RAMREQ: A COMPUTERIZED TOOL FOR THE DEFINITION OF RAM (RELIABILITY, AVAILABILITY, MAINTAINABILITY) REQUIREMENTS OF COMPLEX SISTEM

P.Camerino,S.Sirignano,G.P.Benedetti
Agusta S.p.A. Tradate Italy

Abstract

The paper illustrate the software package RAMREQ developed by AGUSTA to define the RAM requirements of complex systems (helicopters, aircrafts, and space systems) by the integration of appropriate deductive (TOP DOWN) and inductive (BOTTOM UP) techniques. The overal logic flow of RAMREQ software tool can be split into three main process: the TDP (Top Down RAM Apportionment Process), which makes use of preliminary system information i.e. tentative RAM requirements, datum mission, preliminary sub-system architecture etc. and Value Analysis techniques; the BUP (Bottom Up RAM Prediction Process) which makes use of conventional RAM prediction and modeling techniques i.e. series, parallel, stand-by and bayesian models; the CAP (Comparison, Assignement and Feedback Process) which provides the final outputs, and controls the whole model by continuos comparison, feedback and cost-optimization of the variables involved.The software tools has been developed to be run under an IBM PC environment and it has been successfully applied to major rotocraft and space systems.

Introduction

A feasibility study to derive RAM quantitative requirements at system level must be carried out having in mind what is possible as well as what is desired. This implies a continuous comparison between what is known (and therefore possible in the timeframe of the technology evolution) and what is desired/required but not proved to be feasible within cost constraints. Clearly, before a comparison can be made, it is essential to have the two components to compare.
Making a decision as far as the feasibility of a RAM quantitative requirement is a difficult problem to solve given many incomparable parameters involved because of the wide spectrum of existing possibilities.

The possible operative philosophies go from a completely subjective decision which is very easily made but completely dependent upon the experience of a particular "decision maker", to the completely objective decision which does not depend upon (and it is not influenced by) the capacity of the decision maker but very difficult to make.

The method illustrated here is a

tentative to balance between the speed of the first approach, given the relative small data requirement to perform the analysis, and the high reproducibility and correctness/confidence in the result of the second.

This approach will allow the analyst to adopt in the same process a coherent structure containing:

i. Objective mathematical models.

ii. Tools which minimise the subjectivity of inputs based on personal judgement.

A furher benefit of this model is that by the integration of two very different logical process, a higher degree of confidence can be placed in the result.

Model description

In the case of the RAM requirements of complex systems there are two components to be considered during the comparison process of very different types, they can be assembled into two groups:

GROUP A : GLOBAL RAM FIGURES (GRF) from the system requirements i.e. system availability, mission success probability, probability to successfully complete corrective maintenance needs, servicing intervals and servicing times etc.

Note: During the preliminary phase of the system project these global RAM figures are not completely defined in the classical quantitative RAM terms (i.e. MTBF, MTTR etc.) therefore a first "tentative value" is assumed for them to start the process. The iterative nature of the process will ensure the convergence to feasible figures to assume as requirements.

GROUP B : INDIVIDUAL RAM FIGURES (IRF) from past experience and the current technology level i.e.component/equipment reliability figures, IVA/EVA and ground maintenance task times, diagnostic time assessments etc. Therefore, given their different nature, it becomes necessary to obtain two pairs of numbers from the above groups as follows:

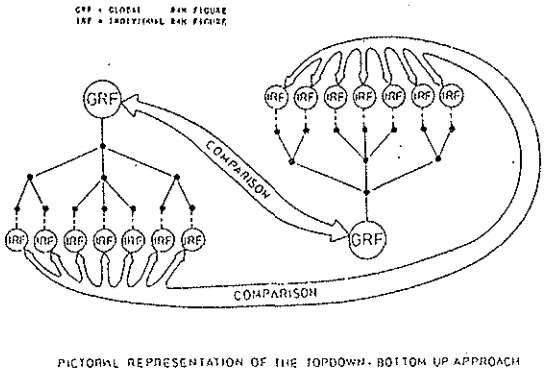
(i) COMPARABLE NUMBERS to make a comparison between GRF the first taken directly from Group A and the second derived from Group B.

