BEACON/GUARDIAN: INSTALLATION MANAGEMENT SOLUTIONS IN SEARCH OF ELUSIVE PROBLEMS

WAYNE E. HOLT
WHITMAN COLLEGE

I SYNOPSIS

Solutions in search of problems? It sounds odd but really isn't. These two software modules are solutions to a very specific type of installation management problem, and frankly would <u>not</u> be useful to most shops.

Guardian is a security module designed to increase the security provisions on both data files and program files in an environment where multiple users work in the same group and account, yet each have varying, overlapping, and often contradictory responsibilities. It is especially designed for the type of site where Users are totally responsible for data entry and report generation, both on-line and batch, rather than a site where the central DP shop handles everything.

BEACON (<u>B</u>atch <u>E</u>ntry <u>A</u>ccess <u>CON</u>trol) is a companion module to Guardian. If the site allows Users to stream jobs at will, someone or something has to play policeman and handle the traffic according to pre-defined rules (or in-flight instructions) in order to prevent the system from becoming clogged and dying. BEACON handles all job schedules, both regular cyclic and demand-mode jobs; it controls the number of jobs allowed to run simultaneously based upon system load and time of day; it modifies quantum, tpri, cpri, and dpri based upon time of day; it provides the DP Center with a concise monitor of all computer activity in a convenient format on any designated terminal; and it provides an easy to use method of rescheduling or expediting jobs.

These software modules are designed to operate in a User-oriented data processing center. The application packages that are used are specifically designed to enable a non-sophisticated User to very quickly learn to operate the terminals and become productive. Smart terminals (2645A's) with softkeys provide the tools needed to allow this type of user to request jobs with little or no understanding of the events set in motion by the request. Without BEACON/Guardian, such a shop would be difficult to manage successfully.

II DATA PROCESSING PHILOSOPHY

Whitman College is typical of many of the current wave of "emerging" computer users. It had enjoyed a very modest degree of involvement with data processing for many years, primarily for business and financial uses. Such involvement called for few resources to be expended since the needs were correspondingly few. The times changed and it found itself unprepared to compete with those colleges that were using modern information technology to pursue a diminishing supply of student applicants.

Because of its size, it would be quite difficult to create and staff a large Data Processing Center, complete with both Systems and Operations personnel. The only viable alternative was to establish a Center whose philosophy was oriented toward total control by the User of the actual "data processing" function. The systems design and programming functions were retained by the Center, with the User required to know little more than :HELLO and :BYE in order to be fully effective.

A smart terminal, the HP2645A, is the hardware link that allows the User-oriented software to function smoothly. Application programs using either DEL or VIEW enable sophisticated screen techniques to facilitate data entry by non-DP personnel. These programs download the terminal softkeys with run instructions in such a manner that the User needs only to press a single key to initiate a job.

The actual technique is relatively simple (refer to Figure 1). After first loading the softkeys by means of a cassette tape or computer program, the User presses the key that best describes the type of work needed to be performed. This causes a program to be initiated that presents the User with a menu (Figure 2). The User selects the job function to be performed by keying an "X" in the appropriate box. Note that the menu lists items in plain English, using phrases that the User understands. The menu program then downloads a softkey with the necessary :RUN or :STREAM command, along with generating any required :FILE statements. The User then presses that key when ready to begin work.

The drawbacks to implementation of this type of philosophy is that it causes problems in two major areas: security and system performance. For these reasons, Guardian and BEACON were designed. They are the software tools that make this kind of environment manageable.

III THE GUARDIAN SYSTEM

The MPE file management system provides levels of security that perform quite well in most situations, particularly where a central organization is responsible for production job set-up and related processing. It still functions in a user-controlled environment, but not with the same degree of effectiveness.

Consider, for example, a Registrars Office. This type of office usually has a mixture of regular and part-time staff, each with varying degrees of responsibility. Even among the regular staff, certain functions such as grade adjustments can only be performed by specific individuals within the office. When this office is automated in a manner described earlier, the traditional approach of adding lockwords can cause more security breaches than it prevents. Requiring a User to learn multiple passwords usually tends to promote a casual attitude toward such things.

