FAULT PROTECTION ARCHITECTURE FOR THE COMMAND AND DATA SUBSYSTEM ON THE CASSINI SPACECRAFT

Thomas K. Brown
James A. Donaldson

Jet Propulsion Laboratory, California Institute of Technology
Pasadena, California

ABSTRACT

Cassini is a NASA/JPL spacecraft that is planned to be launched in 1997 for a 10.7 year mission to the planet Saturn. This paper focuses 'on one subsystem on the spacecraft, the Command and Data Subsystem (CDS).

The paper will present overviews of the Cassini and CDS avionics and then describe, the fault protection architecture for the subsystem. This description will cover fault detections, error filtering, event activation rules, and response triggering for the following key subsystem items:

1 Command and Data Electronics Assemblies (CDEAs)
2 1553B Bus (CDS bus) and Remote Terminal Communication Interface Units (RT ClUs)
3 Remote Engineering Units (l< Us)
4 Solid State Recorders (SS1{s)

INTRODUCTION

In order to put the Cassini CDS fault protection architecture and design into perspective, the following two sections will describe the spacecraft and CDS avionics architectures [1].

CASSINI SPACECRAFT

The Cassini spacecraft is approximately 4 by 6.6 meters in size, **5,655** kilograms in mass, operates on between 650 and 825 watts of power, and contains twelve engineering and twelve science instrument **subsystems.**

Cassini Spacecraft Avionics Architecture

The spacecraft's avionics architecture is shown in Figure 1. This viewpoint captures end-to-end uplink command reception through downlink telemetry transmission. Ground operations arc not included in the figure.

Of the twelve engineering subsystems, the six that arc avionics oriented and their principal services arc as follows:

1 Radio Frequency Subsystem (RFS) - uplink command reception and downlink telemetry transmission,
2 Command and Data Subsystem (CDS) - uplink command processing, spacecraft intercommunicat ion, and downlink telemetry collection and packetization,
3 Attitude and Articulation Control Subsystem (AACS) - attitude determination, attitude control, thrust vector control, and main engine control,

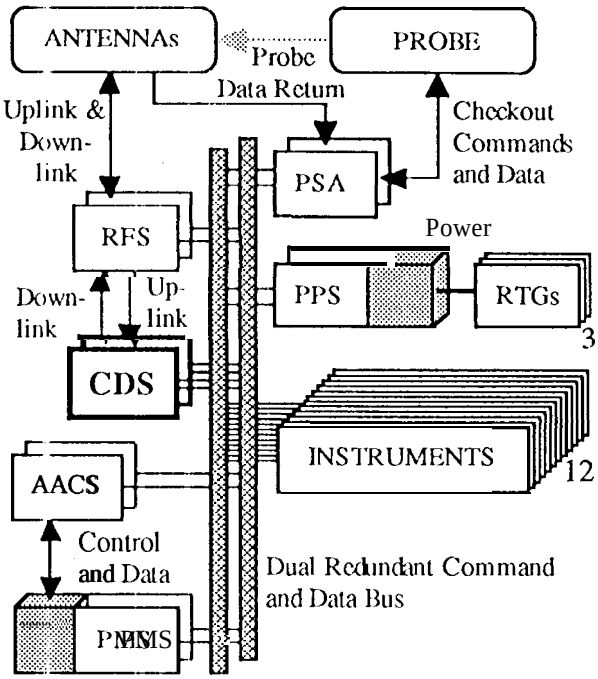

Figure 1 Cassini Functional Block Diagram

4 Propulsion Module Subsystem (PMS) - propellant tanks, thrusters, and main engines for spacecraft maneuvers (controlled by AACS),
5 Power and Pyrotechnics Subsystem (PI'S) - power supply, conditioning, and control along with pyrotechnic firing circuitry, and
6 Pi obc Support Avionics (PSA) - probe checkout during cruise and data return during the probe's encounter with the Saturn moon Titan.

In essence, the RFS, CDS, and AACS arc completely dual redundant. The RFS operates with only onc unit powered. The CDS and AACS must operate with either onc unit powered and onc unit cold-spared or both units powered. The I'PS and PMS both have redundant communication interfaces, but each also contains nonredundant elements. This non-redundancy is shown by gray shading in Figure 1 and by dashed lines in Figure 2.

The remaining six engineering subsystems arc cabling, structures, mechanical devices, antennas, thermal control, and electronic packaging.

