EVALUATION OF ON-BOARD IN-FLIGHT CHECKOUT ON MILITARY AIRCRAFT

J. L. Johnson

IBM Electronics Systems Center
Federal Systems Division
Owego, New York 13827

The potential benefits of On-Board In-Flight Checkout (OBIFCO) were evaluated through computer simulation: This evaluation included fighter and transport aircraft operations within both wartime and peacetime environments to determine the effects of early failure warning, reduced diagnosis time, and reliability of diagnosis. A RAND-developed Monte Carlo Simulation, SAMSOM II, was modified for use in the study.

In February 1967, IBM began a study for the Air Force to evaluate the role of On-Board In-Flight Checkout (OBIFCO) in Air Force operations. OBIFCO is a maintenance technique featuring automatic analysis of data monitored during operation of aircraft subsystems. Preventive and other maintenance is performed on the basis of events and trends evident from the data. OBIFCO, related to AIDS (Aircraft Integrated Data System), has been assessed for feasibility and state of the art in Reference 1.

The study applied to two types of aircraft: a small tactical fighter bomber of the FX category and a multiengine intra-theater support aircraft similar to the C-130. Peacetime and war-emergency operations were considered. The techniques used in the study are presented in detail in Reference 2.

The military model used in the simulation is a combination of bases containing a number of one type of aircraft. Each type was assigned its own subsystems and mission requirements. Failure rates were incorporated for unscheduled maintenance that tends to interrupt schedules. Maintenance concepts were represented by varying diagnosis time and the probability of detecting failures before they could degrade subsystem performance.

Several potential benefits of OBIFCO were defined in Reference 1, and are summarized in Table 1. A computer simulation including all the benefits would be prohibitively complex, so the most significant characteristics were selected for detailed study. Based on these potential benefits, three factors were selected to reflect OBIFCO capability:

- Detection of critical malfunctions before they cause loss of subsystem utility.
- Reduction in time and man-hours required for diagnosis and fault isolation.
- Confidence in diagnosis (i.e., fewer maintenance actions on systems generating suspicious symptoms, but shown by detailed diagnosis to be healthy).

Table 1

POTENTIAL OBIFCO BENEFITS

On-line testing
Detection of intermittent failures
Continuous monitoring
Preflight checkout
Rapid fault isolation
Breakdown prediction
Optimum maintenance scheduling
Increased confidence in airworthiness
Knowledge of present capability
Reduction in:
Crew monitoring responsibilities
Redundancy in crew displays
Redundancy in equipment
Testing man-hours
On-ground time
False removals
Fleet size
Secondary damage
Damage from manual inspections
Field support:
Personnel
Skill levels
Spares inventories
Test equipment

Cost evaluation is difficult, especially in wartime conditions. However, a practical evaluation in terms of improved operations can be obtained. A multiparameter analysis was used to isolate several key areas that might benefit from some degree of OBIFCO application. The computer program employed was SAMSOM II (Support-Availability Multi-Systems Operations Model), written in SIMSCRIPT for the IBM 7044/7094 computers.

The SAMSOM II simulation program provides a general flexible model of airborne weapon system basing, maintenance, and logistics operations. The RAND Corporation developed SAMSOM II as part of Air Force Project RAND. It is described in References 3 and 4.

The model simulates operations events (alert requirements, sortie-generation capabilities, and readiness postures) and associated logistics support requirements (manpower, equipment, facilities, and, to a limited extent, parts) for one or more aircraft at one or more bases. At the user's option, the model can also take into account weather constraints, resource shortages, flying schedules, alert commitments, sortie-configuration requirements, abort rates, attrition and battle damage estimates, and operations policies governing sortie cancellation and makeup practices.

Therefore, the model can supply diversified information, including such management planning data as: (1) the capability that can be expected from a specified number of aircraft, given some set of supporting resources; (2) the resources that are required to support a group of aircraft operating at some specified level of effectiveness; or (3) the impact that specified operations changes have on total system effectiveness. Variations in maintenance policy were of greatest interest in OBIFCO analysis.

Figure 1 is a simplified logic diagram of events simulated by SAMSOM II. The simulation centers around three status pools, from which all missions are flown: the flyable pool, ready pool, and alert pool.

All aircraft are first put into the flyable pool. Depending on the sortie and alert requirements specified on the inputs, suitable numbers of aircraft are