Some Users, when faced with such an array of mumbo-jumbo, have actually been known to post the passwords on the Terminal itself!! A User should only be required to know his/her own unique Log-on, and it should NOT be shared with anyone else or known by anyone else except, of course, the System Manager.

The approach taken by Guardian is very simple. The Security File contains an entry for every valid User and the corresponding programs that he/she is allowed to run. All production software calls Guardian to validate each request to run a program. Guardian also checks the authorized run time and User terminal number as well, treating any violations as potential security breaches.

Further, all important data files are lockworded. Users do NOT know these lockwords; rather, they are known only by Guardian, which supplies them when the files are opened. This implies that Users have no access to their data except through approved software that interfaces with Guardian. Unfortunately, this also provides an obvious path for security breaches, since only a single password is needed to run any particular program to which a given User has access. On the other hand, all approved production software creates standard audit trails that can be analyzed for irregularities. Thus, the "most obvious path" is also the most dangerous for any potential security violator.

This software technique, oriented toward the specific User, is coupled with normal MPE methods to provide a high level of security for all situations. There are two program modules in the Guardian system. Refer to Figure 3 for the interrelationships with the BEACON system.

Guardian. This module is a called subroutine, executed by any program in production mode. it is responsible for

- o Authorizing a specific User to have access to a given program,
- o Verifying the time of day of use of the program,
- o Verifying the location of the terminal being used, and
- o Opening all required files for the calling program.

It returns status flags and file buffers to the calling program upon successful termination.

Sentinel. This module is a stand alone program that is used to build entries in the Security File. It is used to create and modify the records that provide Guardian with the information it uses to verify log-on requests by Users. Using prompt-answer techniques, it presents all known programs and jobs to the Manager for yes-no responses in relationship to a User's capabilities. It uses the Activity File in the BEACON system as a source for this information.

IV THE BEACON SYSTEM

When Users have the capability of performing almost any task (data entry, report generation, file maintenance, etc.) whenever they choose to do so, the computer system is not going to perform well. The Computer Center could set the system limits down to ensure good response time in a FIFO (First In First Out) situation, but overall throughput would suffer at the hands of such arbitrary measures. Constant operator intervention would be required to manage such a situation, thus defeating the whole purpose.

BEACON is designed to remedy this situation. Fundamentally, its primary purpose is to take a batch job request from a User and run it at the appropriate time in an efficient manner. In order to do this, BEACON actually performs a wide variety of tasks. These tasks are split between various program modules. Please refer to Figure 3.

Gopher. This module is a called subroutine, executed by any program that requests a job to be launched. The calling program passes the name of the job to be launched and any associated run time parameters to Gopher, which in turn builds a Job Request Record in the Activity File. It builds the record based on the contents of a Priority Record permanently resident in the Activity File, which includes

- o Execution priority on a scale of 1 (high) to 9 (low),
- o Literal Description of the Job,
- o Normal start time (HH:MM), and
- o Average run duration in minutes.

The Job Request Record is built as a "one time" record and will eventually be deleted when launched by BEACON. A Wizard number is assigned to each Job Request Record and is returned to the calling program by Gopher.

Wizard. This module is a stand-alone program that should be run by the operator or account manager. It is designed to perform a wide variety of tasks associated with running the BEACON module.

It is the tool that is used to modify the contents of the Job Request Records in the Activity File when responding to spot requests by the Users. Any key field can be altered -- date, time, or priority of launch. Also, hot jobs can be placed in a demand-mode launch priority regardless of the System load if the priority is altered to "0".

One of its most important functions is to build a Job Request Record for "permanent" or "cyclic" jobs. These records are identical to the "one-time" Job Request Records except that the run interval prevents them from being deleted by BEACON after launching.