THE COMMAND ANI) DATA SUBSYSTEM

The CDS is the hub of communications (1) for the subsystems on the spacecraft and (2) between these subsystems and the ground. In this role, the CDS provides a set of services to **both the subsystems** and the ground.

CDS Services

The nine services that CDS furnishes along with short descriptions of each arc as follows:

1 Uplink Command Decoding - the ability to process commands and thereby control the spacecraft from the ground.

2 Sequencing - the ability to store sequences of commands from the ground for later execution in order to orchestrate sets of activities. Three of these sequences arc designated as critical (launch, Saturn Orbit Insertion (S01), and probe data relay) where completing the event is more crucial than the safely of the spacecraft itself.

3 Time-keeping - the ability to maintain a unique. spacecraft t i mc in order to coordinate spacecraft activities and synchronize science and engineering subsystems.

4 Downlink telemetry - the ability to provide visibility into engineering subsystem performance and science subsystem data.

5 Bulk data handling - the ability to buffer on-board data when the data collection rate exceeds the downlink telemetry rate.

6 Spacecraft intercommunications - the capability to communicate with the engineering and science subsystems on-board the spacecraft.

7 Control services - the capability to monitor and control oil-board temperatures.

8 System Fault Protection (SFP) - the capability to host algorithms (monitors and responses) to respond to non-CDS spacecraft level faults.

9 CDS Fault Protection (CFP) - the capability to detect and respond to faults that affect the above eight services CDS provides.

All these services arc vital to the operation of the space.craft during all mission phases.

CDS Architecture

T'hc dual redundant architecture of the CDS is shown in Figur c 2. There arc four primary regions associated with **the** CDs:

1 Onc pair of Command and Data Electronics Assemblies (CDEAs),

2 Onc pair of MIL-STD-1553B buses (CDS buses) connected to twenty-eight Remote Terminal Communication Interface Units (RTCIUs),

3 Four pairs of Remote Engineering Units (REUs), and

4 Onc pair of Solid State Recorders (SSRs),

The CDEAs arc referred to as the central items and the Bus/R TCIUs, REUs, and SSRS as the peripheral items.

