Embedded Diagnostics in Combat Systems

Christopher Miles, Elena Bankowski

US Army TACOM, Warren, MI 48397-5000

ABSTRACT

Diagnostics capability of combat systems shall be compatible with the Army Diagnostic Improvement Program. Present systems are capable of performing health monitoring and health checks using internal embedded resources. They employ standard sensors and data busses that monitor data signals and built-in test (BIT). These devices provide a comprehensive source of data to accomplish an accurate system level diagnostics and fault isolation at line replaceable unit (LRU) level. Prognostics routines provide capability to identify the cause of predicted failure and corrective action to prevent unscheduled maintenance action. Combat system's health status and prognostic information are displayed to operator, crew, and maintenance personnel. Present systems use common data/information interchange network in accordance with standards defined in the Joint Technical Architecture (JTA) to provide access to vehicle's health data. The technologies utilized in present systems include embedded diagnostics, combat maintainer, schematic viewer, etc. Implementation of these technologies significantly reduced maintenance hours of combat systems. Health monitoring, diagnostics and prognostics of future systems will utilize federated software and probes approach. Gauges will determine if the system operates within acceptable performance bands by monitoring data provided by the probes. Health monitoring system will use models of missions to make intelligent choices considering tasks criticality.

Keywords: Prognostics, diagnostics, embedded systems

1. INTRODUCTION.

Prognostic is the process of predicting the future state of a system. Prognostic systems comprise of sensors, a data acquisition system, and microprocessor based software to perform sensor fusion, analysis, and reporting of results in real time. The Army is marching forward on embedded technology. The goal of prognostic is to provide vehicles and soldiers readiness assessment capability. This goal has been established to create a fighting force that achieves victory through superior striking speed, agility, and mobility rather than massive power. This level of responsiveness is enabled by information and communication technologies that provide all echelons of the fighting force with real-time situation awareness and rapid reaction capabilities. The ability to react much more rapidly than the enemy is on of the keys to achieving a decisive victory across the full spectrum of operations. Another goal is to reduce the logistics footprint in order to enhance the sustainability, deployability, readiness and reliability of ground systems. The vision for the future ground systems is based on moving accurate data through the command and control systems in real time. The end state objective is to move logistics data without soldier input. Embedded diagnostics and prognostics capabilities are key to making the battlefield command and control work as envisioned.

The capability of a system to monitor its own health and logistics status and self report this information can be leveraged to support other Army objectives such as readiness reporting, rapid deployment, reduced workload on soldiers, minimized logistics infrastructure on the battlefield, and lower life cycle costs. This requires new logistics processes and dramatic changes in current business processes to support the future force. These processes are focused on weapon systems and must be readiness driven, lean, and agile. They must detect and correct problems early, allocate resources where they are most needed, and continuously reduce labor requirements and cost. Combat vehicles are becoming progressively more dependent on software systems which run them (engine, transmission, power pack interface, gauge cluster unit and

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **10 FEB 2004**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE **Embedded Diagonostics in Combat Systems**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) **Christopher Miles; Elena Bankowski**	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **USA TACOM 6501 E 11 Mile Road Warren, MI 48397-5000**	8. PERFORMING ORGANIZATION REPORT NUMBER **13999**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) **TACOM TARDEC** 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES

U.S. Government Work; not copyrighted in the U.S. Presented at the 2006 SPIE Conference, The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**SAR**	**8**	

others) and these systems are becoming increasingly complex. Consequently, system reliability, a rapid development cycle, and the ability to predict system failures are critical. The lifecycle of a combat system consists of numerous iterations of requirements specification, implementation, testing, deployment, use and maintenance. These activities can run in parallel, and there is feedback between the steps as information is gathered. This feedback is crucial to the quality of the overall system.

2. TECHNICAL DISCUSSION.

Accomplishing this requires several new pieces of technology. These are listed below:

- **Probes**

Probes are able to capture events that occur in the software systems. These must span multiple software technologies used in the target software (e.g., Ada, CORBA, C/C++, etc.). They must be able to be inserted automatically to avoid software rework simply to install the probes. A variety of probes will be necessary in order to capture the various kinds of information desired. For example, probes that inspect a software environment checking for the presence or absence of required software are likely to be very different that probes that capture functions calls or messaging information. The probes must have negligible impact on the behavior of the software; particularly they must not interfere with operations by slowing response to a noticeable extent. While this is an issue in conventional testing where the act of monitoring changes the timing behavior, in testing environment it is misleading, whereas in operational environments it could be catastrophic. The solution to this is three fold:

1) Probes must be engineered to have minimal space and performance impact.

