resultarían en el fracaso del proyecto de desarrollo del sistema experto. Entonces, este método de prototipeamiento rápido puede imponer limitaciones a la habilidad y robustez del sistema experto desarrollado.

De modo que se puede examinar con provecho su contraparte: El método de arriba-hacia abajo, para un sistema experto más robusto que sea capaz de habérselas con casos nuevos y que no se degrade bruscamente en la periferia de su conocimiento (el método de arriba-hacia abajo parte de métodos de solución bien estructurados y de alto nivel, que tienen algún grado de independencia del dominio de conocimiento y a partir de los cuales se puede obtener conocimiento específico del dominio.- Páginas 21 y 22).

6.- La programación con reglas inversas es la más versátil; cualquier paradigma de representación de conocimiento y su conexión con la interfase al usuario, se puede manejar con ellas. Además poseen facilidades de predicados exclusivos, y el mismo formato básico para incluir razonamiento inexacto. No sucede lo mismo con las reglas directas, que son menos flexibles.

7.- Es muy importante valerse de las posibilidades que brinda el emparejamiento de muestras del motor de inferencia -en especial usando variables- para automatizar el proceso de inferencia, ahorrar código y memoria.

4.4 RECOMENDACIONES GENERALES

1.- Es deseable asegurarse el financiamiento necesario desde el comienzo y antes de echar a andar el proyecto. Por ejemplo, en el desarrollo de un sistema experto, el computador es una herramienta de trabajo indispensable a todo tiempo para el diseñador, por lo que se debería buscar su disponibilidad continua. Si no es así, el trabajo puede verse retardado y perderse la continuidad.

2.- Todos los prototipos exitosos de investigación y desarrollo de sistemas expertos se han logrado a través de un trabajo en equipo.

En el presente trabajo se experimentó esta necesidad y se pudieron haber alcanzado aún mayores objetivos al trabajar en equipo.

Por ello se sugiere que al emprender un proyecto para desarrollar un prototipo de éxito, se organice y se comparta el trabajo entre dos o más graduandos. De esta manera se trabaja en proyectos de investigación y en la industria.

3.- Para otras aplicaciones se sugiere involucrar bases de datos relacionales en el proceso de razonamiento. Esta es una técnica que brinda perspectivas muy interesantes con su álgebra relacional que incluye: Unión, intersección, diferencia, selección, proyección, y equijunción de relaciones.

4.- La programación semi-exacta ofrece atractivas posibilidades para manejar factores de certeza de una manera hábil. Esta facilidad también se la puede considerar con provecho en otros prototipos; por ejemplo en aplicaciones de

lógica difusa que cada vez son más numerosas [21], [22], [24], [33].

5.- Sería interesante también, llevar adelante un trabajo que se centre en la tarea de adquisición de conocimiento, ensayando técnicas, por ejemplo con ayuda de herramientas comerciales para este fin [19]. Y que pueda continuarse con otro trabajo que demuestre la validez del conocimiento adquirido, en el diseño de un sistema experto práctico.

La razón es que en el futuro cercano la ingeniería de conocimiento aún seguirá siendo una actividad experta y consumidora de tiempo, debido a la ausencia de herramientas mejoradas para la adquisición de conocimiento. Pues el comportamiento cognoscitivo y las estructuras de conocimiento en las tareas humanas de resolución de problemas, aún necesitan ser entendidas mejor y formalizadas por técnicas apropiadas de adquisición de conocimiento.

Con esto también se facilitaría mucho la construcción de sistemas expertos que apoyen a diferentes clases de usuarios, con diferentes grados de experiencia y conocimiento.

6.- Se ha dicho que la interacción entre un ingeniero de conocimiento y un experto, puede ser una sociedad y una búsqueda conjunta de conocimiento. No obstante, el diseñador debe estar preparado para que así suceda. Y para tener éxito en este objetivo le convendrá prepararse en ciertas técnicas para reunir información (ver págs. 23 y 24). Se ha visto la necesidad de esta clase de preparación específica.

Para este prototipo se ha hecho uso de las entrevistas y los recorridos a pie (en especial cuando había que recorrer el parque de mantenimiento de los aviones, o era necesario ingresar al avión).

Esta será una preparación productiva, básica y previa, para la segunda etapa de diseño: el Diseño Conceptual.

En conexión con lo anterior, se conoce que los expertos se desempeñan mejor como críticos antes que en la tarea de dilucidar su experiencia. Por ejemplo, en el desarrollo de un sistema experto llamado CLOT [3] para consultar desórdenes circulatorios, el prototipo inicial se construyó en 3 días usando el conocimiento de un estudiante de medicina. Y después se presentó este prototipo a un consultor experto.

Es mejor hacer esto, antes que involucrar desde el principio al experto en un proceso intensivo de adquisición de conocimiento. Así, la construcción de un sistema inicial básico provee un excelente marco de referencia para que las formalidades y complejidades del conocimiento del experto, se puedan adquirir más fácilmente [3].

7.- De la investigación en torno al desarrollo de este sistema experto, se encontró que la aviación es uno de los campos en el que existen varios prototipos exitosos de sistemas expertos con diversos propósitos, así por ejemplo:

- En control de vuelo tolerante a la fallas, este prototipo está siendo aplicado a la automatización de procedimientos de emergencia en aviones. Estas situaciones requieren de respuestas rápidas y

decisiones complejas. Puesto que sus procedimientos contienen numerosos puntos de evaluación y decisión; se prestan para una representación declarativa del conocimiento en tiempo real [16].

- Otro prototipo involucra un sistema inteligente para el aterrizaje por instrumentos (IILS, '87); el cual está diseñado para supervisar el estatus de la aproximación, y detectar la aparición de situaciones peligrosas en el avión durante los aterrizajes por instrumentos. Aquí se vio la necesidad de que los sistemas expertos asistan a la tripulación de vuelo en estos casos, debido a que según un estudio, las dos terceras partes de los accidentes parecen ser causados primariamente por la tripulación de vuelo, y casi la mitad de todos los accidentes ocurren durante la fase del aterrizaje [30].
- Un sistema para manejo del tráfico aéreo en una terminal desarrolla un plan de vuelo aceptable para cada avión que llega. Supervisa y corrige las desviaciones del acercamiento, y desarrolla planes de acercamiento para aviones inesperados [34].
- El Auxiliar de Planificador de Carga de American Airlines (AALP) se encarga de la planificación y operación segura de cada vuelo, antes del despegue. Con este sistema experto, la aerolínea incrementó su capacidad de planificación de carga en 300 vuelos, en 6 meses. Así se ha descargado en parte la planificación de carga al AALP, incrementando la productividad de los agentes de carga [29].

Estos antecedentes pueden servir como punto de partida para investigar posibles prototipos experimentales.

8.- Se encuentran muchas aplicaciones cruciales y fascinantes de sistemas expertos en tiempo real [6], [20]. Esta también es otra perspectiva para desarrollar proyectos.

Se conocen muchos paquetes que son capaces de trabajar en tiempo real: RESCU, PICON, HEXSCON, MUSE, y ART por citar los más importantes. Una de sus aplicaciones industriales son para el control de procesos. Estos shells son muy caros. Sin embargo, no hace falta tenerlos para trabajar en aplicaciones exitosas de tiempo real. Por ejemplo, utilizando programación mezclada con lenguajes de procesamiento simbólico y numérico (LISP y PASCAL en este caso) se han implementado controles basados en reglas (para toma de decisiones complejas), en computadores AT corrientes. La base de conocimiento de uno de estos controladores desarrollado, contenía sólo 34 parámetros y 34 reglas. La base de conocimiento en PASCAL (se hace una traducción automática de LISP a PASCAL) requirió de 16 Kbytes de RAM (parte para código, parte para datos), y el motor de inferencia requirió de 13 Kbytes de código [16]. Consecuentemente, se ha concluido que las reglas declarativas con código de procedimientos encajado, suministran una base sólida para la construcción de sistemas de control complejos y sin embargo económicos.

9.- A su vez, CONEX es un sistema experto para el diseño y análisis de sistemas de control [2]. En CONEX, el núcleo de conocimiento está formado por los diversos algoritmos de la

ingeniería de control para llevar a cabo las tareas de análisis y diseño. Dependiendo de un diseño particular o meta de análisis, de la descripción del sistema, de las especificaciones y restricciones de diseño, etc., se necesita un algoritmo o un conjunto de algoritmos particulares para alcanzar la meta. En el proceso para esta selección requiere que se consideren varios parámetros tales como la estabilidad numérica, la fortaleza, y la complejidad computacional de los algoritmos. Desafortunadamente estos parámetros a menudo no son bien especificados, por lo que el juicio y la experiencia de los expertos juegan un papel importante en la selección de los algoritmos. Esta experiencia se ha incorporado en CONEX, en la forma de una base de reglas. Además, estas reglas también guían al sistema experto para que efectúe las interacciones necesarias con el usuario.

Un motor de inferencia controla el entero proceso de toma de decisiones y puede adoptar diferentes estrategias de búsqueda, para diferentes espacios de reglas. En cada etapa de operación selecciona las reglas aplicables, usa las estrategias apropiadas para la resolución de conflictos, y activa las reglas para inferir nuevos hechos [2].

Esta es otra posibilidad para proyecto de un sistema experto que podría centrarse en el diseño y análisis de sistemas de control continuos o discretos.

10.- Un área promisoria del control basado en reglas, es la de los controles difusos; los cuales son pequeños sistemas basados en reglas que razonan hacia adelante para deducir

controles usando lógica difusa o multivaluada para manejar la incertidumbre [33], [21], [22], [24].

Se conoce también que las aplicaciones de control de lógica difusa a los procesos industriales; con frecuencia han producido resultados superiores a los controladores clásicos. Hay planteamientos para analizar la equivalencia entre controladores difusos y no difusos que aplican igualmente bien a controladores lineales PI o PD, lineales de multibanda, y controladores relevadores de multi-nivel. También se ha presentado una herramienta sencilla para ayudar a resolver el problema del sintonizamiento de los controladores difusos [33].

Este podría ser otro enfoque para proyectos, en el área de los controles lógicos.

11.- Otra posibilidad para investigación en este mismo sentido, tiene que ver con el uso de controles lógicos que usan matrices de reglas boleanas equivalentes a dígrafos de implicación que transforman los estados de verdad asociados con las condiciones de las reglas, en estados de verdad deducidos o abducidos [23]. Estas transformaciones efectúan lógica proposicional para deducir controles en sistemas no interactivos complejos de tiempo real. Ejecutan un control de tipo estímulo-respuesta que los humanos utilizan para desempeñar tareas como volar aviones, montar bicicletas, o controlar reactores nucleares.

El tipo corriente de sistemas expertos basados en reglas, es más poderoso y más lento de lo necesario en su ejecución para complejos controles en tiempo real, debido a que usan lógica de predicado (la cual puede desordenar y fragmentar

la memoria). Y el proceso debe suspenderse ocasionalmente debido al "aglutinamiento de basura", lo cual es un error fatal en tiempo real. Además la lógica de predicado es interminable si la solución no existe en el espacio de búsqueda de la información. Es incómoda también para organizar y controlar la información, y requiere de engorrosos emparejamientos y búsquedas. A su vez, esto requiere de búsquedas lineales o listas enlazadas que chequean el primer objeto, después usan un puntero para el segundo objeto, etc., hasta que se empareja el esquema completo.

La búsqueda binaria no es posible, de modo que la lógica de predicado es ineficiente en cualquier lenguaje de programación (alrededor de n comparaciones versus n/2exp(n). Las heurísticas pueden reducir el espacio de búsqueda, pero son limitadas (el diseñador debe conocer una técnica que acorte un tipo de búsqueda en particular).

Se presenta así la alternativa de la lógica proposicional, en vez de la lógica de predicado. Se considera un trueque. La primera es más simple, más rápida, y aún suficiente para muchas clases de sistemas de control que necesitan de alta velocidad. La última es más poderosa porque las condiciones del antecedente y consecuente de la regla tienen variables mas bien que objetos fijos (por ejemplo TEMPERATURA(X,Y)). El predicado no puede ser variable.

La lógica de predicado es impresionante, y algunos sistemas expertos interactivos con el usuario necesitan esto, sin embargo también necesita masivamente de arquitectura paralela y algoritmos que la acompañen para control de alta

velocidad en sistemas complejos de tiempo real. Ninguno de estos requerimientos resulta económico todavía.

No obstante, con este método, se trata de incrementar la automatización de una manera no costosa en sistemas tales como: Líneas de producción, plantas de procesamiento de alimentos o químicos, toma de decisiones en la cabina del piloto de un avión, comunicaciones y control de filtramiento adaptivo, sistemas de encendido y mezcla de combustible en un motor automotriz, etc.

12.- Otro punto de vista interesante tiene que ver con el MDI -un sistema experto de diagnóstico, para dominios eléctricos y mecánicos- descrito en la págs. 47-51 [11].

13.- En el apartado 1.4.1 completo, se trata sobre técnicas de razonamiento de diagnóstico, estructuras múltiples de conocimiento que tienen la habilidad de edificar y razonar alrededor de modelos profundos que abarcan una amplia extensión de información, tal como aprendizaje de la experiencia, información probabilística, y aprendizaje de ejemplos [27].

Hasta ahora, una buena parte de los sistemas de diagnóstico, se han construido usando alguna forma de sistemas de producción basados en reglas. Sin embargo, últimamente han emergido nuevas estrategias para apoyar un razonamiento mucho más complejo en el diagnóstico.- También se pueden tomar en cuenta estas técnicas.

14.- Puesto que llevar a cabo futuros proyectos de investigación depende en gran medida de trabajar con las herramientas apropiadas, se podría analizar la posibilidad de adquirirlas.

Por ejemplo, NEXPERT Object, es uno de los mejores sistemas expertos que hay disponibles en el mercado, a un precio razonable en comparación con otros; con muchas facilidades para interfases gráficas al usuario [14], [29].

Conociendo esto, se escribió a nombre del Departamento de Electrónica y Control de la Escuela Politécnica Nacional pidiendo más información a la empresa que lo diseñó. Se recibió una respuesta que ofrecía un descuento del 30 % por destinárselo a una institución educativa. El precio final era de U$ 5.600 (toda la correspondencia informativa enviada acerca de NEXPERT Object, se la puede consultar en el Laboratorio de Control de la Facultad de Ingeniería Eléctrica).

BIBLIOGRAFIA

1. Alty, J. L. and G. Johannsen, Knowledge-based Dialogue for Dynamic Systems, Automatica, Vol. 25, No. 6, pp. 829-840, 1989.

2. Basu, A., A. K. Majumdar and S. Sinha, An Expert System Approach to Control System Design and Analysis, IEEE Trans. Syst. Man Cybern., Vol. SMC-18, No. 5, pp. 685-693, Sep./Oct. 1988.

3. Bedford, D. R. and P. H. Buck, A Method of Automating the Knowledge Acquisition Task for Process Plant Expert Systems, Ferranti Computer Systems Ltd. U. K., pp. 47-51, 1989.

4. Bernard, J. A., Use of a Rule-Based System for Process Control, IEEE Control Systems Magazine, pp. 3-13, Oct. 1988.

5. Bitácoras de Vuelo DC-10, Ecuatoriana de Aviación, 1990-1992.

6. Chantler, M. J., Real-Time Aspects of Expert Systems in Process Control, Intelligent Automation Laboratory, Department of Electrical and Electronic Engineering, Heriot-Watt University U. K., pp. 4/1 - 4/8, 1988.

7. DC-10 Maintenance Manuals, McDonnel Douglas.

8. DC-10 ATA 100 Breakdown, McDonnel Douglas.

9. DC-10 Monthly Reliability Reports, KSSU, 1992.

10. Faught, W. S., Applications of AI in Engineering, Computer, pp. 17-27, July 1986.

11. Fink, P. M. and J. C. Lusth, Expert Systems and Diagnostic Expertise in the Mechanical and Electrical Domains, IEEE Trans. Syst. Man Cybern., Vol. SMC-17, No. 3, pp. 340-349, May/June 1987.

12. García Alegre, M., Inteligencia Artificial en el Control de Procesos.- Controladores Borrosos, Mundo Electrónico, 214, pp. 42-49, Feb. 1991.

13. Gertler, J., Survey of Model-Based Failure Detection and Isolation in Complex Plants, IEEE Control Systems Magazine, pp. 3-11, Dec. 1988.

14. Gevarter, W. B., The Nature and Evaluation of Commercial Expert System Building Tools, Computer, pp. 24-41, May 1987.

15. Gorman, C., Putting Brainpower in a Box, Time Magazine, August 15, pp. 40, 1988.

16. Handelman, D. A. and R. F. Stengel, Perspectives on the Use of Rule-Based Control, Princeton University, Department of Mechanical & Aerospace Engineering, Princeton New Jersey 08544, pp. 21-26, 1989.

17. Hari Narayanan, N and N. Viswanadham, A Methodology for Knowledge Acquisition and Reasoning in Failure Analysis of Systems, IEEE Trans. Syst. Man Cybern., Vol. SMC-17, No. 2, pp. 274-288, March/April 1987.

18. Hayes-Roth, F., The Knowledge-Based Expert System: A Tutorial, Computer, pp. 11-27, Sep. 1984.

19. Johannsen, G. and J. L. Alty, Knowledge Engineering for Industrial Expert Systems, Automatica, Vol. 27, No. 1, pp. 97-114, 1991.

20. Laffey, T., The Real-Time Expert, Byte, pp. 259-264, Jan. 1991.

21. Li, Y. F. and C. Lau, Development of Fuzzy Algorithms for Servo Systems, IEEE Control Systems Magazine, pp. 65-71, April 1989.

22. Looney, C. G., Fuzzy Petri Nets for Rule-Based Decisionmaking, IEEE Trans. Syst. Man Cybern., Vol. SMC-18, No. 1, pp. 178-183, Jan./Feb. 1988.

23. Looney, C. G. and A. Alfize, Logical Controls via Boolean Rule Matrix Transformations, IEEE Trans. Syst. Man Cybern., Vol. SMC-17, No. 6, pp. 1077-1081, Nov./Dec. 1987.

24. Maiers, J. and S. Sherif, Applications of Fuzzy Set Theory, IEEE Trans. Syst. Man and Cybern., Vol. SMC-15, No. 1, pp. 175-189, Jan./Feb. 1985.

25. Malin, J. T. and N. Lance, Processes in Construction of Failure Management Expert Systems from Device Design Information, IEEE Trans. Syst. Man Cybern., Vol. SMC-17, No. 6, pp. 956-967, Nov./Dec. 1987.