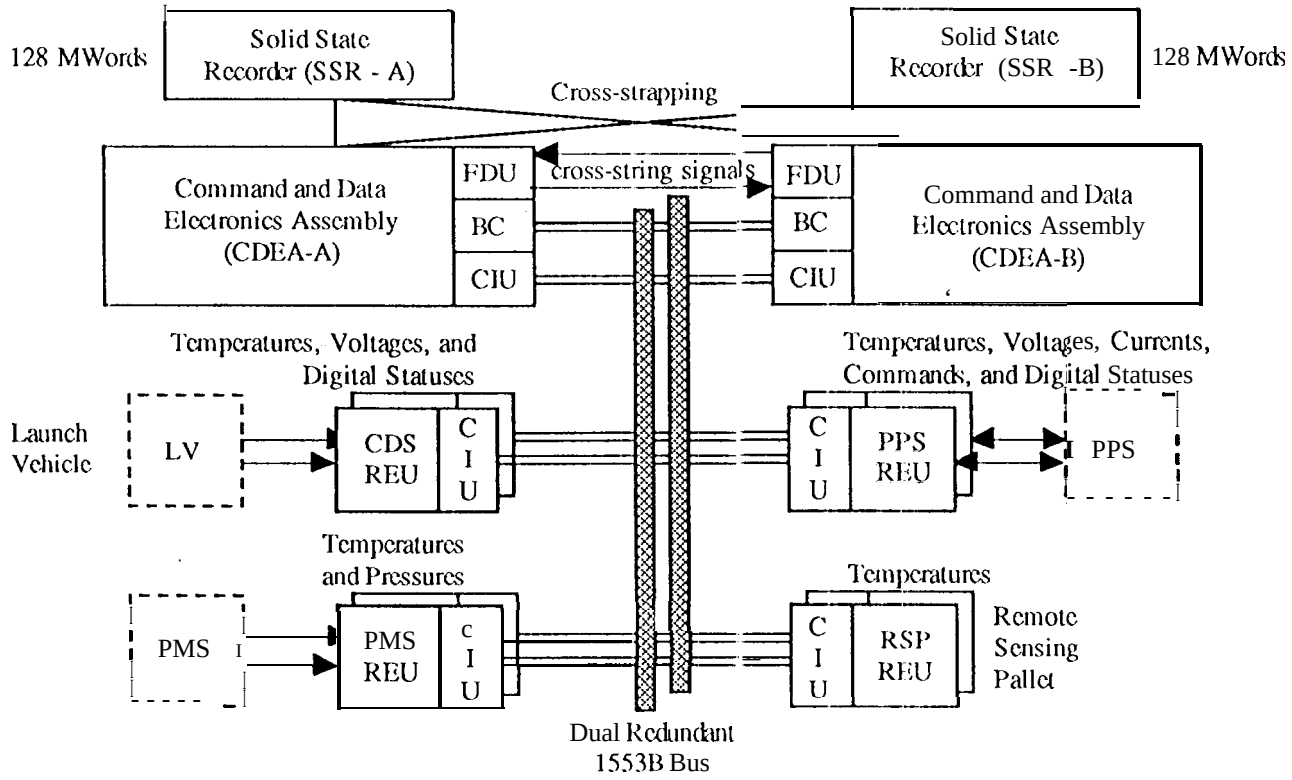

Figure 2 Cassini Command and Data Subsystem Functional Block Diagram

Each central CDEA contains a set of components that are listed and described below::

1 An engineering flight computer with 512K of Random Access Memory (RAM) and 8K of Programmable Read Only Memory (PROM) for supporting CDS services and redundancy management,
2 A hardware command dccodcr for uplink command reception from the RFS,
3 A Reed Solomon downlink unit to encode telemetry for the RFS,
4 A timing unit for spacecraft time maintenance and synchronization signal generation,
5 An SSR interface unil to command, record to, playback from, and status the SSRs,
6 A Bus Controller (BC) to manage 1553B bus activities (only onc CDEA BC is active at a time),
7 A 1553B Remote Terminal Communication Interface Unit (RTCIU) so the BC can communicate with the redundant CDEA, and
8 A Fault Detection Unit (FDU) with a special interface to the redundant CDEA to help manage both CDEA and SSR redundancy.

Each CDS peripheral item is listed and described below:

1 The 1553B bus and RTCIUs provide for inter-communications between a BC and tile twenty-cigi~t remote terminals (RTs) on the spacecraft. The front-end interface to these RTs is called an RTCIU.
2 The REUs arc used primarily to collect engineering measurements (temperatures, pressures, voltages, currents, and digital statuses) from around the spacecraft. In addition to these standard functions, tile CDS REU is used to convey launch vehicle commands and separation indicators to the CDS, and the PI'S REU is used to convey commands from the CDS to the PPS.
3 The SSRS each contain 128 megawords (16 bit words) of Dynamic Random Access Memory (DRAM) and arc cross-strapped 10 both CDEAs. (see Figures 1 and 2).

The CDS dual redundant configuration provides for single fault tolerance. To support this, the CDS internal fault protection (CIFP) must detect and respond to all single faults that can render any onc unit of any of its redundant pairs inoperable. The CIFP accomplishes this using hardware and software fault detection mechanisms.

At the center of the CDS arc the CDEAs. Only onc CDEA can bc prime at any onc lime because only the prime unit controls access 10 the SSRS and the1553B bus. Tile other CDEA is backup (it can access the SSR not being used by the prime CDEA but its 1553B BC is inhibited). Fault protection and redundancy management for the CDEAs arc accomplished using the interface signals passing between the CDEA FDUS coupled with both PROM and RAM code residing in the CDEA.

CDS INTERNAL FAULT I'RO"IRCTION

On Cassini, CDS internal fault protection (CIFP) refers to those flight and ground based hardware, software, and procedural elements that avoid, detect, and respond to perceived spacecraft faults. It has denotations of fault intolerance and fault tolerance [2]. In the former, the goal is to prevent or minimize the probability of failure through the usc of conservative design practices, etc. In the, latter, the goal is to nullify the effects of failure, e.g. through the usc of redundancy. Consequently, the primary purpose of fault protection is to provide a highly available spacecraft espec i all y for iaunch, Saturn Orbit Insertion (S01), and the Probe data relay critical sequences. This goal is supported by both the on-board, autonomous systems that ensure spacecraft system integrity in the presence of anomalous conditions and by the. ground nonautonomous operations when time and circumstances permit. A secondary objective of fault protection is to support a highly reliable spacecraft that will survive the entire 10.7 year mission. However, this goal must bc rnct entirely by each individual piece of equipment on the spacecraft regardless of the dual redundant architecture and fault protection actions.