(ii) COMPARABLE NUMBERS to make a comparison between IRF the first derived from Group A and the second taken directly from Group B.

Intuitively it is possible to say that the two derivations introduced hereabove are obtained by the use of two different logics:

i. a TOP-DOWN deductive logic to determine the IRF's from the GRF's.

ii. a BOTTOM-UP inductive logic to determine the GRF's from the IRF's.



PICTORIAL REPRESENTATION OF THE TOPDOWN - BOTTOM UP APPROACH

Fig. 1

The methodological approach to make comparison is illustrated in Fig. 1, where the trees represent suitable deductive TOP-DOWN or inductive BOTTOM-UP techniques.

The overall logic of derivation of the RAM requirements is split into three main processes as follows (Fig. 2):

(i) A Top Down RAM Apportionment Process (TDP)

(ii) A Bottom Up Prediction Process (BUP)

(iii) A Comparison, Assignment and Feedback Process (CAP)

The first process makes use of current system information (i.e., requirements, datum mission, preliminary sub-system-architecture etc.), a functional approach to assess complexity and importance factors necessary to carry out the analysis and "value analysis" techniques.

The second process makes use of conventional RAM prediction and modelling techniques.

The last process provides the final outputs, and controls the whole model by continuous comparison, feedback and optimisation of the variables involved.

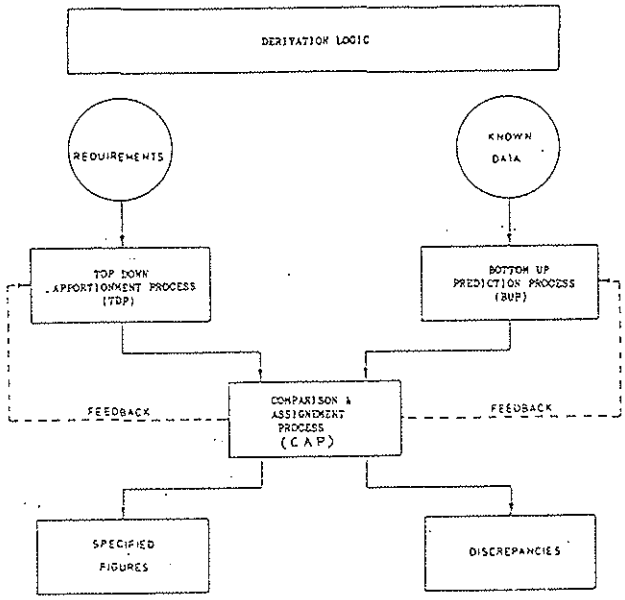


Fig. 2

The logic flow of the apportionment process (TDP) is illustrated in Fig 3.

The explanation of each block is given below:

- Block 1 Primary Inputs

Beside the "tentative" system RAM requirements, data concerning both system architecture and mission profile provide inputs to the TDP.

At the initial stage of a complex system project the "tentative" start value is at highest level (i.e. availability level and mission success

probability level) because of the continuous fluctuation of such essential parameters like mission duration, system configuration and architecture, system dimensions etc. Only the mission objectives are more clearly defined and fixed enough.

Always,at the initial stage of the analysis performed on a complex system its system architecture is not completely defined at the level necessary to perform a definitive assessment. However, in order to proceed with a first stage of the process which can subsequently evolve with the architecture definition level, a functional approach may be used.

The functional approach assumes that each system function can be identified with a set of hardware. Clearly, identification between function and hardware is related to the system definition level (DL). In this context, if the system definition level DL is 100%, then the identification between function and hardware is, of course, 100%.

Therefore, from the above assumptions, the system architecture is assumed to be the set of its functions as currently defined.

The systematic overview is carried out following the classical Miles approach to Value Analysis that it is possible to synthesise in the following steps:

a) Function definition. If we take a complex idea and condense it into two words, say a verb and a noun, we lose information. If we do it with a purpose, we have to decide what information we can afford to lose. Then we are left with the most significant aspect of the item's purpose - the primary function or "What does it do?".

b) Functional dependence. By asking iterative and interconnected questions like "Why?" and "How?" we look for the functional dependences to set up the logical links between the input functions and the output functions. To raise the level of abstraction we ask "Why?". To lower the level of abstraction we ask "How?".

c) Reference functions selection. A selection of reference functions is made as much balanced as possible in terms of functional contents.