26. McCoy, M. S. and R. Levary, Augmenting Knowledge Acquisition Processes of Expert Systems with Human Performance Modeling Techniques, IEEE Trans. Syst. Man Cybern., Vol. SMC-18, No. 3, pp. 467-472, May/June 1988.

27. Milne, R., Strategies for Diagnosis, IEEE Trans. Syst. Man Cybern., Vol. SMC-17, No. 3, pp. 333-339, May/June 1987.

28. Ogard, O. and E. Woods, Intelligent Alarm Handling, SINTEF - The Foundation for Scientific and Industrial

Research at the Norwegian Institute of Technology, Division of Automatic Control, N-7034 Trondheim Norway, pp. 69-74, 1989.

29. Neuron Data, Inc., NExpert Object Correspondence, Palo Alto CA, 1991.

30. Parsaye, K. and M. Chignell, Experts Systems for Experts, John Wiley & Sons Inc., 1988.

31. Rouse, W. B. and S. H. Rouse, Analysis and Classification of Human Error, IEEE Trans. Syst. Man Cybern., Vol. SMC-13, No. 4, pp. 539-549, July/Aug. 1983.

32. Shirley, R. S., Some Lessons Learned Using Expert Systems for Process Control, IEEE Control Systems Magazine, pp. 11-15, Dec. 1987.

33. Tang, K. L. and R. J. Mulholland, Comparing Fuzzy Logic with Classical Controller Designs, IEEE Trans. Syst. Man Cybern., Vol. SMC-17, No. 6, pp. 1085-1087, Nov./Dec. 1987.

34. Tobias, L. and J. L. Scoggins Jr., Time-Based Air-Traffic Management Using Expert Systems, IEEE Control Systems Magazine, pp. 23-29, April 1987.

35. Yoon, W. C. and J. M. Hammer, Deep-Reasoning Fault Diagnosis: An Aid and a Model, IEEE Trans. Syst. Man Cybern., Vol. SMC-18, No. 4, pp. 659-675, July/August 1988.

36. Intelligence Compiler: User's Manual, IntelligenceWare, Inc., Los Angeles, CA, 1989.

NOTA: Las referencias bibliográficas [3], [6], [16] y [28] se las puede encontrar en el Laboratorio de Electrónica y Control de la Facultad de Ingeniería Eléctrica.

SHELL INTELLIGENCE COMPILER (I/C)

B.1 ESPECIFICACIONES TECNICAS

Intelligence Compiler es el nombre de un sistema para desarrollo de sistemas expertos, registrado por IntelligenceWare, Inc. de Los Angeles, CA. que presenta excelentes atributos en programación de alto nivel y software de Inteligencia Artificial.

Intelligence Compiler (versión 3.0 , 1989) se distribuye en 12 diskettes de 5 ¼ DS DD para IBM PC's y compatibles. Para PS/2 y compatibles se disponen de 6 diskettes de 3 ½ DS DD. Los archivos del sistema vienen empaquetados.

Requiere de IBM o PC-DOS 3.3 en adelante, y ocupa aproximadamente 2.2 Mb en el disco duro.

Funciona en dos modos: En el modo básico (640 K) y en el modo de memoria extendida.

El modo básico requiere de 570 Kb de memoria RAM libre como mínimo, si no es así, entonces se puede hacer disponible la memoria faltante desactivando la del software residente (estos programas se instalan por lo general a partir de una invocación en el autoexec.bat).

A su vez, la versión de memoria extendida, si está disponible en el sistema utiliza esta clase de memoria para manejar bases de conocimiento más grandes. Si no se cuenta

con memoria extendida, entonces se debe seleccionar el modo básico durante la instalación. Este modo no permite el uso de predicados C para ahorrar espacio de memoria.

B.2 CARACTERISTICAS GENERALES

El I/C es un ambiente para desarrollar sistemas expertos y compilar conocimiento que permite:

a. Desarrollar bases de conocimiento usando un editor inteligente para IBM PCs y PS/2 (y compatibles).

b. Combinar varios métodos de representación de conocimiento dentro de un solo ambiente.

c. Compilar el conocimiento para su ejecución.

El I/C recopila eficientemente los métodos fundamentales de inferencia, control y representación de conocimiento que se usan en aplicaciones de Inteligencia Artificial.

Otras características del I/C incluyen:

1. Programación de alto nivel.
2. Edición de Hipertexto.
3. Formatos múltiples para la representación de conocimiento.
4. Inferencia de encadenamiento hacia adelante, hacia atrás, y en modalidad mixta.
5. Inferencia para incertidumbres.
6. Diseño y manejo de base de datos relacionales.
7. Sistema para desarrollo de interfases, y
8. Generación de aplicaciones de operación independiente.

Además el I/C permite al diseñador de software utilizar inferencia exacta, inexacta y semi-exacta, así como lógica basada en marcos. Para manejar el conocimiento, el Administrador de Conocimiento incluye la facilidad de "master-scope" que rastrea todos los predicados y sus definiciones, permitiéndole al programador ver de inmediato la definición de cada predicado usado en una regla.

El editor tiene dos modos de operación: El de edición y el de compilación. El primero permite construir y editar una aplicación mientras que el segundo permite ejecutarla y depurarla.

B.3 PARADIGMAS DE COMPUTACION

La mayoría de las personas se dan cuenta de que no existe un paradigma de computación único que sea el mejor en todos los casos, así como no existe una estructura de datos que sea la mejor para todos los problemas. Por eso es necesario integrar y combinar varios paradigmas libremente.

En el ambiente del I/C se combinan técnicas tradicionales y avanzadas de IA; además se pueden combinar reglas y marcos con programas tradicionales escritos en Microsoft C o Microsoft Pascal.

El I/C provee facilidades para:

a. Programación basada en reglas:

   Se representa el conocimiento como un conjunto de reglas IF-THEN con premisas y conclusiones lógicas. En formato basado en reglas se cuenta con encadenamiento hacia adelante, hacia atrás, y razonamiento inexacto, que se pueden combinar libremente.

b. Programación basada en marcos:

Cuenta con conocimiento estructurado para expresar relaciones recurrentes. Cada marco incluye un cierto número de ranuras con herencia, predicados agregados, y valores activos.

c. Programación basada en lógica:

Trata con predicados y aserciones lógicas. Incluye un sistema de representación lógica de conocimiento que combina inferencia de encadenamiento hacia adelante y hacia atrás con recursión y atajos.

d. Programación de procedimientos:

Provee interfases para programas en Microsoft C o Pascal. Cualquier predicado dentro del I/C puede incluir una llamada a una función convencional de C o Pascal, y puede pasar o recibir valores de C o Pascal.

B.4 ARQUITECTURA

La construcción de una aplicación con I/C se relaciona con los 6 módulos siguientes:

- Generador Inteligente de Diálogos/ Interfase al Usuario
- Editor/ Base de Conocimiento
- Administrador de Conocimiento
- Compilador
- Ambiente de Ejecución
- Base de Datos Relacional

A su vez, cada uno de estos módulos incluye varios sub-componentes integrados.

B.4.1 LA INTERFASE AL USUARIO

Presenta al usuario menúes, ventanas de diálogos, etc. a medida que se van activando las reglas.

Diseñar la interfase al usuario antes de empezar a desarrollar la lógica de una aplicación puede ser de gran ayuda para organizar los objetivos de esa aplicación, pues hacer esto requerirá que se especifiquen claramente sus entradas y salidas, y así el código de dicha lógica será más eficiente.

Generador Inteligente de Diálogos (IDG)

Es una herramienta gráfica, que puede manejarse con ratón y que permite construir interfases elegantes para el usuario. Con esta herramienta, lo que antes solía requerir semanas se puede hacer ahora en menos de una hora.

Editor (ICE)

Sirve para ingresar conocimiento al I/C. Es un editor orientado a la pantalla con menúes que se despliegan hacia abajo. Puesto que el editor incluye un analizador, detectará cualquier error de sintaxis para corregirlo a tiempo y con facilidad.

Base de Conocimiento

Los hechos, las reglas y las estructuras de conocimiento que se necesitan en un sistema experto se almacenan en forma modular y en algunos archivos. Una base de conocimiento especifica los nombres de los archivos que se utilizan en una aplicación particular.

Administrador de Conocimiento

Maneja hechos, marcos, tablas, y archivos de reglas; además registra la base de conocimiento que se está usando en

determinado momento y maneja sus archivos e inter-relaciones. Incluye la facilidad de Master-scope que se mantiene al tanto de las definiciones de predicados y verifica sintaxis y ortografía.

Compilador (ICP)

Procesa las reglas y marcos antes de su ejecución. Se responsabiliza del trazado y depuración del programa junto con la generación de explicaciones. Después de escribir las reglas en el editor, se puede invocar el compilador para compilarlas y ejecutarlas.

Ambiente de Ejecución (ICX)

Ejecuta la base de conocimiento compilada y además permite depurar la aplicación usando trazados flexibles y facilidades de edición de variables.

Base de Datos Relacional

Permite almacenar hechos o marcos en disco. Sus aplicaciones pueden incluir grandes cantidades de datos que se almacenan en unidades secundarias.

B.5 DESARROLLO DE APLICACIONES

El I/C suministra las herramientas que se necesitan para implementar cualquier clase de programa basado en conocimiento, ya sea que se traten de:

- Aplicaciones convencionales basadas en datos.
- Aplicaciones de bases de datos inteligentes.
- Sistemas de recuperación de información basados en hipertexto.
- O cualquier otra clase de aplicación de software.

El lenguaje de programación del I/C es de muy alto nivel.

B.6 DOCUMENTACION

La documentación del manual se divide en 8 secciones:

Parte I: Introducción

Parte II: Generador Inteligente de Diálogos

Parte III: Compilador Inteligente

Parte IV: Editor

Parte V: Ambiente de Compilación y Ejecución

Parte VI: Tutorial de Sistemas Expertos

Parte VII: Definiciones de Predicados

Parte VIII: Glosario e Indices

GUIA DEL USUARIO

C.1 REQUERIMIENTOS DEL SISTEMA

Para utilizar este sistema experto se necesita de un computador IBM PC/XT/AT o PS/2 (o compatibles), con 640 Kb de memoria RAM como mínimo, que tenga al menos 1 drive de 3½'' para diskettes de alta densidad (1.44 Mb). Y de 2.5 Mb libres en el disco duro. Se requiere MS-DOS 3.3 en adelante.

C.2 INSTALACION

Para la instalación del software se debe ejecutar el archivo INSTALL, que se encuentra en el diskette Nº 1 y seguir las instrucciones que se van presentando a medida que la instalación se completa.

El procedimiento es el siguiente:

- En el ambiente del DOS inserte el diskette Nº 1 en cualquiera de las unidades: A o B (según esté disponible un drive de 1,44 Mb).
- Transfiérase a la unidad en la que insertó el diskette Nº 1 escribiendo A: o B:
- Escriba INSTALL y presione ENTER.

  Automáticamente en la unidad C se creará un directorio llamado DC-10 donde se copiarán todos los archivos de los 10 componentes del sistema experto (C:\DC-10).
- Después que se copien todos los archivos del diskette Nº 1, se exhibirá la frase INSERTE EL DISKETTE Nº 2. Retire el diskette Nº 1 e inserte el Nº 2. Presione ENTER.

- Una vez finalizado el proceso de transferencia de archivos al nuevo directorio DC-10, se presentará la frase: INSTALACION COMPLETA. Entonces puede retirar el diskette Nº 2 y el sistema estará listo para funcionar.

C.3 COMO CORRER EL PROGRAMA

Para que se despliegue la pantalla de presentación de menús, con todas sus 10 opciones, escriba DC-10. En esta pantalla, presione la tecla numérica correspondiente al sistema que desea ejecutar y luego ENTER. Entonces se presentará la barra horizontal del menú respectivo al sistema que haya elegido.

Cada vez que escoja la opción de ABANDONAR-SALIR dentro de cada sistema ejecutable, se volverá a presentar esta pantalla de selección general.

Para salir de esta pantalla al DOS, presione la tecla numérica 0.

Dentro de cada sistema ejecutable puede acceder al sistema de ayudas presionando F1 y salir de él con ESC.

El sistema experto de diagnóstico cuenta con 8 subsistemas expertos de diagnóstico (uno por cada subsistema del sistema eléctrico, según se indica en el manual de mantenimiento), que se invocan separadamente desde la pantalla se selección. El funcionamiento del sistema entero es muy sencillo, y puede manejarse libremente solo con el ratón.

La utilización del sistema de diagnóstico se reduce fundamentalmente a 2 componentes de interfase que interactúan con el usuario:

C.3.1 MENUS DE SELECCION

Después que se ha hecho una selección de entre las 10, en la pantalla de selección general -se cargarán sus archivos correspondientes en la memoria RAM del computador, y acto seguido aparecerá la barra del menú-.

Todos los menús de sistemas de diagnóstico poseen solo dos opciones horizontales: SINTOMAS DE FALLA y ABANDONAR.

La primera opción, una vez escogida, desplegará hacia abajo sus propias posibilidades de selección.

La segunda opción -ABANDONAR- sólo desplegará la selección vertical para SALIR, con la cual se podrá retornar a la pantalla de selección general.

Todas las selecciones del menú, tanto horizontales como verticales se ejecutan ubicándose en ellas con las teclas de flechas y presionando ENTER.

A su vez con la tecla ESC se recogerán las opciones verticales.

Con un click del ratón de acción simple se escogen las opciones horizontales, y con uno de doble acción las verticales.

C.3.2 VENTANAS DE DIALOGOS

Después que se escoja una opción vertical de SINTOMAS DE FALLA, aparecerá en primer lugar una ventana de diálogo, ya sea una de hipertexto o una estándar.

Ventanas de Hipertexto

Una ventana de hipertexto es otra facilidad de interfase al usuario y contiene en su barra superior (en vídeo reverso) la expresión: "HyperText: [Nombre del término marcado como hipertexto]".

Esta ventana contiene información más explícita de los síntomas de falla para el sistema que se ha escogido.

Su característica principal está en la manera cómo se escogen sus opciones:

Para moverse hacia abajo -a la siguiente opción- se lo hace con la tecla TAB. Y hacia arriba -a la opción anterior- con las teclas SHIFT + TAB (manteniendo presionada SHIFT).

Una vez que se ha ubicado en la opción deseada presione ENTER.

Con el ratón, basta hacer un click en la opción deseada para escogerla.

Finalmente, estas ventanas se cierran con un click del ratón en el pequeño rombo del extremo superior izquierdo de la ventana, o con la combinación de teclas ALT + F4 (manteniendo presionada ALT).

Para mirar el texto que se extiende verticalmente fuera de los limites de la pantalla, bastan las teclas de flechas.

Antes de salir a la barra horizontal del menú, para escoger otra opción de SINTOMAS DE FALLA o ABANDONAR. Se sugiere cerrar estas ventanas de hipertexto.

Ventanas Estándar

En estas ventanas, se escogen las opciones ubicándose con las teclas de flechas y presionando F10 (según se indica en la parte inferior izquierda de la ventana).

Para cerrarlas, basta presionar ESC (además de las alternativas que ya se indicó para las ventanas de hipertexto).

Con un click del ratón en la opción deseada se la ubica, y con otro en F10 se la ejecuta.

En las ventanas que aparezca una sola opción de tecla sobre la barra inferior, sea F10 o ESC.- La primera indica que se trata de una ventana de presentación de texto exclusivamente, y que la inferencia seguirá adelante al presionar F10.

La segunda indica que se trata del último nivel del árbol de fallas, a partir del cual sólo se puede retroceder.

Opciones de Ventana

Si el texto de cualquier ventana se extiende más allá de los límites de la pantalla (verticalmente en este diseño), se puede ver el resto de la información utilizando las teclas de flechas únicamente. Además una ventana puede moverse al sitio que se desee. Cada ventana presenta varios componentes en forma de símbolos para uso exclusivo con el ratón.

A continuación se presenta un breve resumen de varios de estos componentes con su opciones respectivas:

OPCION	TECLA(S)	CLICK (CON RATON)
Cerrar	Alt+F4	Símbolo ♦
Mover	Alt+F7	Barra horiz. superior
Alargar (vert.)	Alt+F8	Símbolo ↕
Agrandar (total)	Alt+F10	Símbolo ↑
Achicar (total)	Alt+F9	Símbolo ↓
Listado	Alt+F3	
Invocar ayuda	F1	Símbolo ?

Para mover una ventana, después de mantener abajo Alt y presionar F7, se sitúa la ventana en el lugar deseado con las teclas de flechas y luego se presiona ENTER.

Con el ratón, hay que mantener el click en cualquier lugar de la barra superior -del título de la ventana- mientras se le mueve a la ventana y soltarlo cuando la ventana se encuentra en el lugar deseado.

La opción de LISTADO, es muy interesante porque cuando se la invoca (ALT + F3) presentará en una pequeña ventana una lista completa de todas las ventanas abiertas hasta ese momento. Es decir presenta la trayectoria del árbol de fallas y por ende el nivel de profundidad del árbol hasta ese momento. Presionando ENTER se sale de esta ventana.

Otra cosa relacionada con las ventanas de diálogos es que se puede tener varias ventanas abiertas en la pantalla, pero sólo una estará activa (ventana de trabajo). Para escoger otra ventana de trabajo de las que permanecen en la pantalla, basta hacer un click en cualquier punto del área de esta ventana.

C.3.3 SISTEMA DE AYUDA

Cada uno de los 10 subsistemas cuenta con un sistema de ayuda en-línea. Está disponible a todo tiempo con la tecla F1. (Si se encuentra abierta una ventana, y se tiene un ratón, se invoca el sistema de ayuda con un click en el símbolo ? de su esquina superior derecha).

El sistema de ayuda también se trata de un "árbol de ayudas". Esto significa que una selección se puede ramificar en otras, y así sucesivamente con cada rama (ítem) del árbol hasta que ya no haya más que el texto de la información de dicha rama.

Para cerrar esta ventana de texto, así como las ventanas de selección de ítems, basta presionar ESC.
En la ventana inicial del sistema de ayuda, se profundiza en las ramificaciones sucesivas con la tecla F9, y se mira el texto con F10. En las ventanas sucesivas se consiguen ambas cosas sólo con F10.