CIFP Requirements

The fundamental requirement on CIFP is onc of Single Fault Tolerance (SFT). in other words, no credible Single Point Failure (SPF) shall prevent attainment of the primary mission objectives or result in a severely degraded mission. in conjunction with this requirement is a limitation that fault protection shall bc designed assuming only onc fault occurs at a time. This means that a subsequent fault wiii occur no earl icr than the on-board response lime for the original fault, and that multiple fault detections occurring within the response time of lilt original fault arc symptoms of the original fault 13].

The rule by which faults arc assigned to either the spacecraft for autonomous handling or delegated to ground operations for the.ir intervention is that if a fault can bc handled by the ground, then do not provide for it on-board. Consequently, the set of faults that must bc detected and responded to on-board arc those faults that wiii disrupt the mission objectives or that will degraded the mission and can not bc handled within onc month by the ground [3].

CIFP Approach

The approach to CIFP error handling is based around two concepts:

1 I)esignation
2 Classification

Each error associated with the CDS is assigned a designation that specifics the CDS service or services the error affects. The error classification specifics the location and criticality of an error. The two prime categories of error classification arc noninterfering and interfering. This level of classification is based on whether or not the error interferes with a CDS service.

The. second tier of classification is different for noninterfering and for interfering errors. The noninterfering errors arc classified as either being message only, which means 10 log the error and continue, or action, which requires some noninterfering response must take place in addition to logging the error. Under the interfering category, the two sub-classifications arc temporary and permanent. Some of the peripheral errors, especially, can bc temporary in nature. Once they are resolved, tile affected CDS services can bc restored. However, the class of errors that is permanent in nature must bc resolved by redundancy management, i.e. by switching to the redundant unit of a pair and/or configuring and operating the spacecraft in a safe manner (safing).

In addition, because of the. nature of the. CDS architecture, errors arc also classified as being either central, affecting the CDEA, or peripheral, affecting the SSRs, REUs, or CDS Bus/RTCIU. The full structure of the Cassini CDS error classification scheme is as follows:

- Noninterfering
  - Message only
    - Central
    - Peripheral
  - Action
    - Central
    - Peripheral;
- Interfering
  - Temporary
    - Central
    - Peripheral
  - Permanent
    - Central
    - Peripheral

CIFP Architecture

The architecture of the. CIFP is based around four processes:

1 Fault detection to produce errors
2 Error filtering to produce events
3 Event activation rules to produce triggers
4 Trigger responses to isolate and recover from faults

In other words, the CIFP process begins with a distributed detection (termed exceedence testing) of conditions resulting from faults to produce errors that are recorded in an error Store. If the errors in this store pass persistence limits, an event is generated and recorded in an event store. The events arc acted upon by activation rules in conjunction with the current CDS state to produce fault response triggers. I'riggers are recorded in a trigger store, prioritized, and initiate both primitive and scripted responses to isolate and recover from faults. This process is illustrated proceeding from left to right in Figure 3.

Fault Detection -- The existence of faults is noted by either

1 Thcprcscncc ofananomaiyor
2 The absence of function.

The presence of an anomaly is noted through four types of detectors:

1 Binary	Value = O or 1
2 Compare	Value = Mask
3 Threshold	Value < Limit
4 Range	Limit 1> Value > Limit 2

The absence of function is noted through syndromes for ail nine CDS services. Syndromes arc indicators of periodic actions that arc required to take place in the flight software in order to provide a service. The existence of an anomaly or the absence of function is registered in a detected error store (Error ID, Time, Data).

Error Filtering - Even though the occurrence of an individual error may result in an autonomous response, the goal is to provide some insensitivity to transient errors through persistence checking. Consequently, error detections arc checked periodically to accumulate their persistences. If the occurrences arc contiguous and exceed a rmi t, an event is generated. This event is then registered in an event store (Time Tag, Event ID, Device ID, Data) and 1~.lc.metered to the ground.

Event Activation Rules -- Once an event is registered, activation rules arc applied to dctcrminc an appropriate response. Activation rules arc Boolean expressions applied to events, the condition or state of the spacecraft, and the class (vitalness) of the unit. The decision reached is called a response trigger and is registered in a trigger store (Time Tag, Response ID).