In the context of the analysis by the expression mission profile is intended to indicate a composite reference mission that includes all possible eventualities without overlap between them. This may, of course, not necessarily be a realistic mission. Often the reference mission is the most critical from a performance and/or resources point if view.

- Block 2 Secondary Inputs

Starting from the datum mission and system architecture (primary performance inputs), a set of parameters and tools that bound the problem are derived:

- Complexity factors. To take into account the hardware content of each sub-system

relative to that of the system;

- Importance factor. The sub-system importance factor is defined as the probability that the system will fail to accomplish its mission if the sub-system fails while all others are satisfactory;

- Duty cycles. To take into account the different utilization of each sub-system throughout the mission;

- Operational reliability configuration. To take into account both the functional and reliability inter-relationship between the sub-systems involved during the mission.

- Block 3 Series Apportionment Model

A first apportionment model concerning logistic reliability figures is exercised to sub-system level as defined in the system architecture, and weighted according to complexity factors and duty cycles.

- Block 4 Mission Reliability and Safety Apportionment Model

A second apportionment model concerning mission reliability figures is exercised considering the sub-systems (with their redundancy degree) that affect the mission reliability.

- Block 5 Selection of Appropriate Reliability Figures

The appropriate reliability figures are selected and assigned in the following priority:

(i) Apportioned logistic M.T.B.F. to those sub-systems that only affect the logistic M.T.B.F. requirement.

(ii) Apportioned mission reliability to only those sub-systems that effect the mission reliability requirement, but not safety.

- Block 6 Outputs

Appropriate apportioned reliability, safety and maintainability figures for all sub-systems are obtained.

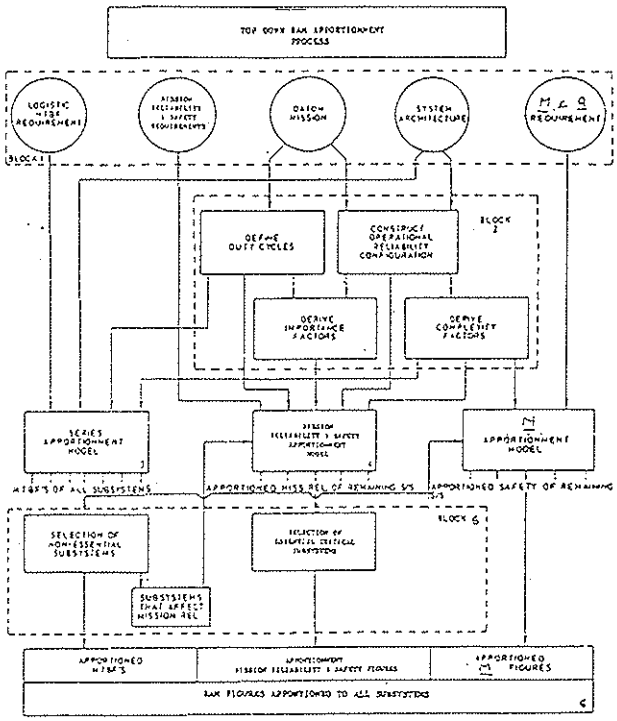


Fig. 3

The logic flow of the prediction process (BUP) is given in Fig. 4.

Each block is explained below:

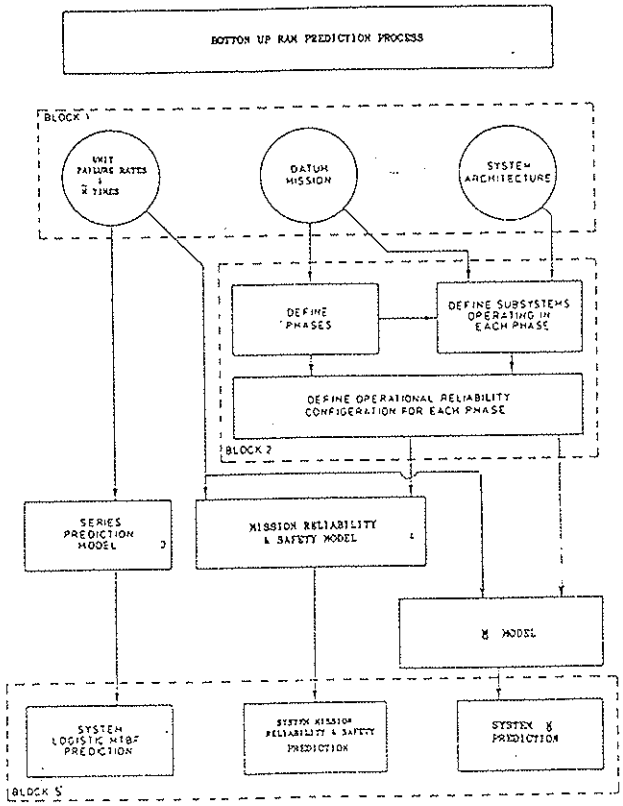


Fig. 4

- Block 1 Input data

Beside unit failure rates, based on past experience or taken from technical literature, data concerning both system architecture and mission profile provide inputs to BUP.

- Block 2 Definition of the process

Starting from the mission profile and the system architecture the different mission phases and the sub-system utilisation in each phase are defined. The system operative configuration in each phase is then built-up taking into consideration the functional and reliability interrelationship between the sub-systems.

- Block 3 Series Model

. A simple series model is used to predict the system logistic reliability figures.

- Block 4 Mission Reliability and Safety Model

The system mission reliability is predicted using the AGREE model.

- Block 5 Outputs

The predicted logistic M.T.B.F., missione reliability, safety and Maintainability are obtained for the system.

The logic flow of the comparison process (CAP) is illustrated in Fig. 5.
The explanation of each block is given below:

- Block 1 Inputs

The inputs and outputs (I.R.F. and G.R.F.) of the two previously described processed provide the inputs to the comparison and assignment process.

- Block 2 Comparison

From a critical assessment of the two comparable figures obtained from the TOP DOWN and BOTTOM UP processes a decision as to the feasibility of the reliability requirements is made.

- Block 3 Selection

Depending upon a "feasible" decision and there being "no overdesign", the most restrictive between the

APPORTIONED and PREDICTED RAM figure is selected and assigned to each sub-system.

- Block 4 Top Down Feedback

Depending upon a "not feasible" decisione a feedback process on the apportionment model is carried out until "saturation" of the model is reached.

- Block 5 Bottom Up Feedback

Depending upon the "saturation" of the apportionment model or the identification of "overdesign" and subject to data on other system constraints (i.e., weight, cost etc.) being available, an optimisation of the system configuration is carried out and then fed back to the prediction model.

- Block 6 Outputs

Appropriate RAM requirements both at system level and at equipment level or any discrepancies which require further agreement are obtained.
A similar model,i. e. a comparison between two different approach (TDP and BUP) is used to derive maintainability figures which are in concert with the derived Reliability figures, will meet the global system Availability requirement.

RAMREQ sw tool description

The RAMREQ software (sw) tool is developed according to the ESA software engineering standard PSS-05-0 Issue 1; the whole therein specific software life-cycle but maintenance phase is followed:

- requirement phase:
  user & software specification
- architectural design phase;
- detailed design phase;
- transfer phase:
  module test,
  sw integration,
  component verification,
  pre-acceptance and acceptance test.