Wizard also maintains the Clock Records on the Activity File. These 24 records control various parameters that affect system performance:

- o Default execution priority for batch jobs,
- o Maximum execution priority for both batch and sessions,
- o Quantum, tpri, cpri, dpri,
- o BEACON delay interval,
- o Video monitor LDEV number assignment, and
- o BEACON status flag.

Wizard can modify any of these items; BEACON will implement them automatically. This allows an unmanned system to "tune" itself as the workload shifts during the 24-hour day.

Snoopy. This module is designed to be run by the User as well as the operator. It produces a formatted display (on either the terminal or the line printer) of all jobs waiting to be launched and their expected launch times. Snoopy takes into account the average run times as well as the System Job limits from the various Clock records in order to create as accurate a forecast as possible.

BEACON. The BEACON module itself is actually quite simple and very efficient in terms of system resource utilization. It is designed to be constantly "up" as a streamed job although it is normally in a "sleep" mode. It could be run in an interactive session if there were an unused port, but this is not likely. It can be started and stopped by Wizard, which also controls the frequency of waking up.

Upon waking up, BEACON checks the System clock and reads the appropriate Clock Record in the Activity File. It then adjusts any system parameters that may have been changed since the last time that it was awake.

Next, it reads the Job Request Records on the Activity File in a sequential mode and determines if a job is ready to be launched by comparing the time fields. If there is a priority "0" (demand-mode) job in the queue it will be launched immediately. BEACON will continue searching the Job Request Records until they are all examined or a "launch ready" status is determined for one of them. If no job is ready to launch, BEACON will skip on to the next step. Otherwise, it checks the system load and determines whether to launch or not.

If a job is launched, BEACON will then log the System Job number and the exact launch time in the Job Request Record and do a KSAM delete. Note that this only "flags" the record for deletion and renders it invisible to BEACON. It will also regenerate any "permanent" Job Request Record with the new launch date/time.

The next step is optional, depending upon a flag set in the current Clock Record. If a valid LDEV has been specified, BEACON will output a current system status display, showing important system performance data. This display could easily be connected to a large video monitor for informational display purposes.

BEACON then goes to sleep for the period of time specified in the current Clock Record.

Reporter. This module is used to generate an Activity Summary report that analyzes BEACON's performance. It is run prior to the daily FCOPY "cleanup" of the deleted records on the Activity File. It finds the Job Request Records of all launched jobs and compares expected launch time with actual launch time and calculates various statistics. Such a report is valuable in setting the values in the Clock Records in the Activity File. It also matches the Wizard number with the final Job number, as an aid in tracking down "lost" jobs.