Trigger Responses -- If onc or more responses have been triggered, they arc prioritized and executed in a semi uninterruptable way. This process is illustrated in Figure 4 (see [4] for statechart conventions). The key features of this process arc (1) that the CDEA DFAR has highest priority and continues to run even in the presence of an interfering peripheral response, (2) that once a peripheral response has been initiated, it runs to compaction in an open loop fashion while excluding fault coverages of the other peripheral items, and (3) that the priority of peripheral responses is Bus/RTCI U, REU, and then S SRS last.

CIFP Design

The CIFP design makes usc of a state matrix that contains the indicators of the conditions of every device covered by CIFP. The indicators arc:

1 Power	On, off, trip, unknown
2 Vitalness	Vitai, nonvital
3 Primeness	Prime, backup
4 RT CIU Health	Well, sick, unknown
5 RT Unit Health	Well, sick, unknown
6 Ex istence	Alive, dead

Items 1,3,4, and 5 arc {ictcr-mined autonomously on the spacecraft and items 2,3, and 6 arc set by the ground.

Fault coverage is provided over all well units and in fact only well units are permitted to provide services,. If a unit fails, its health is degraded to sick. If a unit is powered off, its health is unknown. If the unit has failed, the ground can label it as dead and it is removed from existence. Figure 5 illustrates this health determination in a statechart format.

The difficult problem is that once a unit has failed, its services may still be needed for spacecraft functionality so autonomous restoration of units must be provided. To accomplish this, the, concept of vitalness is used to deter mine which functionality is required to be restored and which can be left for the ground to handle. i.e. the autonomous CIFP recovers the prime vital unit of a redundant pair following a fault.

In the process of restorat ion, succeedingl y more severe levels of response arc handled through higher levels of activation. The last level is always to reset CDS CDEAs. Figur c 6 shows the form that CIFP activation takes.

Acknowledgment

The work dcscribcd in this paper was carried out by the Jet Propulsion Laborator y, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

1 T.K. Brown and J.D. Donaldson, "Fault Protection Design of the Command and Data Subsystem on the Cassini Spacecraft", 13th Digital Avionics Systems Conference, pp. 408-413, October31, 1994.

2 D.P. Siewiorek and R.S. Swarz, Reliable *Computer Systems Design and Evaluation*, Second Edition, Digital Press, 1992,

3 C.E. Kohlhase, "Cassini Project Policies and Requirements Document", JPL PD 699-004 Rev C, June 1994 (JPL internal document).

4 D. Harel, "Statecharts: A Visual Formalism for Complex Systems", Science of Computer Programming 8. Elsevier Science Publishers (North-Holland), p 231-274, 1987.

Faults	Detect. (Exceedence)	Error Store	Filter (Persistence)	Event Store	Actuate	Trigger Store	Respond
Interrupt	Existence		Condition		Condition		Reset
Software	Existence	Error ID	Condition	Time Tag	Condition	Time Tag	Reset
Device	Status	Time	Occurrences	Event ID	State	Response ID	Reset
Syndrome	Function	Data	Duration	Error ID	Condition		Reset
Bus/Remote Terminal	Data		Occurrences	Data	State		Suspend and Reconfigure
REU	Data		Occurrences		State		Suspend and Recover
SSR	Status		Occurrences		State		Reconfigure