2) Performance thresholds must be defined and monitored for critical software interactions.

3) It must be possible to turn off monitoring when these thresholds are in danger of being violated.

Unlike testing environments, in operational environments the testing model is only implied. Thus, in addition to capturing the software interactions, probes must capture external stimuli from the environment and the users. In a test environment, these would have been generated by the test harness and recorded directly from here.

- **Gauges**

Gauges analyze probed event streams and report summaries/conclusions in a more usable fashion. Gauges output can be used in four distinct ways:

1) Gauges can feed back to previous software activities. Example of the feedback included detailed graphs of the developed software configuration compared to design specification, violations of timing constraints, logs of software exceptions thrown, summaries of functions calls, CPU usage of various modules, etc.

2) Gauges can be used to affect previous software lifecycle steps. A prime example is the use of gauges to validate that a testing model conforms to reality as observed by probes of the environment. Specifically, if testing is conducted according to a Marko model defining environmental patterns of stimuli and user responses, it would be desirable to validate that observed patterns are statistically compatible with such a model.

3) Gauges can control the operation of the target software, possible by triggering a reallocation of resources, shutting down nonessential functionality, and even compensating for errors.

4) Gauges can control the data collection activities by activating and deactivating probes to avoid impacting performance or to focus collection activities on suspect or critical areas.

3. VISION.

In the Army, most combat, combat support and combat service support platforms will include embedded diagnostic/prognostics systems. An embedded prognostics system adds additional capabilities to the embedded readiness monitoring system, enabling it to forecast the future state of the equipment. This software includes engineering models of the system, degradation models, sophisticated decision-making routines, and advanced trending algorithms. The current state of the system is compared to the models and the operating conditions to forecast the future state. If performance begins to degrade, the prognostics software predicts the point in time at which a predetermined failure threshold will be crossed. The prediction is updated continuously as new sensor data is analyzed. When the anticipated remaining life of a key component reaches a predetermined reporting level, the prognostics system determines what maintenance action is required and issues an alert/report. The tactical commander will be able to tailor the reporting criteria to the current phase of the operation through the command and control system. The readiness monitoring system also forecasts when the platform will require re-supply (or what kinds of missions it can carry out with current inventories of consumables.) This analysis is carried out by comparing current inventories of fuel, ammunition, and other consumables with consumption rates required to support the OPTEMPO projections inferred from commands issued through the C2 system.

Current technology will not support application of embedded prognostics to all systems/components on a platform. Prognostics can currently only be applied to components that fail through wear-based mechanisms. These types of components are generally mechanical, electric, or hydraulic. Electronics tend to fail when they are exposed to conditions (voltage spike, temperature extremes, shock, vibration, etc) outside their approved operating conditions. Health monitoring can be effective for electronic components, particularly if use and environmental conditions are tracked but the technology to accurately forecast their failure is currently cost prohibitive. For this reason electronic systems may be designed with multiple redundant (parallel) circuit paths, to allow for a circuit failure without loss of the system. Prognostics is the capability to forecast remaining life based on actual operational conditions. Since few legacy ground systems today have the capability to determine remaining useful life this capability is considered a future function more achievable in the Objective Force. It is not cost-effective to install additional sensors and embedded diagnostic/prognostic systems on all SBCT platform components. For example, tracks, structural members, hull components, and similar systems are likely to be outside the "view" of the embedded system, except through direct observations by the crew during preventive maintenance or routine checks and services. Business Case Analysis must consider these limitations when deciding whether it is worthwhile to install a particular capability for a specific application. In addition, the alert provided by a specific capability must be reliable enough, and provide sufficient warning, to enable the logistics system to respond in a manner that prevents or anticipates the resource need.

The Proof of Enablers Demonstration will use an Embedded Predictive capability to support Condition-Based Maintenance (CBM). CBM is dynamically scheduled maintenance based on the condition of the platform. Maintenance is required based on real time evidence to prevent equipment degradation or further degradation. CBM provides information on what and when a component will fail using previous data collected, current conditions, and regression analysis. If performance begins to degrade, the CBM software predicts the point in time at which a predetermined failure threshold will be crossed. The prediction is updated continuously as new sensor data is analyzed on-board the platform. A Modern Weapon System is composed of many processing elements as shown in Figure 1. The processing elements are grouped into seven categories, shown as pie segments. Typically, each processing element represents a software program that operates on a dedicated processor. Most often, the elements are co-located inside of the LRU that contains three to nine processors. Current vehicles have more than half of the total processing activity performed on embedded processors. Figure 2 shows the processing elements, colored blue, that are embedded. At the time of the design of current ground systems and its follow on variants, processor and bus technology did not have the speed or capacity to accommodate true distributed systems.

