HEALTH AFFAIRS

THE ASSISTANT SECRETARY OF DEFENSE

WASHINGTON, DC 20301-1200

August 12, 1997

MEMORANDUM FOR: MILITARY HEALTH SERVICES SYSTEM INFORMATION MANAGEMENT/INFORMATION TECHNOLOGY EXECUTIVE AGENTS

SUBJECT: Policy for Military Health Services System Automated Information Systems Performance Standards

This memorandum establishes Office of the Assistant Secretary of Defense (Health Affairs) (OASD(HA)) policy related to performance standards for Military Health Services System (MHSS) Automated Information Systems (AISs). By setting AISs performance standards, the MHSS will achieve improved and consistent overall system performance and provide a higher level of service for all users of the MHSS Enterprise.

Each MHSS Information Management/Information Technology (IM/IT) Executive Agent will establish performance standards for automated information systems within their business area. Standards should address both functional and technical transactions and should reflect user and operator requirements. At a minimum, Executive Agents must establish standards that are sufficient to measure performance from the user's perspective. Additionally, Executive Agents will monitor and regularly report to the MHSS Information Management Project Review Board (IM PRB) operational system performance against these standards. Executive Agents will develop a Performance Management Plan that documents performance standards and a plan for data collection and reporting against the standards for systems within their business area. Executive Agents will also establish performance levels that determine the need for corrective action and describe the general nature of that action.

Attached are examples of the performance standards which have been developed for existing MHSS AISs. Also attached is an information paper on performance monitoring and reporting developed by Defense Medical Information Management (DMIM).

Should you require additional information, my point of contact for this policy directive is Ms. Clarissa Reberkenny, DMIM, who may be reached at (703) 681-8823 or by electronic mail at creberke@ha.osd.mil.

Edward D. Martin

Edward D. Martin, M.D.
Acting Assistant Secretary of Defense

Attachments:
As stated

Performance Standards:

The following performance goals/standards have been documented by several MHSS AIS. They provide a set of reasonable targets for comparison with the requirements from other user communities.

	Goal / Standard
CHCS	
Display patient appointment	2.5 sec
Cancel nursing order	10 sec
Cancel laboratory order	10 sec
DEERS Check	3 sec
File nursing order	2.5 sec
Complete CHCS logon	30 sec
Read mailman message	5 sec
Send a mailman message	5 sec
Display next screen	1 sec
Enter laboratory order	10 sec
Enter nursing order	10 sec
Retrieve single laboratory result	3.5 sec
Review multiple results, 1 patient	7 sec
Retrieve single pharmacy order	3.5 sec
ADS	
Query one record from CHCS	4 sec
Paint Screen after CHCS query complete	3 sec
Send a walk-in encounter form	30 sec
Print a walk-in encounter form	50 sec
Display a medium (ex. MEPRS)	18 sec
Display a small pick list (Provider)	9 sec
Scanner on application server	2.6 to 8.7 sec per form
Scanner on database server	1.7 to 4.5 sec form
CIS	
Character echo time	<1.5 sec
Patient selection	<1 sec
Review data on another machine	<1 sec
Begin printing a 4 page report	<30 sec
Single-sheet screen copy, including printing	<1 min

Performance Monitoring and Reporting:

Performance monitoring consists of the measurement of system performance (generally by functional transaction as specified in the standards) and the storage of that measurement for use locally or centrally. There are several methods of system performance measurement which may be used.

1. **Manual timing and recording**. This is the simplest method but obviously a labor intensive approach to measurement. It is also one that seldom produces repeatable results for short response times. Additionally, the nature of the collection inevitably translates into data that is collected too infrequently and not over required time periods. Lastly, data collection and storage is often delayed meaning reports are not timely.

2. **Internal application self-timing**. This method of timing (sometimes referred to as "Instrumenting the Application") should be the most thorough as it is always available and can be extended to include new application functionality at the

time that the new functionality is written into the application. In general this method will produce the most data and do so in an automatic fashion. Unfortunately, most existing applications will be extremely difficulty to retrofit. Drawbacks also include the lack of timing of components not in the application proper. In particular, this applies to host-terminal systems where the instrumentation is in the host and neglects to include any timing of the communication channel from the terminal to the host. In a client-server environment where the measurement must be placed in clients, that does not apply. Another negative to this method is that the mere presence of the measurement software often can impact performance. Even if the measurement function can be disabled so that the application need not be affected most of the time, the measurements that **are** made will be affected and of reduced value. The measurement function may be designed to minimize this impact.