Los 2 últimos (de los 10) subsistemas restantes corresponden a una base de datos relacional, y a un sistema para ingreso de memos por los usuarios. Estos dos últimos son los que más interactúan cooperativamente con el usuario, permitiéndole un ingreso flexible de información para enriquecer las capacidades del sistema. En el un caso (la base de datos relacional) la información ingresada podrá ser aprovechada instantáneamente por el usuario, y en el otro (los archivos de memos) será necesaria una intervención natural de reactualización y mantenimiento del sistema.

A continuación se describe en detalle la utilización de estos dos últimos subsistemas, especialmente la base de datos relacional por su gran importancia y autonomía de utilización que ofrece al usuario (cabe señalar que lo que viene a continuación es una transcripción de la información que se encuentra en el sistema de ayudas en-línea, en cada uno de ellos).

C.3.4 BASE DE DATOS RELACIONAL

Esta es una base de datos inteligente que puede usarse para averiguar cualesquier información relacionada con las bitácoras de vuelo del avión DC-10. En principio, desde

febrero del año 90 hasta mayo del 92, y con extensión para seguir añadiendo toda la información que se desee, a fin de llevar un historial lo más completo posible.

Esta base de datos ofrece varias posibilidades de búsqueda de información, según los datos que se ingresen. Así, es posible imponer restricciones convenientes para hacer una clasificación automática de la información que cumpla con ciertos requisitos.

En los siguientes apartados, se tratará en detalle la utilización de cada uno de estos ítems del menú de la base de datos: MIRAR REGISTROS, AGREGAR REGISTROS, MODIFICAR REGISTROS, y SUPRIMIR REGISTROS.

A todo tiempo se puede acceder al sistema de ayuda con F1.

C.3.4.1 MIRAR REGISTROS

El grado hasta el cual se provea la información que el sistema pide determinará las restricciones que se aplicarán a la búsqueda con fines de presentación (lo mismo aplica para fines de modificación y supresión de registros).

Con este propósito se utilizan 6 de los 10 campos de información en los que se ha detallado un registro individual de bitácora de vuelo.

Estos campos son en orden de aparición: FECHA (mm.dd.yy), NÚMERO DE PÁGINA DE BITÁCORA, NÚMERO DE ÍTEM DE LA PÁGINA, NÚMERO DE EXTENSIÓN DEL ÍTEM DE LA PÁGINA (básicamente, se añade un número de extensión a un ítem, en el caso de que la descripción del problema, o del reemplazo/ acción correctiva/ mantenimiento, requiera más de 138 caracteres cada una; o de algún otro caso en el que se desee conservar en un solo ítem cualesquier información relacionada de otros

campos pertenecientes a un mismo registro), NÚMERO DE SERIE DEL DISPOSITIVO dado mantenimiento: REMOVIDO o REEMPLAZADO y, adicionalmente se ha incluído un método normalizado para incorporar información desde una ventana de diálogo denominada CATEGORIAS DE SISTEMAS. Esta ventana contiene alistados los 8 subsistemas del ATA 24 para fines de clasificación automática de fallas por cada uno de estos subsistemas. Así por ejemplo, cuando se desee averiguar toda la información de bitácora que se disponga para el Sistema de Generación AC, se pasa por alto la petición de información para todos los campos excepto para este último. De manera similar se procederá con los otros 7 subsistemas. Antes de presentar esta ventana de diálogo, el sistema le preguntará si desea imponer ésta restricción a la búsqueda.

Si se ingresa sólo la fecha, se presentarán todas las bitácoras que se encuentren con esa fecha.

De manera similar, si sólo se ingresa un número de página de bitácora, se buscarán y exhibirán todos los registros que existan de dicha página.

O si se desea averiguar respecto a un dispositivo en particular, y no se recuerda ninguna otra información pero se cuenta con su número de serie, entonces de dejan en blanco todos los campos y sólo se ingresa el número de serie (en cualquiera de los 2 campos del dispositivo -removido o reemplazado). Con esto se buscará cualquier registro (o registros) de bitácora que contenga información de este dispositivo.

Por otro lado, si no se ingresa un número de extensión de ítem para la exhibición de registros, se presentarán todos los registros que posean extensiones para ese número de ítem.

Por regla general, mientras más completa sea la información que se dé, la búsqueda se volverá más exclusiva; y mientras menos completa sea la información que se dé, la búsqueda se volverá más general.

Si no está seguro de qué información se encuentra en el campo de CATEGORIAS DE SISTEMAS de un registro que desea mirar, es mejor responder que NO, a la sugerencia de incorporar esta información a la búsqueda.
Cuando no se encuentra ningún registro que armonice con la información suministrada para los campos de búsqueda, el sistema simplemente le preguntará si desea seguir mirando más registros. Respondiendo que no, saldrá al menú general.
Además, la exhibición de registros se vuelve cíclica por una respuesta de SI o NO, sin necesidad de volver al menú general para escoger otra vez esta misma opción.

C.3.4.2 AGREGAR REGISTROS

Escogiendo esta opción, se puede agregar más información a la base de datos; para lo cual se presentan en orden cada uno de los 10 campos de información.

No se podrá dejar en blanco los campos de búsqueda: Fecha, número de bitácora, número de ítem, número de extensión de ítem, e incluso una línea de la descripción del problema.

El resto de campos sí se los podrá dejar en blanco si hay necesidad de ello. Esto impide que se agreguen registros sin la identificación básica de búsqueda.

Además, el sistema exhibe una llamada de atención si inadvertidamente se trata de agregar un registro ya existente con los mismos datos de: Fecha, número de bitácora, número de ítem, y número de extensión de ítem; impidiendo así repetir el ingreso de un registro que ya posee esta información básica de búsqueda. Esto ahorra tiempo, memoria y protege la información. En cambio, el sistema le dejará saber que tal registro sólo puede ser modificado.

Finalmente, en el último campo de información de CATEGORIAS DE SISTEMAS, se presentará su ventana de diálogo respectiva, para que escoja e ingrese la información de manera estandarizada sin que tenga que escribir nada. Previamente, se le pedirá una confirmación para incorporar o nó información en este campo. Por eso aquí se debe determinar en primer término a cuál subsistema corresponde el registro que se está ingresando. Si no está seguro de esto, será mejor responder que NO a la petición de agregar información a este campo final -que permanecerá en blanco-. Se lo podrá llenar después utilizando la opción MODIFICAR REGISTROS del menú. Esta información puede ser muy útil en un análisis estadístico de fallas para cada subsistema.

Los campos: DESCRIPCION DEL PROBLEMA, y REEMPLAZO/ACCION CORRECTIVA/MANTENIMIENTO en vista de su naturaleza explicativa ofrecen 2 líneas de 69 caracteres por cada uno. Si este espacio no fuera suficiente, se puede hacer uso de las extensiones de ítems, es decir, agregar nuevos registros con idéntica fecha, número de bitácora, y número de ítem, variando las extensiones solamente. Hacer esto le permitirá continuar la descripción del problema, o el reemplazo, en otro registro con extensiones consecutivas hasta el grado que se desee. Esto también facilitará la búsqueda y observación de estos registros con extensiones de ítem, los cuales se exhibirán uno tras otro, a condición de que se deje en blanco el campo de extensión de ítem (en la opción de MIRAR REGISTROS, que sí permite hacer esto).

El campo OBSERVACIONES de 61 caracteres, también será de utilidad, aunque no se incorpora en la búsqueda.

Finalmente, al igual que en la opción de MIRAR REGISTROS, no hace falta regresar al menú para incorporar un nuevo registro después que ha terminado de agregar ya uno; sino que puede responder que SI a la formulación de si desea agregar un nuevo registro, y continuar así hasta que desee.

C.3.4.3 MODIFICAR REGISTROS

Esta opción permite modificar uno o más campos de información por turno en un solo registro. Se debe ingresar valores no vacíos en los campos de la FECHA, NUMERO DE BITACORA, Y NUMERO DE ITEM; sólo en el campo NUMERO DE EXTENSION DE ITEM, se podrá pasar por alto el ingreso de un valor.

Una vez encontrado el registro que contiene la información de búsqueda suministrada, se lo exhibe. Luego se le pedirá que confirme la modificación de un campo o campos en ese registro. Si su respuesta es negativa el control regresará al menú general; en caso de ser afirmativa mirará una ventana de diálogo con los nombres de todos los campos definidos en los registros de la base de datos. Se puede modificar uno a la vez. El campo seleccionado se presentará para un nuevo ingreso de información. Los campos de doble línea (DESCRIPCION y REEMPLAZO) presentarán ambas líneas otra vez.

Inmediatamente después que ya se ha modificado un campo el sistema le preguntará si desea modificar otro. Si su respuesta es negativa se detiene el proceso cíclico, todas las modificaciones se hacen permanentes y se le notificará esto antes de volver al menú.
Cabe señalar que el registro se exhibe una sola vez al inicio, antes de modificarlo, más no cuando se efectúan las modificaciones. Si desea mirar el registro modificado puede recurrir a la primera opción del menú, MIRAR REGISTROS.

Una notificación importante: Si un campo no está vacío y se desea que lo esté, no basta con no ingresar ninguna información (presionando sencillamente <Enter> cuando aparece el campo para su modificación), en cambio bastará con ingresar un espacio en blanco. Si sólo se presiona <Enter>, no quedará vacío un campo que ya contenga información.

C.3.4.4 SUPRIMIR REGISTROS

Esta es la última opción que ofrece el menú y permite suprimir registros, uno a la vez y previa confirmación, entonces se removerán todos los registros existentes que sean extensiones del mismo ítem, de un solo registro.

Al igual que para modificar registros, para suprimirlos se hace necesario suministrar la información completa de los campos: FECHA, NUMERO DE BITACORA, y NUMERO DE ITEM (el NUMERO DE EXTENSION DE ITEM es opcional). No se podrá suprimir ningún registro sin la información (no vacía) de los 3 primeros campos de búsqueda mencionados.

Una vez encontrado el registro que empareja con la información de búsqueda se lo exhibe, y se le pedirá confirmar su supresión. Si no la confirma, el control regresará al menú principal. Si confirma la supresión, las modificaciones en la base de datos se hacen permanentes y se le notificará de esto. Si no ha ingresado el número de extensión del ítem, se continuará exhibiendo todos los registros que se hallen que sean extensiones del mismo número de ítem suministrado al inicio.

Como se ve, en esta opción también se encuentra un proceso cíclico con una meta definida a semejanza de las otras 3 opciones del menú.

C.3.5 SISTEMA PARA EL INGRESO DE MEMOS

Este es un programa que maneja archivos. Esta ideado con la finalidad de obtener información de parte del usuario, que sirva para la reactualización y mantenimiento del sistema experto de diagnóstico.

Casi en todas las cartas de caza de fallas, se reconoce que pudieran haber síntomas de falla que no se alisten allí. En estos casos se puede escribir un breve memorándum, de acuerdo con el subsistema de que se trate, por medio de este programa (SNL).

Al inicio se presenta un menú general que contiene alistados los 8 subsistemas que se consideran separadamente en el DC-10 MAINTENANCE MANUAL, para caza de fallas. Seleccionando uno de estos, se exhiben indicaciones generales para ingresar la información. El tiempo y la fecha se toman internamente, del reloj digital del computador para incorporarlos al memorándum.

A continuación será necesario ingresar el nombre de quien hace el memorándum, y en pocas palabras el asunto del que se trata —a manera de tema para el mismo—.

Después de esto sí podrá escribir sus observaciones o recomendaciones. Para concluir el ingreso de la información, será necesario que en una línea en blanco escriba la palabra end y presione enter. Con esto se almacenará la información en el archivo (con extensión .MNT) correspondiente al subsistema elegido, y se le exhibirá un acuse de recibo con el contenido del memo ingresado. Presionando <enter> de nuevo podrá volver de nuevo al menú general.

Además se manejan internamente 2 archivos: MEMO y RESPONSE, los cuales se utilizan exclusivamente con propósitos de presentar el memo ingresado.

Será necesario también tomar en cuenta estas sugerencias:

- A pesar de que automáticamente el cursor continúa en la siguiente línea una vez que se han completado 80 caracteres por línea; conviene preveer esto antes de alcanzar el caracter número 80, y presionar <enter> para hacerlo (externamente no hay ninguna diferencia de hacerlo o no así). Previendo esto también se evitará que se queden palabras mal truncadas al final de la línea; en vista de que no se podrá regresar a corregir la línea anterior.
- Finalmente se sugiere utilizar letras mayúsculas para todo, ya que no se trata de un procesador de palabras en sí mismo. Y se tendrá dificultades para poner tildes en las palabras, con lo cual desmejorarán la presentación y comprensión de la información.

Con esta programa y la cooperación del usuario, será posible obtener valiosas sugerencias y nuevos hechos obtenidos de la experiencia, para incorporarlos a la base de conocimiento del presente sistema experto de diagnóstico.

File: ACGEN.TOP

```
% META DE ALTO NIVEL PARA EL SISTEMA DE DIAGNÓSTICO
% EN EL SISTEMA DE GENERACION AC.
AC GEN
```

File: ACGEN.BWD

```
% MANEJO DE SELECCIONES DEL MENU.

AC GEN  IF  RUN-MENU  AC GEN;

MENU-CHOSEN  AC GEN, OPERACION NO PARALELA
        IF   REPAINT-SCREEN  AND
             HYPER-ACTION  OPERACION NO PARALELA  AND  FAIL;

MENU-CHOSEN  AC GEN, OPERACION  PARALELA
        IF   REPAINT-SCREEN  AND
             HYPER-ACTION  OPERACION PARALELA  AND  FAIL;

MENU-CHOSEN  AC GEN, MODO DE TIERRA DUAL
        IF   REPAINT-SCREEN  AND
             HYPER-ACTION  MODO DE TIERRA DUAL  AND  FAIL;

MENU-CHOSEN  AC GEN, POT APU  DISPONIBLE
        IF   REPAINT-SCREEN  AND
             HYPER-ACTION  POT APU DISPONIBLE  AND  FAIL;

MENU-CHOSEN  AC GEN, POT APU EN BUS DE GEN
        IF   REPAINT-SCREEN  AND
             HYPER-ACTION  POT APU EN BUS DE GEN  AND  FAIL;

MENU-CHOSEN  AC GEN, SINTOMA NO ALISTADO
        IF   REPAINT-SCREEN  AND
             HYPER-ACTION  SINTOMA NO ALISTADO  AND  FAIL;

% MANEJO DE HIPERTEXTO

LBODY  CODIGO AD, AD
        IF  (COD OF UNP OP):=AD  AND  DLG APR;

LBODY  CODIGO AP, AP
        IF  (COD OF UNP OP):=AP  AND  DLG BCU;

LBODY  CODIGO DA, DA
        IF  (COD OF UNP OP):=DA  AND  DLG BCU;

LBODY  CODIGO DC, DC
        IF  (COD OF UNP OP):=DCFB  AND  DLG B C U;

LBODY  CODIGO EA, EA
        IF  (COD OF UNP OP):=EAG  AND  DLG EA EG;

LBODY  CODIGO EB, EB
        IF  (COD OF UNP OP):=EB  AND  DLG EA EG;

LBODY  CODIGO EE, EE
          IF  (COD OF UNP OP):=EEH  AND  DLG APU GENER;
```

```
LBODY  CODIGO EJ, EJ
       IF  (COD OF UNP OP):=EJ  AND  DLG FUEL;

LBODY  CODIGO EG, EG
       IF  (COD OF UNP OP):=EAG  AND  DLG EA EG;

LBODY  CODIGO EH, EH
       IF  (COD OF UNP OP):=EEH  AND  DLG APU GENER;

LBODY  CODIGO FB, FB
       IF  (COD OF UNP OP):=DCFB  AND  DLG B C U;

LBODY 'HT', 'HL'
       IF  (COD OF UNP OP):='HL' AND  DLG 'HL';

BODY 'HT', 'HLI'
       IF  (COD OF UNP OP):='HLI' AND  DLG AKW;

% INFERENCIA DE DIAGNOSTICO CON DIALOGOS, MARCOS Y HECHOS.

DLG 'VAR'   IF  OPEN-DIALOG  'VAR'  AND
                RUN-DIALOG   'VAR'  OR
                CLOSE-DIALOG 'VAR';

BUTTON-CHOSEN 'VAR', OK  IF  SAVE 'VAR';

SAVE 'VAR'
       IF  'Y':=(COD OF UNP OP)  AND
           EYE 'Y', 'P', 'VAR'  AND
           'Z':=('P' OF 'VAR')  AND
           SEE 'Y', 'Z', 'VAR', 'V'  AND
           'VAR':= 'V'  AND
           DLG 'VAR';

BUTTON-CHOSEN 'VAR', PERS  IF  SAVE1 'VAR';

SAVE1 'VAR' IF  'Y':=(COD OF UNP OP)  AND
            EYES 'Y', 'VAR', 'V'  AND
            'VAR':= 'V'  AND  DLG 'VAR';
```