Figure 3 Detection, Filter, Activate, and Response (DFAR) Data Flow

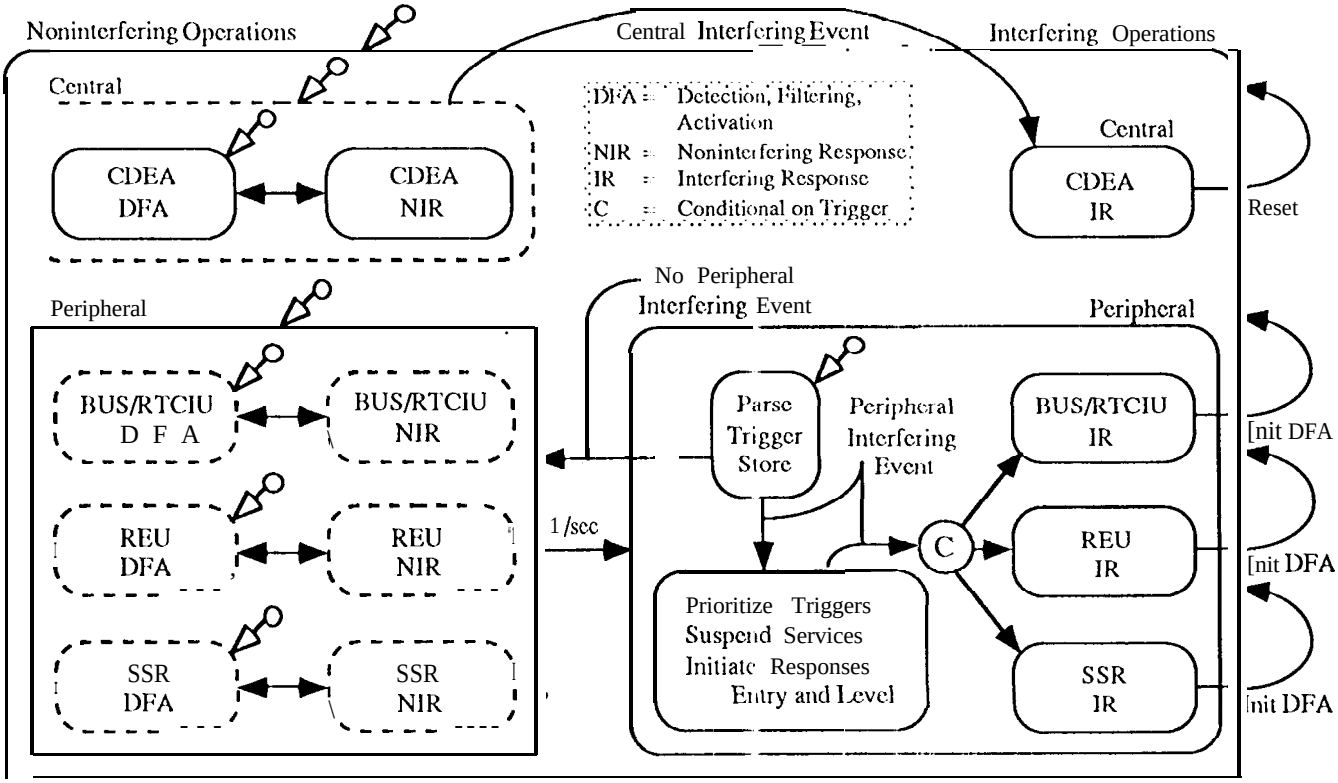

Figure 4 Detection, Filter, Activate, Response (DFAR) Statechart