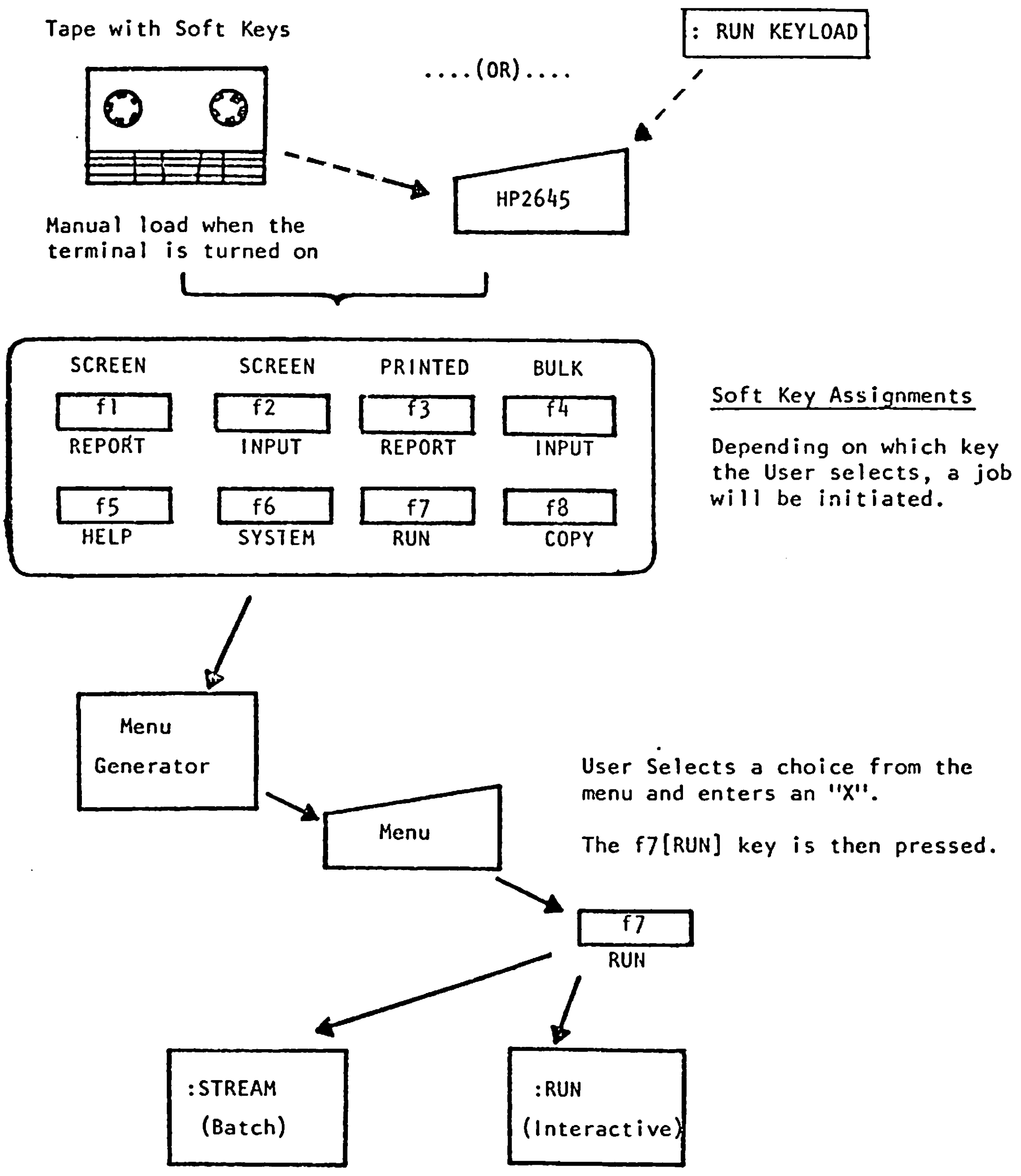