File:ACGEN.FBS

```
% ESTE ARCHIVO CONTIENE LOS ARBOLES DE FALLA CODIFICADOS
% COMO HECHOS, EN BASE A LOS PREDICADOS: EYE, EYES y SEE.

%                                          CODIGO AA
EYES AA, AA, GEN

%                                          CODIGO AB
EYES AB, AB, GCU

%                                          CODIGO AC
EYE AC, SEL BY, AC
SEE AC, NEGRO, AC, GR
SEE AC, AMARILLO, AC, GCU
```

```
%                                          CODIGO AE
EYE AE, SEL 1, AE
SEE AE, ENCENDIDA, AE, AE2
SEE AE, APAGADA, AE, BTR

EYES AE, BTR, SUG

EYE AE, SEL 2, AE2
SEE AE, SI, AE2, AE3
SEE AE, NO, AE2, G C U

EYES AE, G C U, USGR

EYE AE, SEL 1, AE3
SEE AE, ENCENDIDA, AE3, AE4
SEE AE, APAGADA, AE3, GCU

EYE AE, SEL 1, AE4
SEE AE, ENCENDIDA, AE4, US
SEE AE, APAGADA, AE4, GR

%                                          CODIGO AG
EYE AG, SEL 3, AG
SEE AG, UNIDAD DE CONTROL DEL GENERADOR, AG, GCU
SEE AG, GENERACION Y DISTRIBUCION, AG, AE2
SEE AG, GENERADOR...O NINGUNO, AG, AE2

EYE AG, SEL 2, AE2
SEE AG, SI, AE2, AG2
SEE AG, NO, AE2, G E N
EYES AG, G E N, GCU

EYE AG, SEL 1, AG2
SEE AG, ENCENDIDA, AG2, GEN
SEE AG, APAGADA, AG2, GCU

%                                          CODIGO AKW
EYE AKW, SEL 2, AKW
SEE AKW, SI, AKW, AKW2
SEE AKW, NO, AKW, SCSD

EYES AKW, SCSD, LC

EYE AKW, SEL FREQ, AKW2
SEE AKW, CORRECTA, AKW2, LC
SEE AKW, DEFECTUOSA, AKW2, GOV

EYES AKW, GOV, CSD

%                                          CODIGO AL
EYE AL, SEL 4, AL
SEE AL, CORREGIDA, AL, NF
SEE AL, NO CORREGIDA, AL, AL1

EYE AL, SEL 5, AL1
SEE AL, 115 VAC, AL1, AGSTR
SEE AL, 0   VAC, AL1, AL2
```

```
EYE AL, SEL 5, AL2
SEE AL, 115 VAC, AL2, GSPR
SEE AL, Ø   VAC, AL2, AGSTRC

%                                        CODIGO AM
EYE AM, SEL 2, AKW
SEE AM, SI, AKW, AM1
SEE AM, NO, AKW, ABSR

EYES AM, ABSR, AC RCCB

EYE AM, SEL 1, AM1
SEE AM, ENCENDIDA, AM1, AM2
SEE AM, APAGADA, AM1, NFU

EYE AM, SEL 5, AM2
SEE AM, 115 VAC, AM2, ABS
SEE AM, Ø   VAC, AM2, AC RCCB

%                                        CODIGO AN
EYE AN, SEL 6, AN
SEE AN, APAGADAS, AN, NA
SEE AN, ENCENDIDAS, AN, AN1

EYE AN, SEL 7, AN1
SEE AN, 22 A 34 VDC, AN1, BTR2
SEE AN, CERO    VDC, AN1, EPR

%                                        CODIGO AQ
EYES AQ, AQ, RCCB ICU

%                                        CODIGO AR
EYES AR, AR, RCCB ICU

%                                        CODIGO AS
EYE AS, SEL 2, AKW
SEE AS, SI, AKW, AS1
SEE AS, NO, AKW, AC VM

EYES AS, AC VM, FUSE

EYE AS, SEL 8, AS1
SEE AS, EL FRECM DEFLEXIONA Y EL VOLTM NO, AS1, AC VOLT
SEE AS, EL VOLTM DEFLEXIONA Y EL FRECM NO, AS1, AC FRE
SEE AS, AMBOS MEDIDORES NO DEFLEXIONAN, AS1, VFF

%                                        CODIGO AU
EYE AU, SEL 2, AKW
SEE AU, SI, AKW, AU1
SEE AU, NO, AKW, G C U

EYES AU, G C U, GEN

EYE AU, SEL 2, AU1
SEE AU, SI, AU1, GCU
SEE AU, NO, AU1, GEN
```

```
*                                        CODIGO AV BE
EYE AVBE, SEL 2, AKW
SEE AVBE, SI, AKW, AV BE 1
SEE AVBE, NO, AKW, L C

EYES AVBE, L C, SER CSD

EYE AVBE, SEL FREQ, AV BE 1
SEE AVBE, CORRECTA, AV BE 1, LC
SEE AVBE, DEFECTUOSA, AV BE 1, SER CSD

*                                        CODIGO AT
EYE AT, SEL 2, AKW
SEE AT, SI, AKW, AT1
SEE AT, NO, AKW, G C U

EYES AT, G C U, LC GEN GR

EYES AT, AT1, AT2

EYE AT, SEL 9, AT2
SEE AT, SOLO LECT DE VOLTAJE ES DEFECTUOSA, AT2, AT21
SEE AT, GEN OFF ILUM Y LECS DE VOL Y FRE OK, AT2, AT22
SEE AT, GEN OFF ILUM Y LECS DE VOL Y FRE MAL, AT2, AT23

EYE AT, SEL FREQ, AT21
SEE AT, CORRECTA, AT21, GCU
SEE AT, DEFECTUOSA, AT21, GEN

EYE AT, SEL 1, AT22
SEE AT, ENCENDIDA, AT22, AT221
SEE AT, APAGADA, AT22, NAC

EYE AT, SEL 1, AT221
SEE AT, ENCENDIDA, AT221, GR
SEE AT, APAGADA, AT221, GCU

EYE AT, SEL FREQ, AT23
SEE AT, CORRECTA, AT23, LC
SEE AT, DEFECTUOSA, AT23, SER CSD

*                                        CODIGO BA
EYE BA, SEL 2, AKW
SEE BA, SI, AKW, BA1
SEE BA, NO, AKW, L C

EYES BA, L C, SER CSD

EYE BA, SEL 10, BA1
SEE BA, CORRECTA, BA1, LC
SEE BA, AUN FLUCTUA, BA1, GOV

EYES BA, GOV, CSD

*                                        CODIGO BD
EYE BD, SEL 2, AKW
SEE BD, SI, AKW, BD1
SEE BD, NO, AKW, BTR OBS
```

```
EYES BD, BTR OBS, OBS BTR

EYES BD, OBS BTR, GCU

EYE BD, SEL 11, BD1
SEE BD, APAGADA, BD1, DUPL
SEE BD, AUN ENCENDIDA, BD1, BTR

EYES BD, BTR, SUG

*                                       CODIGO BF
EYES BF, BF, GCU

*                                       CODIGO DC FB
EYES DCFB, B C U, TRAN

*                                       CODIGO DB
EYE DB, SEL 2, AKW
SEE DB, SI, AKW, DB1
SEE DB, NO, AKW, TRP

EYES DB, TRP, GCU

EYE DB, SEL 1, DB1
SEE DB, ENCENDIDA, DB1, NA
SEE DB, APAGADA, DB1, DB2

EYE DB, SEL 07, DB2
SEE DB, 22 A 34 VDC, DB2, TR
SEE DB, CERO    VDC, DB2, OBSERV

*                                       CODIGO EA EG
EYE EAG, SEL 1, EA EG
SEE EAG, APAGADA, EA EG, SLA
SEE EAG, ENCENDIDA, EA EG, APU GCU

EYES EAG, SLA, OBS PLUG

*                                       CODIGO EB
EYE EB, SEL 1, EA EG
SEE EB, APAGADA, EA EG, SLAVE
SEE EB, ENCENDIDA, EA EG, APU GCU

*                                       CODIGO EC
EYE EC, SEL 1, EC
SEE EC, APAGADA, EC, APU GCU
SEE EC, ENCENDIDA, EC, APU GEN

*                                       CODIGO EF
EYE EF, SEL 12, EF
SEE EF, SISTEMA No 1, EF, APU GENER
SEE EF, SISTEMAS No 2 y 3, EF, BCU APU

*                                       CODIGO ED
EYE ED, SEL 02, ED
SEE ED, SI, ED, GCU
SEE ED, NO, ED, APU GENCU
```

```
EYES ED, APU GENCU, SLAVE

%                                        CODIGO EJ
EYES EJ, FUEL, APU BOX

%                                        CODIGO FA
EYE FA, SEL 01, FA
SEE FA, APAGADA, FA, B C U
SEE FA, ENCENDIDA, FA, FA2

EYES FA, B C U, SYS GCU

EYES FA, SYS GCU, SYS BTR

EYES FA, FA2, FA21

EYES FA, FA21, FA3

EYE FA, SEL 13, FA3
SEE FA, EL BUS DEL GEN REPORTADO TIENE POT, FA3, CIA CKT
SEE FA, EL BUS 1 DE GEN NO TIENE POTENCIA, FA3, GCU APU
SEE FA, BUSES 2 o 3 DE GEN SIN POTENCIA, FA3, BCU

%                                        CODIGO FC
EYE FC, SEL 2, AKW
SEE FC, SI, AKW, FC1
SEE FC, NO, AKW, GCU REP

EYES FC, GCU REP, FOUR

EYE FC, SEL FREQ, FC1
SEE FC, CORRECTA, FC1, GCU
SEE FC, DEFECTUOSA, FC1, FC2

EYE FC, SEL 14, FC2
SEE FC, OK Y LUZ DE APU EN USO NO ILUM, FC2, GCU APU
SEE FC, FALLOSA Y LUZ APU EN USO ILUM, FC2, A P R

EYES FC, A P R, WIRE


File: ACGEN.FRM

% ESTE ARCHIVO CONTIENE UN MARCO QUE PROPORCIONA
% UNA VARIABLE GLOBAL EN LA RANURA COD.

Frame: UNP OP
Parent: Thing
Slot: COD    Value:
```

NOTA: EN LAS CARTAS DE TROUBLE-SHOOTING QUE SE ADJUNTAN PARA ESTE SISTEMA DE GENERACION AC, SE PUEDE TENER UNA COMPROBACION DEL SISTEMA DE DIAGNOSTICO.

File: AIRLOG.TOP

```
% METAS DE MAS ALTO NIVEL PARA LA BASE DE DATOS INTELIGENTE.

% ESTA META SE LA EJECUTA UNA SOLA VEZ AL PRINCIPIO PARA
% CREAR EL ARCHIVO DE BASE DE DATOS RELACIONAL  *.DAI
% CORRESPONDIENTE:
% create first

  esdb
```

File: AIRLOG.BWD

```
% ESTE ARCHIVO DISEÑA Y MANEJA UNA BASE DE DATOS RELACIONAL
% PARA LAS BITACORAS DE VUELO DE UN AVION DC-10, UTILIZANDO
% RAZONAMIENTO CON REGLAS HACIA ATRAS.

% LOGICA DE APERTURA Y CIERRE DEL MENU:

esdb    if    run-menu ESDB and APP-CLOSE ESDB;

APP-STARTUP ESDB
IF
open-relations AIRLOG.DAI, 30000, 2048  and
REPAINT-SCREEN;

APP-CLOSE ESDB
IF
close-relations AIRLOG.DAI;

% LOGICA DEL MENU PARA AGREGAR, MIRAR, SUPRIMIR Y MODIFICAR
% REGISTROS.

MENU-CHOSEN ESDB, AGREGAR REGISTROS
if  check db and
    database and retmenu and
    fail;

check db
if
retract database or  true;

MENU-CHOSEN ESDB, MIRAR REGISTROS
if  show SRMset and retmenu and  fail;

MENU-CHOSEN ESDB, SUPRIMIR REGISTROS
if  not del SRMrec and  retmenu and  fail;

MENU-CHOSEN ESDB, MODIFICAR REGISTROS
if  mod SRMrec and retmenu and  fail;

retmenu
if      show-message '  ** REGRESA AL MENU ** .';

% ESTA REGLA DEBE EJECUTARSE UNA SOLA VEZ, COMO UNA DE LAS METAS
% DE ALTO NIVEL, EN EL ARCHIVO .TOP; SIRVE PARA CREAR E INICIALI-
% ZAR LA BASE DE DATOS.
```

```
create first
if
load-factbase AIRLOG.FBS and
create-relations AIRLOG.DAI, 2048;

% ESTA SECCION AÑADE NUEVOS REGISTROS

database
if
show-message 'INGRESE UN NUEVO REGISTRO A CONTINUACION.' and
read for input 'Dt','Lg','IT','EX','D1','D2','Dv1','Dv2','SN','S2','Rm2'  and
decide knd 'Knd' and
assert SRMset  'Dt','Lg','IT','EX','D1','D2','Dv1','Dv2','SN','S2','Rm2','Knd'  and
commit-relations AIRLOG.DAI and  continues and  no-backtrack or

database and no-backtrack;

read for input 'Dt','Lg','IT','EX','D1','D2','Dv1','Dv2','SN','S2','Rm2'
if
read-non-empty  'Dt',8 , 'FECHA (MM.DD.YY)' and
read-non-empty  'Lg', 6, 'No. DE BITACORA ' and
read-non-empty  'IT', 2, 'ITEM NUMERO     ' and
read-non-empty  'EX', 2, 'ITEM EXTENSION  ' and
not non repeat 'Dt','Lg','IT','EX' and
read-non-empty  'D1', 69, 'DESCRIPCION DEL PROBLEMA' and
read-input  'D2', 69, 'DESCRIPCION DEL PROBLEMA-CONTINUACION' and
read-input 'Dv1',69,'REEMP/ACCION CORRECTIVA/MANTENIMIENTO' and

read-input 'Dv2',69,'REEMP/ACCION CORRECTIVA/MANTENIM.-CONT' and

read-input  'SN', 15, '# SERIE REMOV' and
read-input  'S2', 15, '# SERIE REEMP' and
read-input  'Rm2', 61, ' OBSERVACIONES ' ;

non repeat 'Dt','Lg','IT','EX'
if
'F':= 'Dt' and  'B':= 'Lg' and
'T':= 'IT' and  'E':= 'EX' and
SRMset 'F','B','T','E' and
pop-text

(YA EXISTE UN REGISTRO CON LA
MISMA FECHA, PAG, E ITEM.EXT;
LO SIENTO, NO PUEDE AGREGARLO
DE NUEVO, SOLO MODIFICARLO.
Presione <RETURN>.) and
assert database;


decide knd 'Knd'
if
read-yes 'AGREGA CAMPO DE CATEGORIAS?' and
acq knd 'Knd' and no-backtrack or
no-backtrack;
```

```
acq knd 'Knd'
if
open-dialog  kind and
run-dialog   kind and
close-dialog kind and
'Knd':= (Category of Kind);

BUTTON-CHOSEN kind, ok
if
(Category of Kind):= (selk of kind) and  fail;

continues
if
read-no 'DESEA AGREGAR MAS REGISTROS? ';

% ESTA SECCION EXHIBE REGISTROS.

show SRMset
if
pop-text
(INGRESE UNA O VARIAS DE LAS SIGUIENTES
ESPECIFICACIONES PARA LOS REGISTROS QUE
DESEA MIRAR.

Presione <RETURN> para continuar.) and

readings 'Dt','Lg','IT','EX','SE' and
decide knd 'Knd' and
SRM 'Dt','Lg','IT','EX','SE','Knd' and
continuous and no-backtrack or
show SRMset and  no-backtrack;

readings 'Dt','Lg','IT','EX','SE'
if
read-input  'Dt',8 , 'FECHA (MM.DD.YY)' and
read-input  'Lg', 6, 'No. DE BITACORA ' and
read-input  'IT', 2, 'ITEM NUMERO     ' and
read-input  'EX', 2, 'ITEM EXTENSION  ' and
read-input  'SE', 15, 'NUMERO DE SERIE ';

both series 'SE','SN','S2'
if
bound 'SE' and  'SN':= 'SE' or
bound 'SE' and  'S2':= 'SE' or
not bound 'SE';

SRM 'Dt','Lg','IT','EX','SE','Knd'
if
both series 'SE','SN','S2' and
view record 'Dt','Lg','IT','EX','SN','S2','Knd' and
ask for return or  no-backtrack;


view record 'Dt','Lg','IT','EX','SN','S2','Knd'
if
SRMset  'Dt','Lg','IT','EX','D1','D2','Dv1','Dv2','SN','S2','Rm2','Knd' and
```

```
pop-text {^      ~** REGISTRO DE BITACORA DE VUELO, AVION   DC-10, ATA-24 **~
                   ~** ECUATORIANA DE AVIACION **~
 ~FECHA:~ %s
 ~BITACORA:~ %g
 ~ITEM #.EXT:~ %g.%g
 ~DESCRIPCION DEL PROBLEMA:~
 %s
 %s
 ~REEMPLAZO/ACCION CORRECTIVA/MANTENIMIENTO:~
 %s
 %s
 ~No. SERIE REMOVIDO:~ %g
 ~No. SERIE REEMPLAZADO:~ %g
 ~OBS:~ %s
 ~ATA 24.-CATEGORIA:~ %s

               **      PRESIONE      ~<RETURN>~       PARA      CONTINUAR
**},'Dt','Lg','IT','EX','D1','D2','Dv1','Dv2','SN','S2','Rm2','Knd';

ask for return
if
no-backtrack and  fail ;

continuous
if
read-no 'DESEA MIRAR MAS REGISTROS?';

% ESTA SECCION SUPRIME REGISTROS UNO POR UNO.

del SRMrec
if
pop-text

{ ~NOTA~: SOLAMENTE SE PODRA SUPRIMIR UN ITEM
 COMPLETO DE UN REGISTRO A LA VEZ.- POR EL
 CASO DE PROTECCION DE LA INFORMACION.
 - A CONTINUACION, SERA NECESARIO INGRESAR
 AL MENOS 3 DE LAS CONDICIONES QUE SE SOLI-
 CITAN (FECHA, BITACORA e ITEM / EXT).

 Presione <Return> para continuar.} and

read for del 'Dt','Lg','IT','EX' and
no-backtrack and
extension 'Dt','Lg','IT','EX','SN','S2','Knd';

read for del 'Dt','Lg','IT','EX'
if
read-non-empty  'Dt',8 , 'FECHA (MM.DD.YY)' and
read-non-empty  'Lg', 6, 'No. DE BITACORA ' and
read-non-empty  'IT', 2, 'ITEM NUMERO     ' and
read-input  'EX', 2, 'ITEM EXTENSION  ';

extension 'Dt','Lg','IT','EX','SN','S2','Knd'
if
view record 'Dt','Lg','IT','EX','SN','S2','Knd' and
ask for del 'Dt','Lg','IT','EX';
```

```
ask for del 'Dt','Lg','IT','EX'
if
read-yes "ESTA SEGURO QUE DESEA SUPRIMIRLO?" and
no-backtrack and
retract SRMset 'Dt','Lg','IT','EX' and
commit-relations AIRLOG.DAI  and
show-message "  REGISTRO(S) DEL MISMO     ITEM SUPRIMIDO(S). " and  fail;

% ESTA SECCION MODIFICA CAMPOS EN LOS REGISTROS.

mod SRMrec
if
pop-text

{ ~NOTA~: .SOLAMENTE SE PODRA MODIFICAR UN RE-
 GISTRO A LA VEZ -EN UNO O VARIOS CAMPOS.
 A CONTINUACION, SERA NECESARIO INGRESAR AL
 MENOS 3 DE LAS CONDICIONES QUE SE  SOLICI-
 TAN (FECHA, BITACORA e ITEM / EXT).

 Presione <Return> para continuar.} and

read for del 'Dt','Lg','IT','EX'  and
show-message "     MIRE SU REGISTRO          CORRESPONDIENTE.   " and

view record 'Dt','Lg','IT','EX','SN','S2','Knd' and
get-identity 'id'  and
read-yes "DESEA MODIFICAR CAMPO(S)?" and
repeat  and
sel field 'sf', 'valf', 'sf1', 'valf1' and
modifying 'id', 'sf', 'valf', 'sf1', 'valf1' and
read-no "MODIFICA UN NUEVO CAMPO?"  and
commit-relations AIRLOG.DAI  and
show-message "** REGISTRO MODIFICADO **" and no-backtrack;

modifying 'id', 'sf', 'valf', 'sf1', 'valf1'
if
bound 'sf1' and
relation-put-field 'id', 'sf1', 'valf1' and
fail or
relation-put-field 'id', 'sf', 'valf';


sel field 'sf', 'valf', 'sf1', 'valf1'
if
open-dialog  rfield  and
run-dialog   rfield  and
close-dialog rfield  and  fail  or
'sg':=~(Field of Kind) and
field sel 'sg', 'sf', 'valf' and  no-backtrack  or
'sg':= (Field of Kind) and
field selec 'sg', 'sf', 'valf', 'sf1', 'valf1' and
no-backtrack;


BUTTON-CHOSEN  rfield, ok
if
(Field of Kind):= (fsel of rfield)  and  fail;
```

```
field sel FECHA, 'sf', 'Dt'
if
'sf':= Dt  and
read-input  'Dt',8 , "FECHA (MM.DD.YY)";

field sel BITACORA, 'sf', 'Lg'
if
'sf':= Log  and
read-input  'Lg', 6, "No. DE BITACORA ";

field selec ITEM.EXT, 'sf', 'IT', 'sf1', 'EX'
if
'sf':= IT  and  'sf1':= EX and
read-input  'IT', 2, "ITEM NUMERO     "  and
read-input  'EX', 2, "ITEM EXTENSION  ";

field selec PROBLEMA, 'sf', 'D1', 'sf1', 'D2'
if
'sf':= D1  and  'sf1':= D2  and
read-input  'D1', 69, "DESCRIPCION DEL PROBLEMA"  and
read-input  'D2', 69, "DESCRIPCION DEL PROBLEMA-CONTINUACION";

field selec REEMPLAZO, 'sf', 'Dv1', 'sf1', 'Dv2'
if
'sf':= Dv1  and  'sf1':= Dv2  and
read-input  'Dv1', 69, "REEMPLAZO/ACCION CORRECTIVA/MANTENIMIENTO" and
read-input  'Dv2', 69, "REEMPLAZO/ACCION CORRECTIVA/MANTENIMIENTO-CONT";

field sel No.SERIE RMOV, 'sf', 'SN'
if
'sf':= SN' and
read-input  'SN', 15, "# SERIE REMOV";

field sel No.SERIE REEM, 'sf', 'S2'
if
'sf':= S2  and
read-input  'S2', 15, "# SERIE REEMP";

field sel OBSERVACIONES, 'sf', 'Rm2'
if
'sf':= Rm2  and
read-input  'Rm2', 61, " OBSERVACIONES ";

field sel CATEGORIA, 'sf', 'Knd'
if
'sf':= Knd  and  acq knd 'Knd';
```