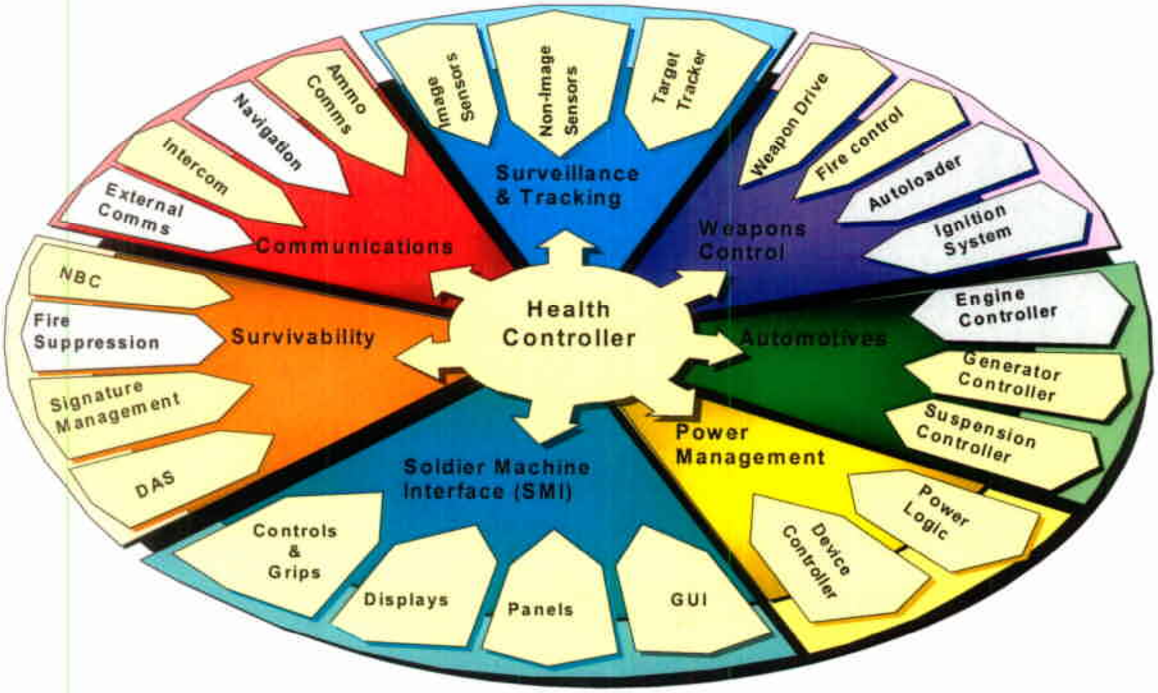

Figure 1. Embedded Processing System's Health Controller.