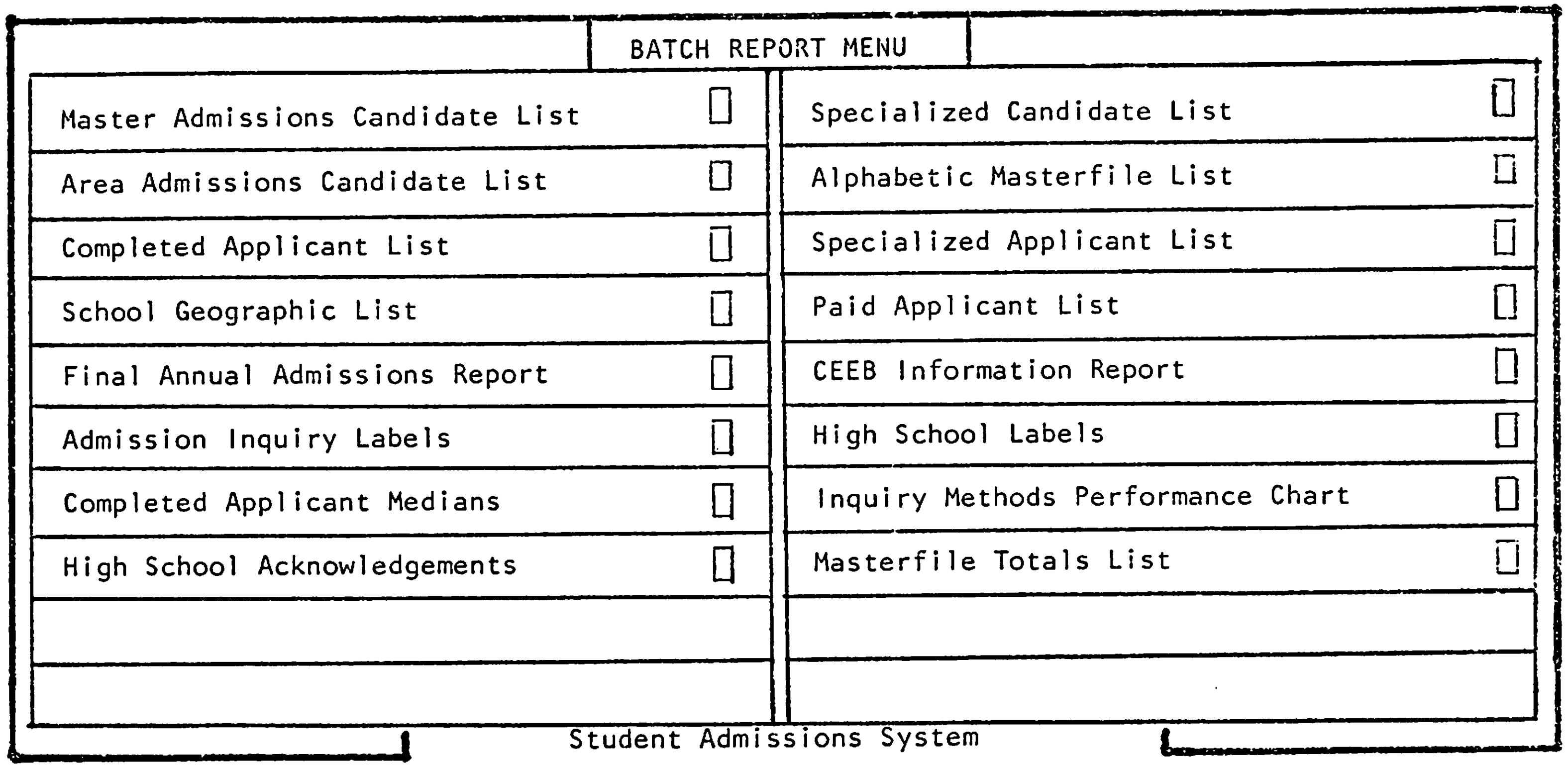