File: AIRLOG.FBS

```
% AQUI SE DEFINE LA RELACION SRMset Y SUS INDICES
% EN LA MODALIDAD DE PREDICADOS, DE AQUI SE PUEDE
% DEFINIRLA FACILMENTE EN LA MODALIDAD DE MARCOS.

relational SRMset

Dt , String, 8 is-a-field SRMset
Log , Real, 1 is-a-field SRMset
IT , Integer, 1 is-a-field SRMset
EX , Integer, 1 is-a-field SRMset
D1 , String, 69 is-a-field SRMset
D2 , String, 69 is-a-field SRMset
Dv1 , String, 69 is-a-field SRMset
Dv2 , String, 69 is-a-field SRMset
SN , Real, 1 is-a-field SRMset
S2, Real, 1 is-a-field SRMset
Rm2, String, 61 is-a-field SRMset
Knd, String, 20 is-a-field SRMset

% INDICES

Dat is-an-index SRMset
Dat has-index-field Dt

Logo is-an-index SRMset
Logo has-index-field Log

IDX is-an-index SRMset
IDX has-index-field IT

SN1 is-an-index SRMset
SN1 has-index-field SN

SN2 is-an-index SRMset
SN2 has-index-field S2

KIND is-an-index SRMset
KIND has-index-field Knd
```

File: KIND.FRM

```
% ESTE MARCO CONTIENE DOS VARIABLES GLOBALES.

Frame: Kind
Parent: Thing
Slot: Category   Value:
Slot: Field      Value:
```

File: SNL.TOP

```
% META DE ALTO NIVEL PARA EL SISTEMA DE INGRESO DE MEMOS
% PARA LOS SINTOMAS NO ALISTADOS.

SNL
```

File: SNL.BWD

```
% ESTE PROGRAMA AGREGA Y MANEJA ARCHIVOS DE MEMOS PARA
% CADA UNO DE LOS SUBSISTEMAS DE DIAGNOSTICO.

% LOGICA PARA EL MENU Y SUS ITEMS.

SNL IF RUN-MENU SNL;

MENU-CHOSEN SNL, CSD GENERADOR
        if  repaint-screen  and cls  and
            sys files GENDRIVE.MNT  and  fail;

MENU-CHOSEN SNL, GENERACION AC
        if  repaint-screen  and cls  and
            sys files ACGEN.MNT  and  fail;

MENU-CHOSEN SNL, POT EMER IZQ
        if  repaint-screen  and cls  and
            sys files LEFT.MNT  and  fail;

MENU-CHOSEN SNL, POT EMER DER
        if  repaint-screen  and cls  and
            sys files ADG.MNT  and  fail;

MENU-CHOSEN SNL, GENERACION DC
        if  repaint-screen  and cls  and
            sys files DCGEN.MNT  and  fail;

MENU-CHOSEN SNL, POT EXTERIOR
        if  repaint-screen  and cls  and
            sys files EXTPWR.MNT  and  fail;

MENU-CHOSEN SNL, POT COCINAS
        if  repaint-screen  and cls  and
            sys files GALLEY.MNT  and  fail;

MENU-CHOSEN SNL, DISTRIB CARGA
        if  repaint-screen  and cls  and
            sys files LDIS.MNT  and  fail;

% MANEJO DE ARCHIVOS.

sys files 'Fil'
if
explain example 'Fil' and
generate memo 'Fil','Date','Tim','Name','Matter','M' and
generate response RESPONSE,'Date','Tim','Name','Matter','M' and
show file RESPONSE;
```

```
explain example 'Fil'
        If
write and
write "A continuación, sírvase ingresar la información pertinente en el" and
write "formato de memorando que se le presenta." and
write "Tenga presente también que una vez escrita una línea, no se po-" and
write "drá regresar a corregirla." and
write and
write "Su información y observaciones se almacenarán en el archivo " and
write 'Fil', " para el mantenimiento correspondiente del Sistema" and
write "Experto." and
write and
ask for return and cls ;


generate memo 'Fil','Date','Tim','Name','Matter','M'
        If
        get information 'Date','Tim','Name','Matter' and
        generate file   'Fil',memo,'Date','Tim','Name','Matter','M';


show date 'Date'
if
date 'Ye', 'Mo', 'Day', 'DW' and
concatenate 'Mo',":",'Day',":",'Ye',":",'DW','Date';


show time 'Tim'
if
time 'Ho', 'Mi', 'Se' and
concatenate 'Ho',":",'Mi',":",'Se','Tim';

get information 'Date','Tim','Name','Matter'
If
show date 'Date' and
show time 'Tim' and
write and
write-word "Ingrese su nombre: " and
read 'Name' and
write-word "Asunto general de que se trata: " and
read 'Matter' and
write ;


generate file 'Fil',memo,'Date','Tim','Name','Matter','M'
If
write and
write "Ahora puede escribir el texto de la información que posee." and
write "Además, escriba 'end' sobre una línea en blanco y presione" and
write "<enter> para concluir el ingreso de su información." and
write "-Gracias-." and
write and
Open-Append 'Fil' and
Open-Write memo and
head file 'Fil','Date','Tim','Name','Matter' and
head file memo,'Date','Tim','Name','Matter' and  'M':=0  and
   do read a line 'Fil', memo, 'Line','M'
   until Line 'Line' and
```

```
foot file 'Fil','Name' and
foot file memo,'Name' and
close-file 'Fil' and
close-file memo;


head file 'File','Date','Tim','Name','Matter'
        if
        file-write 'File', "FECHA:  ", 'Date' and
        file-write 'File', "HORA:  ", 'Tim' and
        file-write 'File', "DE: ", 'Name' and
        file-write 'File', "PARA: ", "ING. MANTENIMIENTO" and
        file-write 'File', "ASUNTO: ", 'Matter' and
        file-write 'File', " " ;


foot file 'File','Name'
        if
        file-write 'File', " " and
        file-write 'File', "Atentamente, " and
        file-write 'File', 'Name', "." and
        file-write 'File', " " ;


read a line 'Fil', memo, 'Line','M'
        if
     -- read 'Line' and
        check empty 'Line' and
        write to file 'Fil',memo,'Line','M';

check empty 'Line'
        if
        replace value 'Line' and 'Line' := " " and
           no-backtrack
   or   no-backtrack ;

replace value 'Line'
        if
        not bound 'Line'
   or
        'Line' = "NO-RESPONSE" ;

write to file 'Fil',memo,'Line','M'
        if
        Line 'Line' and no-backtrack
    or
        file-write 'Fil', 'Line' and
        file-write memo, 'Line' and
        'M':='M'+ 1;

Line 'Line'
    if
    'Line'= end  or  'Line'= END  or  'Line'= End;

generate response 'resp', 'Date','Tim','Name','Matter','M'
        if
        Open-Write 'resp' and
        file-write 'resp', "FECHA:  ", 'Date' and
```

DETALLE DEL ATA 24 Y CARTAS PARA CAZA DE FALLAS DEL SISTEMA DE GENERACION AC

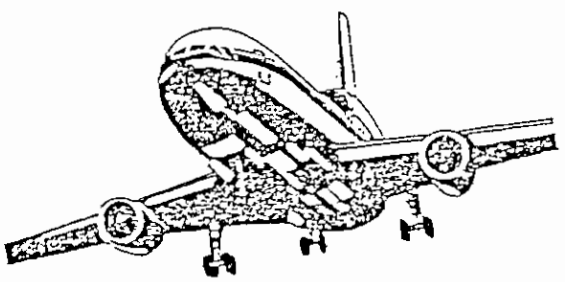
McDonnel Douglas

DC-10

swissair

ATA 100 BREAKDOWN

MAI, IPC, WDM



Documentcenter Swissair TZVF

Key: EFZATA / June 1/91

Figur: F2-01-001C

SWISSAIR SAS

- 30 AUTO THROTTLE/SPEED CONTROL (AT/SC)
- 31 Auto Throttle/Speed Control
- 32 Auto Throttle Warning and Advisory
- 33 Stall Warning
- 34 Spoiler Position Switch and Sensor
- 40 PERFORMANCE AND FAILURE ASSESSMENT MONITOR
- 41 Performance and Failure Assessment Monitor (PAFAM)
- 50 AUTOLAND LOGIC ANALYZER
- 51 Autoland Logic Analyzer

23 COMMUNICATION

- 00 GENERAL
- 10 HIGH FREQUENCY
- 11 High Frequency System
- 19 Calsel Encoder-Calsel System
- 20 VERY HIGH FREQUENCY (VHF)
- 21 VHF
- 22 Selective Calling (SELCAL)
- 23 VHF Satellite Comm (WDM, IPC)
- 30 PASSENGER ACCPESS & ENTERTAINMENT
- 31 Passenger Address
- 32 Recorded Announcent
- 33 Passenger Entertainment/Service Multiplex Syst.
- 34 Portable Communications
- 39 Passenger Entertainment System (Television)
- 40 INTERPHONE
- 41 Service Interphone
- 42 Galley Intercom
- 50 AUDIO INTEGRATING
- 51 Flight Interphone
- 52 Radio Isolation CTS Flt Interphone (WDM)
- 60 STATIC DISCHARGING
- 70 VOICE RECORDERS
- 71 Cockpit Voice Recorder
- 90 CLOSED CIRCUIT TELEVISION

24 ELECTRICAL POWER

- 00 GENERAL
- 10 GENERATOR DRIVE
- 11 Constant Speed Drive and Disconnect Element - Change and Scavenge Oil
- 12 CSD Oil Cooling
- 13 CSD Oil Temperature Indicator (WDM,IPC)
- 14 CSD Oil Pressure Low Lights (WDM,IPC)
- 15 CSD Load Control
- 20 AC GENERATION
- 21 AC Generation and Control
- 22 AC Loadmeters (WDM,IPC)
- 23 AC Protection (WDM,IPC)
- 24 Left Emergency Power (Inverter)
- 25 Right Emergency Power (Air Driven)
- 30 DC GENERATION
- 31 DC Power
- 32 DC Tie Bus Control (WDM,IPC)
- 33 DC Loadmeters (WDM,IPC)
- 34 Emergency DC Power
- 40 EXTERNAL POWER
- 41 Main External Power and Control
- 42 Galley Power and Control (External/Internal)
- 50 ELECTRICAL LOAD DISTRIBUTION
- 51 AC Power Distribution (WDM,IPC)
- 52 DC Power Distribution (WDM,IPC)
- 53 Ground Service Distribution and Control
- 54 Galley Power and Control (External/Internal)
- 55 Electrical Outlets
- 56 Spare Wires (WDM)
- 57 Circuit Breakers - AC (WDM)
- 58 Circuit Breakers - DC (WDM)
- 59 Flight Compartment Panel Grounding (WDM)

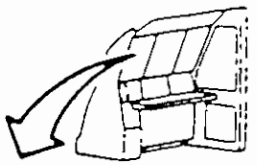
25 EQUIPM./FURNISH.

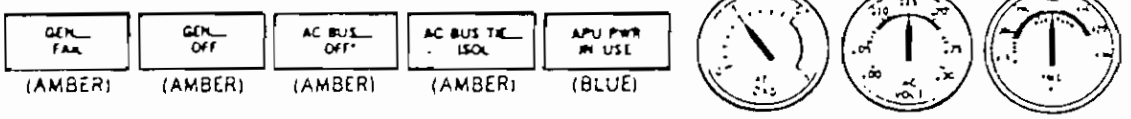
- 00 GENERAL
- 10 FLIGHT COMPARTMENT
- 11 Flight Compartment Seats
- 12 Flight Compartment Accessories
- 13 Flight Compartment Stowage Provis.
- 15 Flight Compartment Lining
- 16 Flight Compartment Floor Covering
- 17 Flight Compartment Insulation
- 20 PASSENGER COMPARTMENT
- 21 Passenger and Attendant Seats
- 22 Forward Cabin Outboard Utilities (WDM, IPC)
- 23 Passenger Compartment Side Panels and Window Shades
- 24 Pass. Compartment Ceiling Panels
- 25 Pass. Compartment Floor Covering
- 26 Movable Partitions
- 27 Passenger Compartment Stowage
- 28 Pass. Compartment Insulation
- 29 Stretchers
- 30 BUFFET/GALLEY
- 31 Service Center
- 32 Lower Galley
- 33 Coffee Bar
- 34 Modules and Carts (IPC)
- 36 Galley Lifts
- 37 Forward Galley Refrigeration System (WDM, IPC)
- 39 Galley Ceiling Panels
- 40 LAVATORIES
- 41 Lavatory Ceiling, Walls and Floor Covering
- 42 Lavatory Equipment
- 50 CARGO AND ACCESSORY COMPARTMENTS
- 52 Cargo Loading System
- 58 Aft Bulk Compartment Flight Kit Container
- 60 EMERGENCY
- 61 Evacuation Equipment
- 63 Emergency Locator Transmitter System
- 65 Evacuation Annunciation

26 FIRE PROTECTION

- 00 GENERAL

NOTE: GEN-SWITCH MUST BE IN ON POSITION AND AC BUS TIE-SWITCH MUST BE IN NORM POSITION FOR FOLLOWING FAULT INDICATIONS





FAULT SYMPTOMS

FAULT CODE AND SYS AFFECTED

UNPARALLELED OPERATION (ENG GENS ONLY)

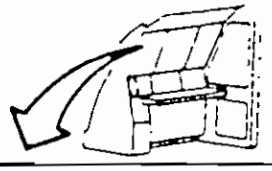
GEN OFF LIGHT-ON AND...

NO AC LOAD READING	24-20	AE	-1/2/3
NO AC LOAD, NO VOLTS, & NO FREQUENCY READINGS	24-20	AG	-1/2/3
FLICKERING & AC FREQ READING FLUCTUATES	24-20	AK	-1/2/3
FREQ READING DIFFERENT ALL THREE GENERATORS	24-20	AW	-1/2/3

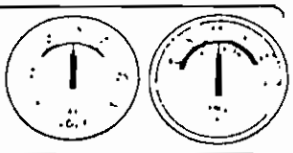
AC BUS OFF LIGHT-ON AND...