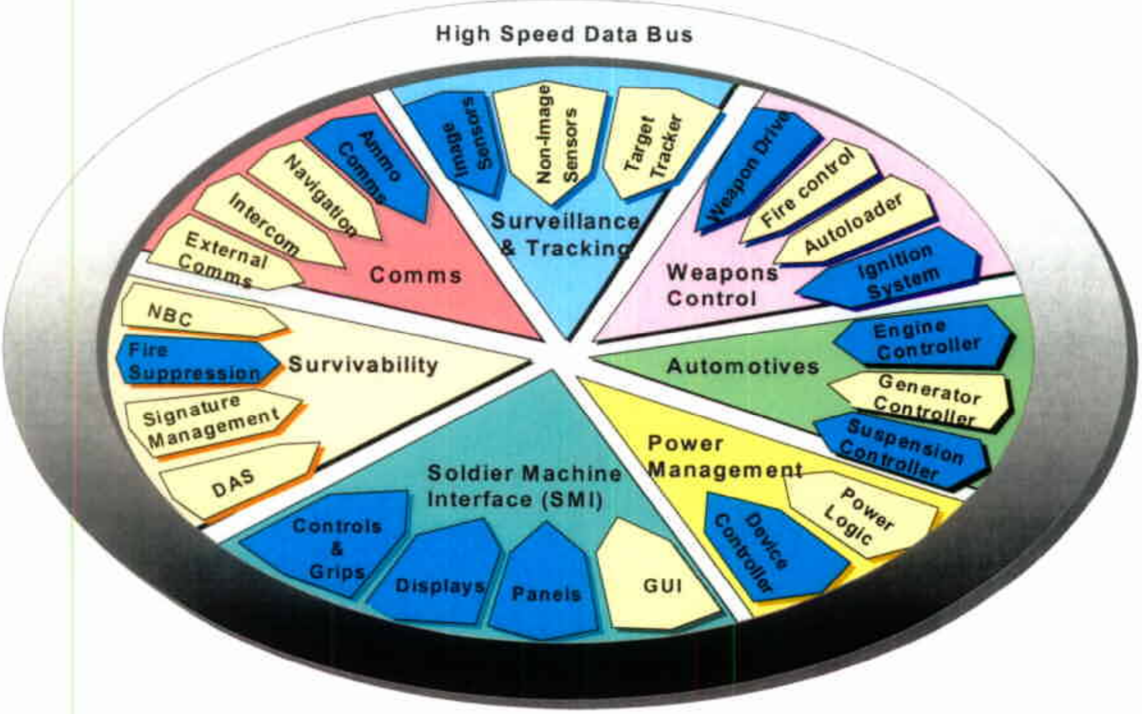

Figure 2. Embedded Processing System's Elements.

The Health and Situation Controller, see Figure 3, provides the Probe/Agent Controller, Health and Criticality Scoring and the Auditor Function. The Health and Situation Controller provides the following capabilities that contribute to system assurance.

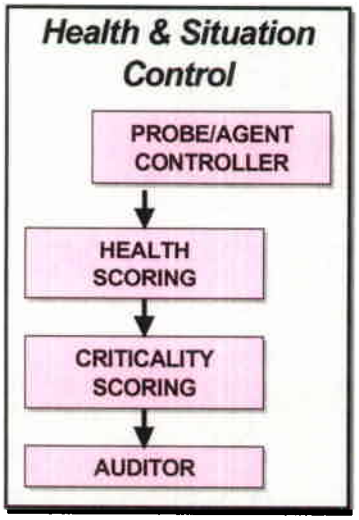

Figure 3. Health and Situation Controller.

- **Probe/Agent Controller:** This activity will determine the frequency, format, type and distribution of queries that the Health Controller will send to the elements so that element health can be determined.
- **Health Scoring:** This activity will evaluate collected and processed data to determine if it lies within the output bands of the element.
- **Criticality Scoring:** Based on the nature of the mission, and the battlefield situation, this activity will prioritize vehicle functions that require processing.
- **Auditor:** This activity will compile and report the collective operational and health status of all of the elements to the Health Controller.

This health monitoring experiment provides software parallels to hardware survivability. The software attacks were encountered in a cyber battlefield. Health monitoring provides protection and compartmentalization of software with the same diligence applied to armor protection of hardware.

The following conditions will require the Health Controller to reconfigure the system:

- Software Failure
- Hardware Failure (Catastrophic such as Battle Damage)
- Hardware/Software Failure (Graceful Degradation)
- Hardware Installation
- Hardware Removal
- Software Installation or Upgrade
- Change in Mission (In-route System Optimization)
- Reorganization of Operations Unit: Change in complexion of Unit. Example:
- Re-assignment of Weapon System Duty Role
- Transfer of TOC
- Training and Simulation

- Figure 4 illustrates the reconfiguration process for the first condition "Software Failure". Reconfiguration due to "Software Failure" has the following steps:

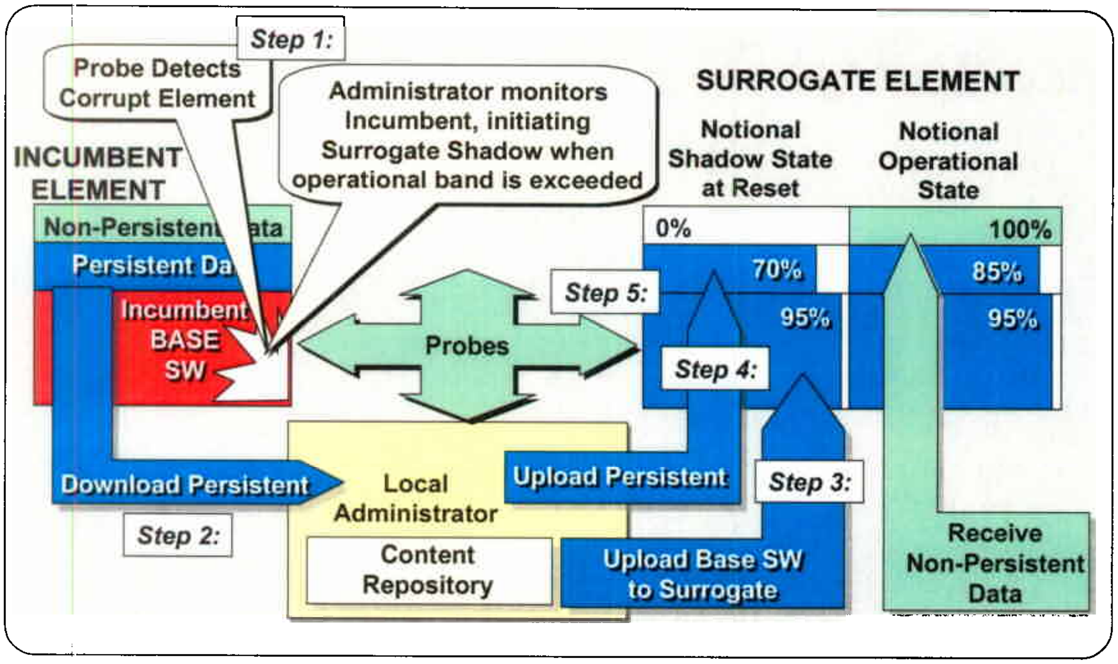
Step 1:
Probe Detects Corrupt Element
Administrator monitors Incumbent, initiating Surrogate Shadow when operational band is exceeded
SURROGATE ELEMENT
Notional Shadow State at Reset
Notional Operational State
INCUMBENT ELEMENT
Non-Persistent Data
Persistent Data
Incumbent BASE SW
0%
100%
70%
85%
95%
95%
Step 5:
Probes
Step 4:
Download Persistent
Local Administrator
Upload Persistent
Step 3:
Step 2:
Content Repository
Upload Base SW to Surrogate
Receive Non-Persistent Data

Figure 4. Reconfiguration Due to Software Failure.

Step 1: A probe detects a symptom of a software-manifested failure. The Health Controller determines that probe reconnaissance has returned data that is out of the operational bands of the processing element. Figure 14 illustrates a flow chart of the Health Check Process.

Step 2: The Health Controller algorithmically computes the next viable state of the software architecture considering the processing assets available and mission requirements.

Step 3: The Health Controller dispatches a new operational publication of the failed software to a new host processor. Once the transfer is complete, the surrogate is brought on line with a replicating processing effecting repair/replacement of the incumbent. The Content Repository issues the operational publication. Contained in the Content Repository are programs or program objects that dedicated processors host (For example: Ballistic Program, Auto Target Tracker, and Engine Controller).

Step 4: Persistent object data is either extracted from the defective element or retrieved from a synchronized replicator and transferred to the new host (surrogate) as part of the replicating process. If the defective element fails catastrophically, object data will have to be reinstated with default values or acquired real time data. The resident programs use Persistent Data. It is unique to the mission or the systems on-board and is maintained in non-volatile memory when the system is powered down. (Example: Round Zero, and Total Engine hours) Non-Persistent Data is time sensitive, it is stored in volatile memory and is typically re-sampled at initialization. (Example: wind speed, vehicle speed, and target-lead angle.)

Step 5: Dispatched Probes validate the performance of the surrogate. The Health Controller will bring the surrogate on-line and retire the incumbent at time consistent with the operational state and usage of the element.

Throughout the reconfiguration process, the Health Controller will employ the following tools:

- **Probe Controller:** The Probe is an agent of the Health Controller, reporting the health of the weapon system elements. Off-vehicle probes are also launched to assess the health of companion vehicles within the Operations Unit.
- **Data Replicator:** A tool for restoration of persistent and non-persistent data from the incumbent element. More complicated issues arise where elements take different approaches in the way they manage persistent and non-persistent data
- **Reconfiguration Tools:** A process that will download data from the content repository, adjusting it to accommodate the new host, and load it into the surrogate element capturing the history of that process. The Reconfiguration Tool requires data matrix algorithms that map data from content repository to the new host. Replicating Tools are also required to control consistent hand-off from incumbent to surrogate.
- **Maintenance Tools:** An automated meta-data population and maintenance tool.
- **Decision Tools:** Will provide mission criticality analysis and the evaluation of Operational Bands limits of the Elements
- **Security Tools:** Access Security embraces Authorization and authentication of the candidate user. Virus Security is very much the same as provided behind the firewalls of corporate enterprise computing systems.
- **Communication Tools:** High performance communication is necessary to provide the portal to the elements on the local data bus. Also required to provide interoperability across domains (to companion weapon systems).
- **Translators:** Translators are used for modifying and adapting distributed elements and object data to conform to a new host. This will provide transparency to the replacement of the incumbent with the surrogate.