3. **External automated timing**. This method uses a separate, external device which looks to the System Under Test (SUT) like a "standard" terminal or client. Traditionally, the external device is called a Remote Terminal Emulator (RTE). Strengths of this method include a relative ease in adding new events to be monitored through the RTE's scripting interface. A large strength is that multiple applications (on the same or different platforms) can be monitored by a single RTE including recording of system resource utilization data (i.e. CPU and disk utilization, etc). Also, the RTE will exert no more load on the SUT than a single user would. Problems with this method include the fact that it generally can only measure events which are triggered by the user-interface (i.e. keyboard transactions) and complexity in aggregating the data for centralized reporting. Also, because transactions from the RTE will appear to the host as regular, valid transactions, they will actually cause entries to be made in the operational data base and will require "dummy" database records to update. CHCS uses this method with an RTE called the Performance Monitoring Tool (PMT).

4. **Internal system resource consumption measurement with extrapolation to response times**. This is not really a separate data collection tool, but rather a different set of data to be collected. This data is the VERY technical data about the internal workings of the computer or operating system. In general, the data will be provided by tools that come standard with the operating system itself as only that lowest level of software is capable of deriving this information from the hardware. This data should be collected in conjunction with application timing data because this is the actual data which will be used to **correct** system performance problems or to forecast problems in the capacity planning process. The data generally only has relevance for a very short time (seconds to minutes) and so need not be kept long except for comparison purposes. Examples of the data are:

    CPU-busy percentages
    Memory utilization
    Disk utilization
    Interrupt frequency
    User/System/I-O state ratios
    I/O channel utilization
    Processor queue lengths
    Port/Controller queue lengths

    If this type of data are collected in conjunction with application level transaction timings for each of the hardware platforms on which the application is running (and assuming that no other major application is sharing that platform), correlation may be drawn between the resource utilization (i.e., system state) and observed application performance. Thus, theoretically, with a sufficiently robust set of data, application performance can be deduced from the easily-collected resource utilization data and need not be collected separately. In practice, this correlation is extremely difficult to derive and changes with each new release of the application and with each hardware platform deployed.

Once the response times are captured by one or more of the methods above, the data representing those times must be stored for later reporting and analysis. Several methods are available for that data storage:

1. Data may be stored on the SUT itself. This makes the data available to any authorized person who has access to that system. However, it means that the data will be resident on the same system as the application with potential security implications. Also, this arrangement may make remote access to the data for centralized performance monitoring more

difficult. Additionally, this means that multiple versions of the data capture, storage and reporting software may be necessary for operation on multiple types of computers. Licenses will be necessary for all SUT platforms.

2. If an external monitor or RTE, as in method 3 above is used, data may be stored on the RTE. This isolates the data from the application it is measuring and thus alleviates some security problems. Assuming the RTE is a PC-class platform, data storage is relatively inexpensive. Only one version of the monitoring, storage and reporting software would be needed and a single RTE and copy of the software would be needed per site.

The following items should be considered for reports of the captured performance data:

- To the maximum extent possible, reports should be available on an exception basis. This would allow reports to be produced only when and where performance was below expectations. This is a critical requirement as periodic (monthly or more often) reports on every system at every installation will produce mind-numbing volumes of data.

- Graphical representation of the data will allow communication of salient results and trends without requiring inspection of large columns of numbers.

- Reports which use indexed data (instead of reporting the number of seconds required for a specific transaction, use the number of seconds divided by the standard for the transaction so that response better than the goal is a negative number and worse is positive) provide an easy method to compare the results for various transactions.

- Composite "scores" of indexed data across time ranges, across transactions, even across sites can provide overall scores for very quick assessment of general system performance. However, these composite or average scores can easily hide localized problems and should not be the only data presented. If composite scores are used, it is highly recommended that the raw timings be kept for later adjustments and "data dredging".

- It is possible to weight performance scores for valid reasons. One type of transaction may be deemed "more important" than others for performance purposes and may therefore be given a larger positive or negative score. Also, some transactions will be used more frequently than others and may therefore rate a higher weight. If weighted scores are used, it is highly recommended that the unweighted scores be kept for later analysis.

- Reporting of various statistical measures, such as maximums, 95th percentile scores, standard deviation, etc. can provide a method to describe the distribution of the data to the statistically knowledgeable user.

- Reports of current data should also include recent past performance so that trends can be recognized and anticipated. Also, system events which took place and would have affected performance should be noted on the reports to explain changes in performance. Where performance upgrades or other changes are planned for a site, those plans should be noted on the performance reports.

- Performance reports should be provided to the management at the site as well as to the Program Managers. In addition to providing the user with feedback from this process, it provides a convenient validity check on the data. If the system is reporting conditions not actually being experienced by the site, they will be quick to correct the data.