NO AC LOAD, NO VOLTS, & NO FREQUENCY READINGS	24-20	AB	-1/2/3
NO AC LOAD READING	24-20	AC	-1/2/3
APU PWR IN USE LIGHT-ON	24-20	AD	-1/2/3

GEN FAIL LIGHT-ON & NO RESET .... CAUTION DISCONNECT CSD, OR EXTENSIVE CSD OR GENERATOR DAMAGE MAY OCCUR	24-20	AA	-1/2/3
AC BUS OFF LIGHT-ON	24-20	AM	-1/2/3
AC BUS 1 OR 3 OFF OR L EMER AC BUS OFF LIGHTS-ON	24-20	AQ	
AC BUS 2 OFF DC BUS 2 OFF LIGHTS-ON	24-20	AR	
ALL AMBER AC BUS TIE ISOL LIGHTS-ON	24-20	AN	
NO POWER TO GROUND SERVICE BUS	24-20	AL	
BUS PARALLELING WITH NO COMMAND	24-20	AP	
VOLT AND/OR FREQ METERS DO NOT DEFLECT	24-20	AS	-1/2/3
GENERATOR WILL NOT ACCEPT ITS BUS	24-20	AT	-1/2/3
GENERATOR BUS DROPS OFF THE LINE	24-20	AU	-1/2/3
GEN OFF LIGHT-ON DIM OR FLICKERING	24-20	AV	-1/2/3
ANY UNPARALLELED OPERATION FAULT SYMPTOM NOT SHOWN ABOVE	24-20	AX	-1/2/3



DUAL LAND SELECTED	DUAL LAND PWR ON	AC BUS TIE ISOL	AC BUS TIE ISOL	APU PWR IN USE	APU PWR AVAIL	APU GEN OFF	APU GEN FAIL
(BLUE)	(GREEN)	(AMBER)	(AMBER)	(BLUE)	(GREEN)	(AMBER)	(AMBER)



FAULT SYMPTOMS

FAULT CODE AND SYS AFFECTED

PARALLEL OPERATION

CHANNEL WILL NOT PARALLEL, AMBER [AC BUS TIE ISOL] LIGHT ON, AND:

- NORMAL AC FREQUENCY READINGS ..... 24-20 BD -1/2/3
- AC FREQUENCY READING FLUCTUATES ..... 24-20 BA -1/2/3

ONE GENERATOR - UNABLE TO PARALLEL OR WON'T STAY PARALLELED ..... 24-20 BE -1/2/3

ALL GENERATORS - UNABLE TO PARALLEL ..... 24-20 BF

DUAL LAND MODE

[DUAL LAND PWR ON] LIGHT ON AND ALL BLUE [AC BUS TIE ISOL] LIGHTS OFF ..... 24-20 DA

ONE BLUE [AC BUS TIE ISOL] LIGHT OFF ..... 24-20 DB -1/2/3

ALL AMBER [AC BUS TIE ISOL] LIGHTS ON ..... 24-20 DC

AVAILABLE APU POWER

[APU PWR AVAIL] LIGHT OFF, [APU GEN OFF] LIGHT ON, AND:

- NO VOLTS AND/OR NO FREQUENCY READINGS ..... 24-20 EA
- VOLTAGE AND FREQUENCY READINGS FLUCTUATING ..... 24-20 EE
- VOLTS AND FREQUENCY READINGS ARE NORMAL ..... 24-20 EB
- FREQUENCY READING STEADY BUT OUT OF TOLERANCE ..... 24-20 EJ

[APU PWR AVAIL] LIGHT ON WITH APU SHUT DOWN & ALL ENGINES OPERATING ..... 24-20 ED

[APU GEN FAIL] LIGHT ON ..... 24-20 EC

APU FAILS TO TAKE OVER BUS FROM ONE CHANNEL OF EXTERNAL PWR ..... 24-20 EF

APU GENERATOR INOP (DE-EXCITED) ..... 24-20 EG

[APU PWR AVAIL] LIGHT OFF WITH APU OPERATING ..... 24-20 EH

APU POWER ON GEN BUS

[APU PWR AVAIL] LIGHT ON, AND:

- MAIN GENERATOR BUS WILL NOT ACCEPT APU ..... 24-20 FA -1/2/3
- WITH MAIN GENERATORS ON, ONE BUS STAYS ON APU PWR ..... 24-20 FC -1/2/3
- ALL BLUE [AC BUS TIE ISOL] LIGHTS ON ..... 24-20 FB

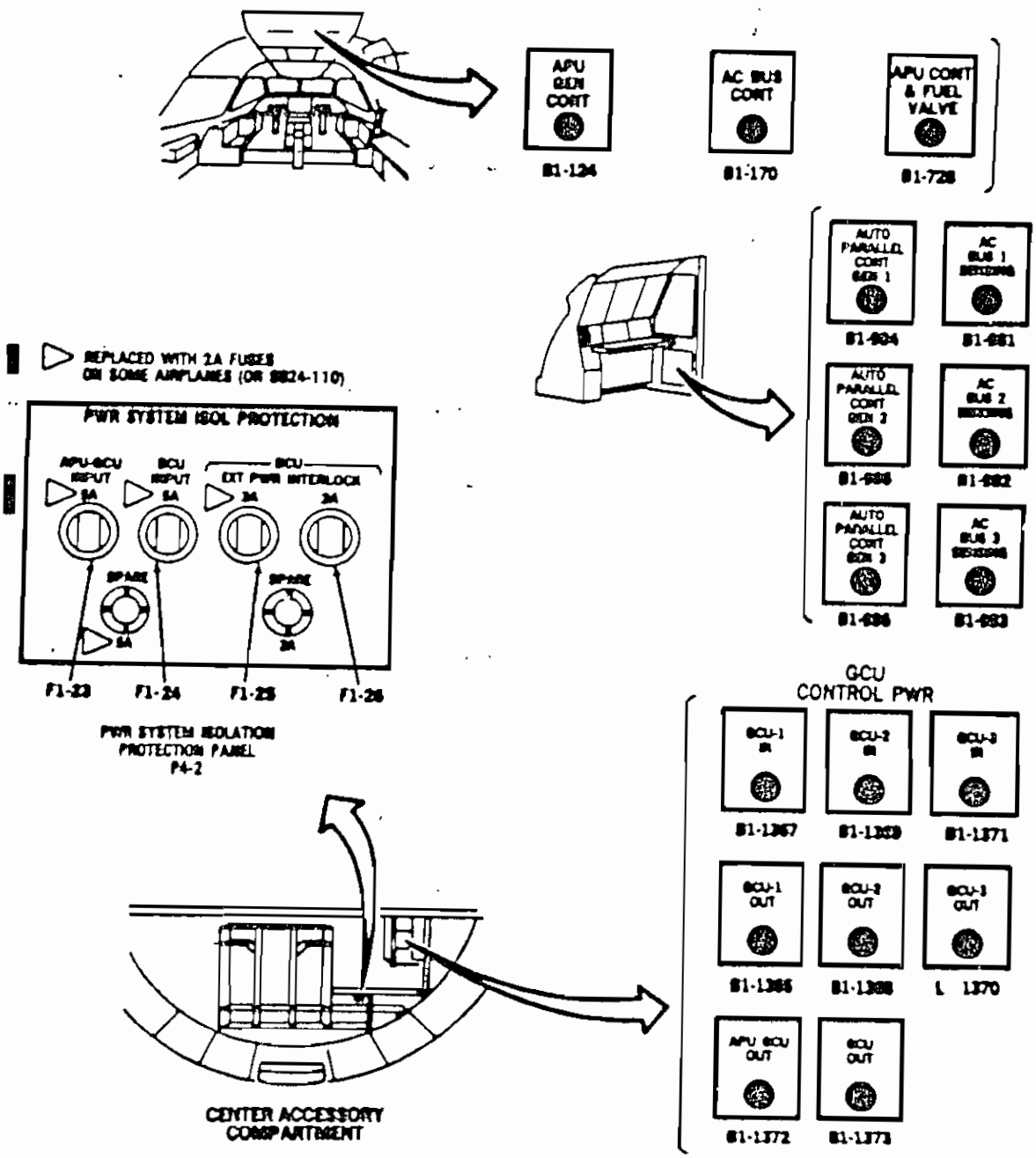
ANY FAULT SYMPTOMS NOT SHOWN ABOVE ..... 24-20 XX -1/2/3

1

CA17-24-38031

Fault Symptoms
Figure 101 (Sheet 2)

PRINTED IN U.S.A.

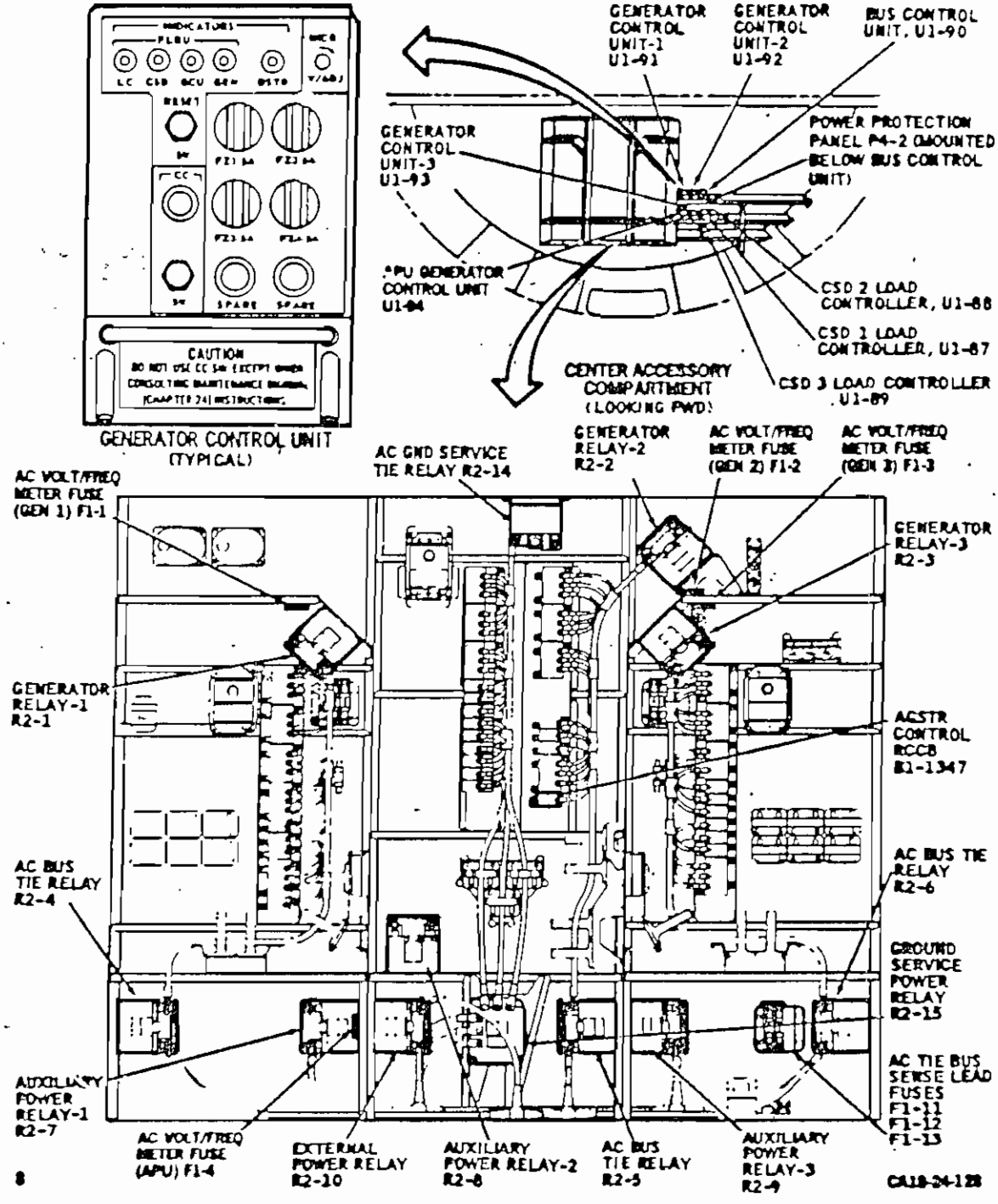


1C

CA17-24-42031

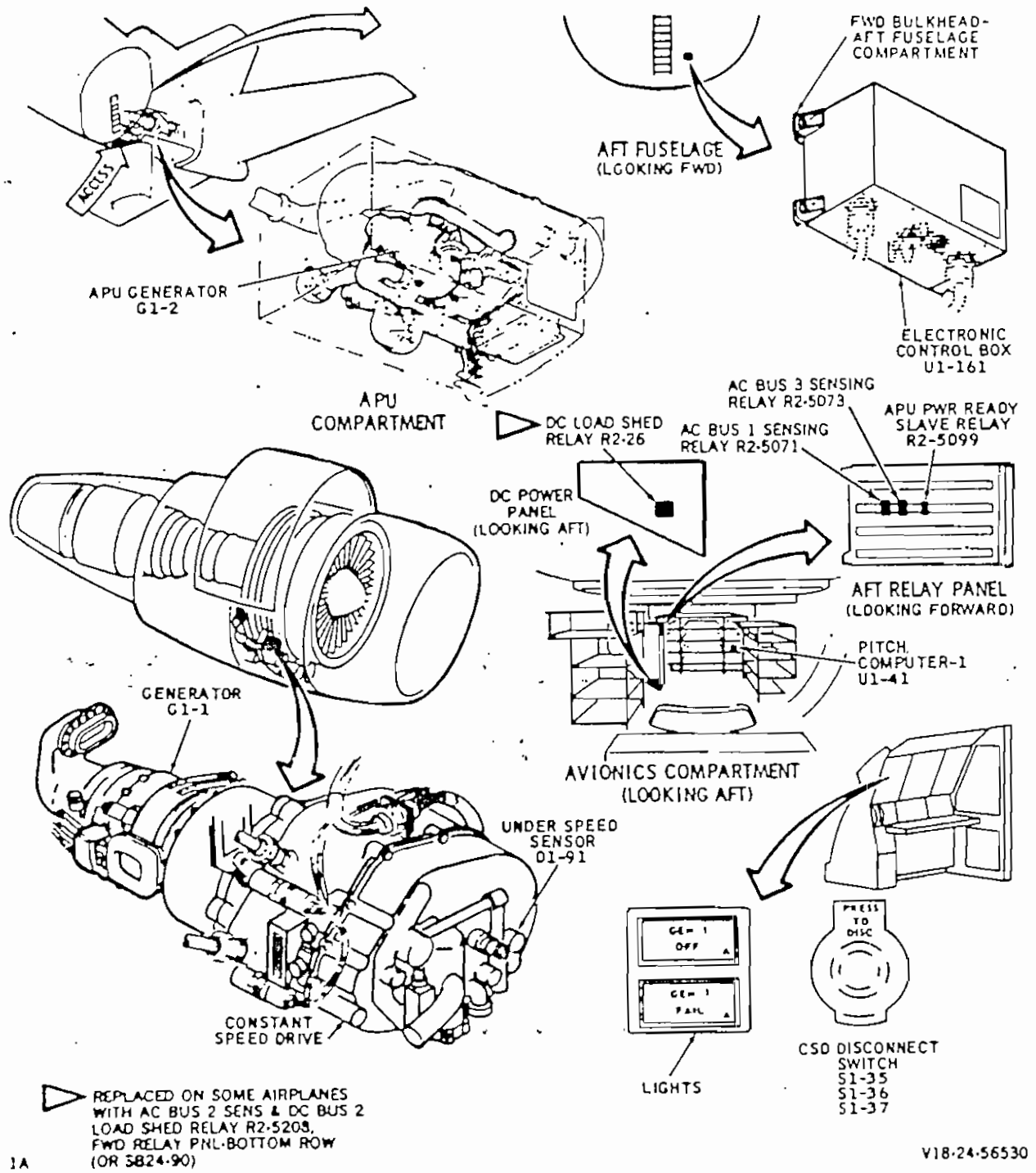
Locator
Figure 102 (Sheet 1)

PRINTED IN U.S.A.



Locator
Figure 102 (Sheet 2)

PRINTED IN U.S.A.



Locator
Figure 102 (Sheet 3)

Supplemental Fault Isolation Data

ELECTRICAL POWER SYSTEM ABBREVIATIONS

AGSTR	AC GROUND SERVICE TIE RELAY	GR	GENERATOR RELAY
APR	AUXILIARY POWER RELAY	GS	GROUND SERVICE
BCU	BUS CONTROL UNIT	GSR	GROUND SERVICE RELAY
BTR	BUS TIE RELAY	ICU	INDICATOR CONTROL UNIT
CIA	CIRCUIT INTERRUPT AUX	ISOL	ISOLATION
DP	DIFF CURRENT PROTECTION	LC	LOAD CONTROLLER
EPC	ELECTRICAL POWER CENTER	MEP	MAIN EXTERNAL POWER
EPR	EXTERNAL POWER RELAY	OE	OVEREXCITATION
FE	FLIGHT ENGINEER	OF	OVERFREQUENCY
FZ	FUSE	OP	OPEN PHASE
GCR	GENERATOR CONTROL RELAY	PFO	PARALLEL FEEDER OPEN
GCU	GENERATOR CONTROL UNIT	PR	POWER READY
GEP	GALLEY EXTERNAL POWER	UBC	UNBALANCE CURRENT
GLCU	GALLEY LOAD CONTROL UNIT	UE	UNDEREXCITATION

FUNCTIONS IN GENERATOR CONTROL UNIT (GCU)

A. For Main Engine Gen Systems

1. Over and Under Excitation
2. Unbalance Current
3. Bus Tie Relay trip and close
4. Generator Relay trip and close
5. Reactive Load Division
6. Auto Parallel Control
7. Bus Temporary Lockout
8. Bus Lockout
9. Identifies failed LRU's for: CSD, LC, GCU, GEN, and DSTR

B. For Main Engine and APU Gen Systems

1. Over and Under Voltage
2. Underspeed
3. Differential Current
4. Open Phase
5. Phase Sequence
6. Parallel Feeder Open
7. Gen Diode and Bearing Fail Ind
8. Overcurrent (Galley Load Shedding)
9. Voltage Regulation
10. Feeder Fault indication
11. Confidence Check of GCU passive functions
12. GCR - opens and closes field circuit to gen
13. Power Ready
14. Overfrequency - on some airplanes (or SB24-111)

C. For APU Gen System:

1. Apr #1 trip and close
2. Identifies failed LRU's for GCU, GEN, and DISTR

FUNCTIONS IN BUS CONTROL UNIT (BCU)

1. Phase Sequence Protection for MEP and GEP.
2. Over and Under Voltage for MEP and GEP.
3. Over and Under Frequency for MEP and GEP.
4. Auto Parallel Command, which arms all paralleling signals to all main channel GCU's.
5. Power Ready for MEP (provides for MEP Power Available Light and MEP switch holding coil). Also provides for control of GSR EP coil and GS power ext switch holding coil.
6. Preferential Circuit with Dead Tie Bus Logic.
7. MEP EPR close and trip.
8. Galley Load Shedding Command to trip galley 1, 2, and 3 relays.
9. APR-2 trip and close.
10. APR-3 trip and close.
11. Tie Bus Differential Trip Protection (trips and locks out all BTR's and APR's).
12. Power Ready for GEP (provides for GEP Power Available Light and GEP switch holding coil). Also enables GTR's to be fed from galley ext power.
13. Dual Land - Provides signal for DUAL LAND PWR ON Light and for relay which transfers AC BUS TIE ISOL lights from amber to blue. Also sends Split Bus logic to Flight Guidance Control System.
14. Confidence Check of BCU passive functions.
15. Parallel Feeder Open for MEP. Open Phase removed on some airplanes (or SB24-51).