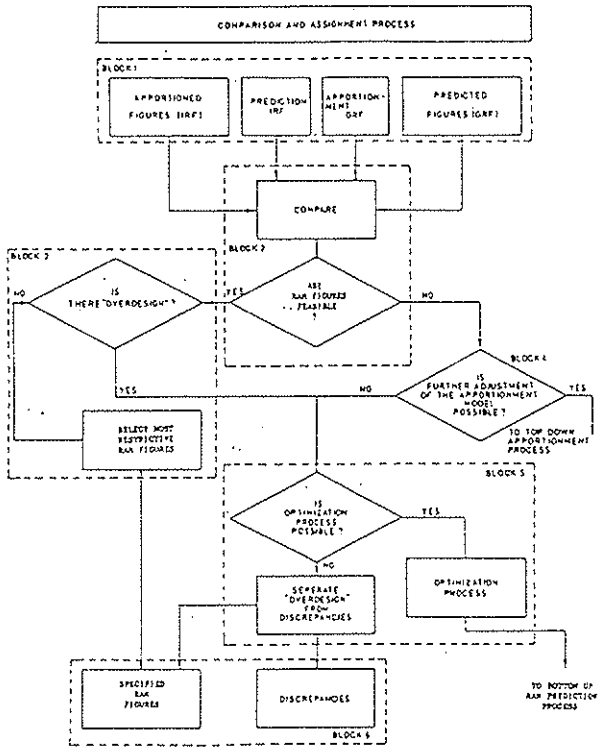


Fig. 5

When necessary an approved tailoring of the standard is followed. All the software design is performed using the CASE tool named "EPOS", all the software documentation is produced automatically by means of this tool.The architectural design is given in terms of "Structure-Analysis" for data,the control flows are also given. The detailed design is given for each module in terms of control-flow including information of all exchanged data.

The RAMREQ sw tool runs on a personal computer (PC) IBM-AT compatible using only the 640K of RAM memory. RAMREQ sw is written in FORTRAN, the Microsoft Version 5.0 of FORTRAN compiler, to enhance the sw tool portability.A large library model and co-processor emulator option are used to compile and debug source code. MS-DOS version 3.1 or higher is used.
At least a VGA colour video-terminal display shall be available.
The application software is the implementation of the model specified above.

The RAMREQ sw tool is mainly made up by application and man-machine-interface (MMI) software.The MMI sw implements the user/application-software/MS-DOS interface as well as the sw system function: data entry, data print, data display, data back-up/restore, RAM process activation (top down, bottom up and comparison/assignment) and help.

The MMI implements a certain number of masks which are built using the PANEL-PLUS II (FORTRAN version) development tool. The user can define/display the system architecture by means of two options: moving through the "system tree", object by object (using the objects name) in a top-down/left-right fashion from higher to lower object levels or displaying on the screen the selected "subtree" of objects (the father and its sons) and then selecting the wanted object. The data entry functions through masks implements an automatic range check on assigned data.The data display function also provide a graphic representation of the modeled system reliability architecture.

The help function is an on-line/context-sensitive function which provides help message boxes partially overlapping the current window; in addition a message line is foreseen at the bottom of each current window where the run-time messages coming from application sw are displayed; these messages are relevant to the status of the validation (see below) and computing session.

The print, the back-up and restore function allow a global and selective data management in order to allow a high degree of modularity when this options are activated; the global option manages the complete mission and/or system architecture data set while the selective option manages either a subtree of the system tree or a specific number of object levels from the tree root level.

In order to fulfill the requirement relevant to the memory limits (640K ram) an "ad hoc" data-base and data-base-manager where developed to store all data relevant to the 4441 different objects which constitutes the maximum size of the modeled system (1 system of 20 functional groups; each functional group made by 20 functional sub-systems; each functional sub-system made up by 10 units). The data-base is made up by four data files which can be read and updated on user request by direct access. To avoid unnecessary data base files opening, a series of "tables" are implemented run time in order to make available the definition status of the data base objects. To ensure safe exit condition from a

computing session in case of system failure the last backed-up situation is maintained: no temporary results are kept.

To avoid any "bad" computing situation, the exception handling is implemented run time into each application sw module since FORTRAN can not give any support in this sense; in addition a validation session on the whole set of data, which will be used in the incoming requested computing session, is always performed.

A consistency check on the results coming from the top-down process (apportionment) and from the bottom-up process (predicted) is also performed to ensure a correct execution of the comparison and assignment process.

References

ESA software engineering standards - ESA board for software standardization and control (BSSC) - NL; PSS-05-0 Issue 1.

Engineering and project management oriented support system (EPOS);GPP - Gesellscheaft fur prozebrechner-programmierung mbH.

PANEL PLUS II FORTRAN version Advanced screen manager Roundhill computers systems - Version 2.