Tape with Soft Keys
....(OR)....
: RUN KEYLOAD
HP2645
Manual load when the terminal is turned on
SCREEN
f1
REPORT
SCREEN
f2
INPUT
PRINTED
f3
REPORT
BULK
f4
INPUT
f5
HELP
f6
SYSTEM
f7
RUN
f8
COPY
Soft Key Assignments
Depending on which key the User selects, a job will be initiated.
Menu Generator
Menu
User Selects a choice from the menu and enters an "X".
The f7[RUN] key is then pressed.
f7
RUN
:STREAM (Batch)
:RUN (Interactive)

Figure 1



BATCH REPORT MENU

Master Admissions Candidate List	▯	Specialized Candidate List	▯
Area Admissions Candidate List	▯	Alphabetic Masterfile List	▯
Completed Applicant List	▯	Specialized Applicant List	▯
School Geographic List	▯	Paid Applicant List	▯
Final Annual Admissions Report	▯	CEEB Information Report	▯
Admission Inquiry Labels	▯	High School Labels	▯
Completed Applicant Medians	▯	Inquiry Methods Performance Chart	▯
High School Acknowledgements	▯	Masterfile Totals List	▯

Student Admissions System

Figure 2

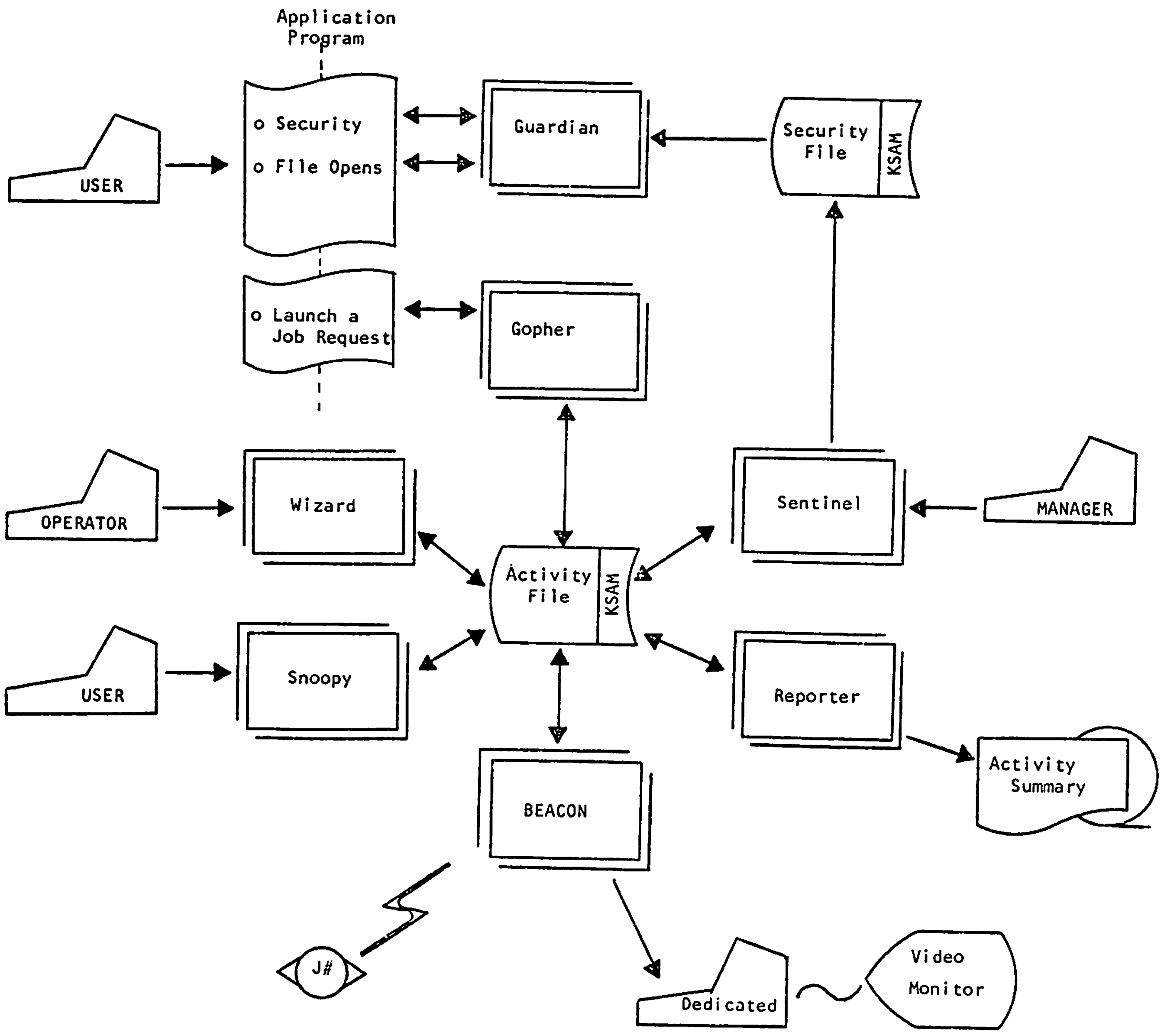

Application Program
o Security
o File Opens
o Launch a Job Request
USER
Guardian
Security File
KSAM
Gopher
OPERATOR
Wizard
Activity File
KSAM
Sentinel
MANAGER
USER
Snoopy
Reporter
Activity Summary
BEACON
J#
Dedicated
Video Monitor

Figure 3