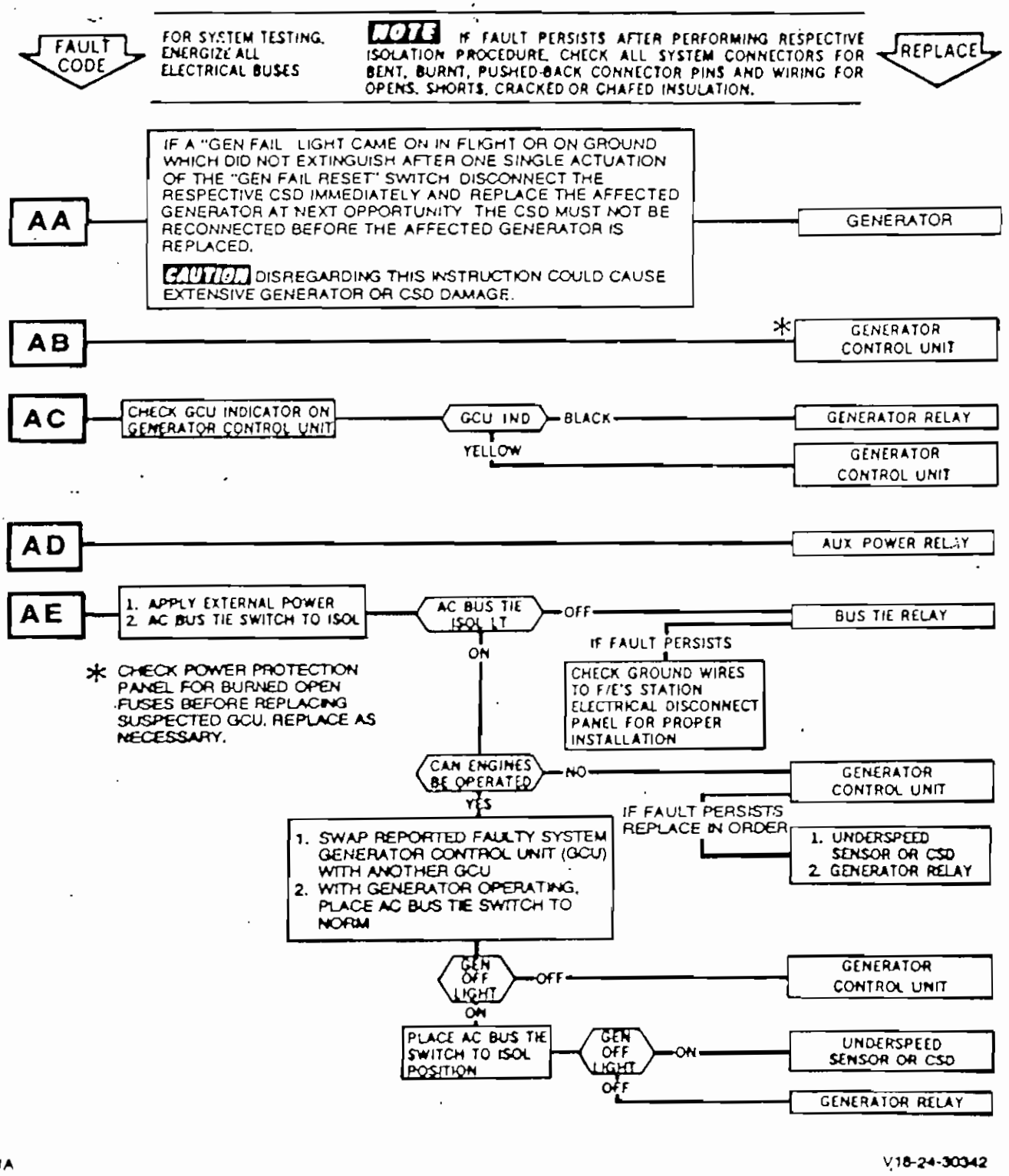
1B

V18-24-84130

Locator
Figure 102 (Sheet 4)

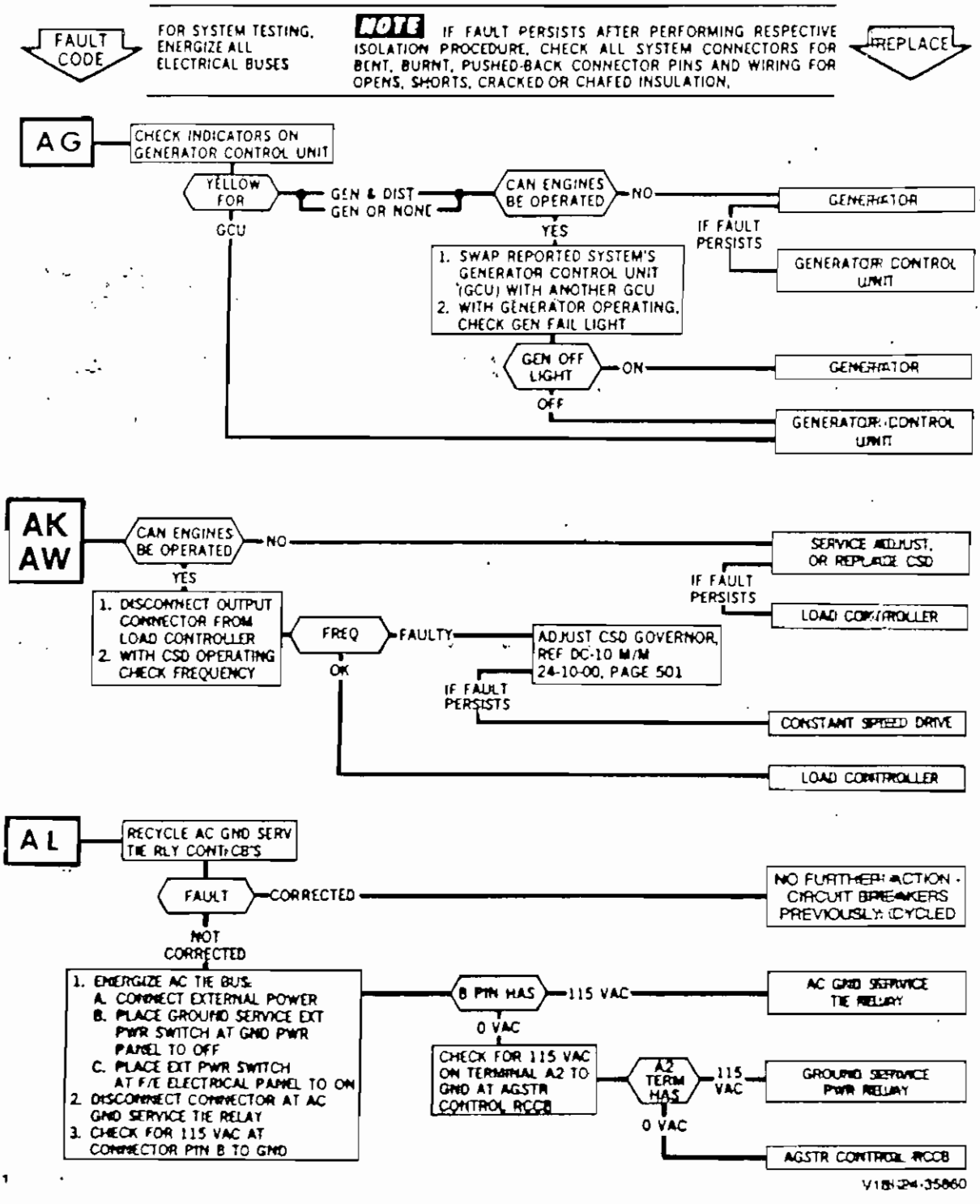
R

PRINTED IN U.S.A.

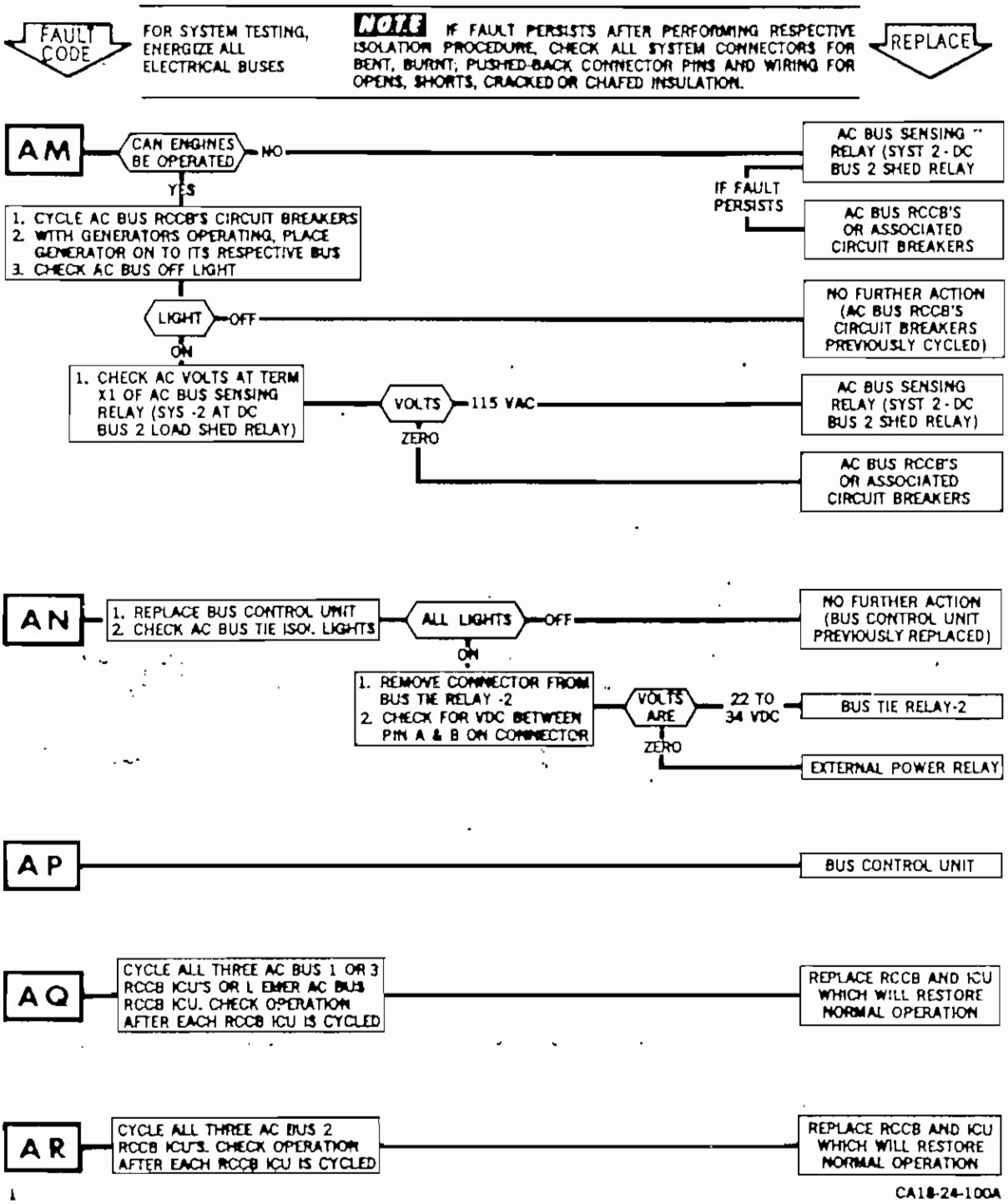


Fault Trees
Figure 103 (Sheet 1)

PRINTED IN U.S.A.



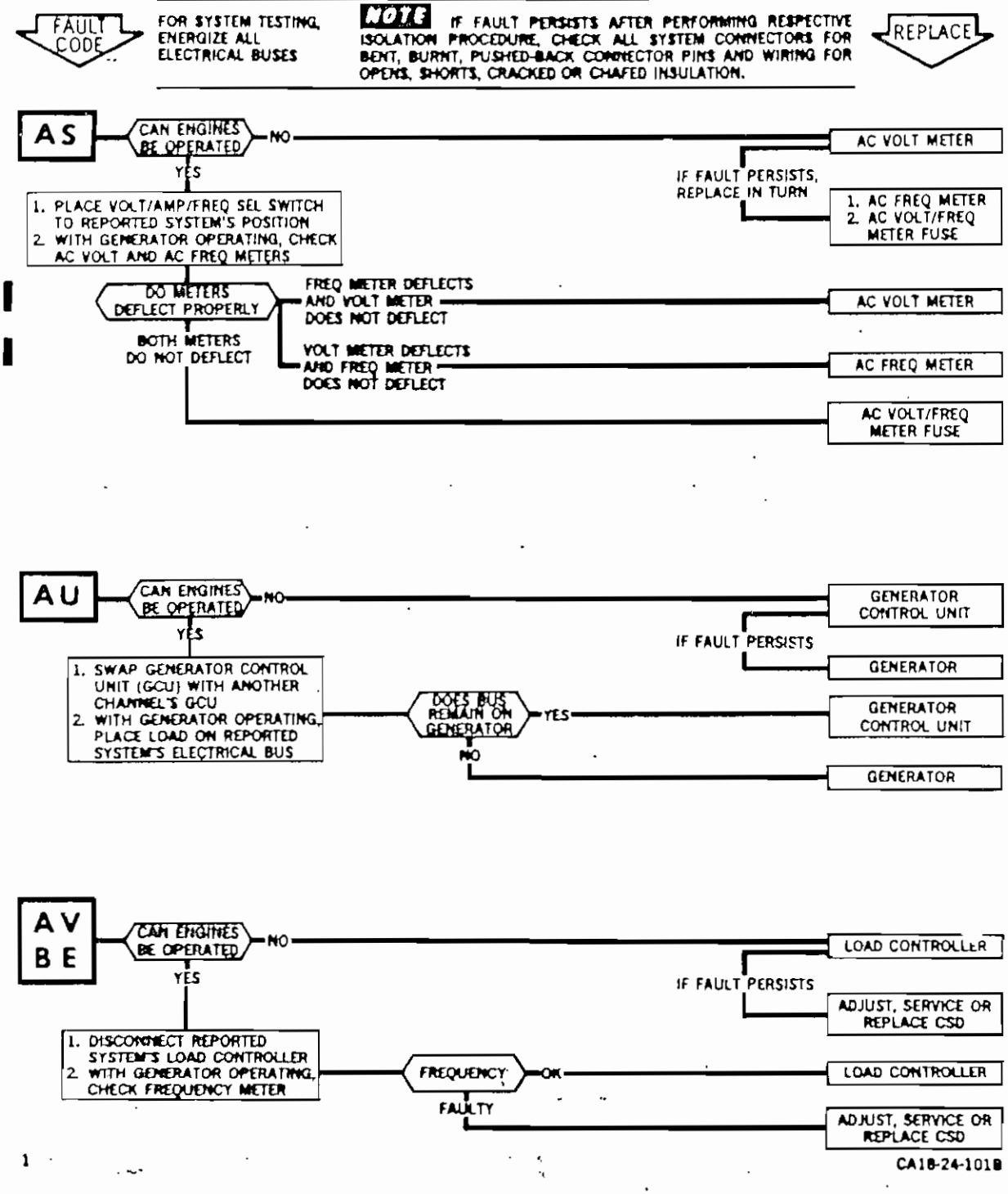
Fault Trees
Figure 103 (Sheet 2)



Fault Trees
Figure 103 (Sheet 3)

EFFECTIVITY:KS ALL

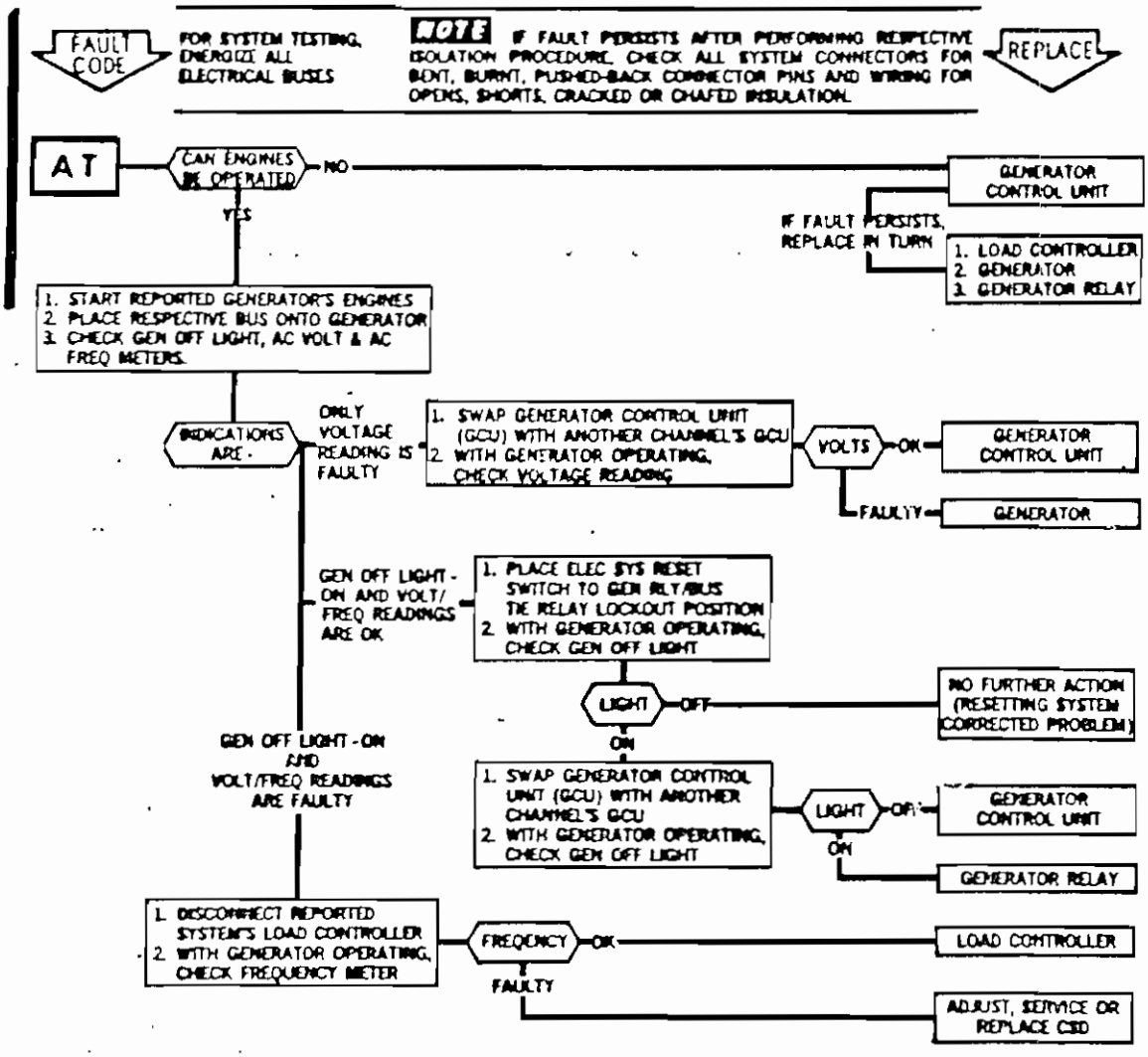
PRINTED IN U.S.A.