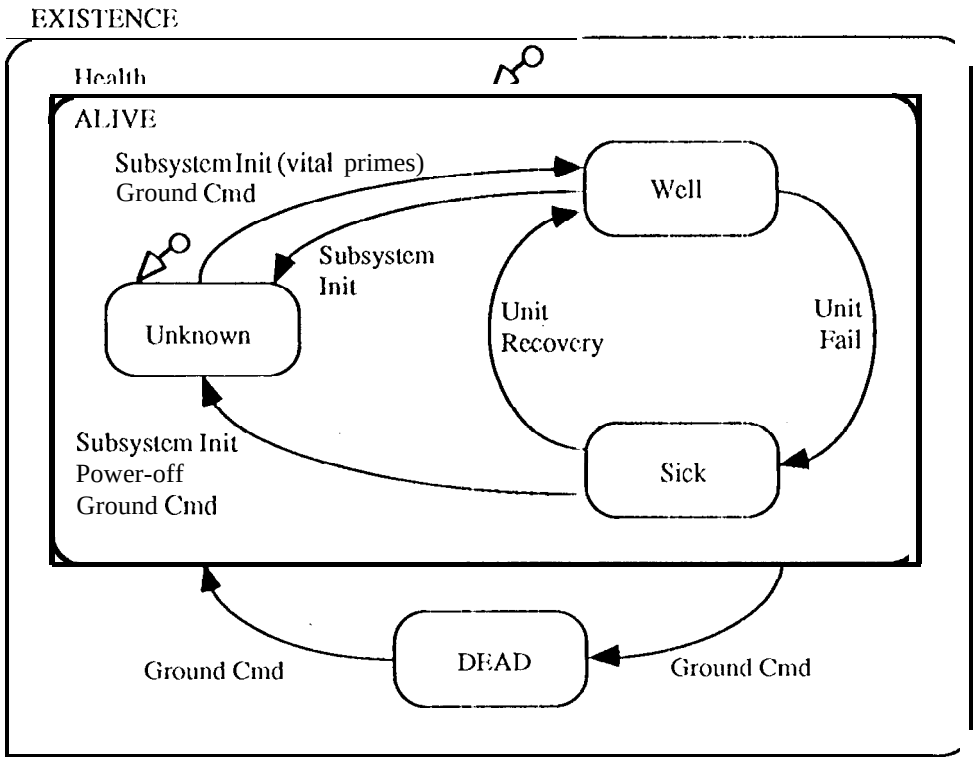

Figure 5 Unit Health Statechart

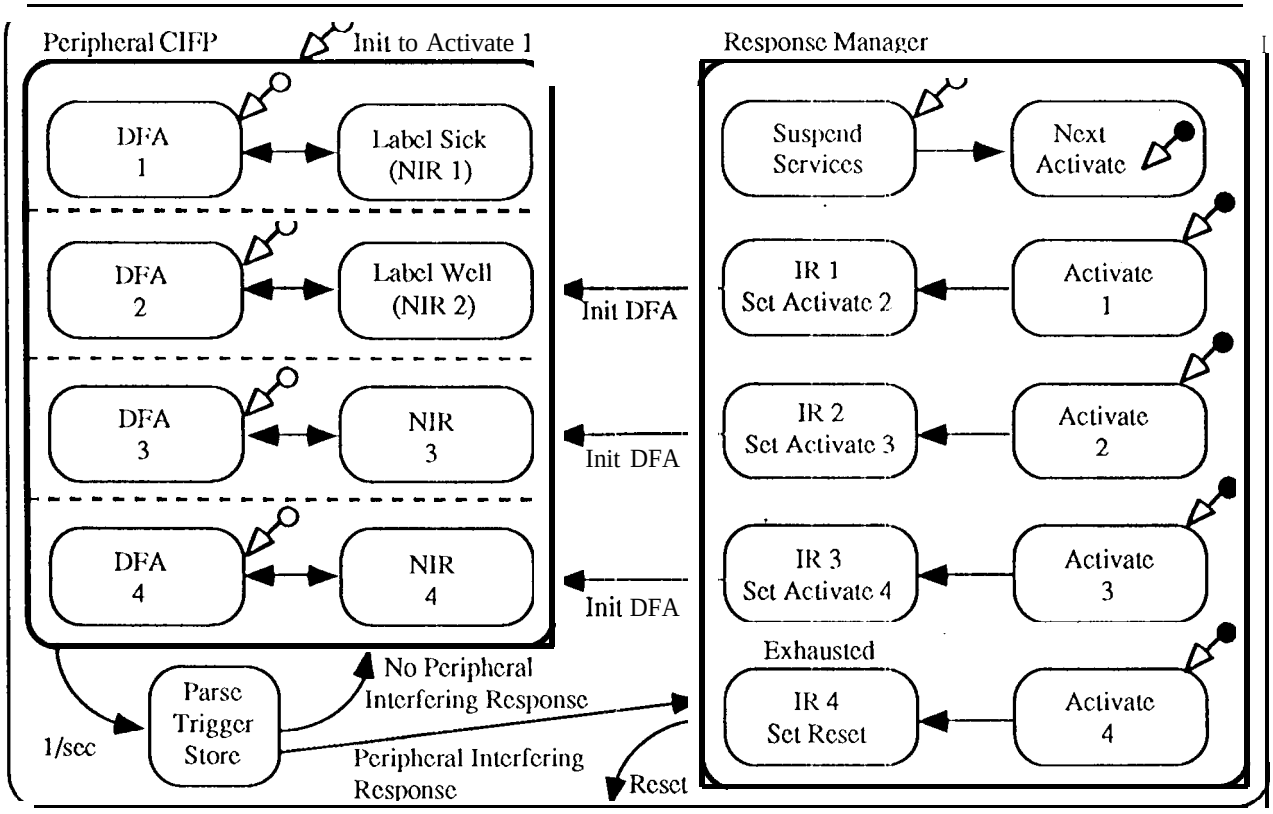

Figure 6 CIFP Response Action Level Statechart