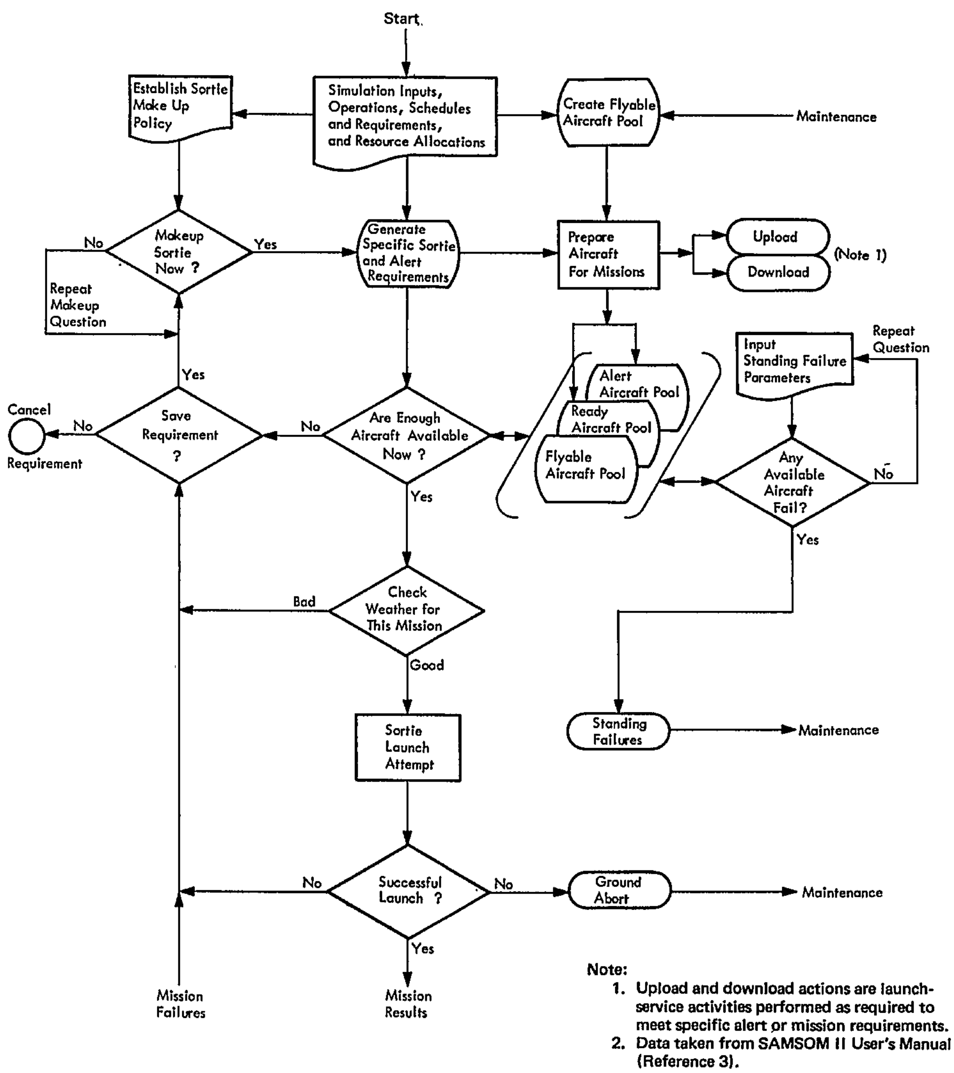

Figure 1. Operations Routines

prepared for missions, i.e., uploaded or configured. Then, the aircraft are put into the appropriate pools for launching to meet the operations requirements.

Individual subsystem failures may occur at landings or takeoff. Their generation, as well as indication of their criticality, are defined through Monte Carlo techniques according to input probabilities. When a failure first occurs, it is defined as latent. In this state the failure does not degrade subsystem performance and is not known to any personnel. If undetected, the failure becomes real after a specified number of flying hours. This interval is known as warning time and is a controllable input for each subsystem. After the warning time has elapsed, the real failure is automatically known, and maintenance is scheduled as soon as possible.

The location, amount of each resource, and the time required for each maintenance activity are specified for the simulation. Resources include personnel, parts, and equipment. Excessive maintenance requirements can lead to problems such as queues of aircraft in maintenance. In addition, weather or a critical failure may prohibit launch. Another aircraft will be substituted, or the sortie will be cancelled or saved until later.

An IBM modification permits specification of additional aircraft to be ready, if needed, at launch. These flight-ready spares are uploaded, placed in the ready pool, and assigned to the mission in the same manner as the aircraft that will fly. If one of the primary aircraft develops a critical failure, any available spare would immediately replace it.

SAMSOM II, as described by RAND, contains 11 summary report formats, described in Section IV of Reference 3. Three additional report formats were developed by IBM. Further information on their development is given in Reference 2.

REFERENCES

1. Johnson, J. L., et al, On-Board In-Flight Checkout, Technical Report AFAPL-TR-67-39, April 1967.

2. Johnson, J.L., McKenzie, H.R., and Moore, D.H., On-Board In-Flight Checkout Evaluation, Technical Report AFAPL-TR-68-54, June 1968.

3. Smith, T.C., Brown, G.D., Mason, R.A., Moulenbelt, R., and Shukiar, H.J., User's Manual for SAMSOM II, U.S. Air Force Project RAND Research Memorandum RM-4923-PR, November 1967.

4. Brown, G.D., Moulenbelt, R., and Shukiar, H.J., A Programmer's Guide to SAMSOM II, The RAND Corporation Report RM-5235-PR, November 1967.

ACKNOWLEDGMENT

This study was conducted by International Business Machines Corporation under USAF Contract No. F33615-67-C-1371. The project was administered for the AF Aero-Propulsion Laboratory of the Research and Technology Division by Mr. Joseph M. Ferguson (APFG), the Project Engineer, under project 8174, task 817412. At the IBM Electronics Systems Center, Owego, N.Y., the project was under the administrative supervison of Ralph I. Daehnke, with James L. Johnson acting as project leader. Messrs. Mitchell P. Rothman and David H. Moore assisted with simulation design, requirements definition, and evaluation. Mr. H. R. McKenzie was responsible for the extensive computer programming activity.