Fault Trees
Figure 103 (Sheet 4)

EFFECTIVITY:KS ALL

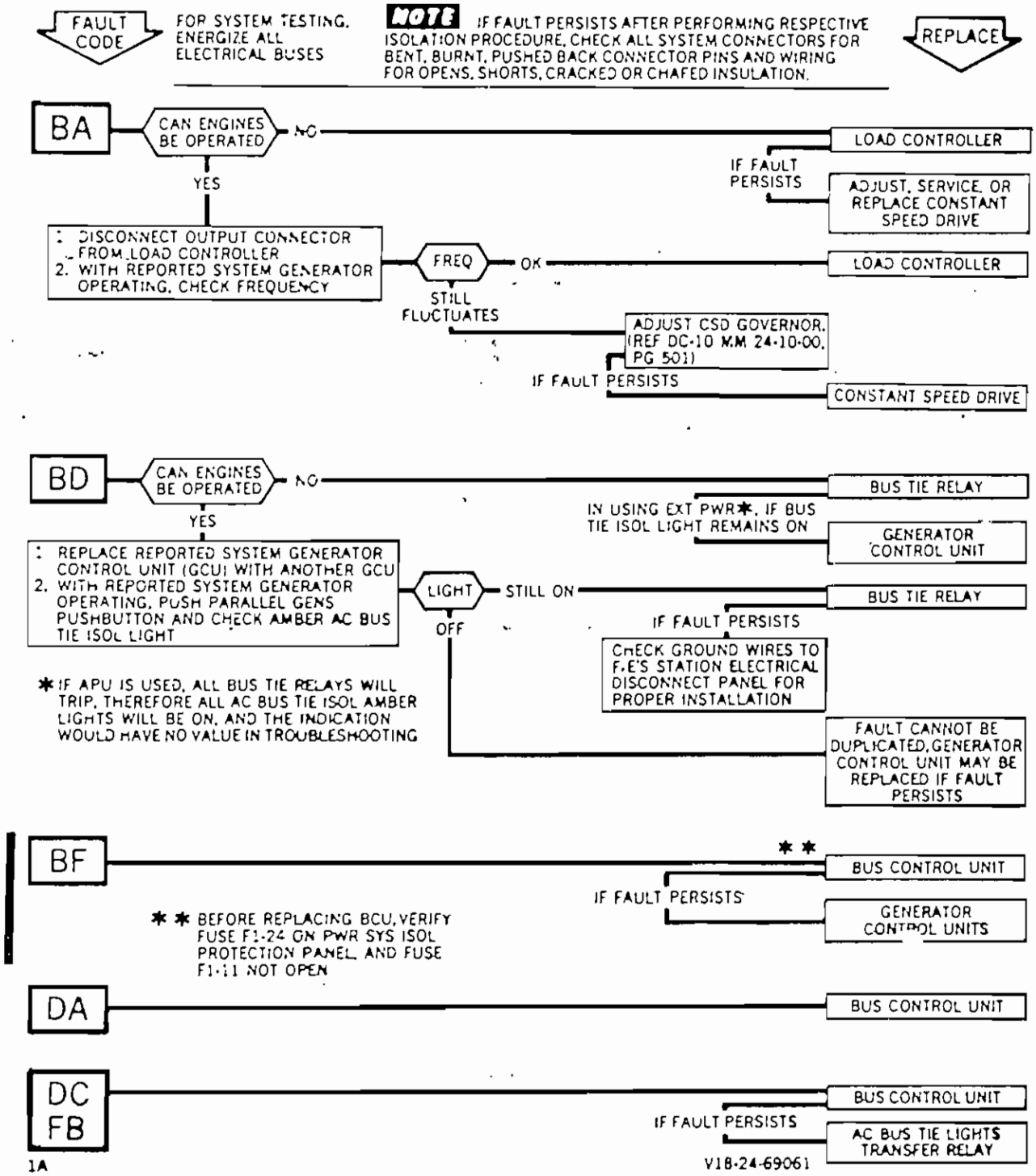
PRINTED IN U.S.A.



1

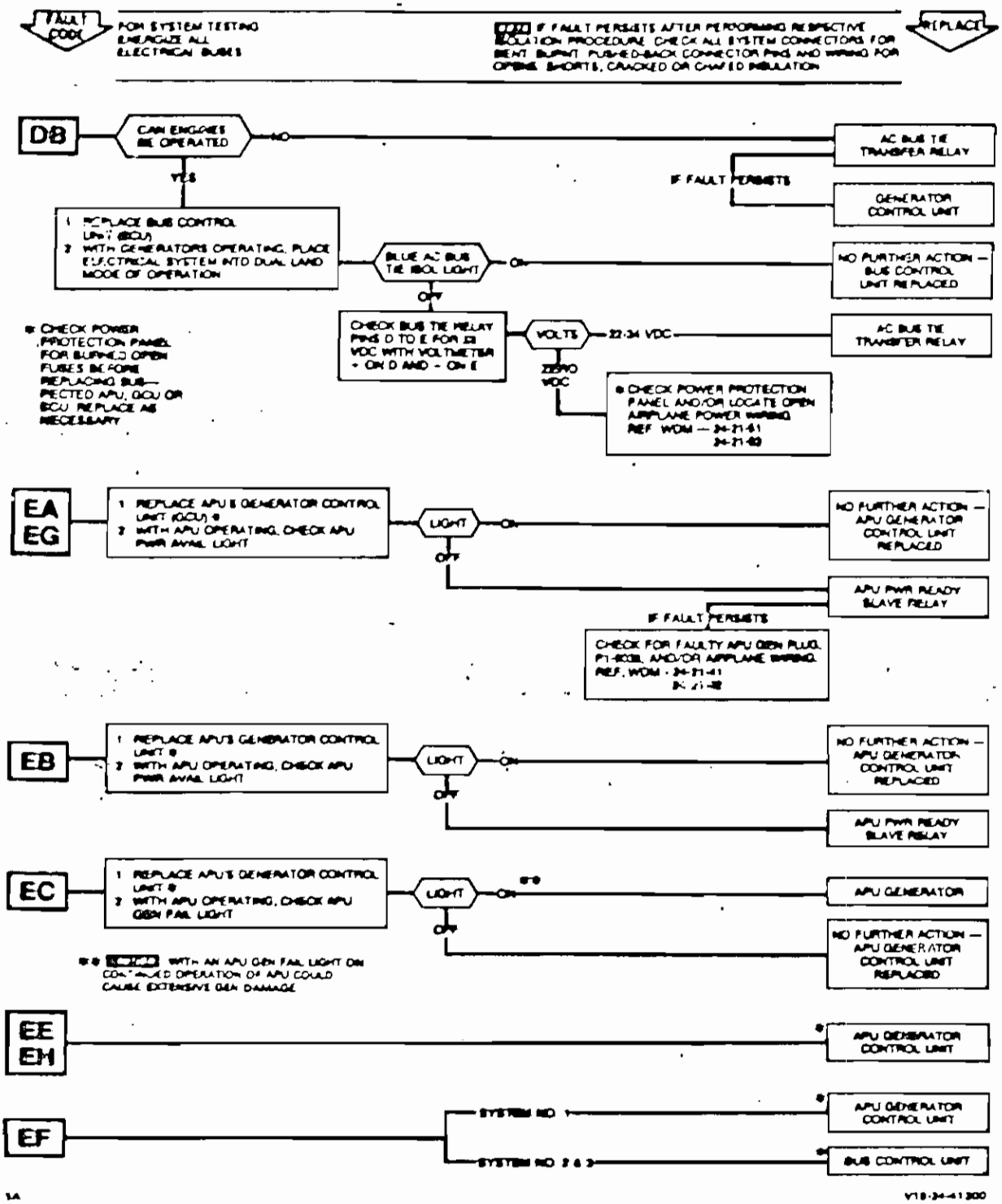
CA18-24-102A

Fault Trees
Figure 103 (Sheet 5)



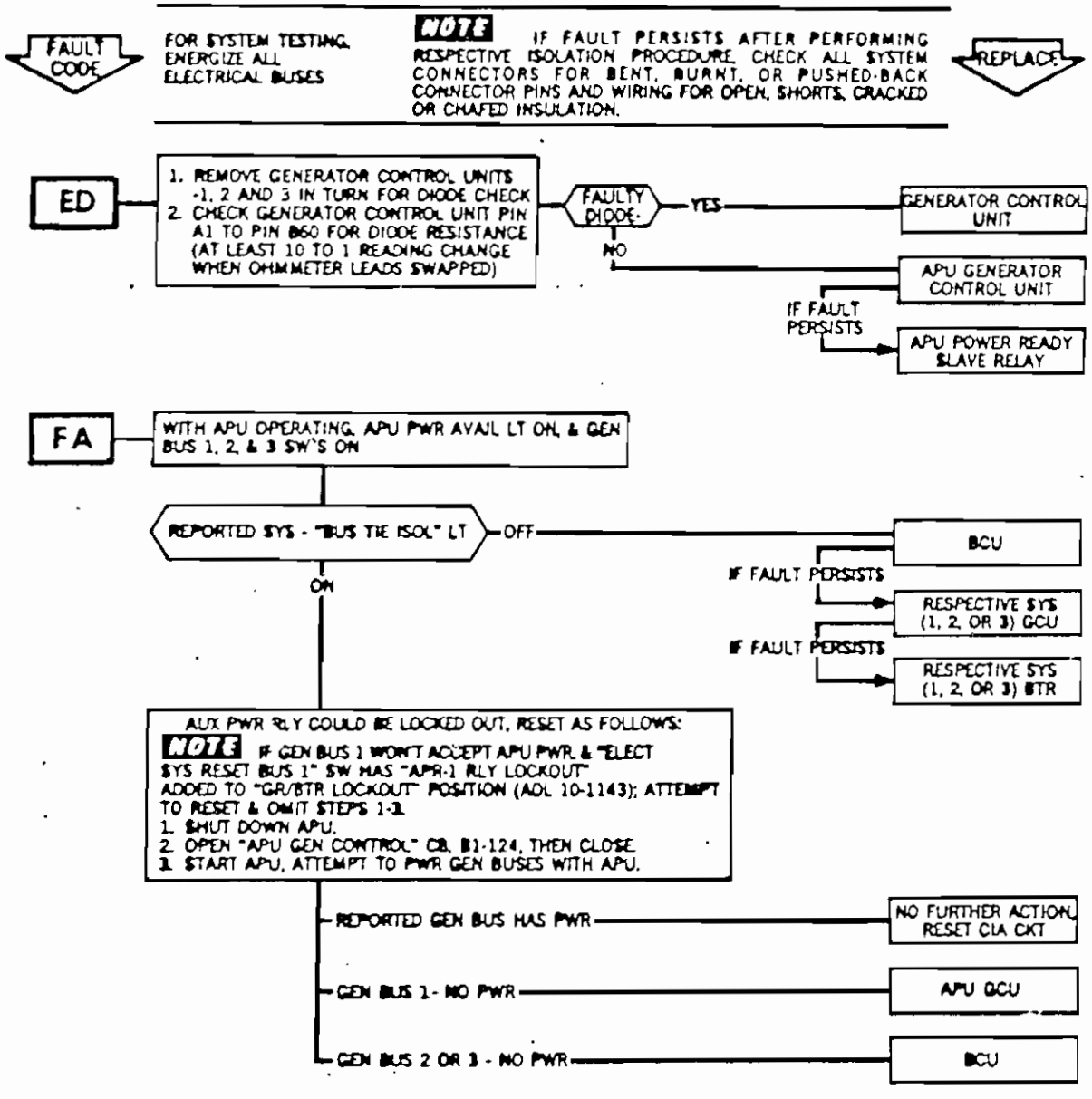
Fault Trees
Figure 103 (Sheet 6)

PRINTED IN U.S.A.



Fault Trees
Figure 103 (Sheet 7)

PRINTED IN U.S.A.

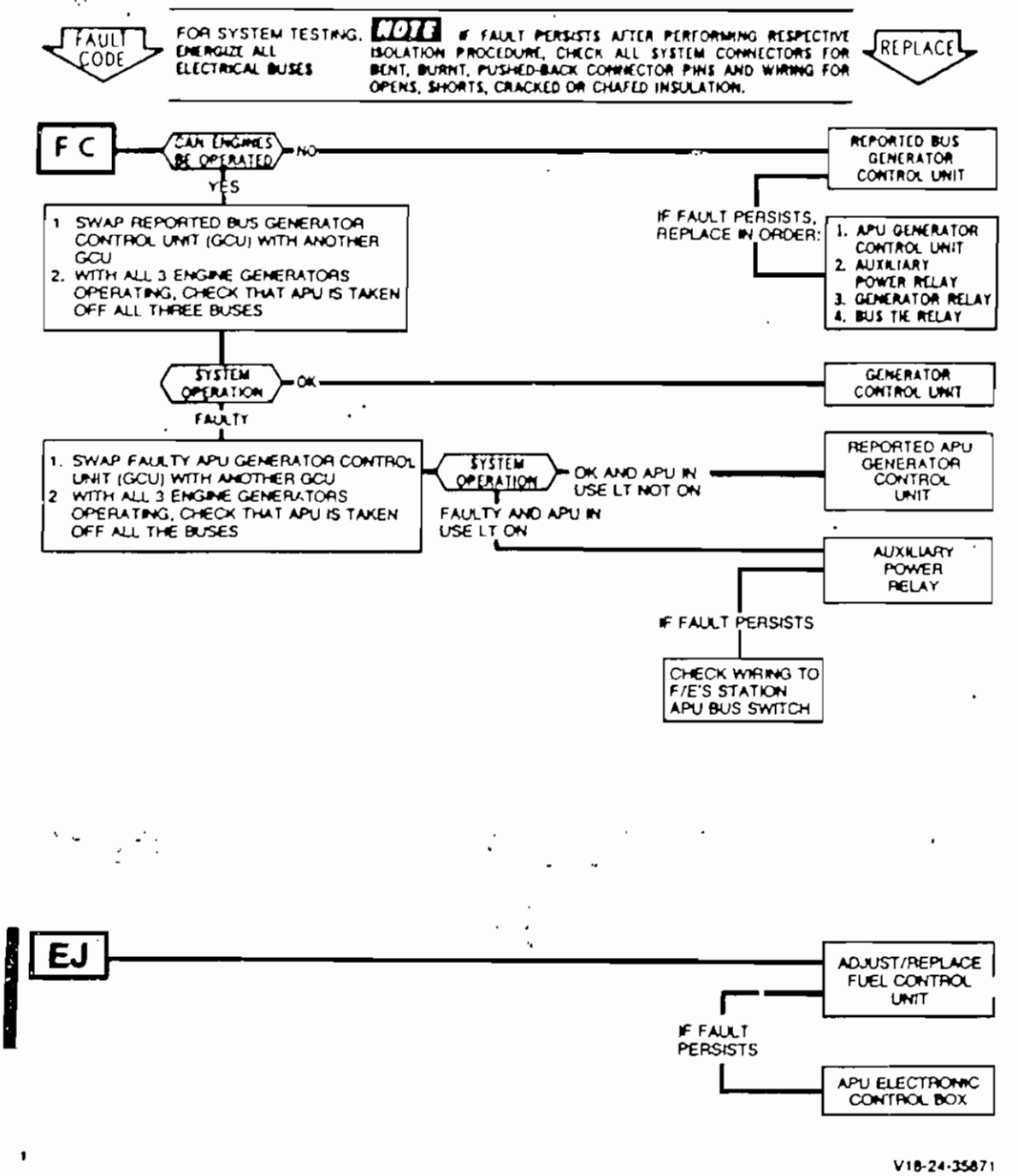


1

V18-24-56511

Fault Trees
Figure 103 (Sheet 8)

PRINTED IN U.S.A.



Fault Trees
Figure 103 (Sheet 9)

PRINTED IN U.S.A.

INDICE DE FIGURAS

CAPITULO I:

1.1 TERMINOS DE LA INGENIERIA DE CONOCIMIENTO.....17

1.2 METODOS DE REPRESENTACION DE CONOCIMIENTO.....27

1.3 POSIBILIDADES DEL MOTOR DE INFERENCIA.....34

1.4 NIVELES DE RAZONAMIENTO DE DIAGNOSTICO.....39

1.5 VISTA GLOBAL DEL MODELO DE DIAGNOSTICO INTEGRADO.....49

1.6 ESTRUCTURA DE LA BASE DE CONOCIMIENTO EXPERIMENTAL...49

1.7 ESTRUCTURA DE UNA HERRAMIENTA PARA SIST. EXP.....53

1.8 POSIBILIDADES PARA LA INTERFASE AL DISEÑADOR.....54

1.9 POSIBILIDADES PARA LA INTERFASE AL USUARIO.....55

1.10 ASPECTOS DE SOFTWARE Y HARDWARE DE HERRAMIENTAS.....57

1.11 EVALUACION GLOBAL DE UNA HERRAMIENTA.....62

CAPITULO II:

2.1 COMPONENTES DEL SISTEMA ELECTRICO DE POTENCIA.....96

2.2 ESQUEMA DEL SISTEMA ELECTRICO DE POTENCIA.....97

2.3 ESQUEMA DEL SISTEMA DE GENERACION AC.....98

2.4 PANEL ELECTRICO DEL INGENIERO DE VUELO.....99

CAPITULO III:

3.1 CICLO DE DESARROLLO DE UN SISTEMA EXPERTO.....106

CAPITULO IV:

4.1 ESQUEMA DE CODIFICACION PRELIMINAR
DE UN ARBOL DE FALLA.....142

APENDICE D:

CARTAS PARA CAZA DE FALLAS DEL
SISTEMA DE GENERACION AC.....213