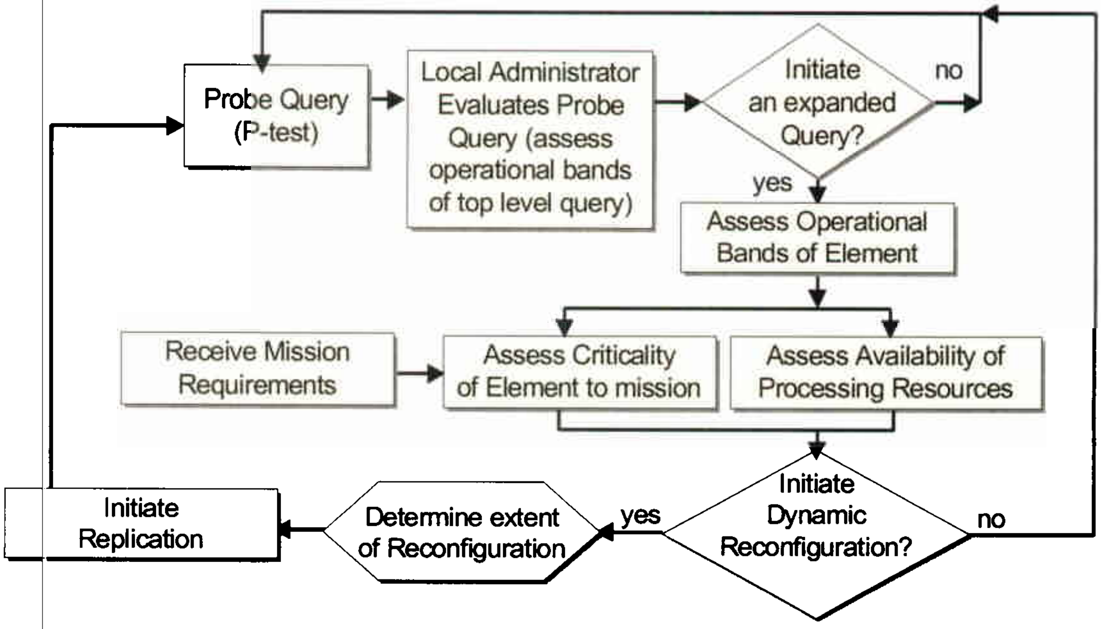
Probe Query (P-test)
Local Administrator Evaluates Probe Query (assess operational bands of top level query)
Initiate an expanded Query?
no
yes
Assess Operational Bands of Element
Receive Mission Requirements
Assess Criticality of Element to mission
Assess Availability of Processing Resources
Initiate Dynamic Reconfiguration?
no
yes
Determine extent of Reconfiguration
Initiate Replication

Figure 5. Controller System Health Check Process.

Figure 5 illustrates the Health System check process employed in Steps 1 and 2 above. A closed-loop system evaluates the operation of the incumbent and the surrogate before handoff.

4. CONCLUSION.

The technologies utilized in present systems include embedded diagnostics, combat maintainer, revised maintenance concept, schematic viewer, etc. Implementation of these technologies significantly reduced maintenance hours of combat systems. Health monitoring, diagnostics and prognostics of future systems will utilize federated software and probes approach. Gauges will determine if the system operates within acceptable performance bands by monitoring data provided by the probes. Health monitoring system will use models of missions to make intelligent choices considering tasks criticality. Prognostics of combat systems LRUs will be based on probes data and statistical usage models.

REFERENCES

1. Elena Bankowski, Christopher Miles, Michael S. Saboe, Peggy Gilbert, "Health monitoring and diagnostics of ground combat vehicles", *Proceedings of the SPIE,* **5049**, 138-145, 2003.

2. M. S. Saboe, P. Gilbert, A. Kouchakdjian, "Applying Statistical Usage Testing (SUT) on a High-Complexity Application", *Proceedings of the Workshop on Statistical Methods in Software Engineering for Defense Systems,* National Academy of Sciences, Washington DC, July 2001.